Analyzing Airbnbs and Bikshare in San Francisco

#### GY476

#### Candidate 50797

#### Word Count: 2998

# Introduction

The last few decades have seen an explosion of new geographic data that describe the physical and digital world around us (Miller 2010). In this paper, I explore spatial data within the San Francisco Bay Area, a region that is home to many companies responsible for the worlds geodata production. Part 1 will analyze how Airbnb listings and prices vary with housing costs and education levels and Part 2 will perform routing analysis to compare car and biking travel times between docks in the Bay Wheels bikeshare system in San Francisco.

# Data

I make use of 4 datasets in this project:

* *Airbnb listing data*: This is a dataset of Airbnb listings produced by [Inside Airbnb](http://insideairbnb.com/). I use the listing data for San Francisco, San Mateo and Oakland, which were scraped in September 2022. It contains variables like price and number of beds for each of the 13415 listings.
* *ACS Demographic Data*: Using library(tidycensus) I download the median rent per bedroom, and the percentage of adults with at least a bachelors degree variables for each Zip Code Tabulation Area (ZCTA) in the Bay Area from the 2016-2020 5 year ACS. The percentage of adults with at least a bachelor’s degree was calculated from Census Table B23006 and I use the number of adults over the age of 25 for the denominator.
* *OSM Bike Rental data*: These are user reported point locations of bike rentals in Open Street Map and includes both private storefronts and bike rental systems/docking stations. Using library(osmdata), I download Amenity data on the 253 bike rental stations in San Francisco.
* *Bay Wheels Trip Data*: Bay Wheels is a subsidiary of Lyft and the regional bikeshare system serving the Bay Area. I use their public [system data](https://www.lyft.com/bikes/bay-wheels/system-data) from San Francisco over the last 6 months, which contains anonymized bike trip data including duration, time and dates, and start/end stations.

I programmatically download our data directly from download URLs wherever possible to make the data import process reproducible.

library(tidyverse)  
library(sf)  
library(tidylog)  
library(osmdata)  
library(mapview)  
library(leaflet)  
library(leaflet.extras2)  
library(leaflet.extras)  
library(tmap)  
# For painless filepath pointing  
library(here)  
library(janitor)  
library(tidycensus)  
library(leaflet.extras2)  
library(paletteer)  
library(ggpubr)  
library(RColorBrewer)  
library(patchwork)  
library(grid)  
library(geomtextpath)  
library(ggh4x)  
library(units)  
library(nngeo)  
library(ggsn)  
library(hereR)  
library(osrm)  
library(dotenv)  
  
# Set HERE API keys  
dotenv::load\_dot\_env()  
set\_key(Sys.getenv("here\_api\_key"))  
  
  
  
# ---- Read in Airbnb data ------  
  
# Helper fxn to download csv.gz file from Inside Airbnb, unzip and read in  
download\_and\_readin\_inside\_airbnb\_data = function(url, city){  
   
 # Set data filepath  
 data\_filepath = here(str\_glue("data/raw-data/{city}.csv.gz"))  
   
 # Download data directly from Inside Airbnb website  
 dir.create(here("data/raw-data/"), showWarnings = FALSE)  
 download.file(url, destfile = data\_filepath)  
  
  
 # Read in data  
 read\_csv(here(data\_filepath)) %>%   
 st\_as\_sf(coords = c("longitude", "latitude")) %>%   
 # Since coords are lot/lng, this is EPSG 4326  
 st\_set\_crs("EPSG:4326") %>%   
 # To get nice and easily workable column names  
 janitor::clean\_names() %>%   
 mutate(city = city)  
}  
  
  
# Download airbnb data, I got these URLs from looking at developer console on Inside Airbnb webpage  
sfo\_data = download\_and\_readin\_inside\_airbnb\_data(url =  
 "http://data.insideairbnb.com/united-states/ca/san-francisco/2022-09-07/data/listings.csv.gz",  
 city = "sfo")  
oakland\_data = download\_and\_readin\_inside\_airbnb\_data(url =  
 "http://data.insideairbnb.com/united-states/ca/oakland/2022-09-18/data/listings.csv.gz",  
 city = "oakland")  
san\_mateo\_data = download\_and\_readin\_inside\_airbnb\_data(url =  
 "http://data.insideairbnb.com/united-states/ca/san-mateo-county/2022-09-19/data/listings.csv.gz",  
 city = "san\_mateo")  
  
# Bind all data into one long dataframe, which is safe bc I have added a "city" column for easy identification  
airbnb\_data = bind\_rows(oakland\_data, sfo\_data, san\_mateo\_data)  
  
