CE 295 Final Report

Newport Beach Tsunami Evacuation Optimization

May 10, 2019

Li-Wei Chen, Jeremy Kho, Zining Wang, Dihan Yang

Abstract

Coastal cities are often prone to flooding due to hurricanes, tsunamis caused by earthquakes, or other natural disasters. In particular for California, there is a high risk of tsunamis caused by Pacific Rim earthquakes, a simulated 9.1 magnitude earthquake in Alaska could cause a five to ten foot storm surge as far south as San Diego in four hours [1]. During these events, city managers have limited time to implement evacuation routes for the displaced population, which can be highly variable due to beach tourism. Our project's focus is the Balboa Peninsula at Newport Beach, an area with a high tourist population, and limited egress. First, we analyzed signal probe data to estimate the population in the area to include day tourism and their likely routes. Next, we applied a traffic model to determine the number of lanes and mix of pedestrian and vehicle traffic to safely evacuate the entire peninsula's population via multimodal methods in under four hours. We were able to recommend lane usage patterns to safely evacuate both a July 4th peak tourism scenario and a summer base case scenario.

1.0 Introduction

1.1 Motivation and Background

Flooding poses a significant risk to coastal cities through the occurrence of tsunamis, heavy rain, or other natural disasters. Specifically, in Los Angeles County, low lying populations in Newport Beach, Marina Del Rey, and Long Beach are vulnerable to tsunamis. Historical examples of tsunami damage include the 1964 Alaskan Earthquake which caused 10 deaths in California, or the more recent harbor damage from the 2010 Chilean Earthquake and 2011 Fukushima Earthquake [1]. From these distances the resulting tsunamis would generate smaller storm surges in the 3 to 10 feet range, but due to the high degree of urbanization near the coast, major flooding would occur affecting over 750,000 people within a few hours [2,3]. Many communities in the Los Angeles Basin have tsunami evacuation plans [4], however, it is unclear if the selected evacuation routes have the throughput capacity to support the evacuating population.

Balboa Peninsula in Newport Beach offers several unique challenges. First, as shown in Figure 1, Balboa Peninsula in Newport Beach only has two major roads leading out of the area. A five foot high storm surge inundate most of the populated areas in the Balboa Peninsula, the majority of which are on islands created by dredging and infill. The peninsula is also highly tourism dependant, with peak populations on the beach in summer months creating additional evacuation strain. It is unclear if the two major transportation routes, as marked by the two red arrows on the top left corner of figure 1, will have enough carrying capacity as their design parameters are not focused on evacuation scale throughput [5]. One other wrinkle in the estimation of evacuees is their mode of evacuations. It may be faster for pedestrians, and people able to abandon their vehicles to attempt to evacuate on foot instead of by vehicle.

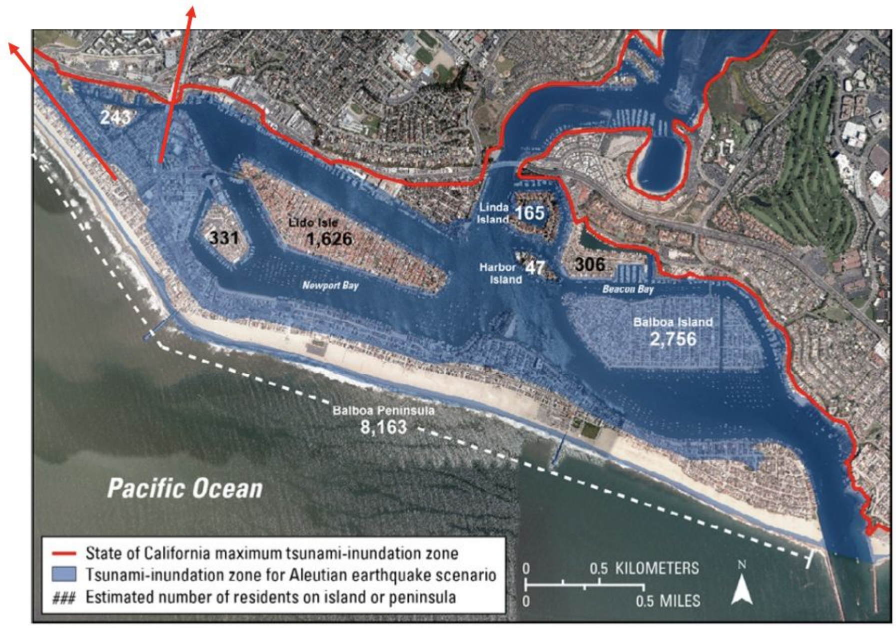

Figure 1. Newport Beach Tsunami Inundation Zone with Evacuation Routes Marked

Unfortunately, city managers have traditionally not been able to access discrete data that can provide them with reliable estimates of population densities and what type of transportation the population is using. The Newport Beach census data only provides information on residents and does not have an accurate estimate of the population on the beaches. However, our team's approach will be to estimate transient population and mode of transport based on the influx of GPS probe points for the period of 2 to 6 July 2018. A visual representation of the probes' density is shown below in Figure 2, which approximates the shape of Balboa Peninsula well.

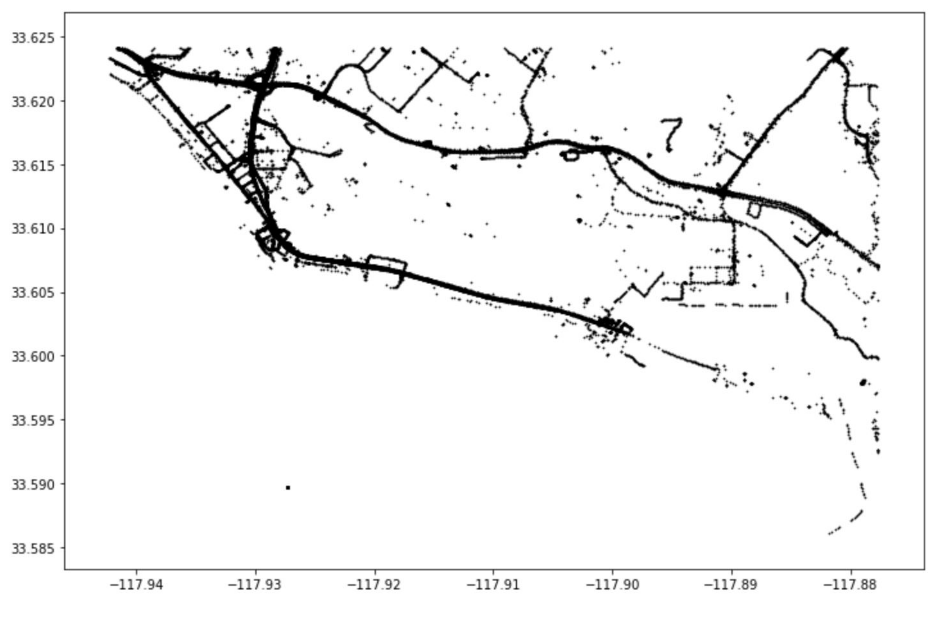

Figure 2. Newport Beach Area Probe Data Provided by CE 263N, 42,785 Data Points

1.2 Relevant Literature

Our team was primarily motivated by the United States Geological Survey Science Application for Risk Reduction published in 2013. The study had broad conclusions for coast communities in California, and particularly highlighted Santa Monica, Marina Del Rey, and Newport beach as communities that could be highly affected with displaced population in the event of a tsunami [2].

