Bay Area Bike Sharing Data 2019

By: Swati Kohli, Nicholas Richmond, and Kaitlyn Robinson

Table of Contents

I. INTRODUCTION

a. BACKGROUND

LYFT BIKE SHARING IN BAY AREA

The Bay Area is a cultural and commercial center of California. The region attracts millions of tourists annually and is recognised as one of the busiest areas in the United States. A bustling city with abundant commuters, tourists, and traffic congestion produces a significant amount of pollution.

However, “Despite its urban character, the San Francisco Bay Area is one of California's most ecologically important habitats.” ("San Francisco Bay Area," n.d.). One creative initiative, “bike sharing”, aims to reduce congestion, noise, and air pollution. The goal is to encourage people to opt for cycling to reach a destination as an extension of mass public transport. Essentially, bike sharing services are not meant to replace mass transit.

Lyft owns and operates a bike sharing service in the Bay Area called Baywheels. They operate in San Francisco, San Jose, and the Oakland/Berkeley area. Lyft is interested in helping to decrease the Bay Area’s negative impact on the health of the environment. Since climate change issues are currently at the forefront of public discourse, and will impact the future of society, we decided to analyze one of the methods being used to offset its effects.

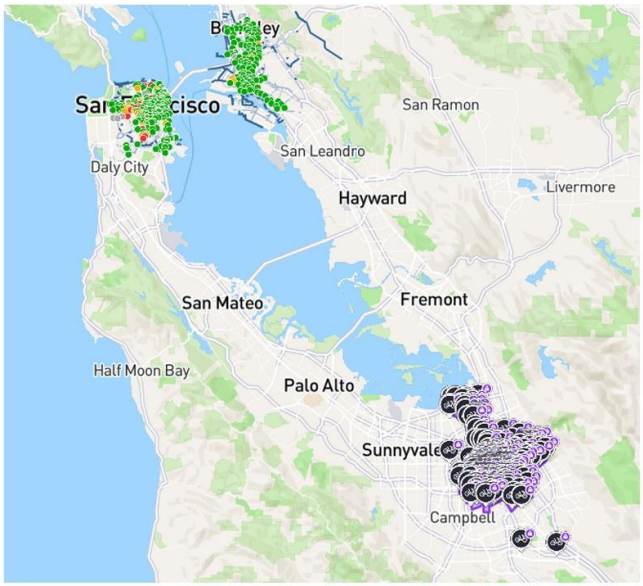


Figure 1: Map of Lyft rideshare locations

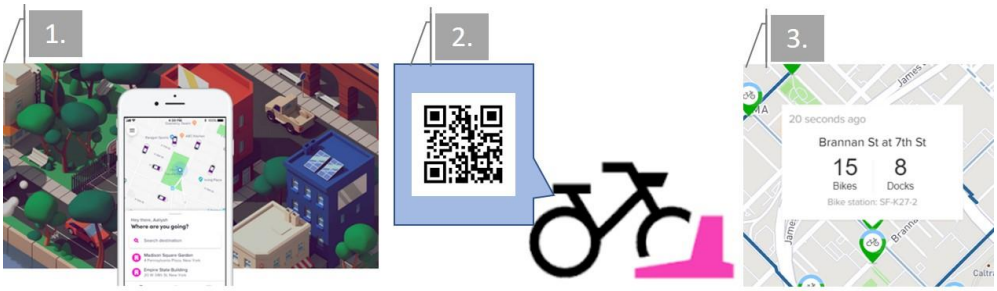
LYFT BIKE SHARING SYSTEM OVERVIEW

The Lyft bike sharing system is straightforward. A 3-step process is as follows (Lyft, Inc.):

i. **JOIN**: Once a user joins the system, he/she inputs a credit card that will ultimately be charged based on user type and usage. No upfront cash payment is required, although this option is available at some physical pay stations. One can use the service as a day user (customer) or a subscriber (monthly/annual to receive free rides up to 45 minutes).