# ---- Read in ACS and ZCTA data -----  
  
# Use get\_acs from tidycensus  
# Census servers often go down and get\_acs fails intermittently. So if that  
# happens read in local copy as backup  
acs\_data\_raw = tryCatch(  
 expr = {  
 suppressMessages(get\_acs(  
 # Get zcta level variables  
 geography = "zcta",  
 # Got these variable names by perusing data.census.gov website  
 variables = c(median\_rent = "B25031\_001",  
 median\_rent\_as\_pct\_of\_income = "B25071\_001",  
 unemp\_rate\_16\_plus = "S2301\_C04\_001",  
 num\_bach\_higher = "B23006\_023",  
 num\_25\_64\_for\_bach\_higher = "B23006\_001"  
 # median\_rent\_no\_bedroom = "B25031\_002",  
 # median\_rent\_1\_bedroom = "B25031\_003",  
 # median\_rent\_2\_bedroom ="B25031\_004",  
 # median\_rent\_3\_bedroom = "B25031\_005"  
 ),  
 year = 2020,  
 survey = "acs5",  
 output = "wide",  
 # Allows us to get ZCTA geometries as sf dataframes  
 geometry = TRUE  
 ) %>%  
 janitor::clean\_names() %>%  
 # Calculate pct bach or higher using num and denom  
 mutate(pct\_bach\_higher\_e = num\_bach\_higher\_e/num\_25\_64\_for\_bach\_higher\_e,  
 pct\_bach\_higher\_m = moe\_ratio(  
 num = num\_bach\_higher\_e,  
 denom = num\_25\_64\_for\_bach\_higher\_e,  
 moe\_num = num\_bach\_higher\_m,   
 moe\_denom = num\_25\_64\_for\_bach\_higher\_m  
 )) %>%  
 rename(zcta = geoid))  
 },  
 error = function (e) {  
 st\_read("data/acs\_zcta\_data\_raw\_saved.geojson", quiet = TRUE)  
 },  
 warning = function (w) {  
 st\_read("data/acs\_zcta\_data\_raw\_saved.geojson", quiet = TRUE)  
 }  
 )  
  
# Drop geometry column for now  
acs\_data = acs\_data\_raw %>%   
 st\_drop\_geometry()  
  
# --- Read in OSM Data ----  
  
# build an overpass query to get bike rental points in Bay Area  
query <- opq(bbox = "San Francisco Bay Area", timeout = 90) |>   
 add\_osm\_feature(key = "amenity", value = "bicycle\_rental")  
  
bike\_rental\_locations\_osm <- osmdata\_sf(query)  
  
# Just extract locations which are encoded as points  
bike\_rental\_points = bike\_rental\_locations\_osm$osm\_points %>%   
 as\_tibble() %>%   
 st\_as\_sf() %>%   
 # These 2 columns seem to be the most useful for identifying bikes in network  
 arrange(network, brand) %>%   
 select(osm\_id, name, network, brand, everything())  
   
# ---- Download SFO County boundary ------  
  
# Download all CA counties  
sfo\_county = tigris::counties(state = "CA",   
 cb = TRUE) %>%   
 clean\_names %>%   
 # filter to San Francisco county  
 filter(name == "San Francisco")   
  
  
# ---- Read in Bay Wheels data -----  
  
# Choose to focus analysis on just one start station ID  
chosen\_station\_id = "SF-F26"  
  
# Helper function to download zip file and read in CSV  
download\_zip\_and\_read\_csv = function(url){  
 last\_part\_of\_url = basename(url)  
 download.file(url = url,  
 quiet = TRUE,   
 destfile = str\_glue("data/raw-data/{last\_part\_of\_url}"))  
 read\_csv(str\_glue("data/raw-data/{last\_part\_of\_url}"))  
   
}  
  
  
bike\_trip\_data\_nov = download\_zip\_and\_read\_csv(url = "https://s3.amazonaws.com/baywheels-data/202211-baywheeels-tripdata.csv.zip")  
bike\_trip\_data\_oct = download\_zip\_and\_read\_csv(url = "https://s3.amazonaws.com/baywheels-data/202210-baywheels-tripdata.csv.zip")  
bike\_trip\_data\_sep = download\_zip\_and\_read\_csv(url = "https://s3.amazonaws.com/baywheels-data/202209-baywheels-tripdata.csv.zip")  
bike\_trip\_data\_aug = download\_zip\_and\_read\_csv(url = "https://s3.amazonaws.com/baywheels-data/202208-baywheels-tripdata.csv.zip")  
bike\_trip\_data\_jul = download\_zip\_and\_read\_csv(url = "https://s3.amazonaws.com/baywheels-data/202207-baywheels-tripdata.csv.zip")  
bike\_trip\_data\_jun = download\_zip\_and\_read\_csv(url = "https://s3.amazonaws.com/baywheels-data/202206-baywheels-tripdata.csv.zip")  
  