One of the limitations indicated in the study was their inability to measure the entire population affected for evacuation planning. In particular, while the researchers were able to use U.S. Census data to baseline population estimates in Balboa peninsula, they were unable to find a reliable way to estimate beachgoers. Currently, tourists in Newport Beach are estimated by volunteer lifeguards and estimates vary widely up and down the coast, and for the USGS study, they estimated 10,000 transients. However, historical police records in Newport Beach indicate that during major holidays almost 100,000 visit Balboa Peninsula [6].

While multiple cities on the California coast have published their tsunami evacuation plans, it is not immediately clear how many of them had rigorously tested their plan based on the physical constraints such as road capacity and bottlenecks such as bridges [5]. Of particular interest was another study done by the US Geological Survey specifically for Alameda, California. In this evacuation scenario, those researchers primarily focused on pedestrian evacuation to safely remove the displaced population from the tsunami hazard [7]. Vehicles were omitted from their simulations due to the limited egress routes off of Bay Farm Island. However, our team felt that focusing on pedestrians would be a significant limitation as it provides limited options for elderly or disabled evacuees unable to make the approximately 4 mile walk off of Balboa Island. Therefore we felt an opportunity to expand on the previous study would be to include pedestrian and vehicle modelling along, to determine an optimal lane usage for the two main routes off of Balboa Peninsula which would allow the entirety of the population to evacuate.

1.3 Focus of this Study

We will use GPS probe data to estimate population density in the Newport Beach area for the July 4th holiday week. Based on the population distribution, we will develop an optimal lane usage framework for Balboa Avenue and Newport Avenue to maximize flow to enable all residents and visitors to Balboa Peninsula to evacuate safely in under four hours while minimizing vehicle losses.

2.0 Technical Description

2.1 Optimization Model

Our model will rely on several assumptions prior to implementation. First, the evacuation speed of pedestrians or vehicles will be constant (time invariant) for the scope of the project. Additionally, we assume that all evacuees will follow Newport Beach's evacuation plans and only follow Balboa and Newport Boulevard off of Balboa Peninsula as described in Newport Beach's evacuation plan [8]. While some pedestrians could use two very limited side streets at the edge of the evacuation error we considered those effects minimal.

We also assumed all roads in Balboa Peninsula will be be opened to only outbound flow, and no traffic would be allowed in the opposite direction of evacuation. While there are multiple dedicated turning lanes, we restricted the lane usage to only four lanes per road. Finally, the population and traffic density in the area beyond the tsunami inundation zone is considered to approach zero, or in other words, there would be no congestion at the edge of our model.

Our model's focus areas will be West Balboa Avenue and Newport Avenue only after the split from Balboa Avenue and will not analyze other sections of the road or other choke points on the island. The goal of the model is to allow for maximum evacuee flow to safely evacuate the whole population on island in under four hours while maximizing vehicle traffic. Our team's motivation to use vehicle traffic as our cost function was twofold, first from a policy standpoint it would be easier to convince residents to evacuate via their cars vice walking, and second, the abandoned vehicles could serve as a property damage indicator.

The models objective function is given in Equation 1, to maximize the vehicles evacuated that travel on Balboa Avenue (A) and Newport Avenue (B). This model is a convex equation with a clear tradeoff between lanes dedicated for pedestrian and vehicle usage per lane, as well as the population assignment to each overall evacuation route.

$$V_{evac} = \sum_{r \varepsilon \{A,B\}} N_{lanes,r} \times Q_{veh,r} \times T_{evac,r} \tag{1}$$

Equation 2 is an expansion of Equation 1 to outline specific relationship for the two roads, and the number of lanes l_A and l_B that are dedicated to pedestrian traffic on Balboa and Newport Avenue respectively.

$$V_{evac} = (4 - l_a) \times Q_{veh,A} \times T_{veh,A} + (4 - l_b) \times Q_{veh} \times T_{veh,B} \tag{2}$$

The objective functions are subject to the following constraints and variable definitions. Each constraint is a separate equation per road, however, for convenience only one equation is shown covering both road cases:

First the pedestrians evacuated via Balboa and Newport Avenue are a function of the flow, dedicated lanes, and evacuation time.

$$P_{ped\ A,B} = (l_{A,B}) \times Q_{ped\ A,\ B} \times T_{ped\ A,B} \tag{3}$$

Next, the vehicles evacuated are also modeled in the same fashion with the number of dedicated vehicle lanes of out the along with an estimated passenger count which was set at five.

$$P_{veh\ A,B} = (4 - l_{A,B}) \times Q_{veh\ A,\ B} \times T_{veh\ A,B} \times Passengers \tag{4}$$

The total population evacuated by road is the sum of the two evacuation modes.

$$P_{out\ A,\ B} = P_{ped\ A,B} + P_{veh\ A,\ B} \tag{5}$$

Next, the population to be evacuated is defined by the different zones on Balboa Island and their likely evacuation routes based on geographical constraints shown in Figure 3.

Figure 3. Newport Beach Area Segmentation

Population evacuating via Balboa Avenue (A) is function of Zone 1 and a percentage of population, given by *k*, from Zone 4.

$$P_{in,A} = P_1 + k \times P_4 \tag{6}$$

Population evacuating via Newport Avenue (B) is the sum of population in Zones 2 and 3, along with the population in Zone 4 not evacuating via Balboa Avenue.

$$P_{in,B} = P_2 + P_3 + (1 - k) \times P_4 \tag{7}$$

Finally, the population in each zone is related to the evacuation capacity, specifically combining equations 5, 6, and 7 so that all evacuees assigned to a certain road will match the population evacuated.

$$P_{in\,A,B} = P_{out,A,B} \tag{8}$$

Additionally physical constraints are added, namely the evacuated vehicles must be less than or equal to the total number of vehicles on island estimated from the 2010 census

$$V_{evac} \le 16900 \tag{9}$$

The time for all evacuations modes must be less than 4 hours, and more than 0.5 hours and 2 hours for vehicles and pedestrians respectively. The lower constraint represents the estimated average time of vehicle or pedestrian travel from the Eastern end of Balboa peninsula off island.

$$0.5 \le T_{veh\,A,B} \le 4 \tag{10}$$
$$2 \le T_{ped\,A,B} \le 4 \tag{11}$$

We then bound the Zone 4 population percentage range, and then discretized the number of lanes for each evacuation mode.

$$0 \le k \le 1 \tag{12}$$

$$l_a,\ l_b \in \{0, 1, 2, 3, 4\} \tag{13}$$

A summary of the variables and parameters are shown in the following table, along with the data source and method for determining the values.