ii. **RIDE-CHECK-OUT**: To ride from any docking station, a QR code scan on the bike or a member key can be used to unlock the bike. Day users pay $2 for the first 30 minutes and then $3 for each additional 15 minutes. Subscribers receive 45 minutes of free ride time and then are charged $3 for each additional 15 minutes.

iii. **CHECK-IN**: The bike can be docked back at any station with empty racks. The system map on the app shows stations with available bikes and racks.



b. DATASET DESCRIPTION

1. Dataset Name and Source:
    a. Bay Area Bike Sharing Trips
    b. Kaggle Website: https://www.kaggle.com/jolasa/bay-area-bike-sharing-trips
    c. We used the Kaggle dataset in conjunction with a larger dataset from Lyft's website.
        i. Lyft website: https://www.lyft.com/bikes/bay-wheels/system-data
        ii. From the Lyft dataset (limited to the same timeframe), we derived a table of unique station information, which also allowed us to pull in geographic data for the beginning and end of each trip.

2. Description:

    The overall dataset tracked each ride that was taken on a Lyft bike from January to May 2019. It contained 15MB of data and had a 10.0 Usability rating on Kaggle. It described users by categorizing them as either a customer or subscriber and provided the rider's gender and birth year. Each ride was tracked by a bike ID, the length of the ride (in seconds), the bike's start station, and the bike's end

station. The data did not contain a unique user identifier. See below for a full listing of initial fields.

3. Observations :
    a. Number of rows: 1,053,067
    b. Number of original columns: 10

4. Fields Within Original Dataset:

    1. Month = when the trip occurred (January to May 2019).
    2. Trip Duration (seconds) = how long a trip lasted.
    3. Bike ID = unique identifier for each bike.
    4. User Type = either customer or subscriber
        a. Customer = one-time rider, 24-hour pass, or 3-day pass user
            i. Usually tourists
        b. Subscriber = annual/monthly member
            i. Usually Bay Area residents
    5. Gender = male, female, or other
    6. Birth Year = self-entered, not validated by a person ID.
    7. Start Station Name = where the trip began
        a. Usually an intersection or street name
    8. End Station Name = where the trip ended
        a. usually an intersection or street name
    9. Start Station ID - unique identifier for each start station
    10. End Station ID - unique identifier for each end station

5. NaN values:

    There are around 50,000 missing values in both member_birth_year and member_gender. Also, there are 745 rides with missing station info. The data cleaning process will describe how these missing values were handled.

```
num_rows = combined_2019.shape[0]
num_missing = num_rows - combined_2019.count()
print(num_missing)
```

```
month                    0
trip_duration_sec        0
start_station_id       745
start_station_name     745
end_station_id         745
end_station_name       745
bike_id                  0
user_type                0
member_birth_year    49376
member_gender        49370
dtype: int64
```

Figure 2. NaN Values

II. PURPOSE OF THE PROJECT (DRIVING QUESTION)

Since bike sharing is becoming a prominent way to get around the Bay Area, it is important to analyze and understand, through a data-driven approach, **how frequently the system is utilized**. This topic is very relevant for any commuter, especially in the Bay Area, as it is a part of routine life. We explore if it is possible to identify usage patterns, whether the bikes are being used at an optimal level, and if this program is a success. It is an opportune time to investigate the publicly available data through Kaggle's website.

The **goal** of this project is to analyze the success of Lyft's bike sharing business based on analysis of travel behavior in the most utilized city/suburb in the Bay Area (see below). This is driven by studying what and how various factors affect the duration and/or frequency of a bike ride. Extensive research went into exploratory data analysis to learn about user composition and how it relates to the trips they take. We endeavour to set a benchmark for future studies following our line of research.

