A "First and Last-Mile" Assessment of California Rail Stations: A Sketch Planning Approach

A Senior Project
presented to
the Faculty of the City and Regional Planning Department
California Polytechnic State University – San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
City and Regional Planning

By
Henry McKay

March 2021

-- This Page Intentionally Left Blank --

Approval Page

Title: A “First and Last-Mile” Assessment of California Rail Stations: A Sketch Planning Approach

Author: Henry McKay

Date Submitted: 03/14/2021

Cornelius Nuworsoo, Ph.D., AICP	__________________________	___________
Senior Project Advisor	*Signature*	*Date*
Michael Boswell, Ph.D., AICP	__________________________	___________
Department Head	*Signature*	*Date*

-- This Page Intentionally Left Blank --

Acknowledgements

Thank you to **Professor Cornelius Nuworsoo**, my Senior Project Advisor, for sparking my interest in transportation planning. Before taking your courses, I had no idea which area of planning I wanted to work in. Your hands-on, quantitative approach to teaching planning concepts heavily shaped me as a planner and I would not be where I am today without your guidance.

Thank you to my various planning colleagues over the years, especially at Caltrans Division of Rail and Mass Transportation. Your guidance and mentorship greatly influenced my planning career thus far and has made me excited to be entering the rail planning field. Furthermore, collaborative work done on station planning directly inspired the topic of this project and feedback from various colleagues was key in its development.

Lastly, thank you to my family, who have been highly supportive of all my endeavors throughout my life, including college. Though I am the only one in the planning field, let alone working in transportation, you all have been incredibly supportive of my work over the years.

Table of Contents

List of Figures

List of Tables

Executive Summary

The purpose of this study is to assess the pedestrian and bicycle accessibility of rail stations in California by providing a sketch planning toolkit enabling planners and decision makers to compare various accessibility factors and to develop and compare metrics.

Pedestrian and Bicycle Accessibility is a highly important area of transportation planning, especially as it pertains to rail stations. While a well-planned rail network can serve many people, travelers must get to and from rail stations on either end of their trip in order for the rail service to be of any use. The mode by which travelers make these trips varies depending on the station, but walking and biking are common, especially in urban areas. The nature of the built environment can play a large role in determining what modes are used to access rail stations. For example, dense urban areas with highly connected street networks and pedestrian and bicycle infrastructure encourage non-motorized station access. However, stations in less-dense areas lacking in street connectivity and pedestrian and bicycle infrastructure tend to encourage more automobile access.

To the degree that these factors can be quantified, rail stations can be assessed on various accessibility factors, and problems can be identified. With this knowledge, planners can better address station area access issues. While this project provides background information on station access and describes some of the most important data in determining accessibility for bicyclists and pedestrians, it primarily provides a technical methodology to quantify station area accessibility as well as a sketch planning toolkit to carry out the analysis. This project is intended to assist practitioners who are already familiar with accessibility issues in their area of work, but who may lack the technical resources and or data to carry out a large-scale analysis.

1. Introduction

In broad terms, this project examines pedestrian and bicycle accessibility for rail stations in California. It also presents a technical toolkit for quantifying and comparing station accessibility.

Origin and Purpose

Developing this project was an iterative process, and it will continue to be updated and enhanced even after it is submitted as a senior project to the City and Regional Planning Department at Cal Poly, San Luis Obispo. The inspiration to assess pedestrian and bicycle accessibility originally arose from work that was being done on the 2022 California State Rail Plan (SRP) by the Planning Branch of the Division of Rail and Mass Transportation at the California Department of Transportation (Caltrans). Part of the State Rail Plan involved the preparation of a statewide station inventory with numerous data points collected for each station, mostly through manual data collection using internet resources and Google Maps. However, this project quickly expanded to look at station area data and to include many GIS-derived quantitative data points. Another aspect of the State Rail Plan was the quantification of station accessibility for various modes of transportation, with fairly-broad technical guidance given from the Federal Railroad Administration (FRA). Given the size and variability of California's rail stations and station areas, the development of a 'one size fits all' metric seemed inappropriate and a more process-oriented, transparent, sketch planning approach was adjudged more suitable. This report documents the development of such a process, the input data used, the tools created, and the analysis results for three different planning scenarios.

The intent of this report is to serve as a guide to the technical set of tools developed to conduct station area analysis and the thought process behind them, as well as a brief introduction to the area of first and last mile planning for rail stations. While the report itself is a static document, the toolkit lives on the internet and will continue to be updated and enhanced when existing data is updated, new data is added, and when new features are developed.

Organization of Report

This report is organized into eight sections (or chapters) including this introductory chapter. The following subsections highlight the contents of the other chapters.

Background and Literature Review

Background on key topics relating to pedestrian and bicycle accessibility for rail stations and a brief review of relevant literature referenced in this report.

Data

A comprehensive look at the various pieces of data gathered and analyzed for this project, including aggregation methodology and sample maps and tables.

Walk Score Analysis

An exploratory data analysis of WalkScore.com's walk and bike access metrics, analyzed in the context of the station area data discussed in the previous section.

Metric Development

An explanation of the metric development methodology used to score rail stations on pedestrian and bicycle accessibility and a discussion of other common methodologies.

Toolkit

An overview of the various technical tools developed for this project, including a spreadsheet-based sketch planning tool, web-based mapping content, and a series of Python and R scripts.

Analysis

An exploration of three hypothetical planning scenarios and their analysis results using the methodology and tools developed for this project.

Conclusion

Summary of key findings and future areas of improvement.

2. Background and Literature Review

This section provides background information on a few key concepts central to this project's focus. It also includes a review of important literature that was highly influential in the development of this project.

Background

Prior to reviewing relevant literature and discussing the project itself, it is necessary to establish a few key concepts and terms that are central to the focus of this project and report.

First & Last Mile Planning

This project broadly addresses issues related to "first and last-mile" planning. In transportation planning, the "first and last-mile" is the portion of public transportation trip where a traveler has to get from the origin to the transit facility or from the transit facility to the destination. In many cases, these legs of the trip are made using non-motorized modes of transportation such as walking or cycling (Los Angeles County Metropolitan Transportation Authority – Metro, Southern California Association of Governments, 2014). "First and last-mile" planning aims to increase the reach of transit service by strategically upgrading infrastructure around rail station areas to make them more accessible to travelers. Though "first and last-mile" planning is a relatively new phenomenon, many transit agencies have developed "first and last-mile" plans or access plans for specific stations, or for their entire networks. This project is different in that it takes a much broader look at station area access but does so for the entire state of California. Furthermore, this project applies a consistent analytical methodology across all stations, enabling a wide variety of comparisons to be made.

Sketch Planning

The technical toolkit developed for this project falls under the broad category of sketch planning. In its simplest form, a sketch planning tool or process is a useful way to present a simplified version of an otherwise highly complex system (Crooks, 2008). Sketch planning often involved the development of simple Spreadsheet or Geographic Information Systems (GIS)-based tools to quantify planning problems and to assess alternative scenarios without performing highly specific engineering analysis. Though sketch planning tools often rely on default, highly generalized parameters, this project allows users to input their own parameters.

Literature Review

In order to better understand "first and last-mile" planning and the various methods by which accessibility is quantified, key pieces of literature were reviewed. In addition to reviewing academic literature, "first and last-mile" and accessibility plans were reviewed, as well as more technically oriented websites.

LA Metro First Last Mile Strategic Plan

The Los Angeles County Metropolitan Transportation Authority (LA Metro) has been one of the most prolific transit agencies in terms of "first and last-mile" planning, largely in preparation for the opening of new rail lines and stations. In 2014, the agency released a "first and last-mile" strategic plan, which established a set of guidelines for future "first and last-mile" planning work. The plan serves as an excellent introductory resource for "first and last-mile" planning and offers clear guidance as to the types of data that should be collected and measured. The plan also does an excellent job of explaining various station area physical site improvements that can increase "first and last-mile" pedestrian and bicycle accessibility. It also clearly (and graphically) lays out the desired outcomes of "first and last-mile" planning efforts, mainly that user access sheds should be expanded for various modes through a well-defined network of access pathways (Los Angeles County Metropolitan Transportation Authority – Metro, Southern California Association of Governments, 2014). However, the methodology set forth by the strategic plan relies heavily on field data that can only be gathered manually and is thus difficult to automate. While carrying out the methodology is practical for a small number of stations, it is less practical when applied to a large number of stations throughout the state. Though this project takes a much broader look at station area access, albeit on a larger scale, LA Metro's "first and last-mile" strategic plan played a key role in determining what data was included in the analysis and how it was analyzed.

Transport Access Manual

The Transport Access Manual was prepared by the committee of the transport access manual, a group of academics and practitioners led by renowned transportation researcher David Levinson. The manual provides a broad overview of accessibility as it relates to transportation, but primarily serves as a guide to quantifying accessibility through various technical methodologies. Though many of the methods discussed are more sophisticated than the ones implemented in this project, the manual was still highly useful in developing metrics and analyzing accessibility data. Specifically, the *Transport Access Manual* discusses potential biases in spatial statistical analysis which are relevant to this project, such as edge effects. When pre-defined cut-offs are used for analysis, important features that exist just beyond the cut-

off can be excluded even though they are important. One recommendation set forth by the manual is to use study areas defined by functional urban areas as opposed to arbitrary boundaries. This was carried out in this project by including accessibility isochrones as measures, which better-represent functional urban areas as defined by the existing network. Furthermore, the study provides an inventory of data sources and tools for accessibility analysis, some of which were used in this project (Committee of the Transport Access Manual (2020).

Manual on Pedestrian and Bicycle Connections to Transit

The Federal Transit Administration (FTA)'s *Manual on Pedestrian and Bicycle Connections to Transit* was referenced in multiple "first and last-mile" plans and serves as a good resource for standards pertaining to "first and last-mile" planning. For this project, it was used to define the primary access sheds for pedestrian and bicycle analysis, one-half mile and three miles, respectively. Though other access shed distances were included in the analysis, it was important to adhere to a standard distance for access sheds, that is used across all "first and last-mile" research and analysis (Transportation Research and Education Center at Portland State University, 2017).

Walk Score Methodology

Walk Score is a company that provides walk, bike, and transit metrics to real estate companies. Though the metrics are calculated using a proprietary algorithm, Walk Score's website discusses the company's metric calculation methodology in some detail, leaving out specific numbers. The methodology developed by Walk Score served as the basis for the methodology developed in this project and also offered insights as to which variables to include in the metric. Overall, the metric methodology outlined by Walk Score emphasized the importance of surrounding amenities in the score and detailed how scores were calculated. Walk Score's website also detailed the methodology of their bike score, which also influenced the development of bike metrics developed for this project (Walk Score, 2021). Section four of this report includes a more detailed discussion of Walk Score's methodology.

Other Literature

Several other pieces of literature were reviewed for this project but were less influential in the development of this project. This literature included academic studies, various "first and last-mile" and accessibility plans, as well as websites and blog posts where technical accessibility quantification methods were discussed. The references section at the end of this report identifies documents reviewed.

3. Data

This section discusses all input data that was collected for this project, how it was collected, how it was quantified, and why it is important in the context of "first and last-mile" planning. For each piece of data discussed, an illustrative map and table show the data in the context of one of California's most important rail stations: Los Angeles Union Station. The map shows how the spatial data is distributed within the various station catchment areas, while the table shows how the data appears when aggregated to each catchment area in tabular form. This tabular data serves as the input for the analysis and toolkit. It is important to note that Los Angeles Union Station was simply chosen as an illustrative example, and that the exact same data is available for all five hundred plus stations in the State. An interactive map version of the complete dataset can be found here, and the complete tabular dataset can be accessed through the tool, which can be downloaded here. Table 3.1 shows the data discussed in this section.

Table 3.1: Data

Data	Source	Included in Analysis
Service	Manual Inventory	Yes
Service Type	Manual Inventory and Classification	Yes
Bicycle Facility Mileage (Exclusive, Shared, All)	Open Street Map	Yes
Population Density	American Community Survey	Yes
Job Density	Longitudinal Employer-Household Dynamic	Yes
Environmental Justice (CalEnviroScreen Scores)	California Office of Environmental Health Hazard Assessment	Yes
Pedestrian and Bicyclist Safety	Statewide Integrated Traffic Records System	Yes
Commute Mode Split	American Community Survey	Yes
Intersection Density	Open Street Map	Yes
Accessibility Isochrones	Open Street Map, ESRI Network Analyst GIS Extension	Yes
Points of Interest	Open Street Map	Yes
Setting	Manual Inventory and Classification	No
Service Frequency	Various Rail Service Schedules	No
Vehicle Parking	Various Service Websites and Manual Survey	No
Bicycle Parking	Various Service Websites and Manual Survey	No
Transit Connectivity	Various Service Websites and Manual Survey	No

For the purposes of this analysis and toolkit, only primary data sources that could be a) publicly obtained online and b) cleaned and aggregated through an automated process were included. This was done to maintain consistency throughout a wide variety of stations in the state and to ensure that data could be

easily updated when new updates became available. Other pieces of data were collected that did not meet the above criteria and thus were not included in the analysis and toolkit. However, they are still important, and should be considered for future improvements to this toolkit. These pieces of data are briefly discussed at the end of this section as secondary data sources.

Framework for Data Collection and Aggregation

Most data included in this analysis is geospatial, meaning that it can be quantitatively measured in terms of its proximity to rail stations. For every piece of data analyzed, cumulative measurements were taken for the following rail station catchment areas:

- Quarter Mile
- Half Mile
- One Mile
- Two Miles
- Three Miles

These five catchment areas were chosen due to their utility in measuring pedestrian and bicycle access and are recommended by the Federal Transit Administration (FTA). Specifically, a half mile catchment area is recommended for pedestrian analysis while a three-mile catchment area is recommended for bicycle analysis (Transportation Research and Education Center at Portland State University, 2017). Other intermediary catchment areas were also measured and included to allow a wider breadth of analysis options.

Primary Data

The following subsections identify pieces of data that are primarily quantitative in nature. These are included in the analysis and toolkit discussed in this report.

Services

For every rail station in the state, any service that stops at the station was recorded. While many stations throughout the state only handle one service, some large multimodal stations such as Union Station in Los Angeles handle several different rail services per day. For this analysis, every station for every type of rail service in the state was analyzed, including:

- Intercity Rail
- Commuter Rail
- Interstate Rail
- Heavy Rail
- Light Rail
- Streetcars

A list of rail services included in this analysis can be found in Table 3.2. For services with multiple lines such as Metrolink, separate lines were analyzed as a single service.

Table 3.2: Rail Services in California

Service	Type	No. of Stations
Capitol Corridor	Intercity Rail	18
San Joaquin	Intercity Rail	18
Pacific Surfliner	Intercity Rail	27
Amtrak Long Distance	Interstate Rail	25
Caltrain	Commuter Rail	32
Altamont Corridor Express (ACE)	Commuter Rail	10
Sonoma-Marin Area Rail Transit (SMART)	Commuter Rail	12
Metrolink	Commuter Rail	64
Coaster	Commuter Rail	8
Sprinter	Light Rail	15
Bay Area Rapid Transit (BART)	Heavy Rail	51
LA Metro	Heavy Rail/Light Rail	93
Sacramento Regional Transit (SacRT)	Light Rail	53
Muni Metro	Light Rail	56
Muni Heritage Streetcar	Streetcar	35
Santa Clara Valley Transportation Authority (VTA)	Light Rail	62
San Diego Trolley	Light Rail	54
Total		***633***

Service Types

Every rail station in the analysis was grouped into service types, describing the kinds of services that serve the station. For stations with only one service type, say a Caltrain commuter rail station, the

assigned type was commuter. For stations with multiple service types, of different types, the assigned type was intermodal. This classification allows for stations to be grouped together and analyzed in relation to one another. Table 3.2 identifies services types.

Bicycle Facilities

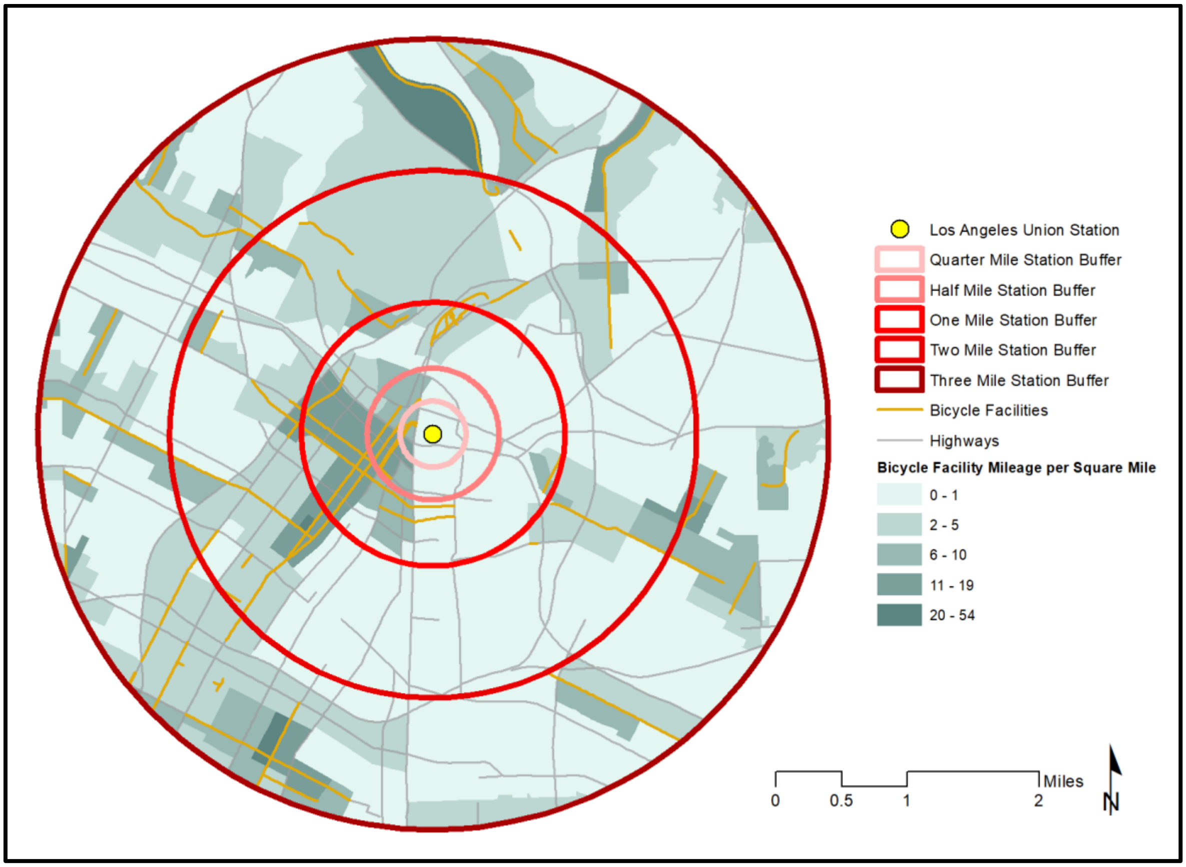
Bicycle facilities include both shared and exclusive bike lanes and are highly important when analyzing a rail station's bicycle accessibility. In this analysis, the total mileage of exclusive and shared bicycle facilities was measured for each station catchment area. Exclusive bicycle facilities are bike lanes that are completely separated from vehicle traffic for exclusive cyclist and pedestrian use. Shared bicycle facilities are bike lanes that are located on streets, sharing right of way with vehicles. Total bicycle facilities are simply the sum of the previous two facility types.

Though many agencies provide GIS data of their bicycle facilities, not all data sets are up to date and certain agencies do not maintain and publish GIS data at all. In order to look at the entire state comprehensively, Open Street Map (OSM) bicycle facilities data was utilized. Open Street Map is a platform which allows any user (including government agencies) to upload and update data, so the statewide bicycle facilities network is fairly comprehensive. However, the data is not perfect, with certain bicycle facilities missing depending on the location. If the study area were restricted to a single region, say Southern California, it may make sense to use data from a regional planning agency since it would likely be more precise.

Mileage of bicycle facilities surrounding rail stations is an important piece of data to assess bicycle accessibility. For bicyclists, it is much safer to ride in a dedicated lane than it is to ride in traffic with automobiles. Adding bicycle facilities can increase the actual and perceived safety of bikers in an area, increasing the bicycle access mode share for a station.

In Figure 3.1, bicycle facilities within a three-mile radius of Los Angeles Union Station are shown. Census block groups are also shaded to reflect the total mileage of bicycle facilities per square mile. There is a higher density of bicycle facilities to the immediate West of the station itself, in downtown Los Angeles.

Figure 3.1: Bicycle Facilities near Union Station, Los Angeles



Data Source: Open Street Map, 2020, Map by Henry McKay

Table 3.3 shows the total mileage of bicycle facilities within each catchment area. These figures are cumulative, meaning that there are 1.59 miles of bicycle facilities within one quarter mile of Union Station, 4.14 miles of bicycle facilities within a half mile of Union Station, and so on… Furthermore, the figures are broken down by type of bicycle facility, Exclusive, Shared, and All, as they are in the data. Detailed technical information on obtaining and cleaning bicycle facility data is in Appendix 2, Scripts.