Optimization Variables	**Data Source** **Method**
l_r	Pedestrian Lanes, Optimized
k	Population of Zone 4 traveling on Balboa Avenue, Optimized, Baselined by **LA Probe Data** using **trajectory splitting**
Model Parameters (data driven)	
$Q_{veh,r}$	Vehicle flow. Estimated from **Macroscopic Fundamental Diagram** from slowest hourly average vehicle speeds from **LA Probe Data**
$Q_{ped,r}$	Pedestrian flow. Estimated from **Cheng 2012** simulating subway station evacuation
P_i	Population. Estimated from **2010 Census** expanded by **LA Probe Data**
Constraint Parameters (scenario driven)	
$T_{veh,r}$	Vehicle evacuation time. Limited to 4 hour maximum based on **USGS Scenario**
$T_{ped,r}$	Pedestrian evacuation time. Limited to 4 hour maximum based on **USGS Scenario**
V_{avail}	Vehicles on island. Estimated based on **2010 Census** vehicles per household (1.69/hh)

Table 1. Optimization and Parameter Variables, Origin and Description

After finishing our model for evacuation, the team then analyzed the LA probe dataset to obtain the data needed as input parameters.

2.2 Flagging/Data Cleaning

Our team then preprocessed the LA probe data using a variety of techniques and heuristics taught in CE 263N. Specifically, erroneous speeds (in excess of 160 kph) were corrected, gaps in data were either discarded or linearly interpolated, data points outside of our model's physical limits in Newport Beach were removed, and duplicated data was deleted [9].

2.3 Map Matching

To facilitate data analysis, we first split the target area into four sections as previously described in Section 2.1 “Optimization model,” and for convenience show it again here.

Figure 3. Newport Beach Area Segmentation

The areas chosen are based on proximity to the three major roads and geographical barriers such as bodies of water. We then matched the probe data to West Balboa Avenue (red) and Newport Avenue (green)leading out of the Balboa Peninsula as well as the rest of Balboa Avenue running through the entirety of the peninsula as shown in blue in figure 4.

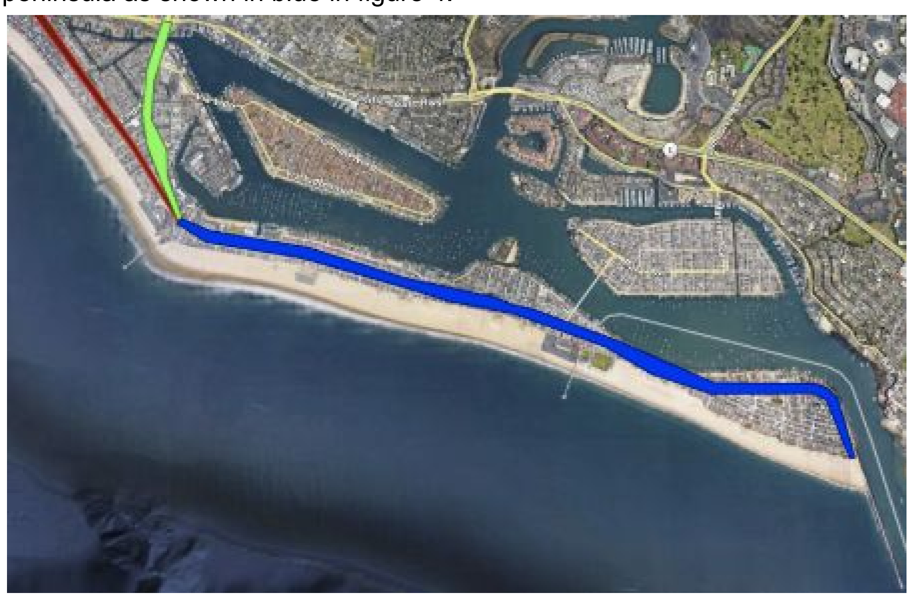

Figure 4. Probe data matching to target roads

As we were not able to find shapefiles for the roads in the area, we manually created shapefiles around the target areas in Google Earth. We then used techniques used in Assignment 0 of CE 263N to determine if a certain probe data point falls within one of our shapefiles. The python library "Shapely" is used to convert the coordinates of each probe into a Shapely Point object. Each of these point objects are then matched to the shapefile that it belongs to and labelled accordingly for easier identification. A snippet of the labeled dataframe is shown in table 2.

SPEED	PROBE_DATA_PROVIDER	LOCAL_TIME	coord	section	street
16.0	FLEET51	2018-07-01 17:00:59-07:00	POINT (-117.9194113 33.60690579999999)	4	w_balboa
0.0	FLEET51	2018-07-01 17:02:29-07:00	POINT (-117.9201456 33.607024)	4	w_balboa
18.0	FLEET51	2018-07-01 17:02:45-07:00	POINT (-117.9203801 33.6070871)	4	w_balboa
3.0	CONSUMER21	2018-07-01 17:03:33-07:00	POINT (-117.92847 33.60822)	4	N/A
4.0	CONSUMER21	2018-07-01 17:03:38-07:00	POINT (-117.9284 33.60815)	4	N/A

Table 2. Snippet of dataframe containing labelled probe data points

The dataframe is labelled by the section and street the point belongs to. A street label of "N/A" signifies that the point does not belong to any of the three targeted roads, and was then excluded from our speed and density analysis.

2.4 Trajectory Segmentation

To further structure our data, we looked into splitting the data collected per person per day into multiple trips. Instead of one trip that took an entire day to finish, we would have multiple trips that the person took that day. To do so, we split the data into separate chunks if the time between the points was longer than 2 minutes. The 2 minute interval was chosen at random but is reasonable for the most scenarios where drivers would stop and then head out for a new destination or if the source of the probe data stopped transmission. Trajectory segmentation allowed us to have a more realistic representation of the traffic patterns and to have more accurate values for average speeds.

Our main focus for trajectory splitting is on which exit road someone leaving the peninsula is more likely to take, W Balboa Blvd or Newport Blvd. Based on the paths identified per probe identification value, we are able to determine the origin and destination of each trip that each probe id took. We then focused on trips that originated from section 4 shown in Figure 3 and ended in one of either W Balboa Blvd or Newport. Through this analysis, we determined that when leaving the peninsula, 64% of travelers would normally choose Newport Avenue over W Balboa Avenue. A visual representation of the increased traffic flow is shown below in figure 5.