III. DATA CLEANING PROCESS

The Kaggle data was mostly clean, with the exception of several fields that had missing or unhelpful values. We began the cleaning process by first combining separate monthly dataframes into one combined dataset. Each month of data began as a separate CSV file on the Kaggle website. Once we combined the data, we displayed the row counts of the new combined dataframe, as well as each month's original dataframe, to confirm data was not lost. We then displayed the "NaN" counts of each column to determine which fields were missing data. From this step, we discovered station info and member demographic information were missing on several thousand rows. Since the entire dataset contained more than one million rows, we decided that removing rows with missing data would not significantly affect our analysis. Also, we knew that we wanted to analyze how gender and age affected trip duration, so we needed to ensure those fields were complete and accurate. We also ran a groupby count operation on member_gender and discovered that around 20,000 rows contained a value of "Other". We also removed these rows because, as stated above, we planned on using this field in our analysis and needed to keep our main variables as complete and explicit as possible. We ran a command to determine each index with a gender value equal to "Other" and dropped those rows.

While conducting initial research, we discovered that the Kaggle dataset was derived from a more complete dataset on Lyft's website. We were able to access the Lyft data and found that it contained latitude and longitude coordinates for each start and end station. Since we wanted to determine the most utilized city, we decided to incorporate the coordinate data with our Kaggle dataset. We began by importing each month's CSV from the Lyft website and combined the data into a separate dataframe. Our main goal with this dataset was to extract coordinates for each unique station, so we created new dataframes for both start and end stations and only kept the station_id, station_latitude, and station_longitude. A separate

dataframe was needed for both start and end stations to ensure that we accounted for all possible stations. We then dropped duplicate rows from each dataset to ensure we were only keeping distinct combinations of these three station-related fields. Finally, we renamed each column in both datasets, combined the dataframes, and removed duplicates one more time. We then used a groupby and lambda function to count any station_id occurrences greater than one to verify that all duplicates were removed.

Unfortunately, there were 5 station_ids that appeared more than once in the dataset with slightly different coordinate values. We decided to remove one occurrence of the duplicate station record to ensure we only kept one distinct row per station. Luckily, the “duplicated” station coordinates were essentially equal. We ran the groupby and lambda count function again to verify that each station only appeared once. See below for a code snippet of how this was determined.

```
combine_stations = pd.concat([all_start_stations, all_end_stations], ignore_index = True)
unique_stations = combine_stations.drop_duplicates()

group_count = unique_stations.groupby('station_id', sort=True).filter(lambda x: x['station_id'].count() > 1)
group_count.sort_values(by=['station_id'])
```

	station_id	station_latitude	station_longitude
19	25.0	37.787522	-122.397405
362	25.0	37.786928	-122.398113
229	37.0	37.785000	-122.395936
332	37.0	37.785377	-122.396906
45	130.0	37.757718	-122.391813
326	130.0	37.757288	-122.392051
294	316.0	37.330165	-121.885831
364	316.0	37.331168	-121.886938
136	345.0	37.766474	-122.398295
324	345.0	37.766483	-122.398279

Figure 3: Code Snippet of Concatenating to Determine Longitude and Latitude

The next step in data cleaning involved merging the unique station list with our Kaggle dataset to add the station coordinates to our main dataframe. We performed a “left” merge once to pull in the start_station coordinates and another time to pull in the end_station coordinates. Since we used the station_id as the join “key”, we ended up with duplicate occurrences of this field for both start and end station columns. We dropped the duplicate columns so only one version remained for both start and end stations. Next, we needed to rename the latitude and longitude fields to be more descriptive. We created a dictionary with keys equal to the original column names and the values equal to desired column names. We then used the “get” method and a for-loop to run through the dataframe and used the dictionary to replace the old column names with our new ones. Below is a snippet of the code used to determine city information. The coordinate conditions were determined by analyzing a map and learning the behavior of geographical coordinates.

```
def determine_start_city(df):
    if df['start_station_longitude'] < -122.354860:
        val = 'San Francisco'
    elif df['start_station_latitude'] < 37.4258460:
        val = 'San Jose'
    elif df['start_station_latitude'] > 37.7492920 and df['start_station_longitude'] > -122.3390910:
        val = 'Oakland/Berkeley'
    else:
        val = 'Unknown'
    return val
```