# Combine rows together, filter to just one start station  
bike\_trip\_data\_combined = bind\_rows(  
 bike\_trip\_data\_jun,  
 bike\_trip\_data\_jul,  
 bike\_trip\_data\_aug,   
 bike\_trip\_data\_sep,  
 bike\_trip\_data\_oct,   
 bike\_trip\_data\_nov  
 ) %>%   
 # filter to chosen station  
 filter(start\_station\_id == chosen\_station\_id)  
   
  
bike\_trip\_data = bike\_trip\_data\_combined

## Background on old and new spatial data

The ACS data we are using is a classic example of “old” spatial data, which are traditional, detailed datasets specifically designed for social science research. However, they are often very costly, published infrequently and only available at coarse resolutions.

This is in contrast to the Airbnb and Baywheels data, which are “new” forms of rich spatial data. These newer data are often collected automatically for business purposes and purely by accident are available for research (Arribas-Bel 2014). These data can be very frequent (Airbnb data for example were all scraped within the last month), extremely granular, and are often much larger in size and scale (Miller and Goodchild 2015).

The disadvantages of new spatial data are that they can be biased (O’neil 2017), often do not measure uncertainty, and may have significant privacy concerns. In the case of Airbnb data, it only contains a small sliver of the rental market and its possible for fake listings to be included. In this paper I combine old and new forms of spatial data to understand how Airbnbs are distributed throughout the Bay and how travel times within San Francisco vary.

## Projections

I use EPSG:7131, or the NAD83 State Plane projection for San Francisco, in all our datasets as this was one of two projected coordinate reference systems specifically created by the San Francisco City and County GIS department. This is a projected 2D conformal coordinate system that:

* preserves angles between locations
* is accurate for distance based calculations like buffers and travel time
* is suitable for the entirety of the Bay Area.

## Data Cleaning

After exploratory analysis, I drop the top 0.5% most expensive Airbnb listings (67 listings with prices greater than $2500 per night) as they are very high outliers. The vast majority of listings have prices between $50 and $500 a night. And upon visual inspection that most of the dropped listings don’t have any reviews and may be fake, justifying our decision.

I filter the Bay Wheels bike data to trips that start at Station SF-F26, or Union Square. Routing analyses can blow up in size with many stations. And this is a relatively central station so results will be relatively transferable.

I also perform other data cleaning steps, including:

* Removing small islands from the SF County boundary for easier to read maps
* Removing probable duplicates from OSM bike rental locations
* Taking the centroid of stations in the Bay Wheels data
* Cropping all bike data to San Francisco
* Assigning ZCTAs to each Airbnb with distance based joins
* Creating a bounding box for our study area to crop our ACS data
* Creating a hexagonal grid of our study area

More details can be found in the code below.

# Part 1: Mapping/Data Viz

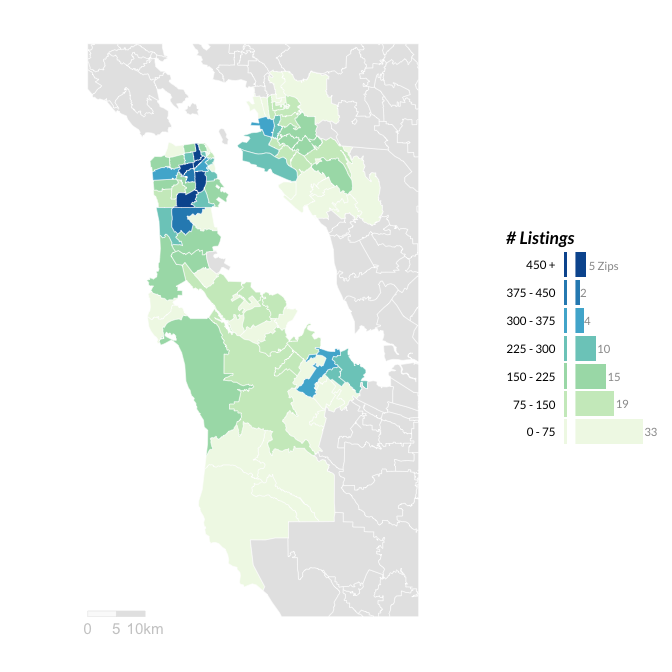
## Mapping Airbnbs

We map the count (Map 1.1) and average price (Map 2.2) of Airbnb listings in the Bay Area. I made several design decisions which are justified below:

1. I bin the continuous data into equal intervals because the ranges of our variables are relatively large and I want to avoid having too many colors. Using equal interval bins makes the resulting ranges easy to interpret.
2. I include the number of zip codes that fall in each bin as a histogram to the right of the legend. This visualizes the actual distribution of the variable and provides context on what may seem like arbitrary bin cutoff decisions.
3. I use the Green-Blue color palette from [ColorBrewer](https://colorbrewer2.org/#type=sequential&scheme=GnBu&n=6), based on Cynthia Brewer’s research (2003) on gradients that are colorblind friendly, and suitable for LCD computer screens.
4. I provide interactive maps using library(mapview) for exploratory purposes in the interactive tabs.

### Map 1.1: Count of Airbnb listings

#### Static

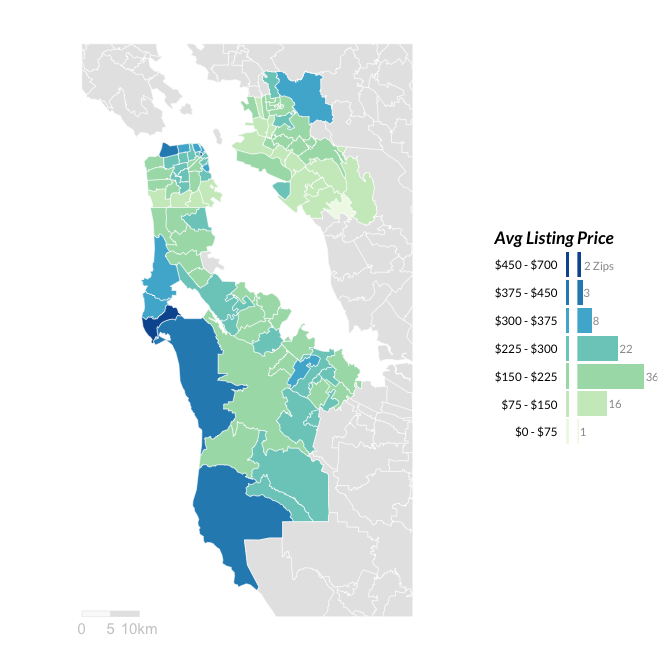


Most ZCTAs have 0-75 listings. The ZCTAs with the highest count of listings are concentrated in downtown San Francisco & right next to the Bay in West Oakland. ZCTAs with the least amount of Airbnbs tend to be in far east Oakland and Southwest Bay Area. This makes sense as these locations are far from common tourist destinations.

#### Interactive

### Map 1.2: Average Airbnb Prices in Bay Area Zipcodes

#### Static



Most Airbnbs tend to be between $150 - $300 a night. The highest priced Airbnbs are along the coastline of the Pacific Ocean and SF. These Airbnbs probably have great views, or access to beaches/other tourist amenities. Conversely, the lowest priced Airbnbs tend to be in Oakland.

#### Interactive

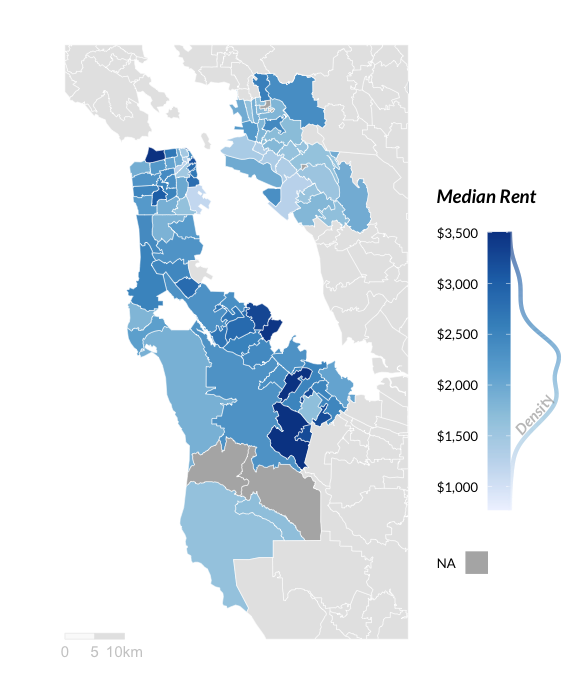
## Mapping Census variables

We map our two Census variables at ZCTA level. Our maps are similar to the previous maps but with a few different decisions:

1. I use a continuous color ramp as we are interested in looking at small differences in the Census variables across ZCTAs.
2. I include a color matched density plot to the right of the legend. This is essentially a smoothed version of a histogram. I choose to omit the axis for the density plot as actual density values can be hard to interpret (they are just meant to integrate to 1) and I want the reader to focus on the overall distribution of the variable.