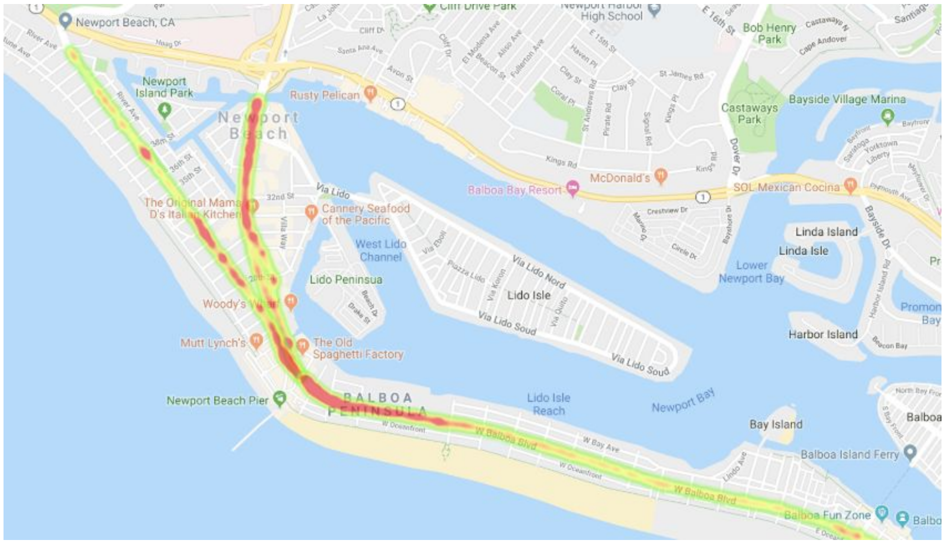

Figure 5. Heatmap of probe densities showing more usage for Newport Avenue

2.5 Average Speed

The average speed of vehicles based on the probe data is needed in order to determine the amount of congestion experienced by the two target roads under normal congestions. We first segmented the data based on the prelabeled roads. We then separated the probe data based on heading. For W Balboa Blvd, we assume that any probe data with a heading value of more than -90 degrees and less than 45 degrees is moving off of the peninsula and any probe data with a value between 145 and -150 degrees is coming onto the peninsula. With Newport Blvd, we assume that any probe data between -140 and 0 is driving off the island and data points with a heading between 90 and -160 degrees is coming onto the peninsula. These values are derived off the average orientation of the roads. We then calculated the average speed of each probe in the targeted road over an hour. Any probes with a mean speed greater than 15 km/hr is assumed to be a vehicle and any that is less than is assumed to be a pedestrian either walking or biking.

Through this analysis, we were able to determine the number of probes and average speeds of probes as sorted by hour, road, trajectory, and vehicle or pedestrian. A snippet of the resulting DataFrame can be seen in table 3.

Hour	Newport_North_Speed_Veh	Newport_South_Speed_Veh	Newport_North_Probe_Veh	Newport_South_Probe_Veh
0	0.000000	0.000000	0	0
1	59.000000	0.000000	1	0
2	30.640351	28.653333	3	3
3	41.842840	41.842840	1	1
4	37.374720	49.000000	2	1

Table 3. Snippet of average speed dataframe

2.6 Calculation of flow rate

2.6.1 Vehicle Flow Rate

Now that the team had determined the worst case congested flow rates for both Newport Beach and Balboa Avenues, we next focused on determining the vehicle flow rate. Initially the team was planning to count probes in order to estimate vehicle densities, but the GPS penetration rate of the probes was too low to make an accurate count. The team instead focused on applying research relating vehicle speed to flow rate.

Vehicle Flow Rate (cars per second) is defined by the following equation [10]:

$$Q = k \bullet V \tag{14}$$

where *k* is the density and *V* is the speed of the mode of transportation. Based on data measurements in the urban core of San Francisco, a macroscopic fundamental diagram can be developed to relate each of the terms in the equation to the other [10]. This relationship can be visualized through the fundamental diagrams shown in figure 6 below.

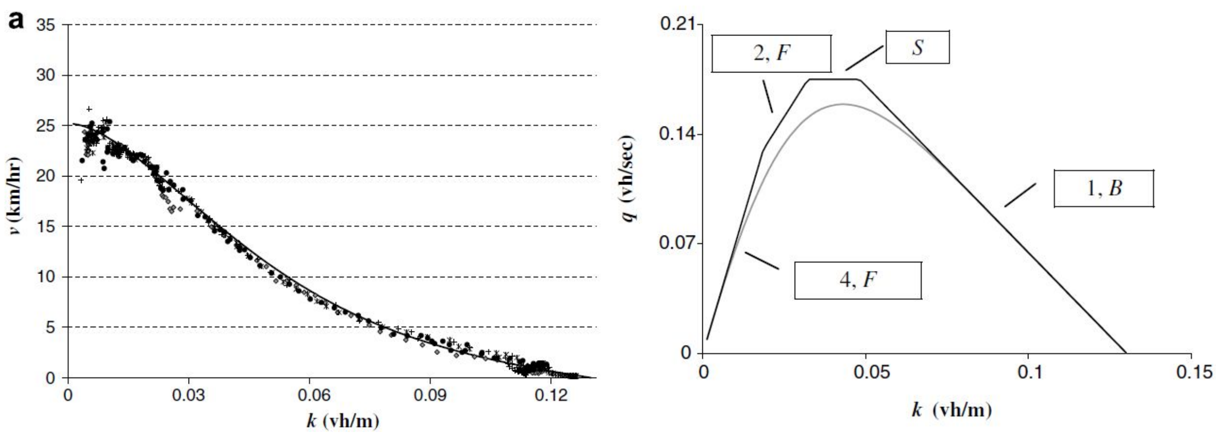

Figure 6. Relationship of vehicle density (k) to vehicle speed (V) and vehicle flow, San Francisco

Specifically in the 2008 study, San Francisco's urban core was analyzed, and after reviewing the number of intersections and lanes, the team believed this model would be a good approximation for Newport Beach's urban core on Balboa Peninsula as well.

Based on our average speed results we then extrapolated our estimates for vehicle flow with results shown below.

Road	Average Speed (kph)	Time of Day	Direction	Estimated Flow (veh/s)
Newport	16	1:00PM	North	0.034
Newport	15.3	12:00PM	South	0.037
Balboa	15	4:00PM	North	0.037
Balboa	15	4:00PM	South	0.037

Table 4. Congested Speed and Flow Estimation Results

2.6.2 Pedestrian Flow Rate

Initially the team planned to measure pedestrian velocity and density using the probes identified as pedestrians based on their behavior, however, the team realized those probes would not likely be exhibiting congested behavior similar to an evacuation. Using a series of subway pedestrian evacuation studies in China, the team selected one study that measured and modeled pedestrians in a 3.5 meter hallway most similar to a road lane. The measurements showed that the peak congested speed for a 3.5 meter wide passageway is 0.502 meters per second at a density of 3.132 people per meters squared [11]. This resulted in a flow of 1.57 people per second and is modeled by the following polynomial equation, where *V* is speed and k is pedestrian density.

$$V(k) = -0.00056k^4 + 0.0009k^3 + 0.0008k^2 - 0.4242k + 1.8267 \qquad (15)$$

2.7 Population estimation

In order to solve the defined optimization problem, we need to determine the population to be evacuated. While census data is available to quantify the resident population, it does not take into account the transient population, of which Balboa Peninsula has plenty. As mentioned in the review of related literature, police roughly estimate a total population of 100,000 during holidays due to the surge in tourist activity. On the other hand, we have access to unlabeled GPS probe data from various sources and devices throughout the Los Angeles Metro Area. The key limitation here, however, is that we do not know the penetration rate of the data, which is protected by intellectual property rights.

In essence, the methodology to estimate population is fairly simple. The trajectories of GPS devices (distinguished in the data through unique IDs) are determined to be either residents or non-residents. Once this is achieved, we assume that the proportion of residents in the GPS data would be a near approximation of the actual proportion of residents to the population. Since we have census data, we can obtain an estimate for the total population.