Table 3.3: Bicycle Facility Mileage near Union Station, Los Angeles

Bicycle Facility Mileage	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
Exclusive	1.12	1.77	4.02	9.63	46.24
Shared	0.47	2.37	9.85	24.10	54.72
All	1.59	4.14	13.87	33.73	100.97

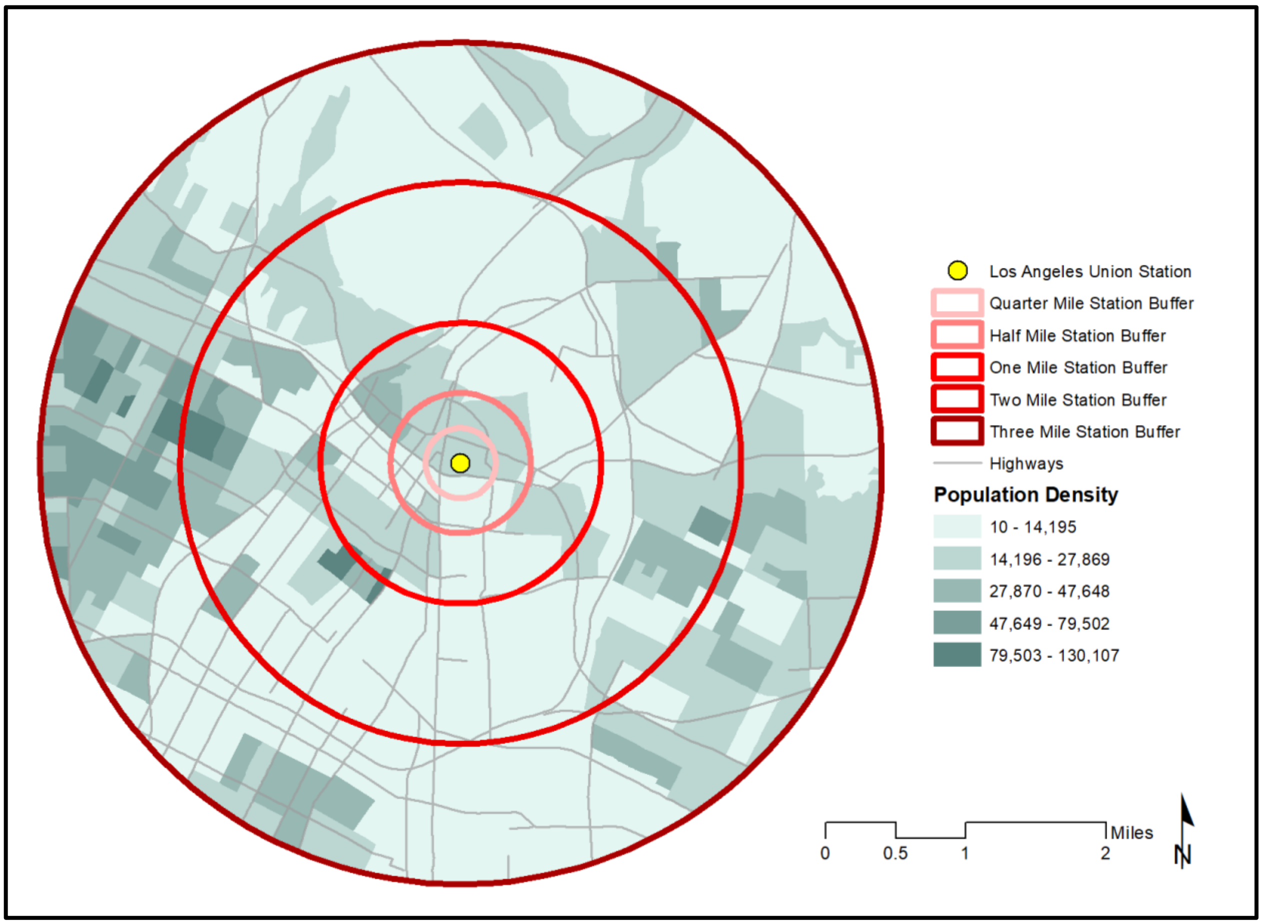
Data Source: Open Street Map, 2020

Population Density

Population density is a measure of how many people live within a certain geographic area. Data was retrieved from 2018 American Community Survey Five Year Estimates at the Census Block Group Level. Total population of all block groups within a catchment area was summed for the area. For block groups that fell partially within the station radius, a proportion of that block group's total population was taken. For example, if twenty five percent of a block group's area fell within the catchment area of interest, twenty five percent of that block group's total population would be included in the sum.

Figure 3.2 shows the population density surrounding Union Station in Los Angeles. Darker shaded census block groups represent higher numbers of people per square mile.

Figure 3.2: Population Density near Union Station, Los Angeles



Data Source: 2018 American Community Survey 5 Year Estimates, Table B01003e1

Map by Henry McKay

Table 3.4 shows the number of people living within each catchment area around Union Station. The number of people living within the immediate vicinity of the station (quarter mile) is fairly low. However, once the radius is extended to three miles, the population estimate is over four hundred thousand, which makes sense since it includes downtown Los Angeles.

Table 3.4: Total Population near Union Station, Los Angeles

Population Density	Catchment Area Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
# of People	3,799	12,753	42,984	168,611	426,786

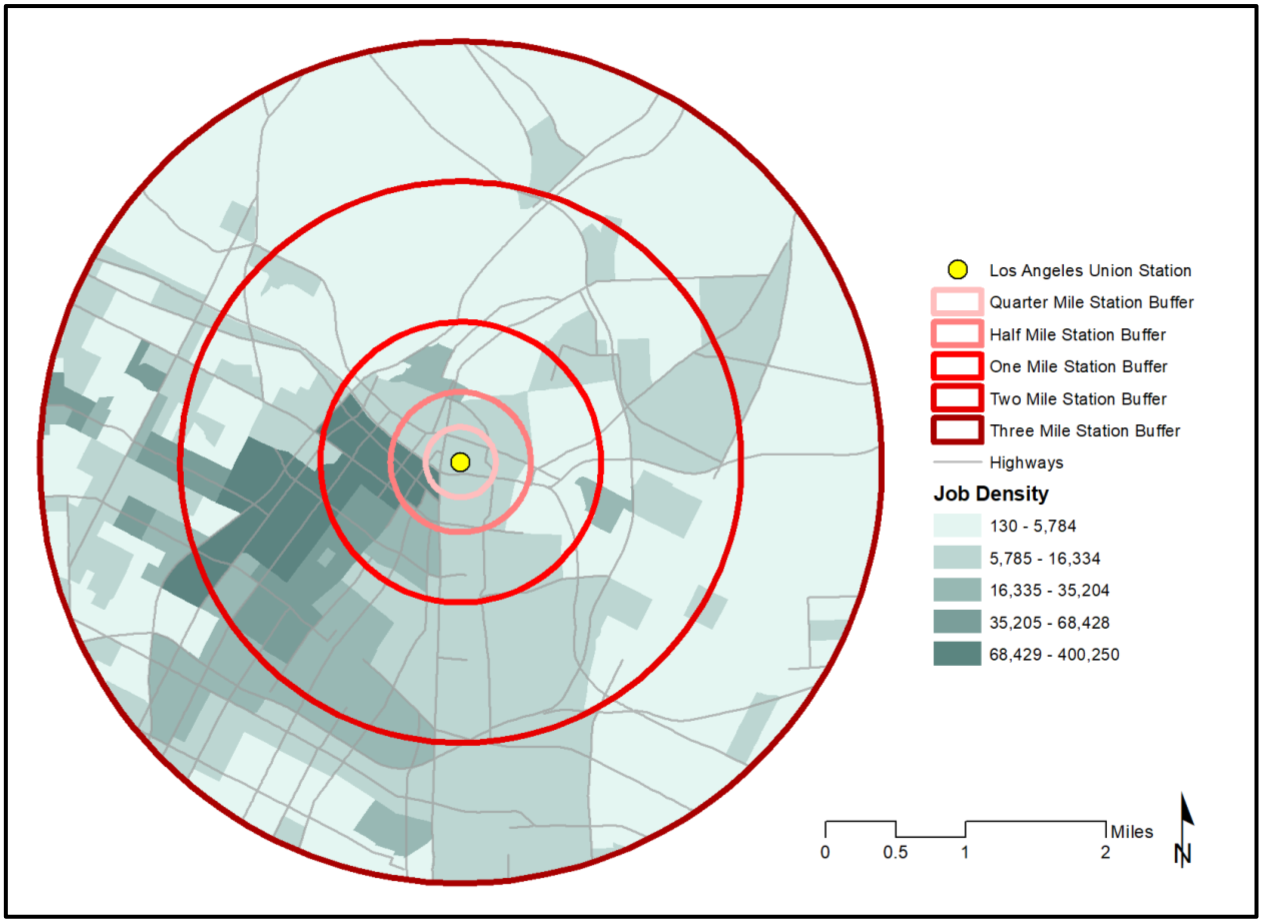
Data Source: 2018 American Community Survey 5 Year Estimates, Table B01003e1

Overall, the stations with the highest surrounding population densities are located in San Francisco and in Los Angeles, along light rail and heavy rail lines. The least dense station areas are not clustered in any specific area of the state. It is important to note that a station's population density ranking can change, sometimes significantly, based on the catchment area. Population density is an important consideration in "first and last-mile" planning because stations with higher surrounding population densities have a greater number of people that can access transit through non-motorized modes, thus increasing the importance of "first and last-mile" planning. A python script for calculating population density can be found in Appendix 2.

Job Density

Job density is the measure of the number of jobs that are located within a certain geographic area. Data was retrieved from the 2017 Census Longitudinal Employer-Household Dynamics Program (LEHD). Since job numbers are provided as geographic points, the number of jobs per point were summed if the point fell within the specified station catchment area. Figure 3.3 shows job per square mile surrounding Union Station in Los Angeles, aggregated to Census block groups. There is a high concentration of jobs to the west of the station in downtown Los Angeles. Table 3.5 shows total jobs within various catchment areas of the Los Angeles Union Station.

Figure 3.3: Job Density near Union Station, Los Angeles



Data Source: LEHD OnTheMap Tool, 2017 Data. Map by Henry McKay

Table 3.5: Total Jobs near Union Station, Los Angeles

Job Density	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
# of Jobs	3,616	51,725	179,808	372,934	460,088

Data Source: LEHD OnTheMap Tool, 2017 Data

As was the case with population density, job density is greatest in places like San Francisco and Los Angeles, and least dense in various places throughout the state. Job density is an important factor in "first and last-mile" planning because it represents the number of jobs that can be reached from a rail station using non-motorized modes of transportation. A python script for calculating job density can be found in Appendix 2.

Environmental Justice

The California Office of Environmental Health Hazard Assessment developed a tool called CalEnviroScreen, which produces environmental justice metrics at the Census Tract level. These metrics have many data inputs and are ultimately aimed at quantifying the environmental burden faced by various communities. Though not directly linked to pedestrian or bicycle accessibility, these metrics can (and should) play a role in the station area analysis process. For example, it may make sense to prioritize active transportation projects in environmentally disadvantaged areas. CalEnviroScreen scores can help determine where these areas are located, and what specific environmental burdens they face. The specific data inputs that make up a CalEnviroScreen score are the following:

Pollution Burden

Exposure

- Ozone Concentrations
- PM2.5 Concentrations
- Diesel PM Emissions
- Drinking Water Containment
- Pesticide Use
- Toxic Releases from Facilities
- Traffic Density

Environmental Effects

- Cleanup Sites
- Groundwater Threats
- Hazardous Waste
- Impaired Water Bodies
- Solid Waste Sites and Facilities

Population Characteristics

Sensitive Populations

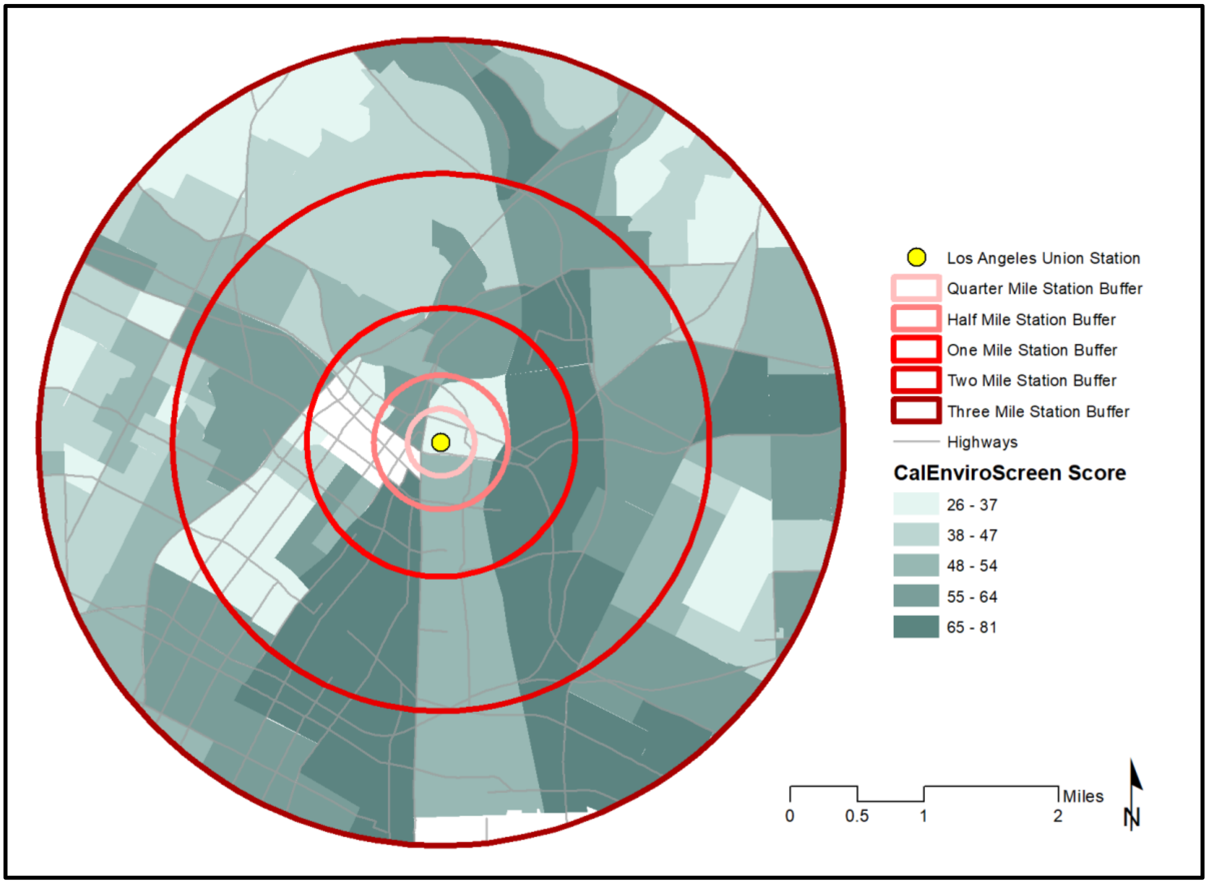
- Asthma Emergency Department Visits
- Cardiovascular Disease (Emergency Department visits for Heart Attacks)
- Low Birth-Weight Infants

Socioeconomic Factors

- Educational Attainment
- Housing Burdened Low Income Households
- Linguistic Isolation
- Poverty
- Unemployment

Figure 3.4 shows CalEnviroScreen scores within a three-mile catchment area of Los Angeles Union Station. Darker shaded census tracts represent higher scores, which correspond to more environmentally burdened areas.

Figure 3.4: CalEnviroScreen for the Los Angeles Union Station Area



Data Source: CalEnviroScreen, 2020. Map by Henry McKay

To calculate CalEnviroScreen scores for station catchment areas, weighted averages were used. An average was taken of the scores for each census tract that fell within the catchment area, with the scores being weighted by the proportion of the total catchment area represented by the respective census tract. Table 3.6 shows the weighted CalEnviroScreen scores for each catchment area for Union Station. A python script for calculating weighted CalEnviroScreen scores can be found in Appendix 2.

Table 3.6: CalEnviroScreen Score for Los Angeles Union Station Area

CalEnviroScreen	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
Weighted Avg. Score	36	39	49	55	53

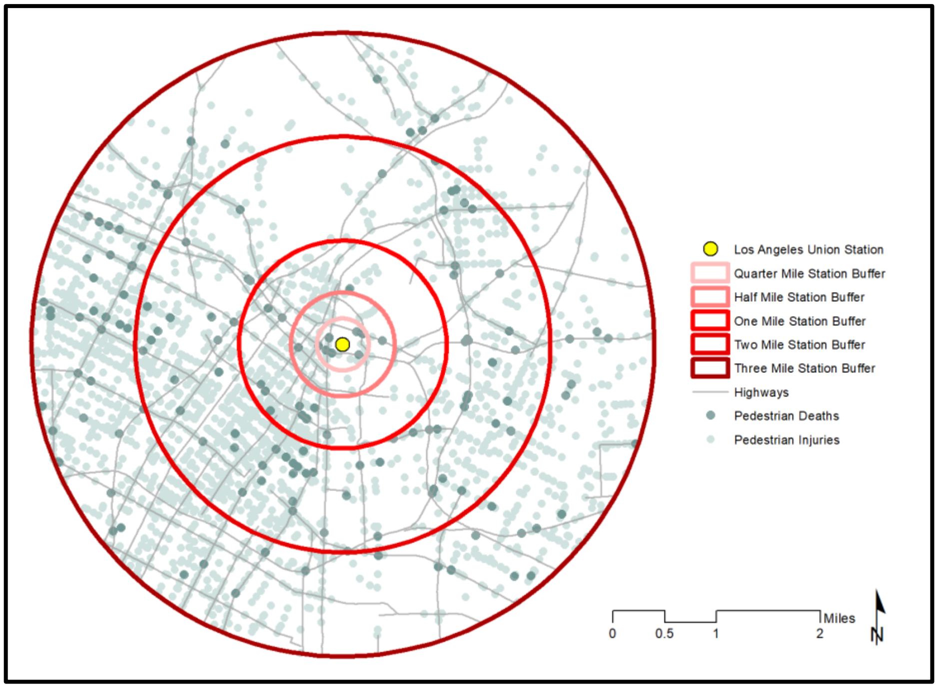
Source: CalEnviroScreen, 2020

Safety

The California Highway Patrol (CHP) maintains a database of traffic safety incidents called the Statewide Integrated Traffic Records System (SWITRS). The database was filtered to only contain incidents involving pedestrians and bicyclists and was mapped to rail station areas. More detailed technical information on how safety data was retrieved can be found in Appendix 2. Pedestrian and Bicycle safety are very important factors in "first and last-mile" planning, as they indicate how safe a station area is for Pedestrians and Bicyclists. However, analyzing safety data is more complicated than measuring other types of data. The raw count or percentile of safety incidents within a station area is not directly comparable across all station areas. For example, stations in highly populated activity centers will always have more safety incidents, mainly due to the fact that there is a much higher rate of pedestrian and bicycle activity than in less urban areas. With that in mind, a more appropriate measure would be the number of incidents divided by a measurement of activity to create a standardized rate. This was not done in this project, although the safety data is still available to explore in the analysis and toolkit. This consideration should be factored in when including safety data in a metric.

Figures 3.5 and 3.6 show the spatial distribution of pedestrian and bicyclist safety incidents respectively around Union Station between 2009 and 2019. The information is broken down into injuries and deaths.

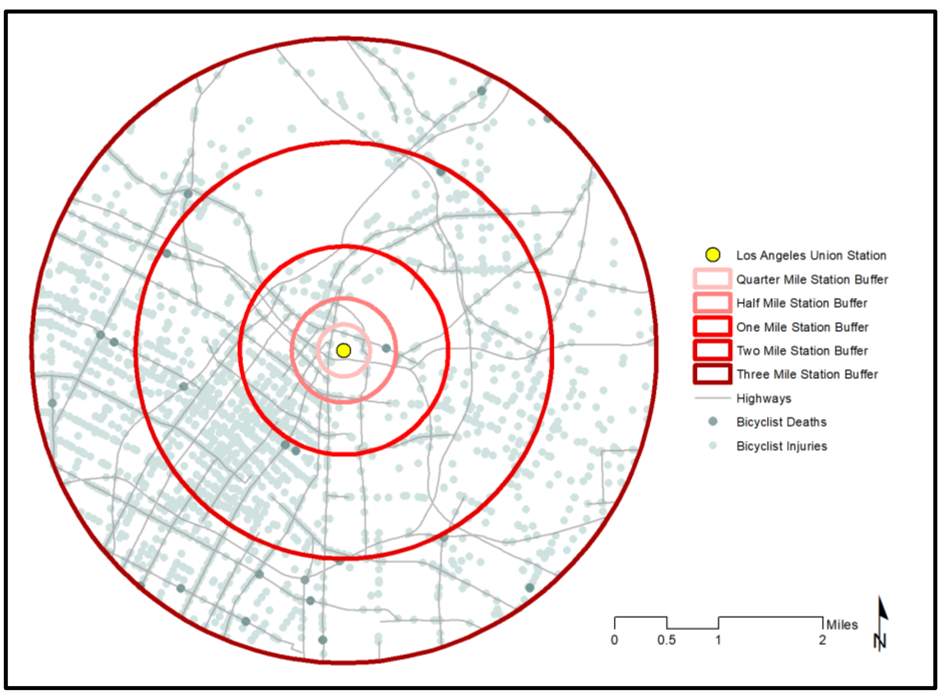
Figure 3.5: Pedestrian Safety near Union Station, Los Angeles



Data Source: Statewide Integrated Traffic Records System, 2009 – 2019

Map by Henry McKay

Figure 3.6: Bicycle Safety near Union Station, Los Angeles



Data Source: Statewide Integrated Traffic Records System, 2009 – 2019

Map by Henry McKay

Table 3.7 shows the cumulative counts of pedestrian and bicycle safety incidents for each catchment area around Union Station. If the incident fell within the catchment area, it was counted.

Table 3.7: Safety

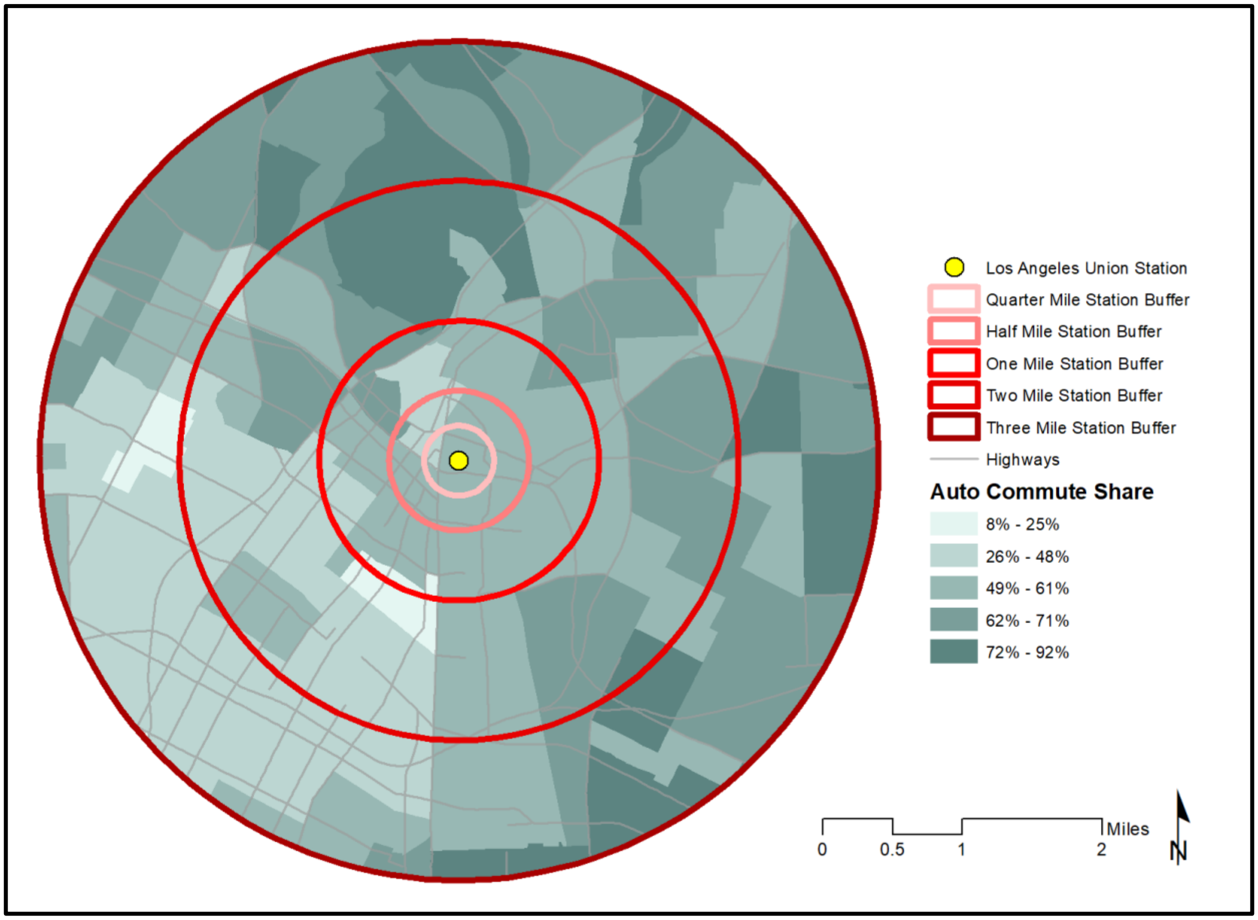
Safety	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
Ped Injuries	44	192	746	2,834	5,256
Ped Deaths	9	11	25	94	161
Bike Injuries	26	117	442	1,972	3,737
Bike Deaths	-	1	1	6	22

Data Source: Statewide Integrated Traffic Records System

Commuter Mode Split

The U.S. Census Bureau records a number of statistics on commute behavior including mode, travel time, and time of departure. Mode is of particular interest as it gives a good indication of how many people actually walk, bike, or take transit to work in a given Census Tract. San Francisco station areas overwhelmingly had the highest share of walk commuters. High bike-commute share station areas were more evenly distributed throughout the state, with Davis being the highest. San Francisco station areas also tended to have the highest transit commute shares. While most station areas had a fairly-high auto share, station areas along the Sprinter line in Northern San Diego County had the highest. Figures 5.7 through 5.10 show the commute mode splits for Auto, Transit, Pedestrian, and Bicycle for Union Station.