### Map 2.1: Median Rent by ZCTA

#### Static

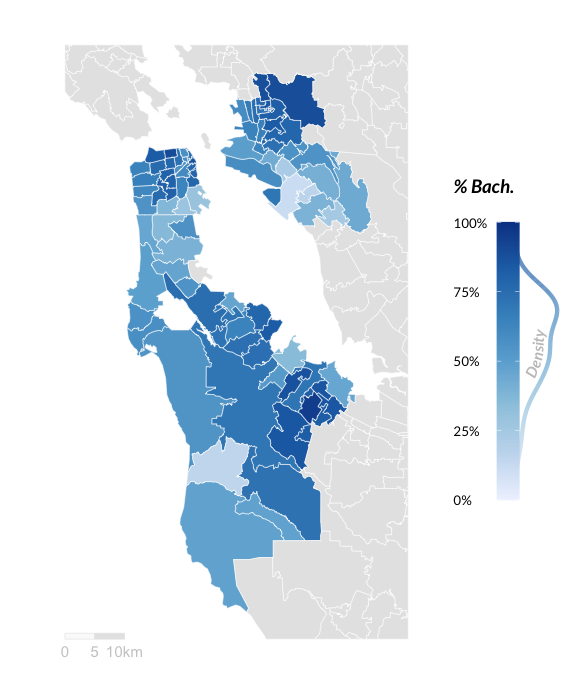


The ZCTAs with the highest rent tend to be in downtown San Francisco and Southeast San Mateo. The ZCTAs with the lowest rent tend to be in Southeast San Francisco, and all throughout Oakland. The density plot reveals that most ZCTAs have average rents between $1500 and $2500 per bedroom.

#### Interactive

### Map 2.2: Percentage of 25+ population with Bacherlors Degree or higher

#### Static



The most highly educated ZCTAs tend to be in East Oakland, West San Francisco, and South San Mateo. The least educated ZCTAs tend to be in South Oakland, and South San Francisco. Generally these maps track the median rent map in that highly educated ZCTAs tend to have high rents and vice versa. These two Census variables have a correlation of 0.57.

#### Interactive

The main types of neighborhoods that we’ve identified in Map 2.1 and 2.2 are high housing costs and highly educated neighborhoods, which tend to be in downtown San Francisco and South San Mateo. I’ll call these tech worker hubs, because these are neighborhoods close to where the headquarters of many Silicon Valley companies are located. It makes sense that high income and highly educated tech workers would drive up rental prices here. Similarly there are less college educated and lower housing cost neighborhoods in South San Francisco and most of Oakland.

My hypothesis would be that Airbnbs cluster near these tech worker hubs (as many workers may need short term housing) and near downtown SF neighborhoods with high rental prices (as these locations are close to attractions and considered safe).

### Map 3: Average Log Price of Airbnb listings and Median Rent

Map 3 shows the average log price of Airbnb listings overlaid on top of a choropleth map of Median rent. Darker orange mean more expensive listings, and darker blue means higher median rent neighborhoods. For this map:

1. I aggregate the 13,000+ Airbnb points into hexagon grids and presented the average log price of listings. Hexagons allow us to visualize relative density effectively within equal area grids and don’t clutter up the map.
2. I use interactivity because the ability to toggle layers on and off is very helpful when comparing two variables. If you hover over the layer button (right below the zoom button) and toggle the Avg Log(Price), you can visually compare where the high priced airbnb neighborhoods overlap the high rent neighborhoods.
3. I set increased transparency of the Median Rent variable to visually send that layer into the background.
4. I turn the map title into a pseudo-legend by making use of colors to prime the reader for the two different variables scales on the map.
5. There are 4 listings which have a list price of $0 a night. I manually increment them to be $1 a night so that log(price) is well defined.