2.7.1 Identifying Resident GPS Data

The key then is to identify which of the trajectories in the GPS data belong to residents. Based on a priori expectations, we can make the simple assumption that trajectories which terminate in Balboa Peninsula at night (from dusk to dawn) are more likely than not to be residents. However, it is important to note that the neighborhood is particularly affluent and residents going home in the afternoon is highly plausible. These would not be included in the initial setup since the stream of GPS data ceases once the navigation device or application has been turned off. Hence, we expand the window to be from noon to dawn of the following day. If a trajectory enters the area after noon and does not have an outbound trip until 6:00 AM, we assume that they are residents.

With this framework, one might think to stretch this window from 6:00AM of each day to 6:00AM of the following day, as long as no outbound trip was made. While subjective, this restriction was placed to reduce the possibility of false positives, as homebound trips during that time are less likely and some transient visitors may possibly use navigation systems inbound but not outbound. Regardless, setting the interval start time to noon provides a more conservative estimate. False positives overrepresent the residents and reduce the estimate for the total population. Because we are dealing with an evacuation scenario, it would be better to overestimate the population, making false negatives the preferred error.

Utilizing data on 6 consecutive nights, from the 1st of July to the 6th of July, there were 32.67 residents on average per night represented in the GPS data. It's important to note that not all the residents are necessarily the same for each night, since navigation systems are not always used and people can have intermittent behaviors. However, the average representation of the residents is what is needed to derive population estimates. For the following section on population expansion, this number was simplified to 33.

2.7.2 Population expansion

We based our estimation on the 2010 US Census Data for Newport Beach [1]. As shown in figure 1, we were able to estimate the residential population distribution to correspond with our four evacuation regions P1, P2, P3, P4, respectively, as 100, 143, 1957, 8163, with 10363 overall residents on Balboa Peninsula. Given there were 33 probes identified in Section 2.7.1 as likely residents, we determined a ratio of 1:314, to be used as an expansion factor to represent probes to actual residents in the region.

The overall number of probes on July 4th was 285. Applying the expansion factor on this day, we estimate the total maximum number of people on Balboa Peninsula was 89,490, which is very close to historical estimates in the region of July 4th visitors and residents [6]. We then took the estimated population and separate it by zone to match the census data proportions. The estimated population on July 4th by zone was used as the population parameters for our program and is shown in Table 5.

Zone	Average probe number residents and visitors	Census residents by zone		July 4th probe number residents and visitors	July 4th population estimation by zone	
1	33 residents 110 visitors	Total Census Population 10363	100	65 residents 220 visitors	Total Estimated Population 89490	865
2			143			1235
3			1957			16900
4			8163			70490

Table 5. Resident estimation via census and probe data

3.0 Discussion

Our simulations were run using the CVX solver add on to MATLAB 2018A on a 2014 MacBook using macOS 10.14 Mojave installed.

3.1 July 4th peak visitors results

For the July 4th scenario evacuating 89,490 total residents and visitors off of Balboa Peninsula, the team found the optimal solution would be to open Balboa Avenue fully to vehicle traffic, and to have only pedestrians travel on West Newport Avenue. A total of 2131 cars would be used to travel off island, and 14% of Section 4's population would need to travel on Balboa Avenue, compared to 36% of trips currently made off island via Balboa Avenue.

These results seemed reasonable as our model had a slightly higher flow rate on Balboa than Newport Avenue. Additionally, the pedestrian evacuees only require three hours to safely evacuate, whereas the vehicles evacuate up to the four hour mark. Vehicle only evacuation solutions were all infeasible.

3.2 Summer base case results

In the case of a Summer base case analysis, we used the population data from the week of July 2nd through July 6th, excluding the July 4th holiday. The optimum solution in this case would be to have Balboa Avenue open to vehicle traffic only, along with one lane for vehicles in Newport Avenue. Three lanes in Newport Avenue would be dedicated to pedestrians.

Since there is less total population to evacuate, more cars are able to leave Balboa Peninsula compared to the July 4th case, with the optimal solution having 2631 cars evacuated and 29% of the population leaving via Balboa Avenue. The timeline for evacuation is the same as the July 4th scenario with four hours needed to evacuate the vehicles and three hours to evacuate the pedestrians. These results are intuitive as the lower total evacuation population of 44,745 people allows for less dense modes of transport to be used. Vehicle only evacuation solutions were all infeasible.

3.3 Future work expansion

For future work expansion our team identified three major areas that would enhance our model.

First, we would examine other points along the route for their flow rates. Instead of just focusing on the exits out of Balboa Peninsula, there may exist local choke points e.g. the Lido Island bridge, which may actually be the main constraint to evacuating one of our population zones. The evacuation model could also be expanded north of the Pacific Coast Highway and account for an increasing evacuee density that would back propagate and affect the flow off of Balboa Peninsula.

Another area of expansion could be determining population density on Balboa Peninsula based on the time of day. While our team was able to estimate a worst case scenario, the distribution of probe data shown in figure 7, indicates that holiday visitors come in two waves and later than the average visitors to Newport Beach.

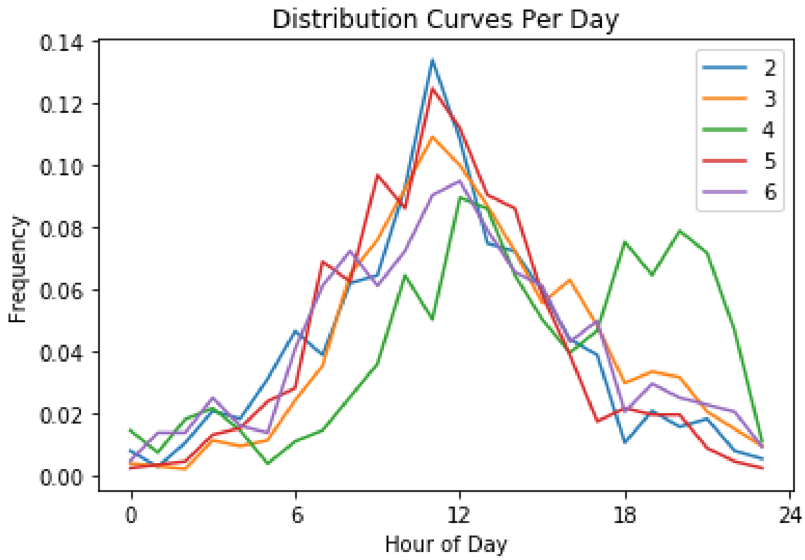

Figure 7. Normalized probe frequency over time of day, July 2nd to July 6th 2018

The varying population distribution could also be applied to the Macroscopic Fundamental Diagram and evacuation scenarios could account not only for differing on island population, but also vehicle flow, based on time of day. Additionally, more time varying components could be added in the analysis. In our optimum evacuation scenarios, pedestrians evacuated completely one hour prior to vehicles so a dynamic programming approach could allow for reoptimization and allocation of those lanes to vehicle traffic to increase vehicles leaving Balboa Peninsula.

A final area to expand our work would be to apply similar tourist population estimation methodologies at different coastal cities in California, such as Marina Del Ray or Malibu, to see if residents and tourists can be distinguished based on their location behavior through time. Additionally, the team's estimation of GPS penetration and probe expansion can be tested against other historical data at other cities.