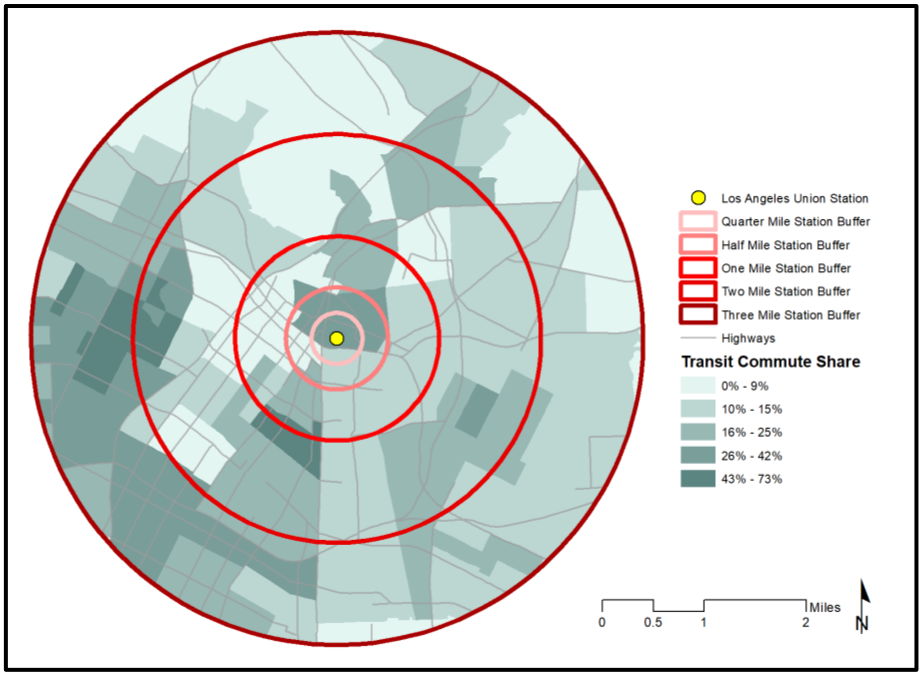
Figure 3.7: Auto Commute Shares for Union Station, Los Angeles



Data Source: American Community Survey 2018 5-Year Estimates, Table B08101e9

Map by Henry McKay

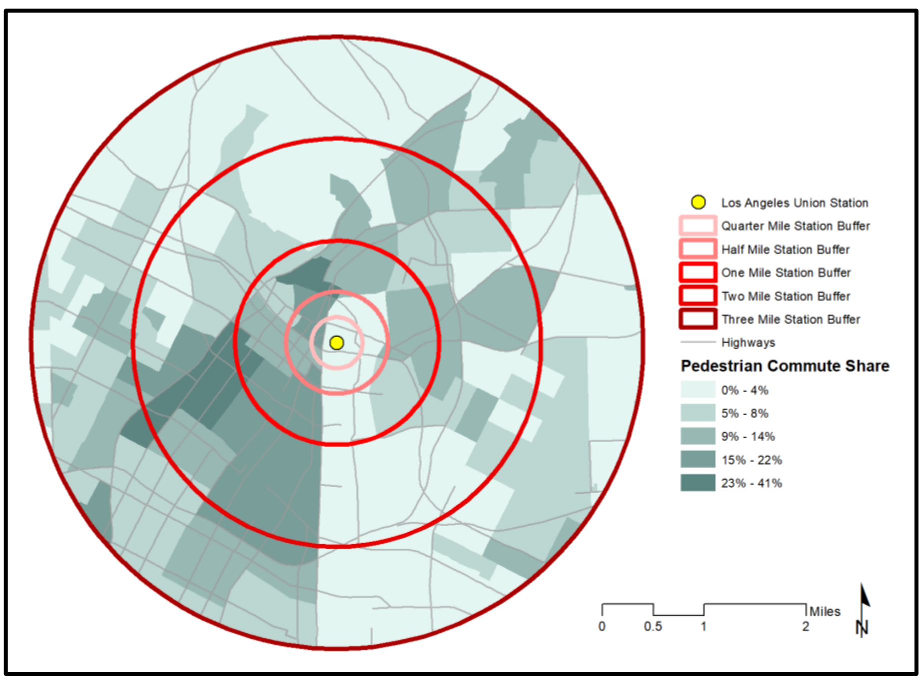
Figure 3.8: Transit Commute Shares for Union Station, Los Angeles



Data Source: American Community Survey 2018 5-Year Estimates, Table B08101e25

Map by Henry McKay

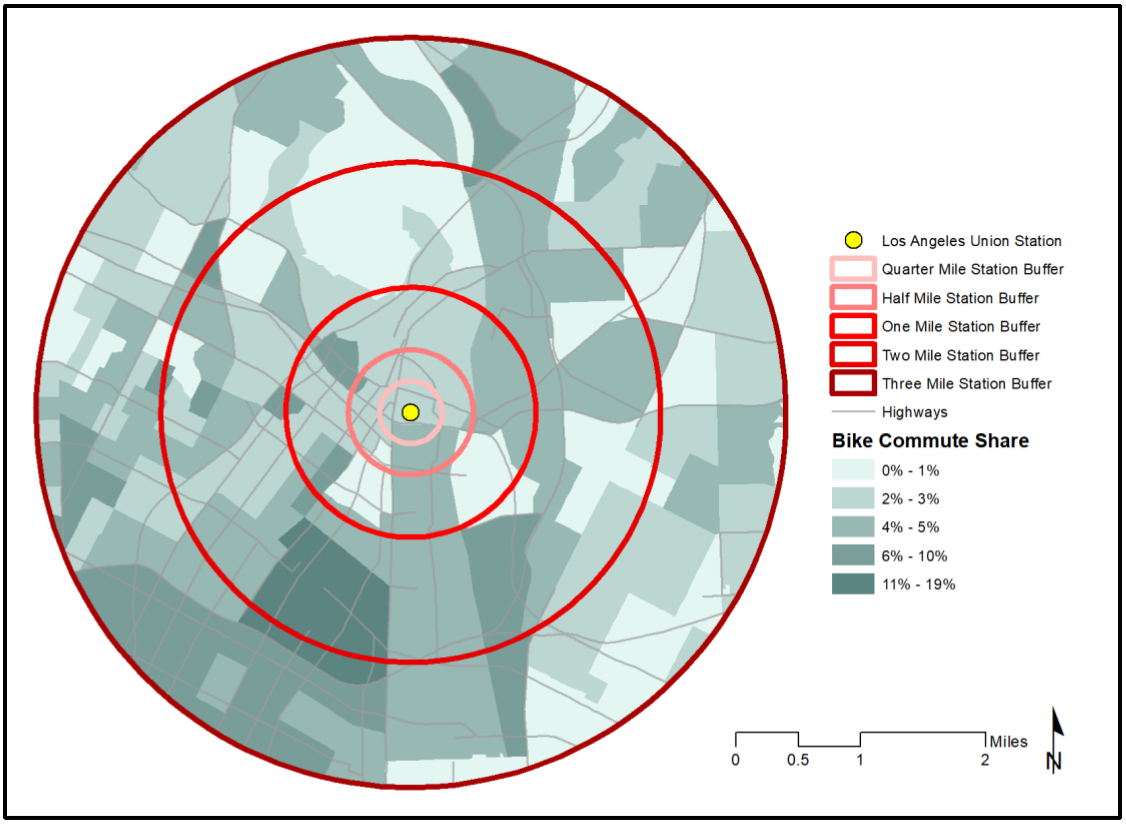
Figure 3.9: Pedestrian Commute Shares for Union Station, Los Angeles



Data Source: American Community Survey 2018 5-Year Estimates, Table B08101e33

Map by Henry McKay

Figure 3.10: Bicycle Commute Shares for Union Station, Los Angeles



Data Source: American Community Survey 2018 5-Year Estimates, Table B08101e41

Map by Henry McKay

Table 3.8 shows the commute mode splits for each catchment area of Union Station. To create mode split proportions for each catchment area, weighted averages were calculated using the same methodology previously described to calculate weighted CalEnviroScreen scores. A python script to calculate commute mode splits can be found in Appendix 2.

Table 3.8: Commute Mode Shares for Union Station, Los Angeles

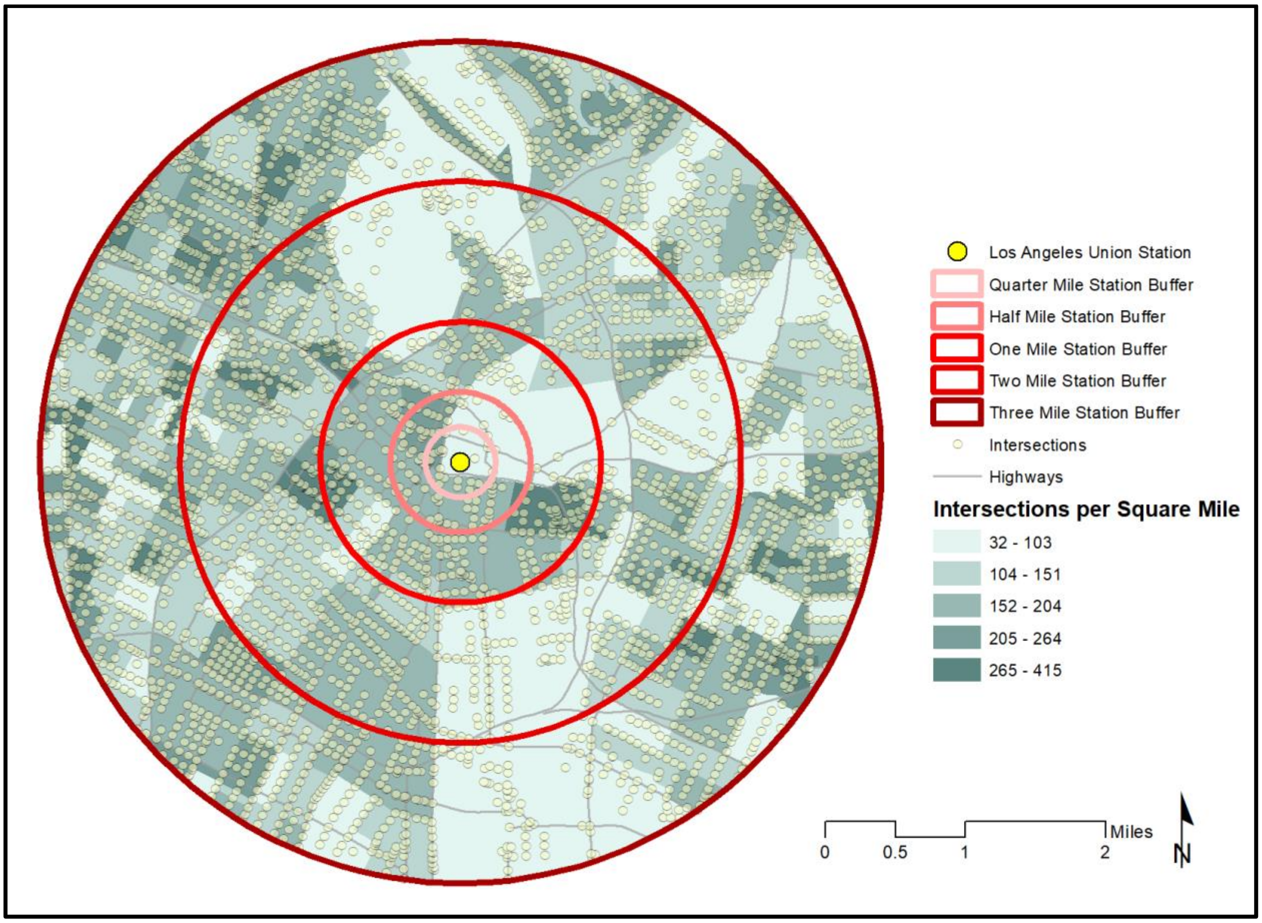
Commute Mode Share	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
Auto %	54.6%	54.0%	54.8%	56.4%	58.4%
Transit %	23.3%	20.0%	15.4%	14.8%	15.5%
Pedestrian %	3.4%	6.4%	10.2%	9.0%	6.7%
Bicycle %	2.7%	2.6%	2.6%	3.4%	3.2%

Source: US Census Bureau

Intersection Density

Intersection density is the count of intersections within a given station catchment area. Since an intersection is a place where multiple links of a network meet, a higher intersection density means that an area is more connected. Intersection density is a good proxy measurement for accessibility. Figure 3.11 shows all intersections within a three-mile catchment area of Union Station. Darker-shaded census block groups indicate higher intersection densities (measured in intersections per square mile).

Figure 3.11: Intersection Density near Union Station, Los Angeles



Data Source: OpenStreetMap 2020, Map by Henry McKay

Table 3.9 shows the intersection count within each catchment area of Union Station. If the intersection was located within the catchment area, it was counted. Intersection density was calculated using input roadway network data from Open Street Map and two Python Scripts. Technical calculation methodology can be found in Appendix 2.

Table 3.9: Intersection Counts near Union Station, Los Angeles

Intersection Density	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
# of Intersections	30	118	464	1,757	4,074

Data Source: OpenStreetMap 2020

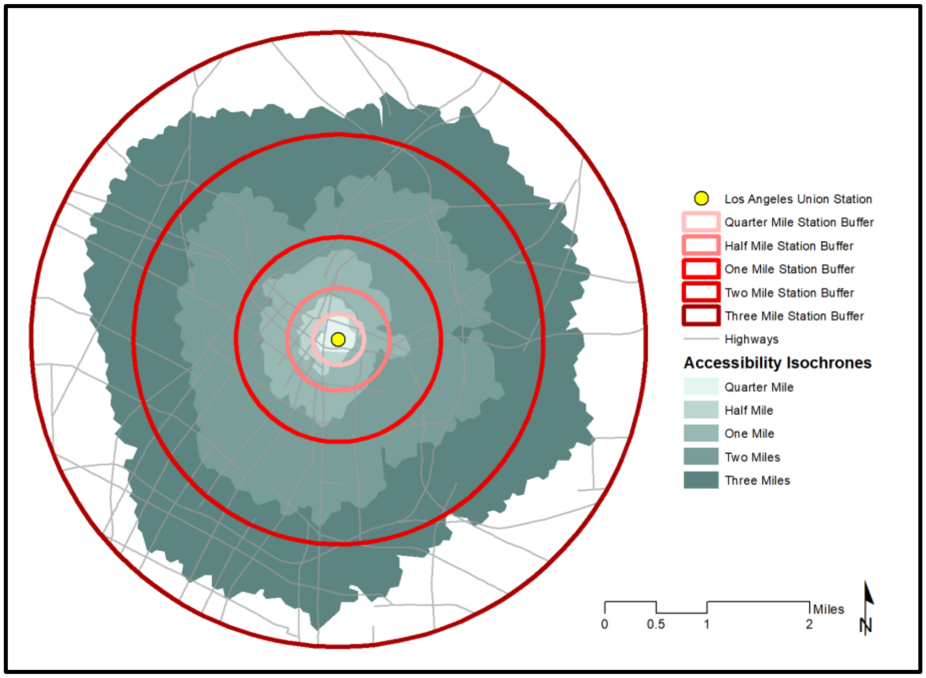
Accessibility Isochrones

Accessibility isochrones were generated for each rail station using ESRI's Network Analyst GIS extension. Accessibility isochrones are shapes representing the spatial area one could traverse by walking or biking, given the existing network of roadways and paths. Travel distance was used to calculate accessibility isochrones instead of travel time. However, the distances calculated directly correspond to travel times for both pedestrians and bicyclists. Table 3.10 shows the conversion between isochrone distance and both pedestrian and bicyclist travel time. For example, a 3 Mile accessibility isochrone is equivalent to a 60-minute walk and a 15-minute bicycle ride. Open Street Map (OSM) data was used to generate the network dataset and isochrones. Furthermore, the area of each isochrone was calculated and used as a quantitative measurement of pedestrian and bicycle sheds. Accessibility isochrones are perhaps the most important measurement of accessibility included in the analysis. Figure 3.12 shows the accessibility isochrones for each catchment area of Union Station. In every case, the accessibility isochrone will not reach as far as its corresponding catchment area radii, as it is constrained by a physical roadway network.

Table 3.10: Isochrone Distance to Travel Time Relation

Isochrone Distance	Avg. Walk Speed (MPH)	Avg. Bike Speed (MPH)	Walk Time (Min)	Bike Time (Min)
.25 Mile	3	12	5	1.25
.5 Mile	3	12	10	2.5
1 Mile	3	12	20	5
2 Mile	3	12	40	10
3 Mile	3	12	60	15

Figure 3.12: Accessibility Isochrones near Union Station, Los Angeles



Data Source: OpenStreetMap 2020, Map and Analysis by Henry McKay

Table 3.11 shows the area measurements (in square miles) of the accessibility isochrones for each catchment area. Technical information on accessibility isochrone calculation can be found in Appendix 3.

Table 3.11: Accessibility Isochrone Areas near Union Station, Los Angeles

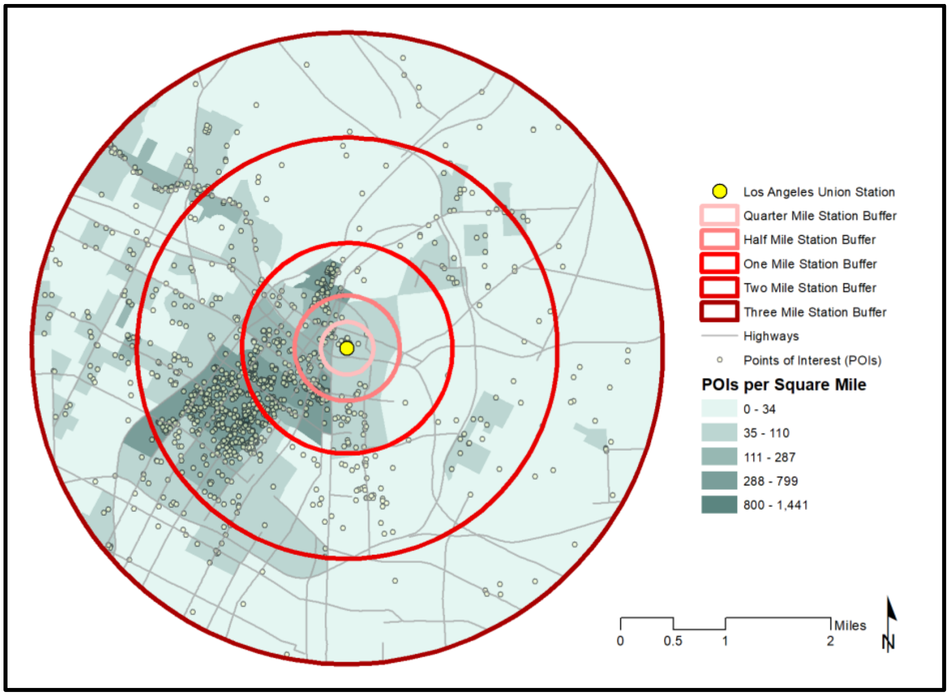
Accessibility Isochrone Area	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
Square Miles	0.1	0.3	1.6	7.9	19.3

Data Source: OpenStreetMap 2020

Points of Interest

Points of interest consist of amenities and include restaurants, banks, businesses, parks, and other places people would want to go. Open Street Maps (OSM) data was used to measure points of interest, as it was the only comprehensive database available. Points of interest are important in "first and last-mile" planning as they represent places that people could potentially reach by transit. Figure 3.13 shows point of interest within a three-mile catchment area of Union Station.

Figure 3.13: Points of Interest (POIs) near Union Station, Los Angeles



Data Source: OpenStreetMap 2020, Map by Henry McKay

Table 3.12 shows the count of points of interest within each catchment area of Union Station. A python script for calculating points of interest can be found in Appendix 2, and more information on included points of interest can be found in appendix 3.

Table 3.12: Counts of Points of Interest near Union Station, Los Angeles

Points of Interest (POIs)	Catchment Area				
	Quarter Mile	Half Mile	One Mile	Two Miles	Three Miles
Count	43	109	416	1,009	1,225

Source: OpenStreetMap

Secondary Data

Secondary data items were not included in the analysis or toolkit but are important to the future direction of this project. They have the potential to be highly useful in assessing the "first and last-mile" potential of California rail stations but were not used in this project. In most cases, this data is qualitative, and must be manually collected on site or through various internet sources. Given this constraint, it was not feasible to compile a comprehensive dataset with these pieces of data for every rail station in California. Future improvements of this toolkit will likely incorporate these data items, as they focus much more on the station itself rather than the station area, enabling more robust and comprehensive metrics to be developed.