Neighborhoods with high listing prices (dark orange) tend to cluster near the coast, and not always near high rent (dark blue) neighborhoods. There are some clusters of high listing price neighborhoods close to high rent neighborhoods, particularly in San Mateo near the Silicon Valley belt. But the highest listing neighborhoods are not generally in the highest rent neighborhoods. The ZCTA level correlation between average listing price and median rent for the 167 ZCTAs in our study area is 0.24 and therefore confirms a weakly positive relationship between median ZCTA rent and Airbnb price.

## Potential Raster Datasets

Some raster datasets I could use to improve to this analysis would be:

1. Air quality data that measures particulate matter or ozone at various grid sizes, like [these data](https://ww2.arb.ca.gov/geographical-information-system-gis-library) from the California Air Resources Board. I could overlay this on top of our listing data to see if more listings or more xpensive listings are correlated with being in good air quality areas.
2. Vegetation data/indexes, that measure tree canopy cover and greenspaces at various grid sizes, like [this raster data](https://gis.data.ca.gov/datasets/CDFW::naip-2020-ndvi-california/about) from the California Department of Fish and Wildlife that provides NVDI imagery at 60 cm resolution. I could use this to look at relationships between the amount of Airbnbs/Airbnb prices, socioeconmic variables, and the amount of greenspace and vegetation available in the neighborhood.

# Part 2: Spatial Analysis

I perform an analysis of travel time during rush hour by bike and car from a bikeshare station in central San Francisco at Union Square. The key question I will be asking is which trips are quicker by bike rather than car and why?

## Visualizing bike rental network

I create a heatmap of bike rentals pulled from OpenStreetMap. I make a few key decisions justified below:

1. I use hexagon binning to aggregate the point level data and present zero bins as grey NA values to easily see the neighborhoods in SF without bike rentals.
2. For exploratory purposes, I provide a point map of bike rentals in a separate tab which maps the point distributions. I overlay the full listing of Bay Wheels Bikeshare stations on this map (in light green) as calculated from the Bay Wheels data, which highlights the limitations of relying on crowdsourced and incomplete OSM data.

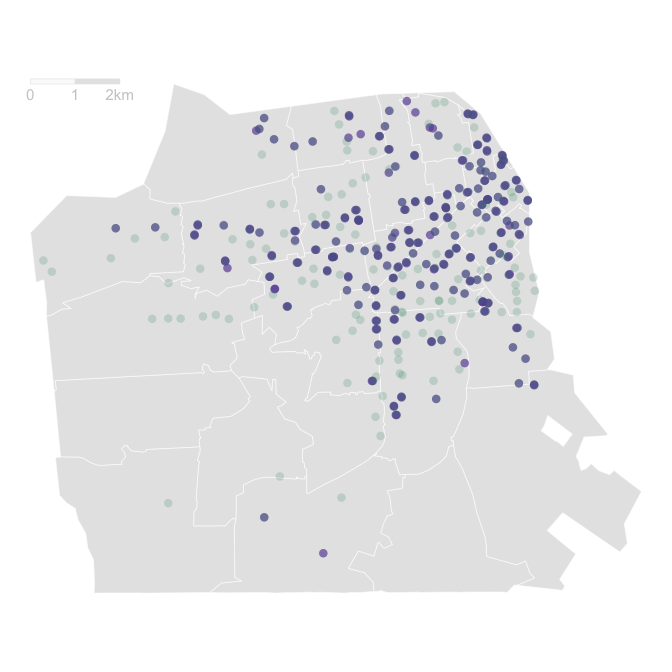
#### Heatmap

#### Map 4: Heatmap of OSM Bike Rentals in San Francisco

Most bike stations are located in North East San Francisco, where the Financial Districts and major tourist attractions (like Pier 39) are located. There are basically no bike rental locations in West or South San Francisco, which makes sense as there are large mountains and hills in that part of the city.

#### Point Map

#### Map 4: Point map of OSM Bike Rentals and Bay Wheels stations in San Francisco



## Creating buffers

I create 200 meter buffers around bike share stations in SF, and perform a spatial join to see which Airbnbs are bike accessible. Overall, 2,051 unique Airbnbs (or 30.5% of Airbnbs in San Francisco) are within the 200 meter buffers. This would help me decide where to rent an Airbnb, but I would have doubts about the effectiveness of this strategy given that we know OSM data is incomplete.

## Generating bike and car travel times

Overall there are 5,679 total trips from the Union Station dock in the Bay Wheels data, and 251 unique trip combinations. For each unique trip combination, I calculate average (mean and median) trip durations of all trips in the Bay Wheels data. Before calculating averages, I drop the top 20% of longest trips for each trip combination. This is because the distribution of biking times has a long right tail, likely representing riders who got lost, or had trouble getting to the destination. These trips are not representative of average trips and heavily skew our grouped calculations, so I drop them. For trip combinations that have less than 5 data points, I don’t do this trimming as this would mean dropping half or more of the data points. And with limited trip information, I want to use all available information.