Figure 4: Function determining city of a station

The resulting dataframe now had almost all of the data we needed to begin our analysis. Instead of simply leaving the coordinate fields as is, we decided to use them to derive two new fields to tell us the start and end station's city. Using specific coordinate ranges derived from a map, we created a function that would take in a dataframe and test whether the station was within San Francisco, San Jose, or Oakland/Berkeley. We created two versions of the function - one for start stations and one for end stations to independently determine the location of each station. We then applied the function on the dataframe twice and added two new columns: start_city and end_city. Finally, we decided to remove “outlier” rides from the overall dataframe. We removed rows where the ride duration exceeded 6 hours because long rides are likely to be erroneous due to incorrect docking or stolen bikes. We also removed rows with a trip duration under 2 minutes with the same start and end station based on the assumption that the user changed his or her mind or the bike was in need of servicing.

Below is a screenshot of the first several rows of our tidy dataframe:

	Unnamed: 0	month	trip_duration_sec	start_station_id	start_station_name	end_station_id	end_station_name	bike_id	user_type	member_birth_year	member_gender
0	4	January	6733	245.0	Downtown Berkeley BART	266.0	Parker St at Fulton St	3532	Subscriber	1994.0	Male
1	5	January	1188	34.0	Father Alfred E Boeddeker Park	146.0	30th St at San Jose Ave	5114	Subscriber	1984.0	Male
2	6	January	1254	318.0	San Carlos St at Market St	314.0	Santa Clara St at Almaden Blvd	3967	Subscriber	1991.0	Male
3	7	January	3153	29.0	O'Farrell St at Divisadero St	70.0	Central Ave at Fell St	4813	Subscriber	1979.0	Male
4	8	January	323	223.0	16th St Mission BART Station 2	129.0	Harrison St at 20th St	1976	Subscriber	1991.0	Male
5	9	January	433	266.0	Parker St at Fulton St	256.0	Hearst Ave at Euclid Ave	4642	Subscriber	1996.0	Male
6	10	January	272	349.0	Howard St at Mary St	60.0	8th St at Ringold St	263	Subscriber	1993.0	Male

Figure 5 First 10 columns of the cleaned dataset

start_station_latitude	start_station_longitude	end_station_latitude	end_station_longitude	start_city	end_city
37.870139	-122.268422	37.862464	-122.264791	Oakland/Berkeley	Oakland/Berkeley
37.783988	-122.412408	37.742314	-122.423181	San Francisco	San Francisco
37.330698	-121.888979	37.333988	-121.894902	San Jose	San Jose
37.782405	-122.439446	37.773311	-122.444293	San Francisco	San Francisco
37.764765	-122.420091	37.758862	-122.412544	San Francisco	San Francisco
37.862464	-122.264791	37.875112	-122.260553	Oakland/Berkeley	Oakland/Berkeley
37.781010	-122.405666	37.774520	-122.409449	San Francisco	San Francisco

Figure 6 Last 6 columns of the cleaned dataset

IV. DATA ANALYSIS

Our data analysis was a progressive process towards the goal we envision. We analyzed with a data-driven approach how various factors were likely to influence the success of Lyft bike sharing based on data of the first five months of 2019. Our intention was not to assume its success or failure, but to give analytics-based results and possible interpretations which can be used by the company to review and base decisions on in the future. Therefore, we set a benchmark for future studies following our line of research.

The factors we considered may influence success are:

1. Usage analysis
2. Travel behavior
3. Bike demand and supply
4. Revenue generation and contribution of highest few routes towards it, and
5. A predictive data model based on various relevant variables from the dataset

The first two questions of our analysis were to set the foundation for our multiple regression and conclusion.