Setting

While the type attribute described what type of services the station had, setting goes further and classifies every station into a spectrum of station typologies, relevant to California's rail network. Firstly, each station was placed into one of three categories; urban, suburban, or rural (also termed small) small community. These three designations follow the State Rail Plan guidance set forth by the FRA. Furthermore, additional classifications were added to the setting to further classify the station type. For example, the BART station at San Francisco International Airport (SFO) would be categorized as Urban-Airport. Park and Ride was another important classification and was applied to every station with a Park and Ride lot.

Service Frequency

For each station, the service frequency for every service was recorded. Service frequency is an important aspect of transit planning and indicates how often a particular service serves a station. Higher service frequencies make transit more attractive and convenient for riders. Service frequency may be further stratified as follows:

Peak-Hour/Peak-Direction:

These are relatively high frequency service in both directions during peak commuting hours. Primarily commuter rail services such as Caltrain and ACE.

Longer than Bi-Hourly:

These are fairly infrequent services throughout the day. Amtrak's long-distance services fall into this category.

Bi-Hourly:

Service that run throughout the day at roughly two-hour intervals. An example of this service frequency is Amtrak's Capitol Corridor between Oakland and San Jose.

Hourly:

Service that runs throughout the day at roughly one-hour intervals. An example of this service frequency is Amtrak's Capitol Corridor between Sacramento and Oakland.

Half-Hourly:

Service that runs throughout the day at roughly half-hour intervals. An example of this service frequency is NCTD's Sprinter, or some less-frequent light rail services.

Shorter than Half-Hourly:

All services that run throughout the day at less than half-hour intervals. These services mostly include light rail and heavy rail services such as BART of LA Metro.

Weekend:

Services that only regularly serve a station during the weekend. The only occurrence of this service type is along the Pacific Surfliner line.

Special:

Special services that only serve a station on special occasions, such as large sporting events. An example of this service type is the Stanford Station along the Caltrain corridor, which is used during large Stanford football games.

Vehicle Parking

Whether or not a station had vehicle parking as well as the approximate number of spaces was recorded. Certain rail services provide an online station parking inventory, with accurate parking counts. When an accurate online count was not available, Google Maps satellite imagery from 2020 was utilized to perform a manual count. In these cases, the count is approximate due to certain limitations such as poor imagery and tree cover but is fairly accurate.

Bicycle Parking

For certain stations, the number of bicycle parking spaces (both bike rack spaces and bike locker spaces) was recorded. Certain stations publish an online bicycle parking inventory, with exact counts of spaces at stations. For stations without an official inventory available, Google Maps Street View imagery from 2020 was used to determine an approximate manual count.

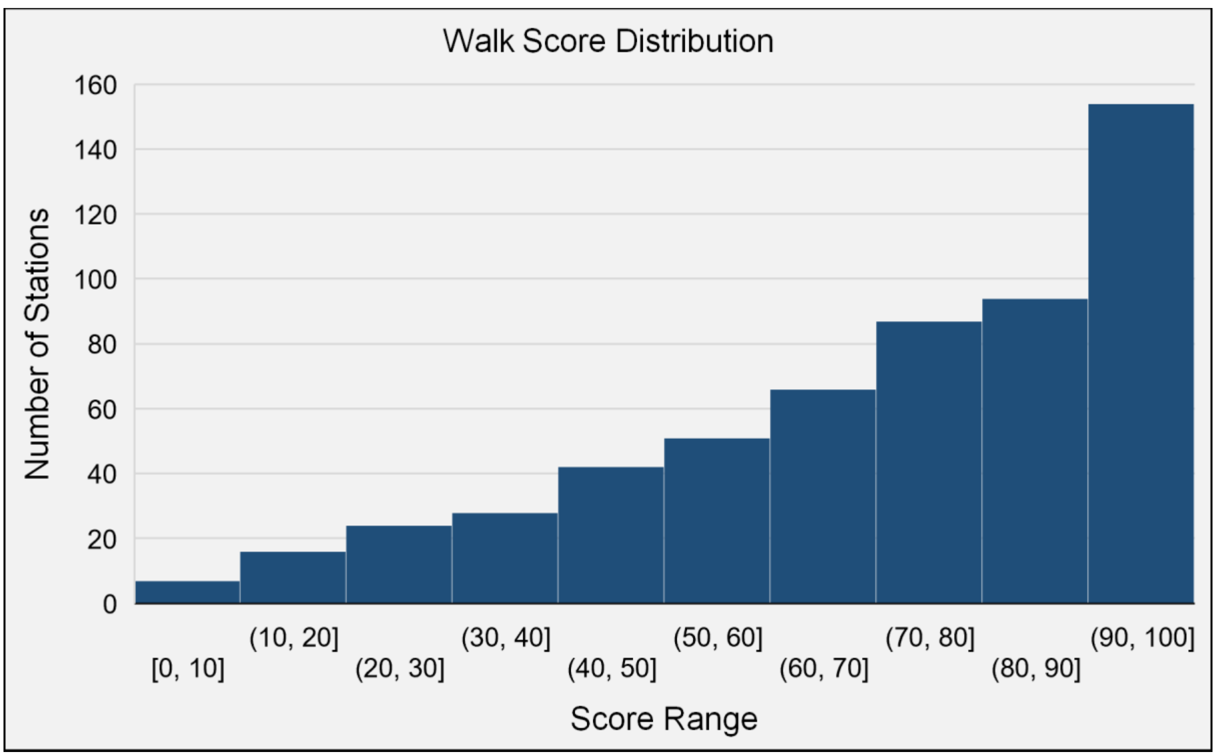
Transit Connectivity

For every station, multimodal transit connectivity was analyzed. Firstly, connection to other rail services was recorded. Many stations in the analysis have multiple services, meaning that they have at least one connection. Secondly, connections to Amtrak thruway bus service were recorded. Amtrak's thruway bus network serves as an essential extension to the state's rail network, filling service gaps and providing service to underserved areas. Third, connections to local bus service were recorded. At many stations, local bus service serves as an essential means of connecting cities to their rail transit stations. Lastly, planned connections to the California High Speed Rail system was analyzed.

4. Walk Score Analysis

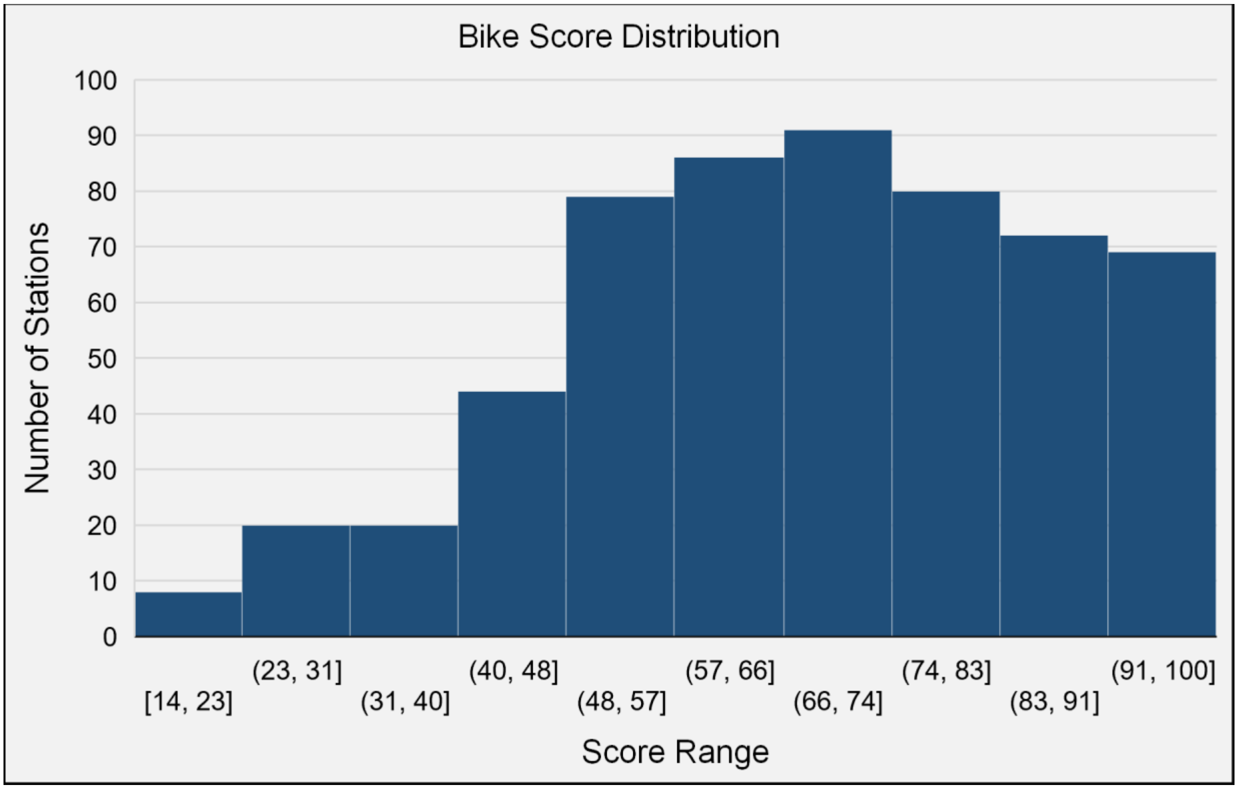
Walk Score is a well-known company that provides walkability, bikeability, and transit metrics for any point in the United States using a proprietary algorithm. While the service is sometimes used by planning researchers and practitioners, it was primarily developed to suit the needs of the real estate industry and is owned by the real estate website Redfin (Walk Score, 2021). Online real estate listings often include a walk score, bike score, and transit score, provided by WalkScore.com. Since the metrics developed by WalkScore.com are the most commonly used and are available for any geographic point, walk scores and bike scores were retrieved and analyzed in order to better inform the development of a new station area "first and last-mile" planning toolkit. Using Walk Score's API and a simple R script, walk scores and bike scores were retrieved for rail stations in California. Figures 4.1 and 4.2 the distributions of walk and bike stores, respectively. For technical information on the score retrieval process, refer to Section 7-2 and Appendix 2, Script #1.

Figure 4.1: Distribution of Walk Scores for Rail Stations in California



Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

Figure 4.2: Distribution of Bike Scores for Rail Stations in California



Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

Considering every rail station in California, the distribution of walk scores was highly left-skewed, meaning few stations have very low scores. There was a high concentration of stations with scores between 90 and 100 than there were stations with lower scores. This is fairly self-evident given the high number of stations in highly urbanized areas such as San Francisco that are very walkable. While slightly left-skewed, the distribution of bike scores was relatively normal in shape, with the most scores belonging to the (66, 74] range.

While these scores can be very useful and are commonly utilized in a wide variety of situations, they have certain limitations, especially when being used to quantify specific problems such as rail station accessibility. The largest issue with WalkScore.com scores is transparency. Though explained in broad terms on Walk Score's website, the company does not disclose exactly how the scores are calculated or provide example calculations with data. This is understandable since Walk Score is a for-profit company, and the scores are their product. Furthermore, there is no method for the user to adjust the scores or change the weight that various types of input data receive in the final score. Given what is publicly known about the scores and the insights that various researchers have uncovered, it is evident that Walk Score's

walk and bike scores heavily weight the number of surrounding amenities, such as restaurants, stores, parks, and other places that people would want to go. While this approach is a good one and is supported by both planning research and practice, accessibility can be looked at in other ways, especially when rail station access and egress is being analyzed. More importantly, it is key that analysts have the ability to determine which variables go into accessibility scores and how each variable is weighted in order to best quantify unique planning challenges.

In order to better understand Walk Score's walk and bike scores in the context of the station area data collected for this project, exploratory data analysis was conducted to determine which variables were most useful in predicting Walk Score's walk and bike scores and how much explanation in score variability each variable provided. Using a stepwise model selection algorithm, all available variables were iteratively added or subtracted with the walk score as the response variable until maximum explanatory power (as measured in R squared) was reached. The resulting model summary was used to break down the exact variation in walk and bike scores that each variable could explain. The input data consisted of all collected station area data, as well as adjusted percentile versions of each variable.

Walk Scores

Using the stepwise regression model's coefficient estimates, walk scores were predicted and plotted against the walk scores obtained from WalkScore.com using the collected station area data, as shown in Figure 4.4. The model was able to explain approximately 80% of the variation in Walk Score walk scores, which is fairly good and offers valuable insights into what variables most influence Walk Score's metrics. However, it is important to note that the station area data used violates multiple regression assumptions, meaning that while useful in breaking down walk scores in the context of other data, the model itself is fairly-crude and would not be suitable for predictive purposes. Figure 4.3 shows the complete regression equation that was used to predict walk scores.

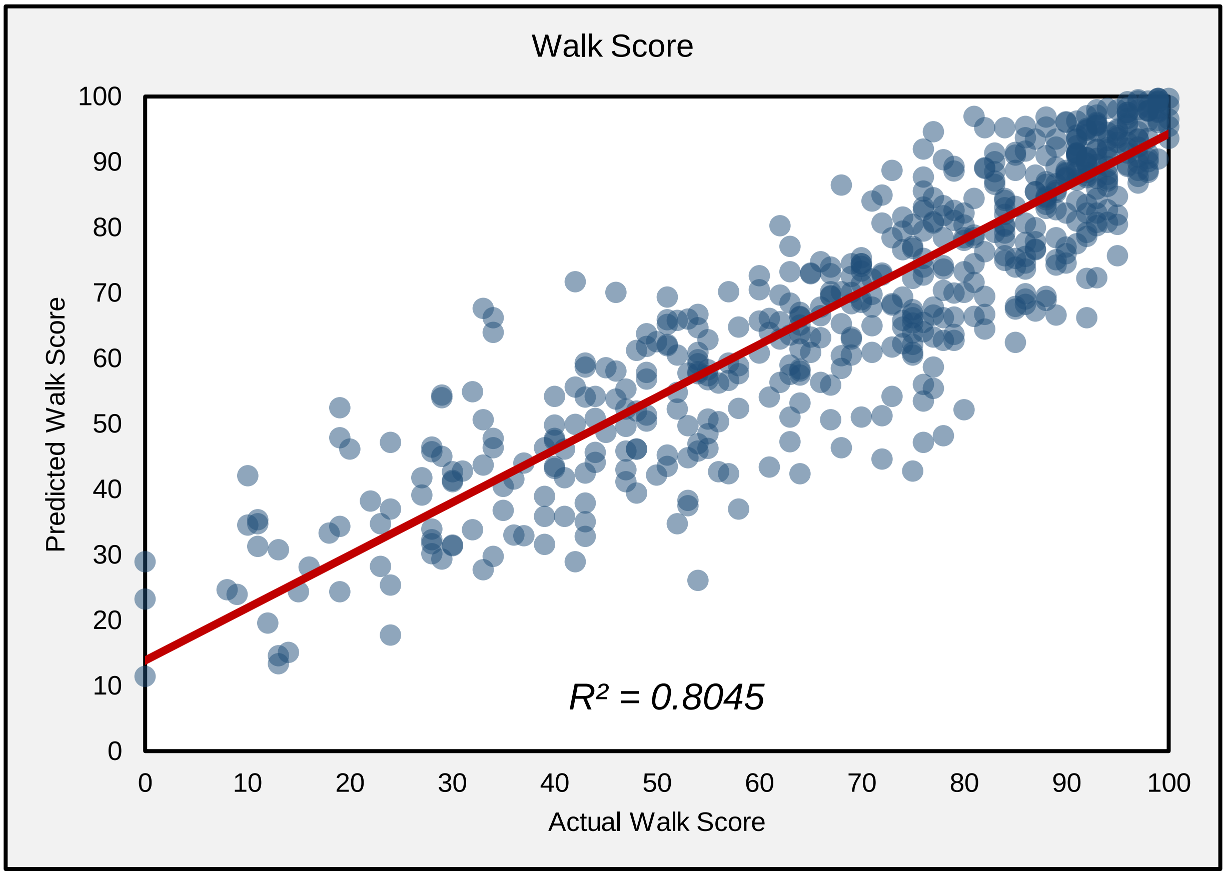
Points of Interest (with a percentile adjustment) explain the most variation in walk scores at 52%. The station area accessibility isochrone value at one mile also explains a decent amount of variation at 15%. Meanwhile other variables such as population density explain under 5% of variation each, although they do add up and explain 13% of variation, which is still fairly insignificant. Overall, it was evident that surrounding points of interest within a half-mile radius had the most influence over variation in walk scores. A complete breakdown of model variables and their corresponding percent of variation explained is shown in Figure 4.6.

Figure 4.3: Walk Score Regression Equation

Predicted Walk Score = *8.745 + 29.37(Points of Interest Percentile - .5 Mile) + 14.17(Accessibility Isochrone Area - 1 Mile) + 17.32(Population Density Percentile - .5 Mile) - 18.26(Points of Interest Percentile - 3 Mile) + 8.816(Walk Commute Percentile - 3 Mile) - .06677(Shared Bicycle Facility Mileage - 2 Mile) + 13.81(Points of Interest Percentile - .25 Mile) - .0266(Points of Interest - .5 Mile) + 5.388(Jobs Percentile - .25 Mile) - 22.59(Transit Commute Percentile - 3 Mile) + 47.67(Transit Commute Proportion - 3 Mile) + .1955(CalEnviroScreen Score - .25 Mile) - 9.35(CalEnviroScreen Percentile - 2 Mile) - 48.58(Transit Commute Proportion - .5 Mile) + 9.647(Intersection Density Percentile - 2 Mile) + 28.22(Transit Commute Percentile - .5 Mile) + 17.38(Intersection Density Percentile - .25 Mile) + .00002594(Jobs - 3 Mile) - 8.887(Jobs Percentile - 1 Mile) - .2159(Intersection Density - .25 Mile) - .00003762(Population Density - 3 Mile) + 12.57(Population Density Percentile - 3 Mile) - 3.185(Shared Bicycle Facility Mileage Percentile - .25 Mile) + .08067(Shared Bicycle Facility Mileage - 1 Mile) + .05817(Intersection Density - .5 Mile)*

Data Source: R Output (Appendix 2, Script #3)

Figure 4.4: Predicted Walk Score vs. WalkScore.com's Scores



Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

Bike Scores

Similarly, regression analysis was performed for bike scores. The explained variability in bike scores was broken down in terms of the collected station area data. As was the case with walk scores, surrounding Points of Interest (with a percentile adjustment) explained the most variation, but to a lesser extent at only 40%. Total mileage of surrounding bicycle facilities within a half-mile radius (with a percentile adjustment) was also fairly-significant at 20%, and station area accessibility isochrones at one mile explained six percent. All other variables were fairly-insignificant.

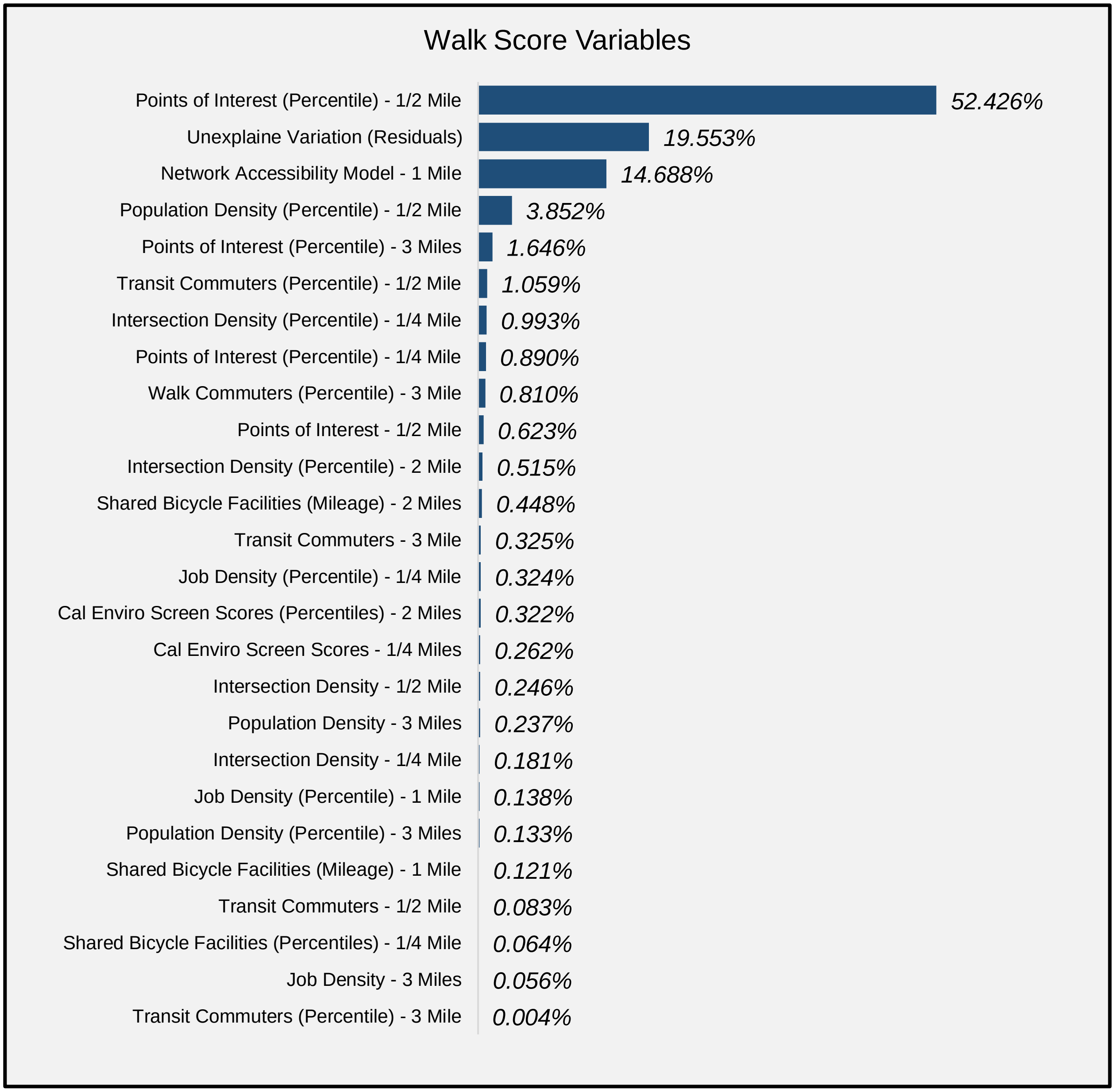
The bike score model had similar predictive power to the walk score model but was slightly less effective with an R-squared value of 78% (compared to 80% for walk scores). A scatter plot showing predicted vs. observed bike scores can be found in Figure 4.7, and a breakdown of model variables explaining bike score variability can be found in Figure 4.8. Figure 4.5 shows the complete regression equation used to predict bike scores.