4.0 Summary

In our project, we used a data driven approach to solve the traditionally difficult problem in estimating tourism population density in order to prepare safe evacuation plans. Using GPS probe data, our team was able to match that data to specific roadways in Newport Beach, as well as identify peak congested average speeds, and behaviors such as route selection. Applying the Macroscopic Fundamental Diagram, our team was then able to use vehicle speed to estimate their flow rate in Balboa Peninsula and applied other research to determine pedestrian flow rate. The team then built an optimization model designed to safely evacuate a peak population estimated on July 4th by allocating dedicated vehicle and pedestrian lanes. Due to the difference in congested flow rates between newport and Balboa Avenue, our optimal solution had all vehicles leaving via Balboa Avenue and pedestrians leaving via Newport Avenue, with the estimated on island July 4th population of 89,940, to all be safely evacuated in four hours. Our model provides a tool for city managers to estimate evacuation flow at certain key points, and can be expanded to analyze different areas, or to account for time delays in arrival and departure from the peninsula.

References:

[1] R. Xia, R. Lin II, ‘California maps will point to tsunami danger zones,’ *Los Angeles Times*, March 21, 2014. Available: https://www.latimes.com/local/lanow/la-me-ln-california-officials-drawing-tsunami-flood-maps-to-aid-future-construction-20140321-story.html [Accessed: February 20, 2019]

[2] United States Geological Survey Science Application for Risk Reduction, 'The SAFRR Tsunami Scenario—Improving Resilience for California', 2013.

[3] United States Geological Survey Science Application for Risk Reduction, 'The SAFRR Tsunami Scenario', 2013.

[4] Ventura County Sheriff’s Office of Emergency Services, 'Ventura County Operational Area Tsunami Evacuation Plan', 2006.

[5] Transportation Research Board, ‘Highway Capacity Manual, 6th Edition,’ Washington D.C., 2016.

[6] T. Martinez, 'Newport Beach’s crowded Fourth of July', 2012 [Online]. Available: https://www.ocregister.com/2012/07/05/newport-beachs-crowded-fourth-of-july/ [Accessed: April 20, 2019]

[7] J. Peters, N. Wood, R. Wilson, K. Miller, ‘Intra-community implications of implementing multiple tsunami-evacuation zones in Alameda, California’ in National Hazards, 2016. DOI 10.1007/s11069-016-2469-8

[8] H. Davis, ‘Big Tsunami could flood large swaths of Newport Beach,’ Los Angeles Times, March 26, 2018. Available: https://www.latimes.com/local/lanow/la-me-newport-tsunami-20180326-story.html [Accessed: February 20, 2019]

[9] J. MacFarlane, 'Data Preprocessing', CE 263N Lecture, University of California Berkeley, 2019.

[10] C. Dagazno, N. Geroliminis, ‘An analytical approximation of the Macroscopic Fundamental Diagram of Urban Traffic’ in Transportation Research, 2008. DOI 10.1016/j.trb.2008.06.008

[11] F. Li, S. Chen, X. Wang F. Feng, ‘Pedestrian Evacuation Modeling and Simulation on Metro Platforms Considering Panic Impacts,’ in International Conference on Traffic and Transportation Studies 2014. DOI: 10.1016/j.sbspro.2014.07.209

Appendix

Optimization Code

%% The purpose of this code is to develop optimization scenario for evacuation off Newport Beach. This code is run in a double nested for loop to test all different lane configurations, or you can index within this code itself. Use CVX SDTP solver, available for download at http://cvxr.com/cvx/download/

```
function [k,P,T,v] = optimize(l_a,l_b)
   % scenario based
   % l_a: lanes for ped on road A
   % l_b: lanes for ped on road B

   % road A is Balboa Blvd
   % road B is Newport Blvd
   % dummy variables/parameters

   q_a_veh = 0.037; % veh/sec
   q_b_veh = 0.034; % veh/sec

   q_ped = 1600/3600; % ped/sec

   psg = 5; % ren/veh

   % population in 4 shapes
   aug = 4.18 % average
   P1 = 865/2;
   P2 = 1235/2;
   P3 = 16900/2;
   P4 = 70490/2;
%     P1 = 865;
%     P2 = 1235;
%     P3 = 16900;
%     P4 = 70490;

   V_on = 16900; % veh

   Q_a_ped = l_a*q_ped; % ped flow on A
   Q_b_ped = l_b*q_ped; % ped flow on B

   T_evac = 4*3600; %3 hours

   % Optimization
   cvx_begin
      variables k(1) T_a_veh(1) T_a_ped(1) T_b_veh(1) T_b_ped(1) P_a_ped(1) P_a_veh(1) P_a_out(1)
P_b_ped(1) P_b_veh(1) P_b_out(1)
```

```
        % k: percentage of people from Zone 4 to exit via Balboa Island
        % ground truth is 0.3555555

        maximize((4-l_a)*q_a_veh*T_a_veh + (4-l_b)*q_b_veh*T_b_veh)
        subject to

            % road A
            P_a_ped == l_a*Q_a_ped*T_a_ped; % people get out in ped on A
            P_a_veh == psg*(4-l_a)*q_a_veh*T_a_veh; % people get out in veh on A
            P_a_out == P_a_veh+P_a_ped;

            % road B
            P_b_ped == l_b*Q_b_ped*T_b_ped; % people get out in ped on B
            P_b_veh == psg*(4-l_b)*q_b_veh*T_b_veh; % people get out in veh on B
            P_b_out == P_b_veh+P_b_ped;

            % in flow = out flow, every one out!
            P_a_out == P4*k + P1;
            P_b_out == P4*(1-k) + P2 + P3;

            % time within 4 hours
            T_a_veh <= T_evac;
            T_a_ped <= T_evac;
            T_b_veh <= T_evac;
            T_b_ped <= T_evac;
            T_a_veh >= 0;
            T_a_ped >= 0;
            T_b_veh >= 0;
            T_b_ped >= 0;

            % count vehicles
            (4-l_a)*q_a_veh*T_a_veh + (4-l_b)*q_b_veh*T_b_veh <= V_on; % cannot use more cars

            % split factor
            k >= 0;
            k <= 1;
    cvx_end

    P = [P_a_ped; P_a_veh; P_a_out; P_b_ped; P_b_veh; P_b_out];
    T = [T_a_veh; T_a_ped; T_b_veh; T_b_ped];
    v = (4-l_a)*q_a_veh*T_a_veh + (4-l_b)*q_b_veh*T_b_veh;
    return
end
```