Then I use library(hereR) to calculate travel times by bike and car for those 251 trip combinations during rush hour (8 AM EST on Dec 8th, 2022). I also generate 8 minute drive time isochrones to represent the areas that travelers could get to in 8 minutes from the chosen station.

## Assessing validity of bike estimates

From our previous calculations, I have 2 estimates of biking time for each of our 251 trips. One estimate from the HERE API and another from the Bay Wheels data. I compare the two to essentially assess the real world validity of the HERE estimates by:

1. Ordering the 251 unique trips from the least difference to the most difference between HERE and Bay Wheels data estimates
2. Splitting the data up into 20 buckets or ntile
3. Calculating a few averages for each ntile

The results are summarized in the below table

### Difference in routing times between HERE and Bay Wheels data

|  | Avg |
| --- | --- |
| Ntile | Distance (m) | # of Trips | HERE | Bay Wheels Median | Diff (min) |
| 1 | 6,215.62 | 0.12 | 27.38 | 51.22 | -23.84 |
| 2 | 5,400.38 | 0.33 | 24.21 | 30.79 | -6.58 |
| 3 | 3,306.15 | 0.20 | 14.78 | 18.16 | -3.38 |
| 4 | 2,720.92 | 0.43 | 11.68 | 13.56 | -1.87 |
| 5 | 2,074.38 | 1.03 | 9.51 | 10.58 | -1.07 |
| 6 | 2,120.46 | 0.43 | 9.22 | 9.66 | -0.44 |
| 7 | 2,467.92 | 0.40 | 10.91 | 10.90 | 0.01 |
| 8 | 2,180.92 | 0.51 | 9.90 | 9.59 | 0.31 |
| 9 | 2,590.92 | 0.42 | 11.61 | 10.81 | 0.80 |
| 10 | 2,925.15 | 0.34 | 13.24 | 12.05 | 1.20 |
| 11 | 3,098.00 | 0.38 | 13.46 | 12.07 | 1.39 |
| 12 | 2,672.83 | 0.47 | 11.71 | 10.00 | 1.71 |
| 13 | 2,897.33 | 0.59 | 13.04 | 10.88 | 2.16 |
| 14 | 3,562.25 | 0.38 | 16.53 | 14.00 | 2.53 |
| 15 | 3,541.92 | 0.33 | 15.77 | 12.96 | 2.81 |
| 16 | 3,903.83 | 0.28 | 17.62 | 14.35 | 3.27 |
| 17 | 3,769.08 | 0.22 | 17.82 | 13.85 | 3.97 |
| 18 | 4,060.00 | 0.36 | 18.82 | 13.55 | 5.28 |
| 19 | 5,445.08 | 0.23 | 24.99 | 18.36 | 6.63 |
| 20 | 7,336.08 | 0.04 | 33.71 | 23.32 | 10.38 |

Overall the expected biking times between HERE and Bay Wheels data are within a few minutes of each other. For Ntiles 7-20 - which corresponds to 65% or 163/251 unique bike trips - the HERE API times are an overestimate, and real world bikers are a little faster. For Ntiles 1-7 - which corresponds to 35% of bike trips - The HERE API times are an underestimate. The travel times that HERE got the most wrong (ie the bottom and top of the table) tend to be routes with a low number of overall trips and relatively long distance trips. These areinfrequent/rare trips and so our limited Bay Wheels data might be skewed.

## Analyzing travel times

Next I map our calculated travel times by mode of transport (car vs bike)