Figure 4.5: Bike Score Regression Equation

Predicted Bike Score = *23.56 + 6.299(Bike Commute Percentile - 1 Mile) + 13.4(Points of Interest Percentile - .5 Mile) + 14.3(All Bicycle Facilities Mileage Percentiles - .5 Mile) + 8.448(Accessibility Isochrone Area - 1 Mile) + 9.577(Bicycle Commute Proportion - .5 Mile) + .2191(CalEnviroScreen Score - .25 Mile) + 15.45(Shared Bicycle Facility Mileage Percentile - .25 Mile) + 5.008(Exclusive Bicycle Facility Mileage Percentile - 2 Mile) - 1.43(Shared Bicycle Facility Mileage - .25 Mile) - .00001103(Population Density - 3 Mile) + 2.691(Exclusive Bicycle Facility Mileage Percentile - .25 Mile) + 9.753(Intersection Density Percentile - 1 Mile) + .00004113(Job Density - 2 Mile) - .005447(Accessibility Isochrones Area - 1 Mile) + 12.83(Transit Commute Percentile - .25 Mile) - 46.01(Transit Commute Proportion - 1 Mile) - 5.665(Job Density Percentile - 1 Mile) - 7.253(Auto Commute Prop - .25 Mile) - .00008206(Job Density - .25 Mile) - 3.787(Population Density Percentile - .25 Mile) - .1445(CalEnviroScreen Score - 1 Mile)*

Data Source: R Output (Appendix 2, Script #3)

Figure 4.6: Walk Score Variable Breakdown

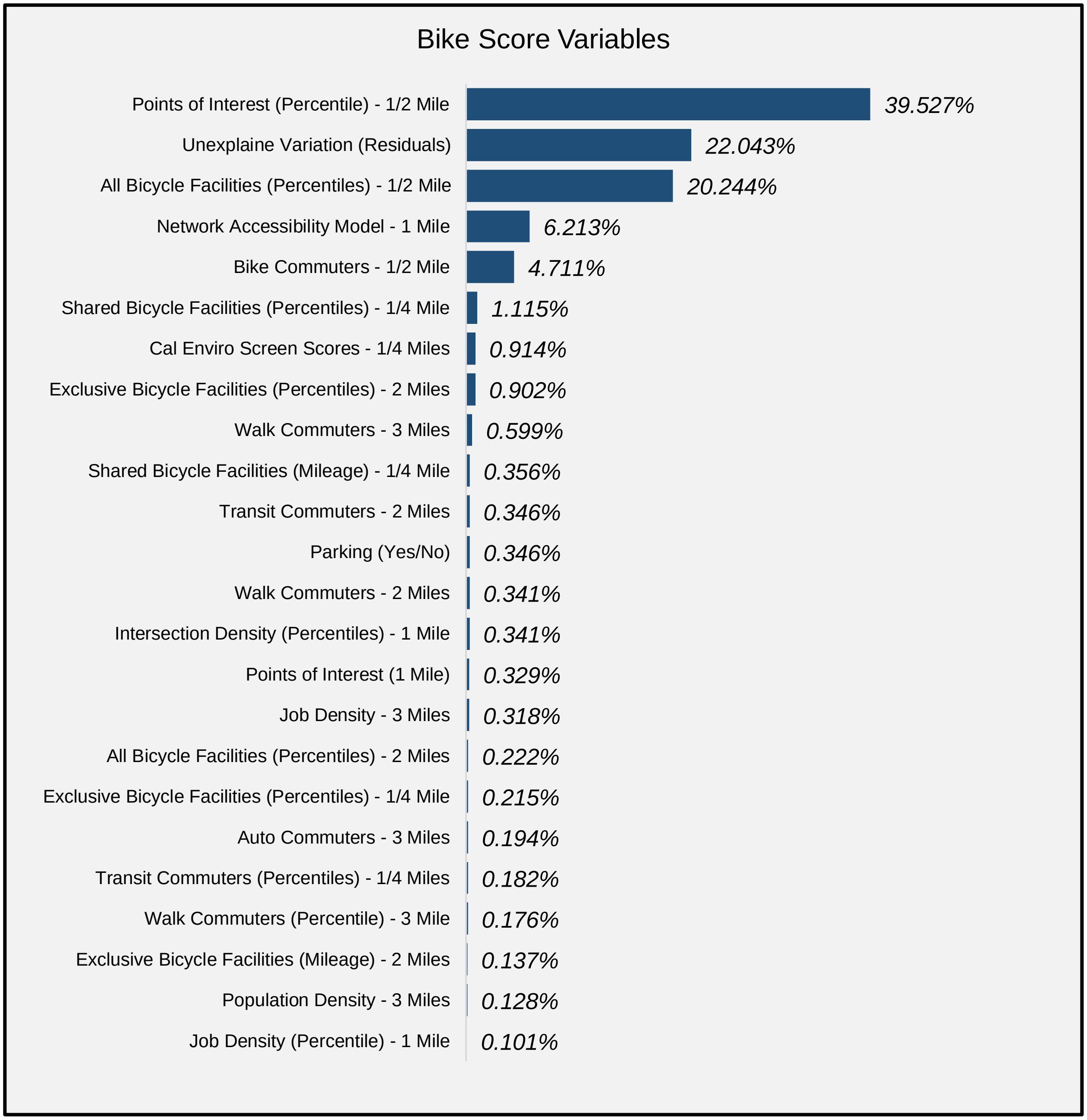


Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

Figure 4.7: Predicted Bike Score vs. Bike Score

Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

Figure 4.8: Bike Score Variable Breakdown



Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

For each rail service in California, the mean walk and bike scores were calculated from every station score along the service's route and are shown in Table 4.1. The highest average walk scores were generally found in the Bay Area with MUNI Metro Light Rail being the highest scoring service, while the lowest average scores belonged to Metrolink in Southern California. This is unsurprising since MUNI

serves a dense urban area and Metrolink serves mostly suburban and even rural areas of the Los Angeles metropolitan area.

Table 4.1: Mean Walk and Bike Scores

	Mean Score	
Service	Walk Score	Bike Score
Muni Streetcar	95.66	87.23
Muni Metro	89.32	78.11
Caltrain	79.55	81.03
LA Metro	78.81	66.14
BART	76.65	71.02
Amtrak Long Distance	74.12	65.56
Capitol Corridor	73.11	71.89
San Joaquin	71.78	63.22
Coaster	70.63	70.88
MTS	70.17	55.76
Pacific Surfliner	66.93	64.22
SacRT	63.30	76.47
ACE	61.80	63.30
SMART	60.83	57.50
VTA	58.65	74.98
Sprinter	57.40	45.33
Metrolink	55.49	52.14

Data Source: WalkScore.com, Graphic and Analysis by Henry McKay

Bike scores were also aggregated in this manner, with fairly different results. While Muni Metro Light Rail was still the highest scoring service on average, several services with lower average walk scores had far higher relative bike scores. For example, VTA Light Rail had a low average walk score of 59, but an average bike score of 75. Anecdotally, VTA has made major investments in bicycle infrastructure in recent years and it shows in the scores (Valley Transportation Authority, 2018). However, VTA serves the Santa Clara valley, which is very suburban helping to explain why its mean walk score was relatively low.

5. Metric Development

The methodology developed to create pedestrian and bicycle accessibility metrics for rail station areas in California is quite simple and is based on a number of other fairly standard methodologies for performing similar types of analysis. Figure 5.1 is a flowchart showing how metrics are assembled.

Input Data

The input data for the metric development methodology consists of tabular data, with a column for all five catchment area measurements for each variable and a row for each rail station.

Select Services

Though this step can be performed at any point in the process without affecting the final scores, it is important to select the rail services that will be included in the analysis. Doing so filters the data set to only include rail stations served by the selected services.

Select Area of Analysis

One of the primary assumptions of the methodology is that for a given analysis, all variables are measured within the same catchment area. Selecting an appropriate area of analysis filters the data set to only include measurements from the selected catchment area.

Add Variables

Variables are selected to be included in the metric itself. At a minimum, one variable must by chosen and the metric can include as many variables as are available.

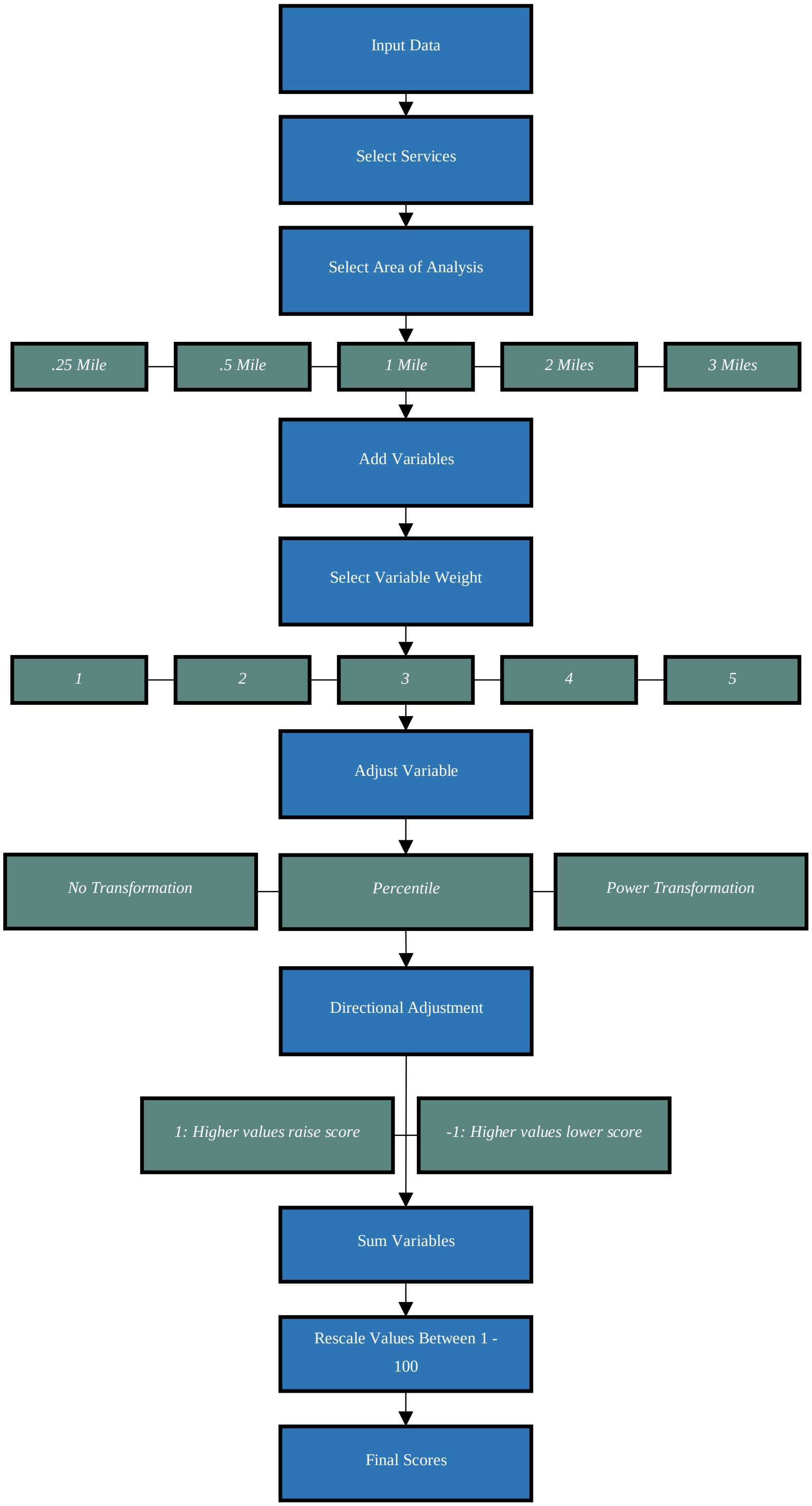
Select Variable Weight

Variable weights between one and five are selected for each included variable to determine how much influence the variable has over the final metric. This is done for each variable by taking a proportion of the variable weight to the sum of all variable weights in the metric.

Adjust Variable

Since the raw input data is often extremely skewed, a metric distribution using only raw input data would also be very skewed. If this is the case, a percentile or a power adjustment can be used. A percentile adjustment simply creates a percentile value for each data point. For example, a percentile value of 0.8 would mean that the data point is greater than 80% of the data in its range. This is an ordinal measurement. A power adjustment raises the data point to the power of an inputted value between 0 and 1, effectively flattening the higher values in the range.

Figure 5.1: Metric Development Methodology



Graphic by Henry McKay

Directional Adjustment

In certain cases, it may be appropriate to subtract a variable's value from a score as opposed to add to it. This could apply to safety data, when a higher value of pedestrian deaths would theoretically work against a higher pedestrian accessibility score. If this directional adjustment is chosen, the value is simply made negative so that it subtracts from the final score instead of adding to it.

Sum Variables

Once all variables have been weighted and adjusted, the values are added together to create a sum.

Rescale

The range of final sums for each station is rescaled between 0 and 100 in order to create a consistent set of scores across the analyses. Furthermore, it is possible to create a negative score if highly-weighted variables are given directional adjustments. Rescaling these values fixes this issue.

Final Scores

These rescaled values represent the final scores for each station. In every case, the scores range from 0 to 100, with a distribution highly dependent on the input data and adjustments chosen.

One key way in which this methodology is different from others is that it does not prescribe a certain set of variable weights. Instead, it allows users to choose their own weights using a spreadsheet-based tool that is discussed in the next chapter. Determining appropriate variable weights could be its own entire project and is fairly subjective depending on the situation. In the previous section WalkScore.com metrics were analyzed using statistical methods and it was determined how much each variable affected walk and bike scores. If 'standard' variable weights are preferred, using the results of the walk score analysis to determine variable weights in the metric would be a good approach.

Lastly, the previously discussed methodology differs from other methodologies in that it relies on pre-aggregated input data. More sophisticated metrics utilize computationally intensive methods to achieve more nuanced scores, while the methodology in this project relies primarily on aggregated and weighted counts. For example, the project's methodology simply counts the points of interest within a certain catchment area and uses that raw number as the input data for each station. In Walk Score's methodology, the walking and biking distances between the selected point and each point of interest are calculated using a routing algorithm and points are assigned based on travel time using a decay function, leading to an aggregated value for points of interest (Walk Score, 2021).

6. Toolkit

One of the primary purposes for this project – in addition to developing a methodology to create station area pedestrian and bicycle accessibility metrics – was to develop a toolkit to assist the user in creating and analyzing the metrics themselves, as well as in retrieving and cleaning the necessary input data. The theory and methodology behind the metrics are not highly complex, nor difficult to grasp. The same is true for the input data, which is simply a measurements of various data points of interest for multiple station catchment areas. However, transforming the various pieces of data from their original form to the necessary tabular structure can be fairly-challenging, and very time consuming. Furthermore, applying the metric methodology to the data is also fairly-difficult without decent knowledge of spreadsheets and GIS. To make this process easier, multiple scripts were written to semi-automate the data retrieval and cleaning process. Additionally, a Google Sheets-based tool was developed to allow the user to easily explore the data, build their own metrics, and analyze results easily. Lastly, an ArcGIS story map was developed to explore the data visually for the entire state network.

Scripts

For this project, multiple Python and R scripts were written to generate the necessary input data for the Google Sheets tool. Completing this work with code as opposed to manually completing it in GIS has many advantages. First, it allows individuals with limited GIS proficiency to gather the necessary data to perform the analysis. Though none of the GIS tasks are particularly advanced, they are fairly tedious and utilize techniques not covered in most introductory GIS education. Secondly, manually retrieving and cleaning the data would be very time consuming and create a large amount of intermediate data that can be cumbersome to manage. Scripting automates this process, greatly reducing the time required to perform the analysis and deleting all intermediary data once it has been used. Furthermore, some datasets are so large that they crash programs such as Microsoft Excel or ArcGIS. Tools such as Python and R can easily handle data sets of this size. Figure 6.1 is a sample Python script.

Figure 6.1: Python Script

```
import arcpy
import os
import numpy as np

# Define analysis variables
arr = np.array(["10 Mile"])
CensusData = "X01_AGE_AND_SEX"
CensusTableField = "B01003e1"
POINT = "CA_Stations"

# Defines file paths and directory
arcpy.env.workspace = "C:\Users\henrymckay\Desktop\GIS\StationArea.gdb"
myGDB = "C:\Users\henrymckay\Desktop\GIS\StationArea.gdb"
TableOut = "C:\Users\henrymckay\Desktop\GIS\Tables"
CensusPath = "C:\Users\henrymckay\Desktop\GIS\ACS\ACS_2018_5YR_BG_06_CALIFORNIA.gdb"
FileName = "Population_Density_v2.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Defines shapefile for TIGER Geometry
TIGER = os.path.join(CensusPath, "ACS_2018_5YR_BG_06_CALIFORNIA")

# Creates a copy of the TIGER Geometry
arcpy.CopyFeatures_management(TIGER, "in_memory/TIGER_Temp")

# Joins specified table to TIGER Geometry
DATETABLE = os.path.join(CensusPath, CensusData)
arcpy.JoinField_management("in_memory/TIGER_Temp", "GEOID_Data", DATETABLE, "GEOID", ["GEOID_Data", CensusTableField])
```

Source: Henry McKay

Python was utilized to prepare most of the geospatial input data. Though separate scripts were prepared for each variable with slight differences between them, the overall structure, functionality, and output of all the python scripts was nearly the same. First, the script created a geospatial layer for whatever was being measured, say population density for example. Next, the script utilized a for loop to measure that variable around each station point, at multiple catchment areas. Lastly, the script joined the data together, creating a simple spreadsheet storing the data in tabular form. In practice, this process is more complex and much of the code is dedicated to tasks such as creating weighted averages and performing other adjustments. For all python scripts and more detailed code instructions, see Appendix 2.

In addition to Python, the programming language R was also utilized to perform two key tasks. Firstly, a simple R script was written to retrieve pedestrian and bicycle data from the Statewide Integrated Travel Records System (SWITRS). Since the data came in 58 CSV files – one for each CA county – the R script was able to combine them into one csv file. While this sounds simple, the sizes of the files were great enough that they were difficult for Excel to handle. Additionally, the R script made it easy to filter the data and remove unnecessary variables to make the data set easier to work with.

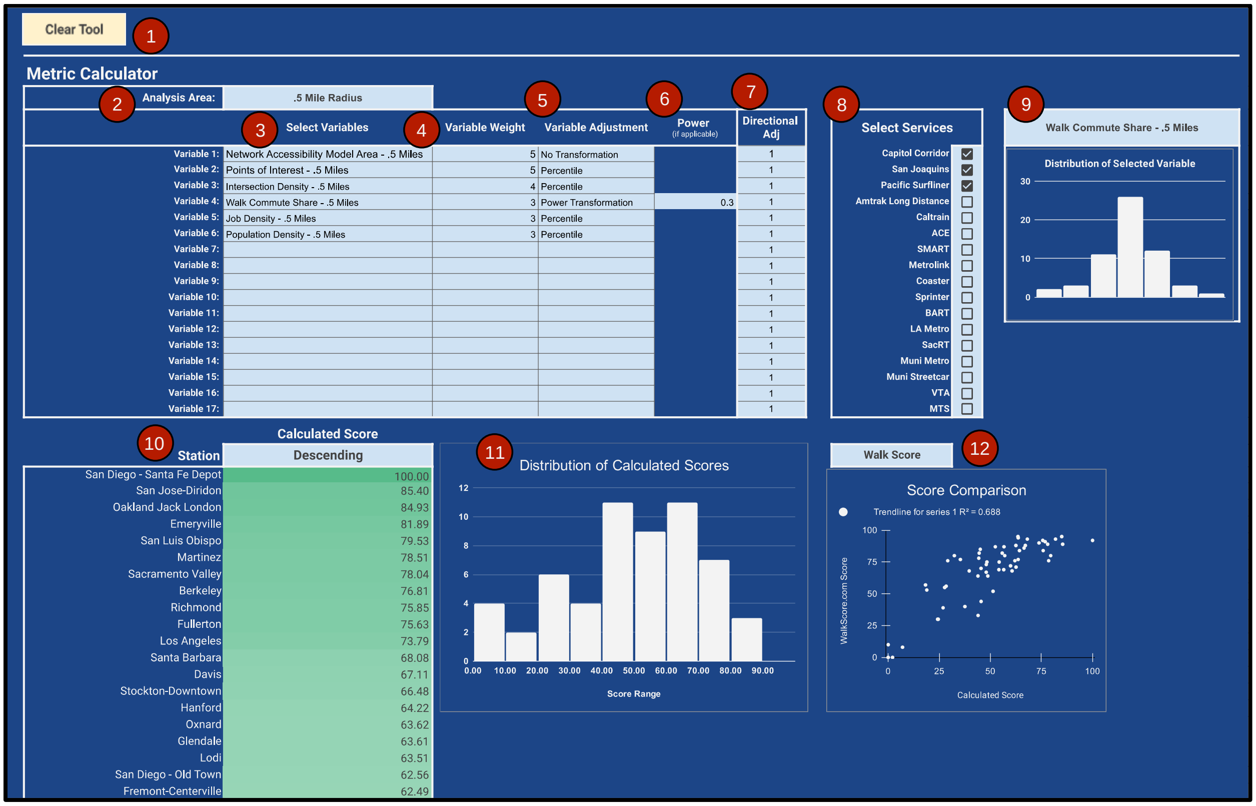
R was also utilized to retrieve walk scores for every station in the state from the Walk Score.com API. Obtaining existing walk scores was necessary to analyze the significance of various data in the context of station areas, and to determine which station area factors had the greatest impact on the scores themselves. This R script took a simple CSV file as an input, with the station name, latitude, and longitude. It then looped through each row, retrieving the walk score corresponding to each point and returning a CSV file with the station name and walk score. It's important to note that, while very effective in obtaining walk scores for five hundred plus stations in only a matter of minutes, the outcome was not perfect. At times, the walk score point selection returned no data, and at other times the selection snapped to a nearby, but slightly different point. In these cases, it was necessary to go in and manually check these scores, although the script did a fairly good job of obtaining the necessary data. Unfortunately, the R package used to obtain the data, walkscoreAPI, was not built to obtain bike scores, so they had to be manually retrieved. In addition, R was used to perform the exploratory data analysis of walk and bike scores, discussed in section 4 of this report. The script used to perform statistical analysis can also be found in Appendix 2.

Google Sheets Tool and Methodology

The primary purpose of this project is not to create a set of pedestrian and bicycle accessibility metrics, but instead to create a series of tools and methods to empower decision makers and stakeholders to easily develop robust, data-driven metrics, informed by their own values and circumstances. This project presents and analyzes a large amount of station area data, most of which is not all that useful in its original form. The tool enables users to focus on which data they want to utilize, adjust the data in a number of ways, and to choose which variables go into the metric and how those variables are weighted.