Data Analysis Code (Average Speed, Path Splitting)

```
import numpy as np
import pandas as pd
import geopandas as gpd
import matplotlib
import matplotlib.pyplot as plt
from shapely.geometry import Polygon, Point, LineString
from datetime import datetime


## Read shapefiles into variables

# Read shape files into geopandas
right_half_tri_SF = gpd.read_file("shapefiles/right_half_triangle/right_half_triangle.shp")[['Name',
'geometry']]
left_half_tri_SF = gpd.read_file("shapefiles/left_half_triangle/left_half_triangle.shp")[['Name', 'geometry']]
island_SF = gpd.read_file("shapefiles/island/island.shp")[['Name', 'geometry']]
newport_SF = gpd.read_file("shapefiles/inundation_map/inundation_map.shp")[['Name', 'geometry']]
# Read Street Shape Files
balboa_blvd = gpd.read_file("shapefiles/Streets/Balboa/Balboa_Blvd-polygon.shp")[['Name', 'geometry']]
w_balboa_blvd = gpd.read_file("shapefiles/Streets/W_Balboa/W_Balboa-polygon.shp")[['Name',
'geometry']]
newport = gpd.read_file("shapefiles/Streets/Newport/Newport_Blvd-polygon.shp")[['Name', 'geometry']]

section_shapefile_list = [right_half_tri_SF,left_half_tri_SF,island_SF,newport_SF]
street_shapefile_list = [balboa_blvd,w_balboa_blvd,newport]

### Read data for each day

# Assign data for each day
d_1_07012018 = 'Data/Probe_Data/2018_07_01_NewportBeach_basic_probe.csv'
d_2_07022018 = 'Data/Probe_Data/2018_07_02_NewportBeach_basic_probe.csv'
d_3_07032018 = 'Data/Probe_Data/2018_07_03_NewportBeach_basic_probe.csv'
d_4_07042018 = 'Data/Probe_Data/2018_07_04_NewportBeach_basic_probe.csv'
d_5_07052018 = 'Data/Probe_Data/2018_07_05_NewportBeach_basic_probe.csv'
d_6_07062018 = 'Data/Probe_Data/2018_07_06_NewportBeach_basic_probe.csv'
d_7_07072018 = 'Data/Probe_Data/2018_07_07_NewportBeach_basic_probe.csv''

data = pd.read_csv(d_1_07012018,
                   names = ["PROBE_ID","SAMPLE_DATE", "LAT", "LONG", "HEADING",\
                   "SPEED","PROBE_DATA_PROVIDER","X","Y", "LOCAL_TIME"])
# This now sorts in date order
data.sort_values(by='SAMPLE_DATE', inplace=True, ascending=True)
data.head(3)
```

```
# # Analyze Data

class Analyze_Data:

    def __init__(self, data_df,section_shapefile_list,street_shapefile_list):
    # Reading data dataframe into class
    self.data = data_df

    # Read shapefiles into class as lists
    self.section = section_shapefile_list
    self.street = street_shapefile_list

    def plot(Long,Lat):
    # Plot Data
    plt.figure(figsize = (12,8))
    plt.scatter(Long, Lat, s = 0.5, c = 'k')
    plt.show()

    def create_point(self,row):
    # Helper function for turning coordinates into shapely points
    return Point(row['LONG'],row['LAT'])

    def append_points(self):

    self.data['coord'] = self.data.apply(self.create_point, axis=1)
    self.data['section'] = 'N/A'

    return self.data

    def get_section(self):
    # For separating data into sections based on defined shapefiles

    self.append_points()

    section_list = []

    for i in self.data.index.values:
    if self.section[0]['geometry'][0].contains(self.data['coord'][i]):
    section_list.append('right_half_tri')
    elif self.section[1]['geometry'][0].contains(self.data['coord'][i]):
    section_list.append('left_half_tri')
    elif self.section[2]['geometry'][0].contains(self.data['coord'][i]):
    section_list.append('island')
    elif self.section[3]['geometry'][0].contains(self.data['coord'][i]):
    section_list.append('4')
    else:
    section_list.append('N/A')
```

```
self.data['section'] = section_list

return self.data

def get_street(self):
# For separating data into streets based on defined shapefiles

self.get_section()
street_list = []

for i in self.data.index.values:
if self.street[0]['geometry'][0].contains(self.data['coord'][i]):
street_list.append('balboa')
elif self.street[1]['geometry'][0].contains(self.data['coord'][i]):
street_list.append('w_balboa')
elif self.street[2]['geometry'][0].contains(self.data['coord'][i]):
street_list.append('newport')
else:
street_list.append('N/A')

self.data['street'] = street_list

return self.data

def clean_data(self):
# Remove useless columns and set timezone
self.get_street()
self.data_cleaned = self.data[self.data.section != 'N/A'][['PROBE_ID','LAT','LONG',
'HEADING',                                      'SPEED',
'PROBE_DATA_PROVIDER',                                  'LOCAL_TIME',
'coord',                                        'section','street']]

self.data_cleaned['LOCAL_TIME'] = pd.to_datetime(self.data_cleaned['LOCAL_TIME'])
.dt.tz_localize('UTC').dt.tz_convert('America/Los_Angeles')
# Separate to hours
self.data_cleaned['HOUR'] = self.data_cleaned['LOCAL_TIME'].dt.hour

return self.data_cleaned

def clean_data_add_heading(self):
# Add heading information per probe
self.clean_data()

heading_fixed = []

for i in self.data_cleaned.index.values:
if self.data_cleaned['HEADING'][i] >180:
```

```
heading_fixed.append(-(360 - self.data_cleaned['HEADING'][i]))
else:
heading_fixed.append(self.data_cleaned['HEADING'][i])

self.data_cleaned['HEADING_FIXED'] = heading_fixed

return self.data_cleaned


def probe_count(self, section):
# For counting number of uniuqe probes
self.clean_data_add_heading()
uniq_probe = self.data_cleaned.loc[self.data_cleaned['section'] == section].PROBE_ID.unique()

return len(uniq_probe)

class road_specific_analysis():

def __init__(self, data_cleaned, street_choice):
self.data = data_cleaned
self.street = street_choice

def road_df(self):
# Filter and create dataframe based on road choice
self.data
self.road = self.data.loc[self.data['street']== self.street].copy()

return self.road

def get_heading(self, df, i, lower_head, upper_head):

# Helper function for differentiating heading
head_df = df.loc[(df['HOUR']==i) &                    ((df['HEADING_FIXED'] >= lower_head) |
(df['HEADING_FIXED'] <= upper_head))][['PROBE_ID','SPEED']]

# This removes all 0 speed instances
head_df = head_df[head_df['SPEED'] != 0]
unique_probe_list = list(set(head_df['PROBE_ID']))

return head_df, unique_probe_list

def get_heading_info(self, limits, speed_limit, trans_type):

'''
Inputs:
df - cleaned up dataframe with hours and streets
limits - list of heading limits, [lower, higher]
trans_type - looking for 'vehicle' or 'pedestrian'
```

```
        Outputs:
        head_speed - list of 24 north heading speeds averaged over an hour
        head_unique_probe - list of 24 counts of unique probe ids in that hour - vehicles
        '''