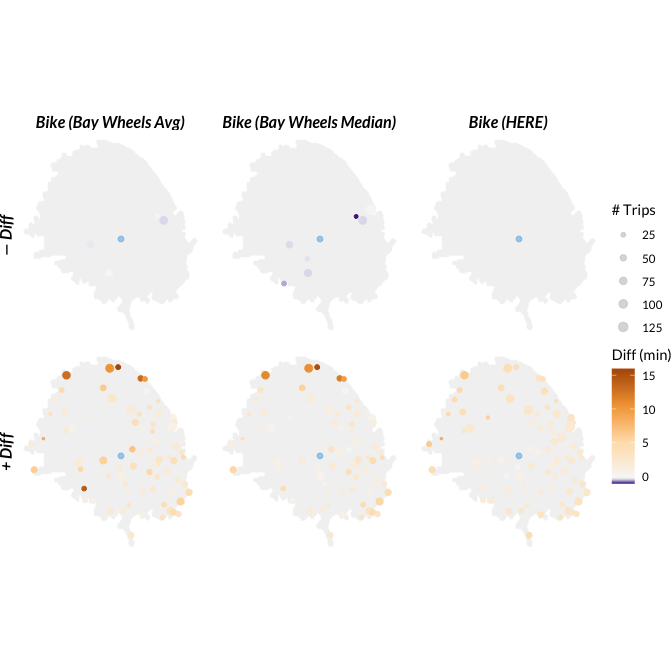
#### Total Time

#### Map 5.1: Travel Time Estimates from Union Station



#### Difference from Car

#### Map 5.2: Travel Time Difference relative to Car from Union Station



In Map 5.1 we see that car trips to all stations are relatively quick. Bike times are comparable at short distances but not at long distances. The quickest trip from the Bay Wheels data is to station SF-F27, which is three blocks down the same street. The slowest trip is to station SF-B15, which is in the Presidio. This interestingly is not the longest distance trip but there is a large hill climb.

In Map 5.2 I only display stations within the 8 minute driving time isochrone and show differences relative to car trips. 8/90 stations are faster by bike than car (ie have negative difference values) according to the Bay Wheels medians. If we expand this criteria to bike trips that are at most 3 minute slower than cars (perhaps taking into account parking time), this includes 62/90 stations. The majority of the 8 bike friendly destinations lie along Market Street and Polk Street, which are both main streets with fully protected grade separated bike lanes. The takeaway here for policy makers is that to incentivize speedy bike trips, build more and better protected bike lanes. Better bike infrastructure here translates to faster bike trips.

# Conclusion

This paper first analyzed Airbnb and Census data for San Francicso, Oakland, and San Mateo. We found that expensive neighborhoods tend to also be highly educated neighborhoods. And there are clear tech worker hubs with large numbers of Airbnb listings in South San Mateo (near the headquarters of Silicon Valley companies) and central San Francisco. The highest priced Airbnbs tend to be along coastlines and close to tourist amenities. Future analyses of the Airbnb data should take into account seasonality, and other rental/housing market data.

I then analyzed travel times by car and bike throughout San Francisco from a central bikeshare dock. There are many station trips that are quicker, or at least equally as fast as car, mostly located along major bikeways. Future analyses of bikeshare data should take into account the locations of bike lanes/bike infrastructure across San Francisco, days/times of trips, and topography of the city.

# References

Arribas-Bel, Daniel. 2014. “Accidental, Open and Everywhere: Emerging Data Sources for the Understanding of Cities.” *Applied Geography* 49: 45–53.

Harrower, Mark, and Cynthia A Brewer. 2003. “ColorBrewer. Org: An Online Tool for Selecting Colour Schemes for Maps.” *The Cartographic Journal* 40 (1): 27–37.

Miller, Harvey J. 2010. “The Data Avalanche Is Here. Shouldn’t We Be Digging?” *Journal of Regional Science* 50 (1): 181–201.

Miller, Harvey J, and Michael F Goodchild. 2015. “Data-Driven Geography.” *GeoJournal* 80 (4): 449–61.

O’neil, Cathy. 2017. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown.

# Appendix

## A note on ZCTAs

Zip Code Tabulation Areas (ZCTAs) are generalized representations of zip codes, and in most cases are very similar to zip codes. But their boundaries can sometimes be a little different due to differences in how the Postal Service and Census calculates zip codes vs ZCTAs respectively. For the sake of standardization, in this paper I always use ZCTA geographies instead of zip code geographies as these are the geography at which ACS data are available and are therefore a standard unit of geography we can join all our point and polygon level data to. Whenever I refer to “zip codes”, these will always mean Zip Code Tabulation Areas. If I had used USPS zip code level geographies (like [this data](https://geodata.lib.berkeley.edu/catalog/ark28722-s7888q) from Berkeley Geodata library) I would have to solve the complex problem of merging zip codes to ZCTA’s and dealing with zip codes that fall into multiple ZCTA’s.

## A note on the ACS

The 5 year ACS survey contains average values from 5 years of data collection in order to provide robust and precise estimates with low standard errors. The tradeoff is that these data are not the most recent and should instead be interpreted as roughly the average demographic composition within 2016-2020. This timeframe does not align with the Airbnb data which was scraped in 2022, but is likely still the best estimate of the demographic composition of these neighborhoods.