A spreadsheet-based tool to store the data and perform the analysis was prepared using Google Sheets. Though the Google Sheets-based spreadsheet tool can be entirely utilized with one tab, a number of hidden tabs perform important calculations which populate the primary tool tab. Figure 6.2 shows the tool interface, with callouts for each primary feature.

Figure 6.2: Google Sheets Tool Interface



Source: Henry McKay

1) JavaScript control button. This button runs some simple JavaScript code which clears the tool settings, defaulting it to its original settings.
2) Analysis Area. Dropdown menu to choose a station catchment area for the analysis. The available options are:
    - 0.25 Mile
    - 0.5 Mile
    - 1 Mile
    - 2 Miles
    - 3 Miles

3) Select Variables. Dropdown menu that allows the user to add up to 17 variables to the metric. Only variables measured at the selected analysis area are available.
4) Variable Weight. Drop down menu allowing users to select a value between 1 and 5, with 1 representing the least important weight and 5 representing the most important weight.

5) Variable Adjustment. Drop down menu allowing users to select one of three variable adjustments:
    - No Transformation: Data is kept in its raw form
    - Percentile: Data is adjusted into a percentile score between 0 and 1. For example, if a datapoint has a percentile value of .66, it means that it is larger than 66% of the data in its range.
    - Power Transformation: Allows the user to raise the datapoint to the power of a value between but not including 0 and 1. This adjustment flattens outlying values in a distribution if it is right-skewed.

6) Power. If the power transformation is selected in the previous step, the cell allows the user to input a power value between 0 and 1.
7) Directional Adj. Allows the user to determine if the variable will add to or subtract from the metric. The default value of 1 means that the variable adds to the metric. A value of -1 will make the variable decrease the score.
8) Select Services. A series of checkboxes allow the user to choose which rail services are included in the analysis. This selection can be changed at any point and results will be automatically updated.
9) Data Distribution Viewer. Allows the user to select any variable and view the data distribution on a histogram. This feature is useful in determining which type of data transformation to use (if any).
10) Calculated Score Results. This table displays the final calculated metrics in ranked order for each selected station. The table can either be set to descending or ascending order with a dropdown menu.
11) Score Distribution Chart. This chart dynamically displays the distribution of calculated scores between zero and one-hundred grouped into buckets representing ranges of ten. If a particular score distribution is desired, this feature is useful, although a specific distribution is by no means necessary.
12) Score Comparison. This feature allows the user to compare the calculated score against WalkScore.com walk and bike scores with a scatterplot and R-squared value. Though this feature is not useful in every scenario, it provides a quick way to validate new metrics against existing metrics.

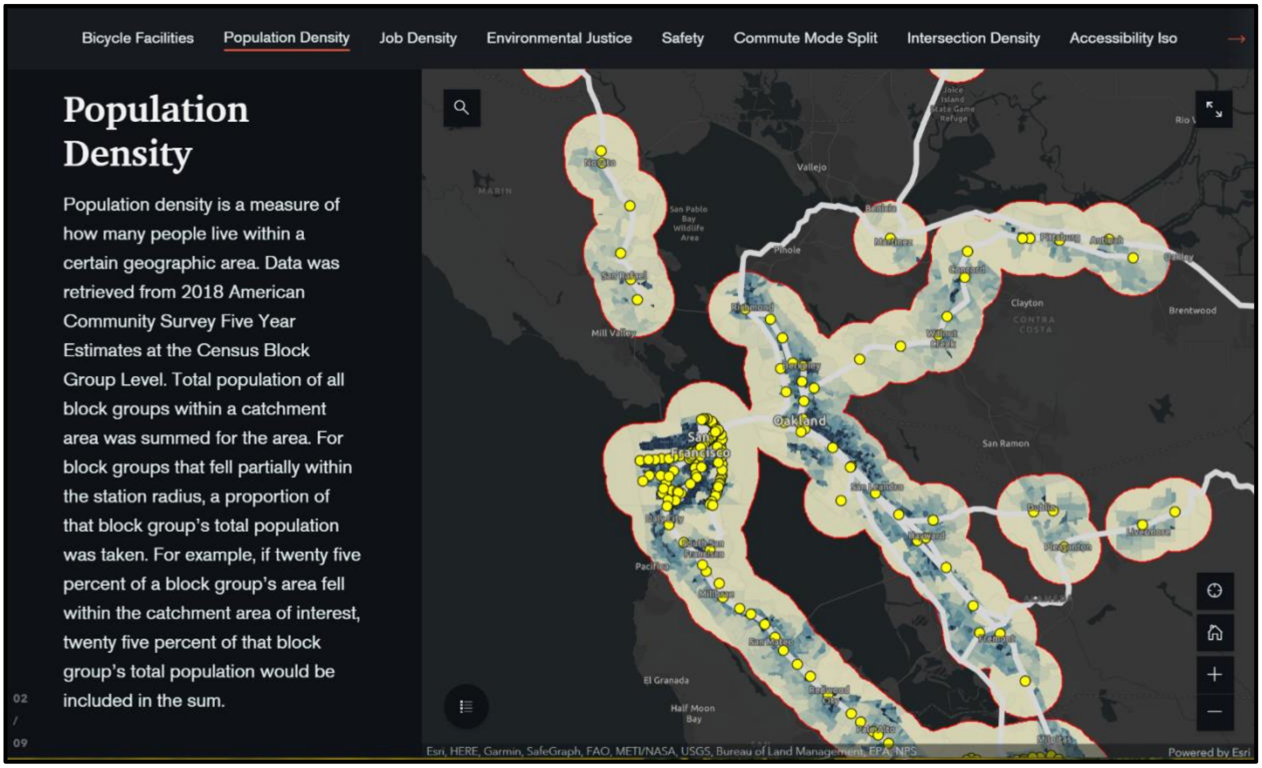
The input data for the tool consists of the data points shown in Table 3.1 in section 3. These are all of the data points in the station area inventory that can be computationally derived from online sources. For

example, the figures for population density can be automatically calculated using ArcGIS and Python every time new Census or American Community Survey (ACS) data is released. Conversely, a statistic like number of bicycle parking spaces, though useful, is something that must be manually updated. Moreover, certain agencies record this type of data, while others do not. For both ease and accuracy, the first version of the tool discussed in this report only considers data that can be automatically derived. In the future, a feature may be added to allow the user to manually upload and include new data which would increase the tool's functionality and usefulness in certain cases. As a result, the tool's metrics are much better suited to assessing a station's area characteristics than they are assessing specific site-level characteristics. Furthermore, the most common forms of input data – census data, Open Street Map bike lane mileage, as well as other station-area data points – are fairly difficult to change quickly or easily. With that in mind, this tool and its resulting metrics provide a starting place for station area planning but are not suitable for comparing project alternatives and or assessing the impacts of site improvements.

ArcGIS StoryMap

Section three of this report discusses the input data collected for this project, but only shows it in the context of Union Station in Los Angeles. The same process that is illustrated for Union Station was carried out for every station in the state. This comprehensive data is available online in the form of an ArcGIS StoryMap, which can be viewed here. A StoryMap is an interactive application which pairs geospatial data with narrative text. Figure 6.3 is a screenshot of the StoryMap displaying population density in the San Francisco Bay Area.

Figure 6.3: ArcGIS StoryMap



Source: Henry McKay, ESRI

7. Analysis

This section applies the metric development methodology and toolkit in three unique planning scenarios, discussing tool parameter choice, results, and on the ground conditions.

Scenario 1: Statewide Walkability Analysis

For the first scenario, the tool was used to assess the walkability of all rail stations in California. Though this broad of an analysis may be uncommon in practice, it is what the tool was designed to do and serves as a good baseline scenario. Table 7.1 shows the tool settings for this scenario.

Table 7.1: Scenario 1 Tool Settings

Analysis Area:	*.5 Mile Radius*	Services Selected:	*All*	
Variable	Weight	Adjustment	Power	Directional Adj
Network Accessibility Model Area - .5 Miles	*5*	*No Transformation*	*N/A*	*1*
Intersection Density - .5 Miles	*3*	*No Transformation*	*N/A*	*1*
Points of Interest - .5 Miles	*2*	*Power Transformation*	*0.15*	*1*
Walk Commute Share - .5 Miles	*3*	*Percentile*	*N/A*	*1*
Population Density - .5 Miles	*2*	*Percentile*	*N/A*	*1*
Job Density - .5 Miles	*2*	*Percentile*	*N/A*	*1*

To analyze walkability, a station catchment area of a half mile was chosen, as recommended by the Federal Transit Administration (FTA). Two variables, Network Accessibility Model Area and Intersection Density, were given the highest weight in the score as they both best represent the physical accessibility and connectivity of a given station area. Points of Interest, Population Density, and Job Density were also included, representing reasons why pedestrians would make a trip. Lastly, Walk Commute share was included to measure the actual level of pedestrian activity in the area. Table 7.2 shows the results of this analysis.

Table 7.2: Scenario 1 Most Walkable Stations

Score	Station	Service(s)
100.0	Market and Guerrero/Laguna	Muni Streetcar
99.4	Market and Gough	Muni Streetcar
99.2	Market and Van Ness	Muni Metro
99.2	Van Ness	Muni Metro
97.8	Market and Dolores/Buchanan	Muni Streetcar

Given the selected tool parameters, five stations with the highest walkability scores in the State were all located in San Francisco, mostly along Market Street. Figure 7.1 shows a photo of the highest scored station, the Market and Guerrero/Laguna Muni Streetcar station.

Figure 7.1: Market and Guerrero/Laguna Muni Station

Source: Wikipedia

Given the score parameters, it is fairly evident why streetcar and light rail stations in San Francisco scored so high. Based on the photo in Figure 7.1, it is clear that the station is in a highly urbanized area, with a high density of housing, jobs, and points of interest. Furthermore, the station itself appears very accessible, as it is located on a highly-active street with high connectivity.

Table 7.3 shows the scores for the lowest scoring stations in the State. According to the tool's output scores, the least walkable rail stations in the State are located primarily in a suburban rail station in Southern California. The Perris – South Metrolink station is the lowest scoring station, with a score of zero. Incidentally, Walk Score.com's score for the station is also zero. Figure 7.2 provides a birds-eye view of the Perris – South station area.

Table 7.3: Scenario 1 - Least Walkable Stations

Score	Station	Service(s)
0.0	Perris - South	Metrolink
2.3	Palm Springs	Amtrak Long Distance
3.0	Vincent Grade/Acton	Metrolink
3.1	Lathrop/Manteca	ACE
10.1	Rancho Cucamonga	Metrolink

Figure 7.2: Birds-Eye View of the Perris – South Metrolink Station

Source: Google Earth

By looking at the birds-eye view of Perris – South station, it is understandable why it scored so low given the selected tool parameters. The surrounding station area has a very sparse roadway network, with little connectivity. Furthermore, the number of surrounding points of interest, people, and jobs, is very low. There is a large parking lot at the station, meaning that it was likely designed primarily with auto access in mind.

Scenario 2: Ranking Caltrain Stations by Bicycle Accessibility

Scenario two focuses on assessing station area bicycle accessibility for stations along the Caltrain corridor in the San Francisco Bay Area. This scenario represents a more focused application of the tool, one which may actually be useful to agencies. If Caltrain were interested in assessing station area bike accessibility in order to better direct "first and last-mile" resources, this analysis would be a good starting place. Table 7.4 shows the tool settings for this scenario.

Table 7.4: Scenario 2 Tool Settings

Analysis Area:	2 Mile Radius	Services Selected:	Caltrain	
Variable	Weight	Adjustment	Power	Directional Adj
Exclusive Bicycle Facilities (mileage) - 2 Miles	*5*	*Percentile*	*N/A*	*1*
Shared Bicycle Facilities (mileage) - 2 Miles	*4*	*Percentile*	*N/A*	*1*
Network Accessibility Model Area - 2 Miles	*3*	*No Transformation*	*N/A*	*1*
Intersection Density - 2 Miles	*3*	*Percentile*	*N/A*	*1*
Bike Commute Share - 2 Miles	*3*	*Percentile*	*N/A*	*1*
Job Density - 2 Miles	*2*	*No Transformation*	*N/A*	*1*
Population Density - 2 Miles	*2*	*No Transformation*	*N/A*	*1*
Points of Interest - 2 Miles	*2*	*No Transformation*	*N/A*	*1*

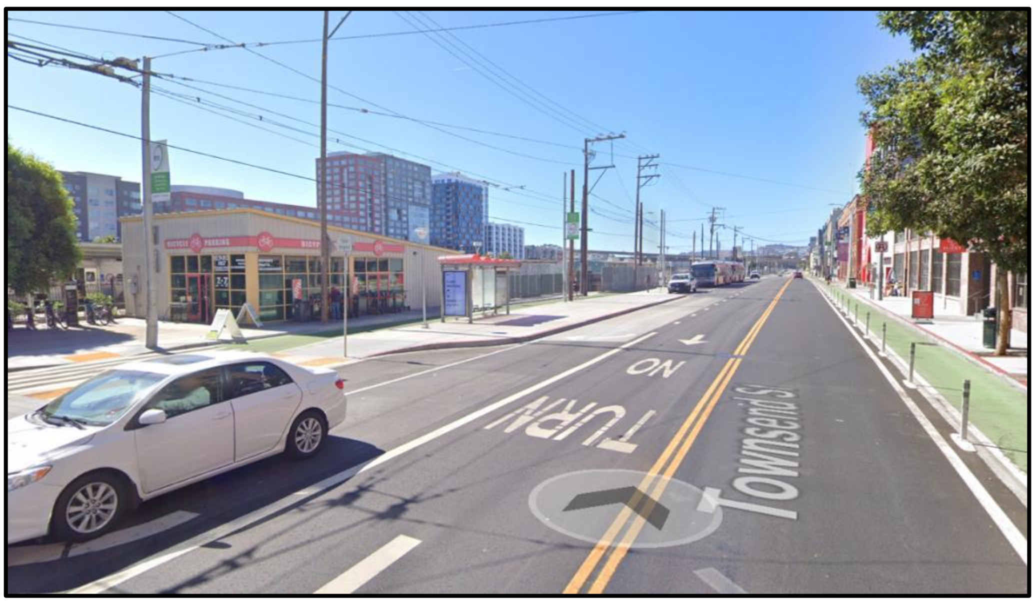
For this scenario, a catchment area of two-miles was chosen since two miles is a fairly average trip length for bicycle trips. While a three-mile radius would also be appropriate for analyzing bicycle accessibility per FTA guidance, there is a great deal of overlap between three-mile station catchment areas along the Caltrain corridor. So, a two-mile catchment area was chosen instead. Bicycle facility mileage was the most important variable in the score and was measured both in terms of shared and exclusive facilities, with exclusive facilities receiving a higher weight. Network Accessibility Model Area and Intersection were included with slightly lower weights than bicycle facility mileage. Bike Commute Share was included and given the same weight of three since it is a good indicator of actual bicycle activity in the station area. Lastly, Job Density, Population Density, and Points of Interest were included, but given lower weights than the other variables. These three variables represent things that individuals would potentially use a bicycle to access. Certain variables in the score were given a percentile adjustment to normalize for large outlying values, which lead to extremely skewed scores. Table 7.5 shows the results of this analysis.

Table 7.5: Scenario 2 – Highest-Scoring Bicycle Accessibility Stations

Score	Station	Service(s)
100.0	San Francisco 4th and King	Caltrain
93.7	San Jose-Diridon	Caltrain
88.3	22nd Street	Caltrain
88.0	College Park	Caltrain
82.3	California Ave.	Caltrain

Given the selected tool parameters, San Francisco's 4th and King Station was the highest-scoring in terms of bicycle accessibility, followed by San Jose Diridon. Looking at both station areas on google maps street view, it is clear why the tool scored them so highly. San Francisco's 4th and King station is surrounded by protected bicycle facilities, with dense urban development and a well-connected street network. Figure 7.3 is a photo of this station area.

Figure 7.3: San Francisco 4th and King Station



Source: Google Maps Street View

The lowest-scoring stations along the Caltrain network were primarily located in suburban and semi-rural areas, often in Santa Clara Valley. The lowest-scoring station, San Martin, has no surrounding bicycle facilities, a fairly limited and unconnected roadway network, and low job and population densities as well as few points of interest. Figure 7.4 shows Monterey Highway, the road on which San Martin's station is located

.

Table 7.6: Scenario 2 – Lowest-Scoring Bicycle Accessibility Stations

Score	Station	Service(s)
0.0	San Martin	Caltrain
9.0	Morgan Hill	Caltrain
10.8	Millbrae Transit Center	Caltrain
18.2	Gilroy	Caltrain
21.7	Broadway	Caltrain

Figure 7.4: San Martin Caltrain Station

Source: Google Maps Street View

As an agency, Caltrain is fairly-proactive in terms of bicycle access planning and has collected data on the number of people who get on and off trains with bicycles at each station. This data was used to compare the calculated bicycle accessibility scores to actual observed bike ridership levels. For both calculated bicycle accessibility scores and bike ridership levels, stations were given a rank with one being the highest level. Table 7.7 shows how these ranks compare across every Caltrain station. Overall, there was fairly weak correlation between station bicycle accessibility score model rank and observed bicycle ridership rank. On average, the stations with the highest and lowest observed bicycle ridership ratings had

fairly-high and low calculated bicycle scores, respectively. However, there were stations that ranked very low in terms of calculated scores but ranked high for observed ridership and vice versa.

Table 7.7: Caltrain Bike Ridership Comparison

Station	Model Rank	Observed Bike Ridership Rank
San Francisco 4th and King	1	1
San Jose-Diridon	2	4
22nd Street	3	7
College Park	4	29
California Ave.	5	9
Tamien	6	17
Stanford	7	N/A
Palo Alto	8	2
Santa Clara	9	16
Mountain View	10	3
San Antonio	11	13
Belmont	12	21
Lawrence	13	12
Sunnyvale	14	6
Hayward Park	15	22
Menlo Park	16	10
Hillsdale (temporary closure)	17	8
Capitol	18	28
San Carlos	19	14
San Mateo	20	11
Blossom Hill	21	26
Redwood City	22	5
Atherton	23	N/A
Burlingame	24	18
South San Francisco	25	20
Bayshore	26	23
San Bruno	27	19
Broadway	28	N/A
Gilroy	29	24
Millbrae Transit Center	30	15
Morgan Hill	31	25
San Martin	32	27

Analysis and Graphic by Henry McKay.

Caltrain data retrieved from Caltrain 2019 Annual Passenger Count Key Findings, 2019

Scenario 3: SacRT Transit-Oriented Development Potential

Scenario three does not focus on pedestrian or bicycle accessibility directly, but instead demonstrates how the tool may be utilized for other sketch planning purposes, such as identifying stations with the most Transit-Oriented Development (TOD) potential given certain criteria. Transit-Oriented Development is the placement of commercial and residential uses within close walking distance of transit stops with the intention of increasing transit usage. In this scenario, the tool is used to identify potential station areas for new transit-oriented development, while addressing issues of environmental justice. Table 7.8 shows the tool settings used for this scenario.

Table 7.8: Scenario 3 Tool Settings

Analysis Area:	*.5 Mile Radius*	Services Selected:	*SacRT*	
Variable	Weight	Adjustment	Power	Directional Adj
Environmental Justice (Weighted CalEnviroScreen Scores) - .5 Miles	*5*	*No Transformation*	*N/A*	*1*
Transit Commute Share - .5 Miles	*4*	*No Transformation*	*N/A*	*-1*
Population Density - .5 Miles	*3*	*No Transformation*	*N/A*	*1*
All Bike Facilities (mileage) - .5 Miles	*3*	*No Transformation*	*N/A*	*-1*

A half-mile catchment area was chosen since it is a distance that many would reasonably walk to access transit. In this scenario, Environmental Justice, as measured as a weighted CalEnviroScreen score for station area census tracts was the most heavily weighted variable. A higher CalEnviroScreen score means that a census tract is more environmentally-disadvantaged, due to a number of factors including pollution burden, median income, and level of traffic stress. This type of data is discussed in greater detail in section 3 of this report. Providing high-quality, affordable Transit-Oriented Development is one way for agencies to address environmental justice issues. Transit commute share was also included but was given a directional adjustment so that higher transit commute shares actually lower the final score. This was done to find station areas where the current transit commute share is relatively low and could thus be improved. Population density was included to ensure that the scores considered current station area population density, with higher densities being more supportive of Transit-Oriented Development. Lastly, total mileage of bicycle facilities was included, but given a directional adjustment, so that top scores reflected lack of adequate bicycle facilities. The resulting scores represent station areas that are currently

disadvantaged in a number of ways and could greatly benefit from Transit-Oriented Development. Table 7.9 shows the highest-scoring stations.

Table 7.9 Scenario 3 Stations with the highest TOD potential

Score	Station	Service(s)
100.0	Meadowview	SacRT
93.0	Roseville Road	SacRT
92.8	Watt/I-80 West	SacRT
91.5	Center Parkway	SacRT
84.0	Florin	SacRT

Given the specified tool settings, Meadowview station in South Sacramento ranked the highest for Transit-Oriented Development potential. It is heavily burdened environmentally, has a low transit commute share, has a fairly high station are population density, and lacks bicycle infrastructure. Though vacant land was not considered in the score, there are several acres of vacant land to the immediate West of the station, as Figure 7.5 shows. The available land already has roads running through it and would be highly suitable for Transit-Oriented Development if other criteria were met. Out of the five top-ranked stations, four had one feature in common: large park and ride lots. When these stations were originally built, they were largely intended for automobile access and still are to a large extent.

Figure 7.5: Sacramento Regional Transit Meadowview Light Rail Station Area

Source: Google Maps Street View

Based on the selected criteria, the SacRT light rail stations with the least TOD potential were primarily located along SacRT's Gold Line, which runs from downtown east into the suburbs. Glenn station, located near Folsom, CA, had the least TOD potential, as Table 7.10 shows. It is located in a fairly wealthy area with little environmental burden. Additionally, it has a fairly-low population density and a decent network of existing bicycle infrastructure.

Table 7.10: Scenario 3 Stations with the lowest TOD Potential

Score	Station	Service(s)
0.0	Glenn	SacRT
6.0	Iron Point	SacRT
13.5	College Greens	SacRT
14.3	Watt/Manlove	SacRT
16.7	Tiber	SacRT

Figure 7.6: SacRT Glenn Light Rail Station

Source: Google Maps Street View

8. Conclusion

Good “first and last-mile” planning can extend the reach of transit services, more fully making non-automotive trips possible. As California continues to invest in its rail transit network and plan for increased service and higher connectivity, it is essential that “first and last-mile” planning also takes place in order to maximize the utility of rail service. Transit agencies and state agencies rarely have control over local land use, which is the domain where “first and last-mile” planning is primarily implemented. However, these agencies do have the power to create certain policy in the form of “first and last-mile” plans and the rail plan at the state level. While these plans do not have direct authority over local land use, they do influence how state funds are awarded and thus carry some weight. However, the state faces a different set of “first and last-mile” planning challenges than those faced by local or regional transit agencies. While a transit agency may only be responsible for a relatively small number of stations, the state is interested in every station within the State, at least for high-level planning and funding purposes.