QUESTION 1: Determine which city has the highest volume and duration of trips.

Part a)

This analysis was performed over the whole cleaned dataset in order to determine which of the routes had the highest volume and duration of riders. In order to successfully determine which city had the highest volume and duration of riders, we added a column that combined the start_city and end_city fields using concatenation. This allowed us to see the path of each ride and determine which route was taken most often. The routes included: Oakland/Berkeley to Oakland/Berkeley, San Francisco to San Francisco, San Jose to San Jose,

Oakland/Berkeley to San Francisco, Oakland/Berkeley to San Jose, San Francisco to Oakland/Berkeley, San Francisco to San Jose, and San Jose to San Francisco. Below you'll find the code that was used to concatenate the start and end city columns.

```
#Add a new row that contains both the start and end city
col_concat = pd.concat([Question_1_2['start_city'] + " / " + Question_1_2['end_city']], axis = 1)
col_concat
```

Figure 7 Code to Concatenate the start and end city

Once this column was added, it was simple to utilize the seaborn countplot to determine which route was used the most frequently.

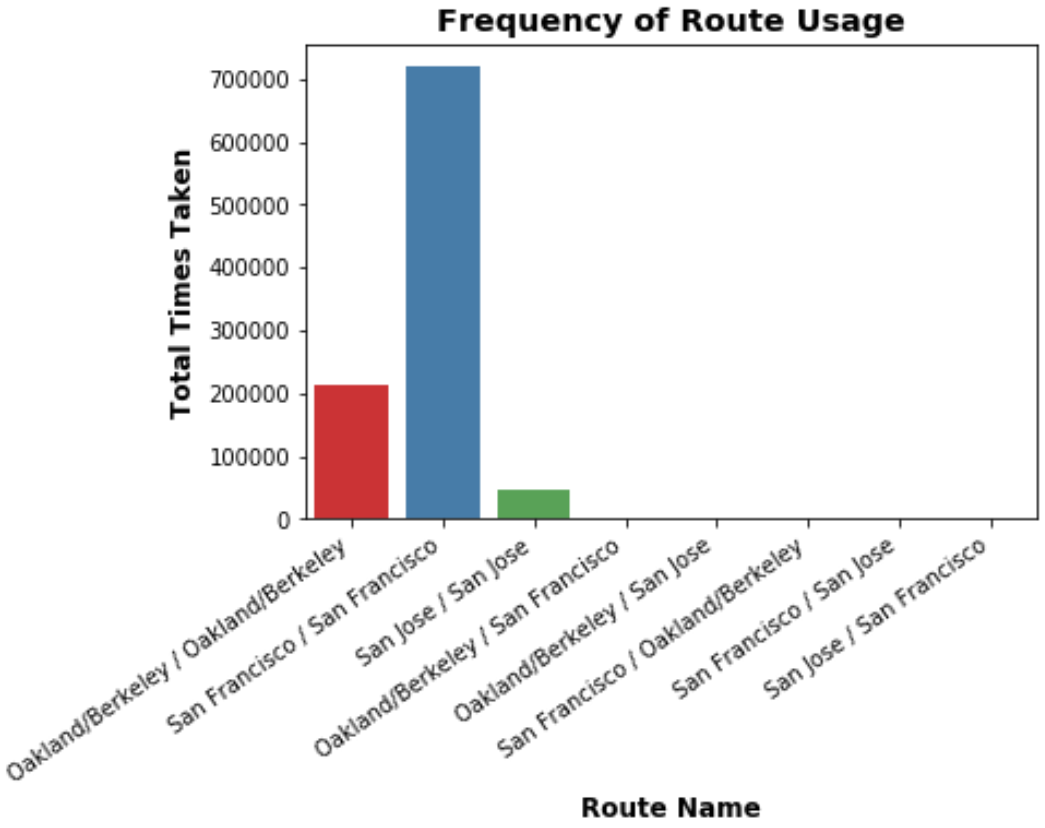


Figure 8 Frequency of Route Usage Chart

As you can see in Figure 8, the rides that both start and end in San Francisco occurred most frequently over the five months, with a total of 719,010 rides. There are 169 docking stations and 3360 bikes in San Francisco, this density of docking stations and bikes reflects the city's heavy usage.