        self.road_df()

        head_speed = []
        head_unique_probe = []

        for i in range(24):
        # Analyze for 24 hours
        head_df, unique_probe_list = self.get_heading(self.road, i, limits[0], limits[1])

        count = 0
        speed = []

        for j in unique_probe_list:

        probe_df = head_df.loc[head_df['PROBE_ID'] == j]
        mean_speed = np.mean(probe_df['SPEED'])

        if trans_type == 'Vehicle':

                if mean_speed >= speed_limit:
                count += 1
                speed.append(mean_speed)

        elif trans_type == 'Pedestrian':

                if mean_speed <= speed_limit:
                count += 1
                speed.append(mean_speed)


        head_unique_probe.append(count)

        if not speed:
        head_speed.append(0)
        else:
        head_speed.append(np.mean(speed))

        return head_speed, head_unique_probe

# # For getting trajectory

traj = Analyze_Data(data, section_shapefile_list, street_shapefile_list)
```

```
traj_df = traj.clean_data_add_heading()
traj_df.head()

uniq_ID = set(traj_df.PROBE_ID)

# Create empty dataframe
columns = ['PROBE_ID','Route_Num','Start_Section','End_Street','Time_Start','Time_End']
traj_route_df = pd.DataFrame(columns=columns)

for ID in uniq_ID:

        probe_df = traj_df.loc[traj_df.PROBE_ID == ID]

        route = [probe_df.section.iloc[0]]
        route_count = 0
        route_time = [probe_df.LOCAL_TIME.iloc[0]]

        for i in range(len(probe_df)-1):

        if (probe_df.LOCAL_TIME[i+1] - probe_df.LOCAL_TIME[i])<pd.Timedelta(minutes = 1):

        route.append(probe_df.street.iloc[i+1])
        route_time.append(probe_df.LOCAL_TIME.iloc[i+1])

        else:
        route_count += 1
        traj_route_df = traj_route_df.append({'PROBE_ID':ID,'Route_Num':route_count,
        'Start_Section':route[0],'End_Street':route[-1],
'Time_Start':route_time[0],'Time_End':route_time[-1]},                    ignore_index=True)

traj_rest = traj_route_df.loc[traj_route_df['Start_Section'] == 'rest_of_new_port']

balboa = traj_rest.loc[traj_rest['End_Street'] == 'balboa']
newport = traj_rest.loc[traj_rest['End_Street'] == 'newport']

balboa = balboa.drop_duplicates(subset = ['Time_Start'],keep=False)
newport = newport.drop_duplicates(subset = ['Time_Start'],keep=False)

print('Number of Cars going to Balboa Blvd:', len(balboa))
print('Number of Cars going to Newport Blvd:', len(newport))

# ## Notes
# ### For unique probe count
# ### Section Options:
# - 'right_half_tri'
# - 'left_half_tri'
# - 'island'
# - 'rest_of_new_port'
```

```
#
# ### Street Options
# - 'balboa'
# - 'newport'
#

# ### Blanket sorting for all data

plt.figure(figsize = (12,8))
plt.scatter(data.LONG, data.LAT, s = 0.5, c = 'k')
plt.show()

data.shape

analysis = Analyze_Data(data, section_shapefile_list, street_shapefile_list)
data_cleaned = analysis.clean_data_add_heading()



street_plot = data_cleaned.loc[data_cleaned.street != 'N/A']

plt.figure(figsize = (12,8))
plt.scatter(data_cleaned.LONG, data_cleaned.LAT, s = 0.5, c = 'k')
plt.show()

plt.figure(figsize = (12,8))
plt.scatter(street_plot.LONG, street_plot.LAT, s = 0.5, c = 'k')
plt.show()

data_cleaned[['PROBE_ID','LAT','LONG','HEADING','SPEED','LOCAL_TIME','section','street','HEADING']
].head()

# ### Analysis of data based on road and transportation type

# Balboa
bal_north_limits = [-90,45]
bal_south_limits = [145,-150]
# Newport
newport_north_limits = [-140,0]
newport_south_limits = [90,-160]

speed_limit = 15

# For Balboa
analysis_balboa = road_specific_analysis(data_cleaned,'balboa')

# Northbound
```

```
veh_balboa_N_speed, veh_balboa_N_probe = analysis_balboa.
get_heading_info(bal_north_limits,                    speed_limit, "Vehicle")


ped_balboa_N_speed, ped_balboa_N_probe = analysis_balboa.
get_heading_info(bal_north_limits,                    speed_limit, "Pedestrian")


# Southbound

veh_balboa_S_speed, veh_balboa_S_probe = analysis_balboa.
get_heading_info(bal_south_limits,                    speed_limit, "Vehicle")


ped_balboa_S_speed, ped_balboa_S_probe = analysis_balboa.
get_heading_info(bal_south_limits,                    speed_limit, "Pedestrian")

# For Newport
analysis_newport = road_specific_analysis(data_cleaned,'newport')

# Northbound
veh_newport_N_speed, veh_newport_N_probe = analysis_newport.
get_heading_info(newport_north_limits,                speed_limit, "Vehicle")


ped_newport_N_speed, ped_newport_N_probe = analysis_newport.
get_heading_info(newport_north_limits,                speed_limit, "Pedestrian")


# Southbound

veh_newport_S_speed, veh_newport_S_probe = analysis_newport.
get_heading_info(newport_south_limits,                speed_limit, "Vehicle")


ped_newport_S_speed, ped_newport_S_probe = analysis_newport.
get_heading_info(newport_south_limits,                speed_limit, "Pedestrian")


# Create a dataframe of road information per hour
road_info_df = pd.DataFrame({'Hour':range(0,24),
'Newport_North_Speed_Veh':veh_newport_N_speed,
'Newport_South_Speed_Veh':veh_newport_S_speed,
'Newport_North_Probe_Veh':veh_newport_N_probe,
'Newport_South_Probe_Veh':veh_newport_S_probe,
'Newport_North_Speed_Ped':ped_newport_N_speed,
'Newport_South_Speed_Ped':ped_newport_S_speed,
```

```
'Newport_North_Probe_Ped':ped_newport_N_probe,
'Newport_South_Probe_Ped':ped_newport_S_probe,
        'Balboa_North_Speed_Veh':veh_balboa_N_speed,\
        'Balboa_South_Speed_Veh':veh_balboa_S_speed,\
        'Balboa_North_Probe_Veh':veh_balboa_N_probe,\
        'Balboa_South_Probe_Veh':veh_balboa_S_probe,\
        'Balboa_North_Speed_Ped':ped_balboa_N_speed,\
        'Balboa_South_Speed_Ped':ped_balboa_S_speed,\
        'Balboa_North_Probe_Ped':ped_balboa_N_probe,\
        'Balboa_South_Probe_Ped':ped_balboa_S_probe})

road_info_df.head()

def find_slowest(df, column,probe):

        slow = []
        for i in df[column]:
        if i>0:
        slow.append(i)
        slowest_speed = min(slow)
        slowest_df = df.loc[df[column]==slowest_speed][['Hour', column, probe]]

        return slowest_df

N_N_slow = find_slowest(road_info_df,'Newport_North_Speed_Veh','Newport_North_Probe_Veh')
S_N_slow = find_slowest(road_info_df,'Newport_South_Speed_Veh','Newport_South_Probe_Veh')

N_B_slow = find_slowest(road_info_df,'Balboa_North_Speed_Veh','Balboa_North_Probe_Veh')
S_B_slow = find_slowest(road_info_df,'Balboa_South_Speed_Veh','Balboa_South_Probe_Veh')

N_N_slow

S_N_slow

N_B_slow

S_B_slow
```