This project takes the key principles of “first and last-mile” planning as practiced by transit agencies on a smaller scale and applies them to the entire state rail transit network to enable analysis at the statewide level. The project takes a sketch planning approach, meaning that the metrics produced are less detailed than the highly site-specific ones created for single-station, “first and last-mile” plans. However, this broad approach enables analysis to be conducted on a larger scale, which is suitable for the type of high-level planning work done by state agencies.

In addition, a technical toolkit was created to semi-automate the data retrieval and cleaning process and to allow practitioners to easily play with and explore the data. Users can create custom metrics that are flexible to their needs and circumstances.

As discussed in the introduction of this report, this is not a static project and will continue to be updated as data is updated, and new features are added to the toolkit. There are also several key areas where the toolkit and methodology can be improved in the future. Firstly, better integration can be incorporated between the data sources and tool by creating a web application that pulls and cleans data directly from an API. This would be much more complex than the current spreadsheet-based system but would enable a higher-performing tool that would be easier to use. Secondly, more variables can be added to the data set in order to create more robust and interesting metrics. Lastly, more complex methods of analysis could be

incorporated to create metrics. For example, routing algorithms could be used to create more data derived from theoretical routes as is the case with walk score's metrics. Many of these methods are more computationally demanding but can lead to more precise measurements of accessibility.

References

Bay Area Rapid Transit District (2020). BART Walk and Bicycle Network Gap Study.

Beale, Linda (2012). Line and Junction Connectivity. ESRI. Retrieved from https://www.arcgis.com/home/item.html?id=3fa41b1f8b764879be8f21b4e7ffbabd

Boarnet, Marlon G, Giuliano, Genevieve, Hou, Yuting, & Shin, Eun Jin. (2017). *"first and last-mile" transit access as an equity planning issue*. Transportation Research. Part A, Policy and Practice, 103, 296–310. https://doi.org/10.1016/j.tra.2017.06.011

Broader, Jacquelyn, Cohen, Adam, Forscher, Edward, Farrar, Emily, Nelms, Alex, Shaheen, Susan (2020). *Understanding Modal Access/Egress for California Rail Stations*. Berkeley Transportation Sustainability Research Center.

Caltrain (2019). Caltrain 2019 Annual Passenger Count Key Findings. Retrieved from https://www.caltrain.com/Assets/Stats+and+Reports/2019+Annual+Key+Findings+Report.pdf

Committee of the Transport Access Manual (2020). *Transport Access Manual: A Guide for measuring connection between people and places*.

Crooks, Andrew (2008). Sketch Planning. GIS and Agent-Based Modeling. Retrieved from https://www.gisagents.org/2008/02/sketch-planning.html

Johnson, Perry (2019). *Reverse Engineering the Walk Score*. Medium, towards data science. Retrieved from https://towardsdatascience.com/engineer-walkscore-4dc1730b976c

Los Angeles County Metropolitan Transportation Authority – Metro, Southern California Association of Governments (2014). *First Last Mile Strategic Plan & Planning Guidelines.*

R Core Team. (2021). R: A Language and Environment for Statistical Computing. Vienna, Austria. Retrieved from https://www.R-project.org/

San Francisco Bay Area Rapid Transit District (2006). *Access BART*.

Semler, Conor & Hale, Chris. (2010). *Rail Station Access - An assessment of options*. ATRF 2010: 33rd Australasian Transport Research Forum.

Transportation Research and Education Center at Portland State University (2017). Manual on Pedestrian Connections to Transit. FTA Research. Retrieved from https://www.transit.dot.gov/sites/fta.dot.gov/files/docs/research-innovation/64496/ftareportno0111.pdf

United States Census Bureau (2018). Longitudinal Employer-Household Dynamics OnTheMap Tool. Retrieved from https://onthemap.ces.census.gov/

United States Census Bureau (2018). TIGER/Line with Selected Demographic and Economic Data. Retrieved from https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-data.html

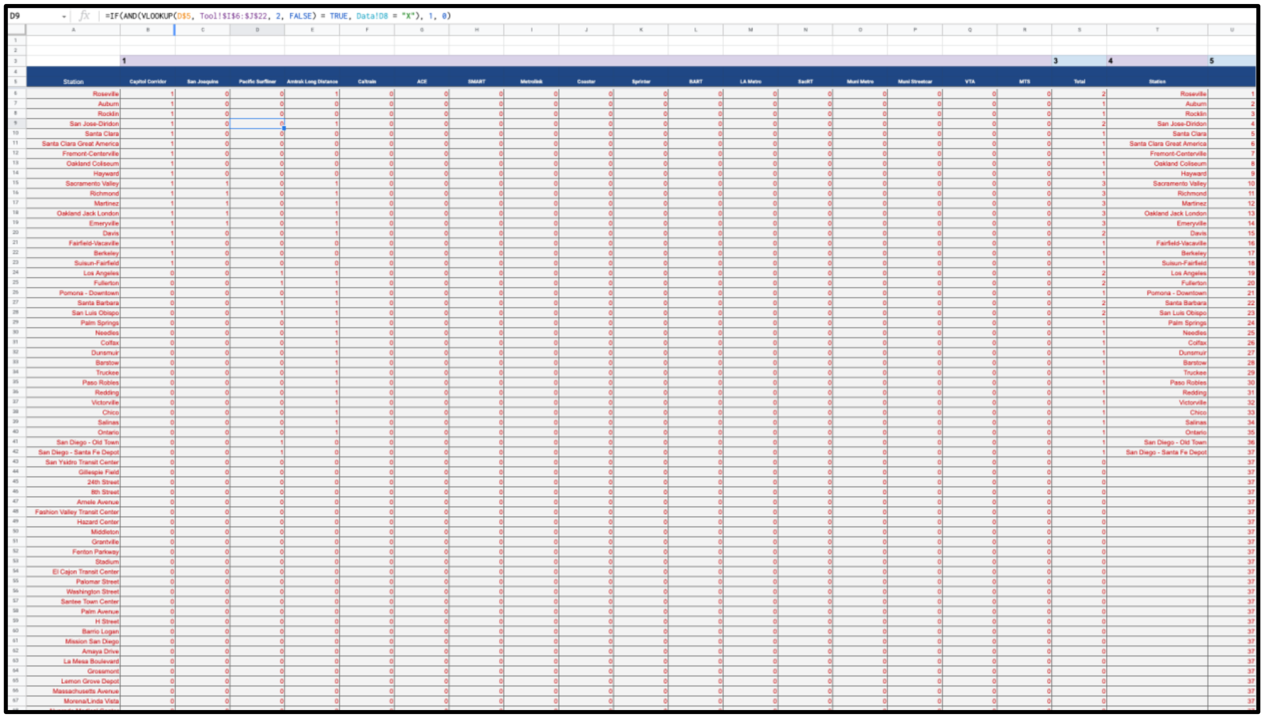
Walk Score (2021). Walk Score Methodology. Retrieved from https://www.walkscore.com/methodology.shtml

Whalen, John (2012). *walkscoreAPI: Walk Score and Transit Score API*. R package version 1.2. https://CRAN.R-project.org/package=walkscoreAP

Appendix 1. Spreadsheet Documentation

This appendix details the calculations performed by the Google Sheets tool. Each bullet corresponds to a number in the Tool_Backend tab, shown below in Figure A-1.1.

Figure A-1.1: Google Sheets Tool Backend



1) First, the tool filters the station area dataset by service. On the tool tab, columns I:J present a series of checkboxes corresponding to each service in the state. In column J, the underlying cell returns TRUE if the checkbox is checked, and FALSE if it not. These checkboxes determine which station will be included in the analysis dataset. The actual filtering of the dataset and metric calculations occur on the hidden Tool_Backend tab.

2) First, every station in the original dataset is referenced in column A. Columns B:R correspond to every service in the original dataset. If the service is checked on the main tool tab, every one of its corresponding station cells returns 1. The cells return 0 if the station is not included in the checked services.

3) Column S returns a sum of the previous range, by row. If the value is greater or equal to 1, then the corresponding station is included in the filtered dataset. If the value is 0, then the station is not included.

4) In column T, a simple if statement is used to reference the station name if it is included in the filtered dataset.

5) Columns U:Y remove the blank spaces between various stations, returning the same previously-filtered list with no blank cells.

6) Columns Z:DF use a VLOOKUP formula to bring in the entire range of data for the filtered stations. This range is rather large, as there are five columns for each variable for measurements at .25, .5, 1, 2, and 3 miles. At the top of this range, there is a column index ranging from 1 to 85. This index plays an important role in later tool functionality.

7) The next range, columns DI:DZ, filters the dataset by area of analysis. The main tool tab allows the user to select an analysis area – .25, .5, 1, 2, or 3-mile Radius – in cell C4. Based on this selection, the filtered dataset is further filtered down to include only the variables corresponding to the selected area. In the Tool_Backend tab. This is accomplished by creating a new index in cells DI4:DZ4, with increments of 5, starting at a value between 1 and 5 determined by the area selected. An HLOOKUP function then retrieves the data values for each station corresponding to filtered variables.

8) Next, in columns EB:ER, the data set if filtered again to only include the variables selected in the main tool tab under 'select variable.' This is accomplished with an HLOOKUP function nested in an IF statement to return the HLOOKUP if the corresponding input cell on the main tool tab is not blank.

9) Columns ET:FJ apply the selected adjustment option that is selected for each variable. On the main tool tab in column E, there is a drop-down menu to select one of three adjustment options for each selected variable. These options are:

    - No Transformation: Data is maintained in its original form

- Percentile: Data points are adjusted into a percentile value relative to the entire variable range. Values range from 0 to 1. For example, a percentile value of .65 would mean that the data point is greater than 65% of the data in its range.
- Power Transformation: When the power transformation option is selected, the corresponding variable cell in column F of the main tool tab turns blue. Values ranging between, but not including 0 to 1 can be manually inputted. In the Tool_Backend tab, the data corresponding to the variables with the power transformation adjustment selected are raised to the power of whatever the inputted power is. This adjustment is useful for when data is highly skewed in one direction and a normal distribution is preferred. For example, the raw data values for job and population density are much higher for station in San Francisco than they are for stations in parts of the central valley by orders of magnitude. However, when comparing these stations, it may be appropriate to adjust the data so that these differences are smaller, while not changing the order of the underlying data.

This task is accomplished by a nested IF statement with multiple VLOOKUPs to return the input values.

10) In columns FL:GB, the inputted metric weights are applied to the dataset. Though this step takes place in one formula, three things are accomplished. First, the data is rescaled to a range between 0 and 1 using the following formula:

$$X_{scaled} = X - X_{min} / X_{max} - X_{min}$$

Secondly, the data is multiplied by the selected weight of its corresponding variable. This selection occurs in column D of the primary tool tab with values ranging from 1 (weighted the least) to 5 (weighted the most). However, the data is not simply multiplied by the weight value itself. In the Tables tab, the selected weight value is divided by the sum of all the selected weights to find a proportional weight. It is this proportional weight that the actual data is multiplied by.

Lastly, the weighted data is multiplied by either 1 or -1, depending on which value is inputted for each variable in column G of the main tool tab. Though the default value is 1, it is sometimes appropriate to have a variable take away from the score as opposed to add to it. For example, a higher value for number of pedestrian incidents in not a good indicator for walkability and should work against the score.

11) Column GD sums the previously calculated range (FL:GB) to create a weighted score between 0 and 1 based on all of the selected variables and assigned weights. However, it is possible for this

value to be negative if the negative variables added outweigh the positive ones. This is dealt with in a subsequent step.

12) In column GE, the station names are referenced in with a simple IF statement.

13) In column GF, the scores calculated in GD are once again rescaled between 0 and 1 to remove any negative values. They are also multiplied by 100 to create a score between 0 and 100.

14) Lastly, a FILTER function nested in a SORT function is used to filter the station names and final scores in columns GE:GF to remove blank cells if not all services are selected and to order by value in either ascending or descending order. The formula is an array formula, meaning that it is only located in cell GG6, but effects the range GG6:GH700. The data range in GG6:GH700 are the final scores that are displayed as output on the tool tab.

Appendix 2. Scripts

The following scripts were written to obtain and clean input data for the google sheets tool. To run R scripts, the following software must be installed:

- R: Download
- R Studio: Download

To run python scripts, a python Integrated Development Environment (IDE) must be installed. PyCharm is a good choice and can be downloaded here. An ArcGIS license must also be installed in order to run the Python scripts, all of which automate GIS tasks.

For all scripts, input data must be placed in the correct folder as specified in the script. Input data is provided with the scripts and can be downloaded here. Comments in the scripts specify the particular function of each block of code.

Folder Structure

A folder containing the scripts discussed in this section and the necessary input data to run them can be downloaded here. The folder structure is as follows:

- HenryMcKay_SeniorProject_Code
    - Data
        - ExcelOutput_Python (Folder that all scripts write their output Excel/CSV files to.)
        - SafetyData
            - County_CSVs (Raw safety incident csv files for relevant CA counties.)
            - Output_File (Location for output of cleaned safety data.)
        - ScoreData (Input data for stations to retrieve walk scores.)
        - SpatialData
            - ACS_2018_5YR_BG_06_CALIFORNIA.gdb (Geodatabase for American Community Survey Block Groups data.)
            - ACS_2018_5YR_TRACT_06_CALIFORNIA.gdb (Geodatabase for American Community Survey Census Tracts data.)

        - CES3_June2018update.gdb (Geodatabase for CalEnviroScreen data.)
        - LEHD_2017 (Folder with shapefiles for LEHD jobs data.)
        - StationAreaData.gdb (Geodatabase containing other relevant spatial data used in this project's scripts.)
    - StepwiseData (Input and output data for the stepwise regression analysis discussed in section 4 of this report.)
- Scripts
    - Python Scripts
        - Script#4
        - Script#5
        - Script#6
        - Script#7
        - Script#8
        - Script#9
        - Script#10
        - Script#11
    - R Scripts
        - Script#1
        - Script#2
        - Script#3

Script 1. Obtain Walk Scores

This R script is used to obtain walk scores from the Walk Score Application Programming Interface (API). In order for the script to run, an input CSV file containing columns with the station name, latitude, and longitude is necessary. Furthermore, it is necessary to obtain an API key from Walk Score's API, which can be requested here.

```
###################################################################################
# Script #1: Obtain WalkScore.com Scores
# *** Change all instances of "YourUsername" in file paths to your user name ***

# Install and load the necessary libraries
if(!require("dplyr"))installed.packages("dplyr")
if(!require("tibble"))installed.packages("tibble")
if(!require("walkscoreAPI"))installed.packages("walkscoreAPI")
if(!require("rjson"))installed.packages("rjson")

library(dplyr)
library(tibble)
library(walkscoreAPI)
library(rjson)

# Set working directory to folder with data
setwd("C:/Users/YourUsername/Desktop/HenryMcKay_SeniorProject_Code/Data/ScoreData")
stations = read.csv("StationScores.csv")
attach(stations)

#create empty list
res = list()

# for loop through the file and retrieve walk score for each station
for(i in 1:500){
  # Obtain API key from Walkscore.com (more info in appendix) and insert below
where instructed:
  res[i] = list(getWS(stations$Longitude[i],stations$Latitude[i],"YourAPIKey"))
}

# Create data frame with stations and scores
res %>%
  sapply(unclass) %>%
  as.data.frame() %>%
  t() %>%
  as.data.frame() %>%
  lapply(unlist) %>%
  as.data.frame(stringsAsFactors = FALSE) %>%
  rowid_to_column("ID") %>%
  remove_rownames() -> df
FinalScores = merge(stations, df, by.x="Index", by.y="ID")
```

```
# Write CSV with stations and scores
write.csv(FinalScores, file = "StationScores.csv")
```

Script 2. Clean Safety Data

This R Script aggregates pedestrian and bicycle safety data into a form than can be analyzed in GIS. To work, CSV files containing safety data for all relevant counties must be placed in the appropriate folder, as specified in the script. These CSV files can be obtained here. An account is required to access data, although it is free. Files used for this project are provided and can be accessed in Appendix 2.

```
##################################################################################
# Script #2: Clean Safety Data
# *** Change all instances of "YourUsername" in file paths to your user name ***

# Install and load the necessary dplyr library
if(!require("dplyr"))installed.packages("dplyr")
library(dplyr)

# Set working directory to folder with csv files
setwd("C:/Users/YourUsername/Desktop/HenryMcKay_SeniorProject_Code/Data/SafetyData/
County_CSVs");
files = dir();

# Function to filter csv by category and only keep relevant variables
combineFiles = function(filename) {
  data_ped = read.csv(file=filename, header=T, as.is=T, na.strings=c("NA")) %>%
    filter(COUNT_PED_INJURED > 0 |
             COUNT_PED_KILLED > 0 |
             COUNT_BICYCLIST_INJURED > 0 |
             COUNT_BICYCLIST_KILLED > 0) %>%
    select(POINT_X,
           POINT_Y,
           COUNT_PED_INJURED,
           COUNT_PED_KILLED,
           COUNT_BICYCLIST_INJURED,
           COUNT_BICYCLIST_KILLED)
}

# For loop to go through each csv file, filter it, and combine into one file
out = NULL
for (i in files) {
  data_temp = combineFiles(i)
  out = rbind.data.frame(out, data_temp)
  print(i)
}

#
write.csv(out,
"C:/Users/YourUsername/Desktop/HenryMcKay_SeniorProject_Code/Data/SafetyData/Output
_File/CA_Collisions.csv",
```

```
          row.names = FALSE)

rm(data_temp, out, i, files, combineFiles)
```

Script 3. Walk/Bike Score Regression Analysis

This R script was used to perform the walk score exploratory data analysis, discussed in section four of this report. The input data files for the script are provided and can be accessed in Appendix 2.

```
####################################################################################
# Script 3: Walk/Bike Score Regression Analysis
# *** Change all instances of "YourUsername" in file paths to your user name ***

# Install and load the necessary dplyr library
if(!require("dplyr"))installed.packages("dplyr")
library(dplyr)

# Read in the csv file containing data
# Change to 'Walk_Score_Data.csv' to analyze walk scores
data2 =
read.csv("C:/Users/YourUsername/Desktop/HenryMcKay_SeniorProject_Code/Data/Stepwise
Data/Bike_Score_Data.csv")
data2 = rename(data2, Score = 1)
data2 = na.omit(data2)

# Perform stepwise regression
FitAll = lm(Score ~ ., data = data2)
FitStart = lm(Score ~ 1, data = data2)
step(FitStart, direction = "both", scope = formula(FitAll))

# Assign the best fit model (output of previous step) to mod object
mod = lm(Score ~ X78 + X122 + X27 + X113 + X72 + X51 + X6 +
           X19 + X1 + X35 + X16 + X108 + X39 + X118 + X86 + X83 + X48 +
           X91 + X36 + X41 + X53, data = data2)


# Obtain summary statistics, coefficients
summary(mod)
anova(mod)
summ = anova(mod)

# Obtain sum of squares and write to csv file
ssquares = summ[2]
write.csv(ssquares,
"C:/YourUsername/henrymckay/Desktop/HenryMcKay_SeniorProject_Code/Data/StepwiseData
/WS_SSquares.csv")

# Obtain predicted scores and write to csv file
predicted = predict(mod)
write.csv(predicted,
"C:/Users/YourUsername/Desktop/HenryMcKay_SeniorProject_Code/Data/StepwiseData//WS_
Predicted.csv")
```

```
# Plot Predicted Scores v. WalkScore.com scores
plot(predict(mod),data2$Score,
     xlab="predicted",ylab="actual")
```

Script 4. Bike Infrastructure

This python script is used to aggregate bicycle facility mileage to station catchment areas. The input data for the script is a shapefile containing all bicycle facilities for the state, which was obtained from Open Street Map (OSM). The OSM data used for this project can be accessed in Appendix 2, in the StationArea geodatabase.

```
###################################################################################
# Script 4: Bike Infrastructure
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array(["1 Mile", ".5 Mile"])
POINT =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\Stati
onAreaData.gdb\CA_Stations"
Analysis_Var = "dist_miles"
Analysis_Type = "SUM"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\Stati
onAreaData.gdb"

# Change end of filepath from 'Shared' to 'Exclusive' to analyze Exclusive bicycle
facilities
BK =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\Stati
onAreaData.gdb\CA_BikeFacilities_Shared"
TableOut =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\ExcelOutput_Pytho
n"
FileName = "CA_Bike_Facilities_Shared.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Creates a copy of the Bike Data
arcpy.CopyFeatures_management(BK, "in_memory/BK_Temp")
```

```
# Defines counter variable
Y = 1


# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

   # Intersects bicycle facilities with buffers
   arcpy.Intersect_analysis(["in_memory/BK_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Creates new output table name
   myDir = "in_memory/"
   New_Path = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Path)

   # Sums bicycle facility mileage by station
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", New_Out,
[[Analysis_Var, Analysis_Type]], "Station_Name")
   BK_SUM = "{}{}{}".format("BF: ", X, " Buffer")
   arcpy.AlterField_management(New_Out, "SUM_dist_miles", BK_SUM, BK_SUM)
   arcpy.DeleteField_management(New_Out, "FREQUENCY")

   print("{}{}".format("Finished calculating bike facilities at ", X))

   Y += 1

# Joins tables from different buffer distances by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
   if table == "Table_1":
       pass
   else:
       arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
       arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)
print("EXPORTED TABLE TO EXCEL")

print("FINISHED:")
```