Part b)

In order to further evaluate the routes, we also analyzed the average duration per route. The results of this analysis were interesting because the routes that appeared most valuable to Lyft Baywheels (based on average duration) did not match our frequency analysis.

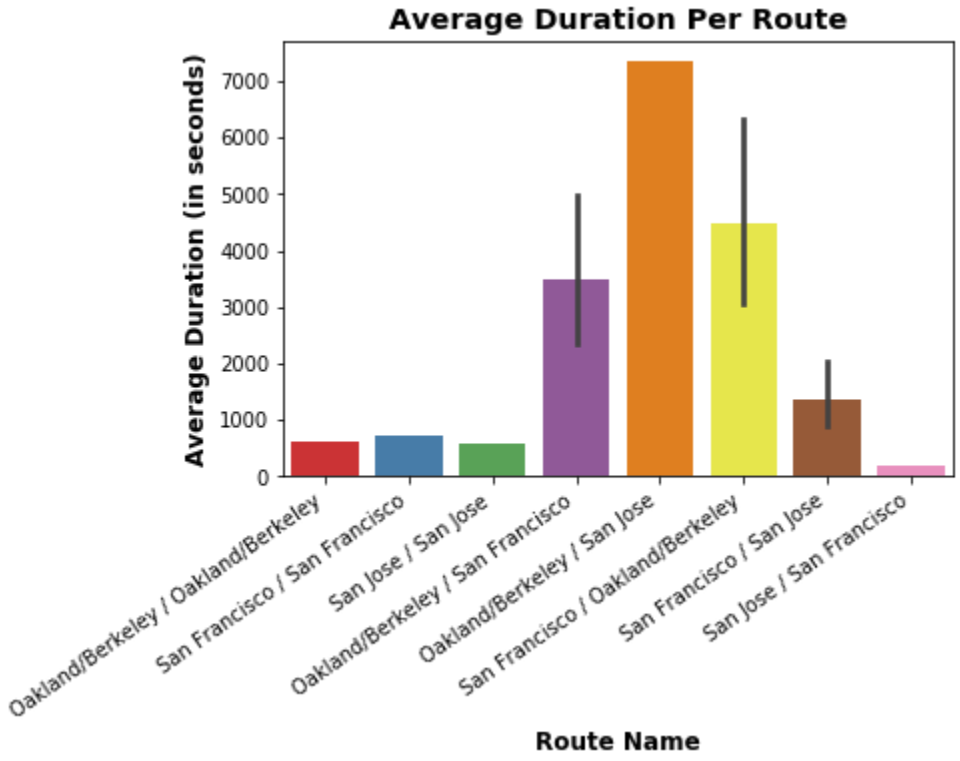


Figure 9 Average Duration per Route

Upon further analysis, it was clear that the higher average duration for these less frequent routes were simply outliers because those routes had only been taken once each. While this chart may suggest these high duration routes could be more valuable to Lyft, with such a low usage rate they are actually far less valuable than the 719,010 shorter rides within San Francisco.

Based off the conclusion that rides within San Francisco city limits have the highest frequency, we decided to further analyze the rides in San Francisco because they have the greatest impact on the success and performance of Lyft Baywheels in the first 5 months of 2019. Therefore, the rest of our questions focus on evaluating the rides within San Francisco.

QUESTION 2: What is the frequency and duration of riders in San Francisco based on gender, age, and user type?

The first step in this question was to create a dataframe that just had the rides within San Francisco. We called this dataframe Rides_SF. Based off this dataframe, we were able to move forward with analyzing our users based on gender, age, and user type.