Script 5. Population Density

This python script calculates population density for each catchment area of specified rail stations. To run, an ArcGIS feature class must exist containing all stations. Furthermore, a Census Geodatabase must be placed in the correct folder, which can be downloaded here. All necessary input data used in this project is provided and can be accessed in Appendix 2.

```
##################################################################################
# Script 5: Population Density
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".25 Mile", ".5 Mile"])
CensusData = "X01_AGE_AND_SEX"
CensusTableField = "B01003e1"
POINT = "CA_Stations"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
myGDB =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
CensusPath =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
ACS_2018_5YR_BG_06_CALIFORNIA.gdb"
FileName = "Population_Density.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Defines shapefile for TIGER Geometry
TIGER = os.path.join(CensusPath, "ACS_2018_5YR_BG_06_CALIFORNIA")

# Creates a copy of the TIGER Geometry
```

```
arcpy.CopyFeatures_management(TIGER, "in_memory/TIGER_Temp")

# Joins specified table to TIGER Geometry
DATETABLE = os.path.join(CensusPath, CensusData)
arcpy.JoinField_management("in_memory/TIGER_Temp", "GEOID_Data", DATETABLE,
"GEOID", ["GEOID_Data", CensusTableField])
# Recalculates area of block groups
arcpy.AddField_management("in_memory/TIGER_Temp", "AREA", "DOUBLE", "", "", "",
"AREA", "NULLABLE", "")
inTable = "in_memory/TIGER_Temp"
fieldName = "AREA"
expression = "!shape.area@squaremiles!"
arcpy.CalculateField_management(inTable, fieldName, expression, "PYTHON")

# Defines counter variable
Y = 1

# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

   # Intersects TIGER Geometry with buffers
   arcpy.Intersect_analysis(["in_memory/TIGER_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Calculates are of intersected TIGER Geometry
   arcpy.AddField_management("in_memory/Intersected_Buffers", "NEW_AREA", "DOUBLE",
"", "", "", "NEW_AREA", "NULLABLE", "")
   inTable2 = "in_memory/Intersected_Buffers"
   fieldName2 = "NEW_AREA"
   expression2 = "!shape.area@squaremiles!"
   arcpy.CalculateField_management(inTable2, fieldName2, expression2, "PYTHON")

   # Calculates new extrapolated population numbers
   arcpy.AddField_management("in_memory/Intersected_Buffers", "NEW_POP", "DOUBLE",
"", "", "", "NEW_POP", "NULLABLE", "")
   arcpy.CalculateField_management("in_memory/Intersected_Buffers", "NEW_POP",
"round((!NEW_AREA! / !AREA!) * !B01003e1!)", "PYTHON")

   # Creates new output table name
   myDir = "in_memory/"
   New_Path = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Path)

   # Sums population estimated by station
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", New_Out, [["NEW_POP",
"SUM"]], "Station_Name")
   POP_SUM = "{}{}{}".format("POPULATION: ", X, " Buffer")
   arcpy.AlterField_management(New_Out, "SUM_NEW_POP", POP_SUM, POP_SUM)
   arcpy.DeleteField_management(New_Out, "FREQUENCY")

   print("{}{}".format("Finished calculating population at ", X))
```

```
    Y += 1

# Joins tables by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
    if table == "Table_1":
        pass
    else:
        arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
        arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)

arcpy.Delete_management("in_memory")
print("EXPORTED TABLE TO EXCEL")
print("FINISHED:")
```

Script 6. Job Density

This python script calculates job density for each catchment area for specified stations. The input data for the script can be obtained from LEHD's OnTheMap tool by selecting the whole state and exporting the results as a shapefile. The OnTheMap tool can be found here. All necessary input data for this script is provided and can be accessed in Appendix 2.

```
##################################################################################
# Script 6: Job Density
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".25 Mile", ".5 Mile"])
POINT =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb\CA_Stations"
Analysis_Var = "c000"
Analysis_Type = "SUM"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
LEHD_2017"
LEHD =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
LEHD_2017\points_2017.shp"
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
FileName = "Job_Density.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Creates a copy of the LEHD Points
arcpy.CopyFeatures_management(LEHD, "in_memory/LEHD_Temp")

# Defines counter variable
Y = 1

# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
```

```
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

# Intersects LEHD with buffers
   arcpy.Intersect_analysis(["in_memory/LEHD_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Creates new output table name
   myDir = "in_memory/"
   New_Path = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Path)

   # Sums jobs by station
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", New_Out,
[[Analysis_Var, Analysis_Type]], "Station_Name")
   JOB_SUM = "{}{}{}".format("JOBS: ", X, " Buffer")
   arcpy.AlterField_management(New_Out, "SUM_c000", JOB_SUM, JOB_SUM)
   arcpy.DeleteField_management(New_Out, "FREQUENCY")

   print("{}{}".format("Finished calculating jobs at ", X))

   Y += 1

# Merges tables by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
   if table == "Table_1":
       pass
   else:
       arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
       arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)

arcpy.Delete_management("in_memory")

print("EXPORTED TABLE TO EXCEL")
print("FINISHED:")
```

Script 7. CalEnviroScreen

This python script calculates weighted averages of CalEnviroScreen scores for each catchment area of rail stations. For the script to run, a shapefile of existing CalEnviroScreen data is necessary, which can be downloaded here. Input data used by this script is provided and can be accessed in Appendix 2.

```
##################################################################################
# Script 7: CalEnviroScreen
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Score: CIscore

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".5 Mile", ".25 Mile"])
CensusTableField = "ES_Prop"
POINT = "CA_Stations"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
myGDB =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
FileName = "Cal_Enviro_Screen_WeightedAverage.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Defines feature class for CalEnviroScreen Census Tracts
TIGER =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
CES3_June2018update.gdb\CES3_June2018updateGDB"

# Creates a copy of the Census Tracts
arcpy.CopyFeatures_management(TIGER, "in_memory/TIGER_Temp")

# Defines counter variable
Y = 1
```

```
# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

   # Calculate the area of buffers
   arcpy.AddField_management("in_memory/POINT_BUFFER", "AREA", "DOUBLE", "", "",
"", "AREA", "NULLABLE", "")
   inTable = "in_memory/POINT_BUFFER"
   fieldName = "AREA"
   expression = "!shape.area@squaremiles!"
   arcpy.CalculateField_management(inTable, fieldName, expression, "PYTHON")

   # Intersects TIGER Geometry with buffers
   arcpy.Intersect_analysis(["in_memory/TIGER_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Calculates area of intersected TIGER Geometry
   arcpy.AddField_management("in_memory/Intersected_Buffers", "NEW_AREA", "DOUBLE",
"", "", "", "NEW_AREA", "NULLABLE", "")
   inTable2 = "in_memory/Intersected_Buffers"
   fieldName2 = "NEW_AREA"
   expression2 = "!shape.area@squaremiles!"
   arcpy.CalculateField_management(inTable2, fieldName2, expression2, "PYTHON")

   # Calculates proportion or new area
   arcpy.AddField_management("in_memory/Intersected_Buffers", "New_Area_Prop",
"DOUBLE", "", "", "", "New_Area_Prop", "NULLABLE", "")
   arcpy.CalculateField_management("in_memory/Intersected_Buffers",
"New_Area_Prop", "(!NEW_AREA! / !AREA!)", "PYTHON")

   # Calculates new prop
   arcpy.AddField_management("in_memory/Intersected_Buffers", "New_ES_Prop",
"DOUBLE", "", "", "", "New_ES_Prop", "NULLABLE", "")
   arcpy.CalculateField_management("in_memory/Intersected_Buffers", "New_ES_Prop",
"(!New_Area_Prop! * !CIscore!)", "PYTHON")

   # Creates new output table name
   myDir = "in_memory/"
   New_Name = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Name)

   # Calculates weighted average of CalEnviroScreen scores within a given station
radius
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", "in_memory/Area_SUM",
[["New_Area_Prop", "SUM"]], "Station_Name")
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", "in_memory/ES_SUM",
[["New_ES_Prop", "SUM"]], "Station_Name")

   arcpy.JoinField_management("in_memory/Area_SUM", "Station_Name",
"in_memory/ES_SUM", "Station_Name")

   arcpy.AddField_management("in_memory/Area_SUM", "Adj_ES", "DOUBLE", "", "", "",
"Adj_ES", "NULLABLE", "")
```

```
    arcpy.CalculateField_management("in_memory/Area_SUM", "Adj_ES",
"(!Sum_New_ES_Prop! / !Sum_New_Area_Prop!)", "PYTHON")
    arcpy.TableToTable_conversion("in_memory/Area_SUM", "in_memory", New_Name)

    PROP_SUM = "{}{}{}".format("Proportion: ", X, " Buffer")
    arcpy.AlterField_management(New_Out, "Adj_ES", PROP_SUM, PROP_SUM)
    arcpy.DeleteField_management(New_Out, ["FREQUENCY",
                                           "SUM_New_Area_Prop",
                                           "Station_Name_1",
                                           "Frequency_1",
                                           "SUM_New_ES_Prop"])

    arcpy.Delete_management("in_memory/Area_SUM")
    arcpy.Delete_management("in_memory/ES_SUM")

    print("{}{}".format("Finished calculating proportion at ", X))

    Y += 1

# Merges tables by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
    if table == "Table_1":
        pass
    else:
        arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
        arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)
print("EXPORTED TABLE TO EXCEL")
print("FINISHED:")
```

Script 8. Safety

This python script calculates the number of pedestrian and bicycle incidents within each catchment area of rail stations. The input data necessary for this script is the output of script #2, detailed earlier in this Appendix 2. The necessary input data is provided and can be accessed in Appendix 2.

```
##################################################################################
# Script 8: Safety
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".5 Mile", ".25 Mile"])
Analysis_Type = "SUM"
Analysis_Var = "COUNT_BICYCLIST_KILLED"
FileName = "{}{}{}".format(Analysis_Var, "_Density", ".xls")
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
POINT =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb\CA_Stations"
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SafetyData\
Output_File"
arcpy.env.overwriteOutput = True

# Create feature class from csv file coordinates
try:
   x_coords = "POINT_X"
   y_coords = "POINT_Y"
   outlayer = "csveventlayer"
   arcpy.MakeXYEventLayer_management("CA_Collisions.csv", x_coords, y_coords,
outlayer)
   filename = os.path.splitext("CA_Collisions.csv")[0]
   myOut = "in_memory/"
   output = os.path.join(myOut, filename)
   arcpy.CopyFeatures_management(outlayer, output)
   os.remove("C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
SafetyData\Output_File\schema.ini")

   # Defines counter variable
   Y = 1
```

```
    # For loop to perform analysis for multiple buffer distances
    for X in arr:
        # Creates buffer to specified radius around all input points
        arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

        # Intersects safety incidents with buffers
        arcpy.Intersect_analysis([output, "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers",
                                 "ALL", "", "")

        # Creates new output table name
        myDir = "in_memory/"
        New_Path = "{}{}".format("Table_", Y)
        New_Out = os.path.join(myDir, New_Path)

        # Sums safety incidents by station
        arcpy.Statistics_analysis("in_memory/Intersected_Buffers", New_Out,
[[Analysis_Var, Analysis_Type]],
                                  "Station_Name")
        Incident_SUM = "{}{}{}".format("Incidents: ", X, " Buffer")
        arcpy.AlterField_management(New_Out, "SUM_COUNT_BICYCLIST_KILLED",
Incident_SUM, Incident_SUM)
        arcpy.DeleteField_management(New_Out, "FREQUENCY")

        print("{}{}".format("Finished calculating incidents at ", X))

        Y += 1

    # Merge tables by station
    arcpy.env.workspace = "in_memory"

    tables = arcpy.ListTables()
    print(tables)

    for table in tables:
        if table == "Table_1":
            pass
        else:
            arcpy.JoinField_management("Table_1", "Station_Name", table,
"Station_Name")
            arcpy.DeleteField_management("Table_1", "Station_Name_1")

    # Creates final Excel output
    myTable = "in_memory/Table_1"
    TableOut2 = os.path.join(TableOut, FileName)
    arcpy.TableToExcel_conversion(myTable, TableOut2)
    print("EXPORTED TABLE TO EXCEL")

    print("FINISHED:")

except Exception as err:
    print(err.args[0])
```

Script 9. Census Data – Commute Mode Split

This python script calculated weighted averages for census data and is used to calculate the average commute mode splits for each station catchment area. To run, the script requires a Census geodatabase, which can be downloaded here. The input data used by this script is provided and can be accessed in Appendix 2.

```
##################################################################################
# Script 9: Census Data - Commute Mode Split
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Variable codes for different commute modes
# Walk: B08101e33
# Transit: B08101e25
# Bike (+ other): B08101e41
# Car B08101e9
# ALL: B08101e1

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".5 Mile", ".25 Mile"])
CensusData = "X08_COMMUTING"
CensusTableField = "Walk_Prop"
POINT = "CA_Stations"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
myGDB =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
CensusPath =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
ACS_2018_5YR_TRACT_06_CALIFORNIA.gdb"
FileName = "Auto_Commute_Prop.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Defines geometry for Census Tracts
TIGER = os.path.join(CensusPath, "ACS_2018_5YR_TRACT_06_CALIFORNIA")
```

```
# Creates a copy of the TIGER Geometry
arcpy.CopyFeatures_management(TIGER, "in_memory/TIGER_Temp")

# Joins specified table to TIGER Geometry
DATETABLE = os.path.join(CensusPath, CensusData)
arcpy.JoinField_management("in_memory/TIGER_Temp", "GEOID_Data", DATETABLE,
"GEOID", ["GEOID_Data",

  "B08101e33",

  "B08101e25",

  "B08101e41",

  "B08101e1",

  "B08101e9"])

# Calculates proportion of commuters for given mode
arcpy.AddField_management("in_memory/TIGER_Temp", "Walk_Prop", "DOUBLE", "", "",
"", "Walk_Prop", "NULLABLE", "")
# Modify the commute mode codes according to the guide at the top of the script
arcpy.CalculateField_management("in_memory/TIGER_Temp", "Walk_Prop", "(!B08101e9!
/ !B08101e1!)", "PYTHON")

# Defines counter variable
Y = 1

# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

   # Calculate the area of buffers
   arcpy.AddField_management("in_memory/POINT_BUFFER", "AREA", "DOUBLE", "", "",
"", "AREA", "NULLABLE", "")
   inTable = "in_memory/POINT_BUFFER"
   fieldName = "AREA"
   expression = "!shape.area@squaremiles!"
   arcpy.CalculateField_management(inTable, fieldName, expression, "PYTHON")

   # Intersects TIGER Geometry with buffers
   arcpy.Intersect_analysis(["in_memory/TIGER_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Calculates area of intersected TIGER Geometry
   arcpy.AddField_management("in_memory/Intersected_Buffers", "NEW_AREA", "DOUBLE",
"", "", "", "NEW_AREA", "NULLABLE", "")
   inTable2 = "in_memory/Intersected_Buffers"
   fieldName2 = "NEW_AREA"
   expression2 = "!shape.area@squaremiles!"
   arcpy.CalculateField_management(inTable2, fieldName2, expression2, "PYTHON")

   # Calculates proportion or new area
```

```
   arcpy.AddField_management("in_memory/Intersected_Buffers", "New_Area_Prop",
"DOUBLE", "", "", "", "New_Area_Prop", "NULLABLE", "")
   arcpy.CalculateField_management("in_memory/Intersected_Buffers",
"New_Area_Prop", "(!NEW_AREA! / !AREA!)", "PYTHON")

   # Calculates new prop
   arcpy.AddField_management("in_memory/Intersected_Buffers", "New_Walk_Prop",
"DOUBLE", "", "", "", "New_Walk_Prop", "NULLABLE", "")
   arcpy.CalculateField_management("in_memory/Intersected_Buffers",
"New_Walk_Prop", "(!New_Area_Prop! * !Walk_Prop!)", "PYTHON")

   # Creates new output table name
   myDir = "in_memory/"
   New_Name = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Name)

   # Calculates weighted proportion
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", "in_memory/Area_SUM",
[["New_Area_Prop", "SUM"]], "Station_Name")
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", "in_memory/Walk_SUM",
[["New_Walk_Prop", "SUM"]], "Station_Name")

   arcpy.JoinField_management("in_memory/Area_SUM", "Station_Name",
"in_memory/Walk_SUM", "Station_Name")

   arcpy.AddField_management("in_memory/Area_SUM", "Adj_Walk", "DOUBLE", "", "",
"", "Adj_Walk", "NULLABLE", "")
   arcpy.CalculateField_management("in_memory/Area_SUM", "Adj_Walk",
"(!Sum_New_Walk_Prop! / !Sum_New_Area_Prop!)", "PYTHON")

   arcpy.TableToTable_conversion("in_memory/Area_SUM", "in_memory", New_Name)

   PROP_SUM = "{}{}{}".format("Proportion: ", X, " Buffer")
   arcpy.AlterField_management(New_Out, "Adj_Walk", PROP_SUM, PROP_SUM)
   arcpy.DeleteField_management(New_Out, ["FREQUENCY",
                                          "SUM_New_Area_Prop",
                                          "Station_Name_1",
                                          "Frequency_1",
                                          "SUM_New_Walk_Prop"])

   arcpy.Delete_management("in_memory/Area_SUM")
   arcpy.Delete_management("in_memory/Walk_SUM")

   print("{}{}".format("Finished calculating proportion at ", X))

   Y += 1

# Merges tables by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
```

```
    if table == "Table_1":
        pass
    else:
        arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
        arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)
print("EXPORTED TABLE TO EXCEL")
print("FINISHED:")
```

Script 10. Intersection Density

This python script calculates the intersection density of station catchment areas. To run, two pieces of input data are required. First, a roadway network is required to define intersections. A comprehensive Open Street Map roadway network for California can be downloaded here. Next, another python tool must be used to create intersections from the roadway network data. This script, called the *Line and Junction Connectivity* tool (Beale, 2012) can be downloaded as an ArcGIS tool here. Once this tool is run, its output serves as the input data for the script discussed in this section. Intersection data used by this script is provided and can be accessed in the StationArea geodatabase accessible in Appendix 2.

```
##################################################################################
# Script 10: Intersection Density
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".5 Miles", ".25 Miles"])
POINT =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb\CA_Stations"
Analysis_Var = "NUM"
Analysis_Type = "SUM"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
ID = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb\Intersections"
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
FileName = "Intersection_Density.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Creates a copy of the intersections feature class
arcpy.CopyFeatures_management(ID, "in_memory/ID_Temp")

# Defines counter variable
Y = 1
```

```
# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)

   # Intersects intersections with buffers
   arcpy.Intersect_analysis(["in_memory/ID_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Creates new output table name
   myDir = "in_memory/"
   New_Path = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Path)

   # Sums intersections by station
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", New_Out,
[[Analysis_Var, Analysis_Type]], "Station_Name")
   ID_SUM = "{}{}{}".format("ID: ", X, " Buffer")
   arcpy.AlterField_management(New_Out, "SUM_NUM", ID_SUM, ID_SUM)
   arcpy.DeleteField_management(New_Out, "FREQUENCY")

   print("{}{}".format("Finished calculating intersection density at ", X))

   Y += 1

# Merges tables by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
   if table == "Table_1":
       pass
   else:
       arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
       arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)
print("EXPORTED TABLE TO EXCEL")
print("FINISHED:")
```

Script 11. Points of Interest (POIs)

This script calculates the number of points of interest within each station catchment area. The input data necessary to run the script can be downloaded here as a shapefile. For the analysis presented in this report, all points of interest were used. Input data used by this script is provided and can be accessed in the StationArea geodatabase accessible in Appendix 2.

```
##################################################################################
##
# Script 11: Points of Interest (POIs)
# *** Replace all instances of 'YourName' in file paths to your username ***

# Import necessary packages
import arcpy
import os
import numpy as np

# Define analysis variables
# Add as many station buffer radii as desired. Must be arranged from largest to
smallest.
arr = np.array([".5 Miles", ".25 Miles"])
POINT =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb\CA_Stations"
Analysis_Var = "VALUE"
Analysis_Type = "SUM"

# Defines file paths and directory
arcpy.env.workspace =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb"
POI =
"C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\SpatialData\
StationAreaData.gdb\POIs"
TableOut = "C:\Users\YourUsername\Desktop\HenryMcKay_SeniorProject_Code\Data\
ExcelOutput_Python"
FileName = "POI_Density.xls"

# Overwrites existing output if name is the same
arcpy.env.overwriteOutput = True

# Creates a copy of the POI points
arcpy.CopyFeatures_management(POI, "in_memory/POI_Temp")

# Defines counter variable
Y = 1

# For loop to perform analysis for multiple buffer distances
for X in arr:
   # Creates buffer to specified radius around all input points
   arcpy.Buffer_analysis(POINT, "in_memory/POINT_BUFFER", X)
```

```
   # Intersects POIs with buffers
   arcpy.Intersect_analysis(["in_memory/POI_Temp", "in_memory/POINT_BUFFER"],
"in_memory/Intersected_Buffers", "ALL", "", "")

   # Creates new output table name
   myDir = "in_memory/"
   New_Path = "{}{}".format("Table_", Y)
   New_Out = os.path.join(myDir, New_Path)

   # Sums POIs by station
   arcpy.Statistics_analysis("in_memory/Intersected_Buffers", New_Out,
[[Analysis_Var, Analysis_Type]], "Station_Name")
   ID_SUM = "{}{}{}".format("ID: ", X, " Buffer")
   arcpy.AlterField_management(New_Out, "SUM_VALUE", ID_SUM, ID_SUM)
   arcpy.DeleteField_management(New_Out, "FREQUENCY")

   print("{}{}".format("Finished calculating intersection density at ", X))

   Y += 1

# Merges tables by station
arcpy.env.workspace = "in_memory"

tables = arcpy.ListTables()
print(tables)

for table in tables:
   if table == "Table_1":
       pass
   else:
       arcpy.JoinField_management("Table_1", "Station_Name", table, "Station_Name")
       arcpy.DeleteField_management("Table_1", "Station_Name_1")

# Creates final Excel output
myTable = "in_memory/Table_1"
TableOut2 = os.path.join(TableOut, FileName)
arcpy.TableToExcel_conversion(myTable, TableOut2)
print("EXPORTED TABLE TO EXCEL")
print("FINISHED:")
```

Appendix 3. Additional GIS Information

Coordinate System

For all GIS Analysis, the NAD 1983 California (Teale) Albers (US Feet) Coordinate system was used.

Accessibility Isochrones

The only piece of data manually created for this analysis was the accessibility isochrones discussed in section three. While automating the process would not be highly difficult, isochrones are highly error-prone, and require careful observation to be created correctly. ESRI's Network Analyst extension was used to generate isochrones, which can be read more about here. Open Street Map roadway network data was used as the input data for the isochrones and can be downloaded here.

Appendix 4. Web Content

Two important components of this project are web-based. This means that these products will continue to be updates as data is updated and as new features are added. A copy of the Google Sheets tool discussed in section six of this report can be accessed at the following link:
https://docs.google.com/spreadsheets/d/1HwoMzF_gV8wKpQP47mQxdMuV7nxerDwmZ9gUx2Vh-Uo/copy?usp=sharing

An ArcGIS StoryMap showing all the geospatial data discussed in this report in an interactive map format can be viewed at the following link:
https://arcg.is/0vb9H4

A zipped folder containing the scripts used in this project and their necessary input data can be downloaded at the following link:
https://drive.google.com/file/d/1AhdwDL9IcRpsKffPbo3q_1-oqoCIz4O3/view?usp=sharing