Part a) Gender

In order to evaluate the frequency and duration based on gender, we determined that the seaborn countplot would be the best method to display the distribution of data and the seaborn barplot would be the best way to display the duration data. We used the member_gender column in the Rides_SF dataframe to create the charts below.

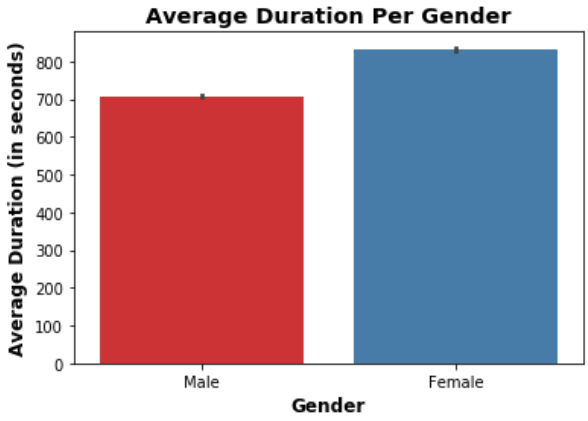


Figure 10 Average Duration per Gender

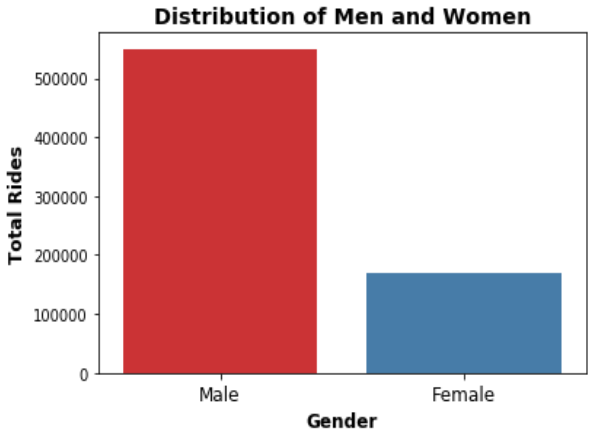


Figure 11 Distribution of Men and Women

Based off of figure 10 and 11, we could clearly see that men make up more of the total amount of rides in San Francisco, but women tend to take longer bike rides. Specifically, men account for 549,901 rides, while women only account for 169,109 rides. Additionally, women's average duration of bike ride was over a minute longer than men's. These were interesting data points to see, especially when you take into consideration that the ratio of men to women in San Francisco is fairly even with 50.8% women and 49.2% men.

Part b) User Type

The user type is broken down into either a customer or subscriber. A subscriber is someone who uses Lyft Baywheels consistently enough to pay on a monthly or annual basis, while a customer is a day user or tourist. In order to evaluate the frequency and duration based on user type, we again determined that the seaborn countplot would be the best method to display the distribution of data and the seaborn barplot would be the best way to display the duration data. We used the user_type column in the Rides_SF dataframe to create the charts below.

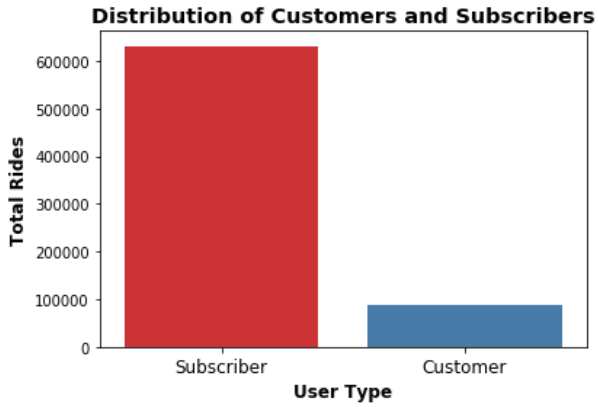


Figure 12 Distribution of Customers and Subscribers

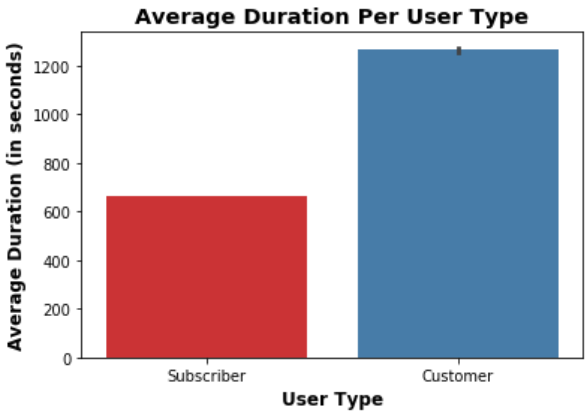


Figure 13 Average Duration per User Type

Based on figure 12, we can see that subscribers make up a majority of the total rides (approximately 600,000 rides), while customers (day users or tourists) only make up about 100,000 rides. From figure 13, we can see that a customer's average duration is about twice as long as a subscriber's. This makes sense since subscribers are most likely going from home or BART to work each day without making leisurely stops, while customers are most likely taking their time enjoying the sites of the city.

Part c) Age

We incorporated the analysis of age because we hypothesized we would see a significant difference in frequency across the ages. In order to graphically display the frequency distribution of ages, we first had to add a column that displayed the rider's age. We added the column by taking the member_birth_year column and subtracting it from 2019. Before creating our final histogram, we decided to create a dataframe that only included the riders that were under 100 years old. We had to do this because some people falsely entered their age - the highest one was 140! In order to clearly display the distribution of rides from age to age, we decided that creating a histogram with seaborn distplot was the best technique to use.

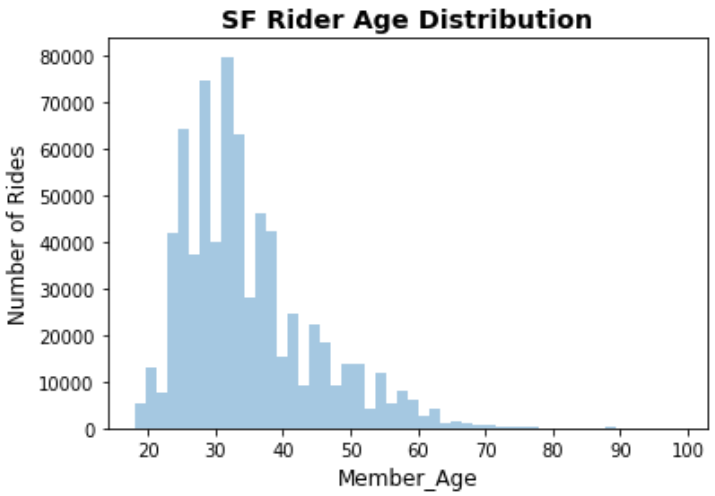


Figure 14 SF Rider Age Distribution

From Figure 14, you can see that riders ages 26 - 35 were the most active with Lyft Baywheels. After age 35, you see a gradual decrease in frequency stopping almost entirely at age 70. This histogram confirms our hypothesis that we'd see a significant difference across the ages. This distribution accurately reflects the population of San Francisco with 37.5% of its population being between the ages of 25 and 44.

QUESTION 3: Analysis of bike demand and supply

The bike sharing system has a good network spread throughout the San Francisco area. Therefore, a contributing factor to the success of the system is to assess the demand and supply of bikes at bike stations. Since the bikes keep rotating within the system, this is an important factor to determine which stations have an excess or shortage of bikes. Further, this information can be utilized to strategize where bikes can be relocated from and where they need to go to ensure optimum bike availability and open docks at each station to accommodate their users.

STRATEGY #1

1. **Analysing popular <u>stations</u> based on usage**

Due to data unavailability on a daily/hourly basis, our approach is to analyze the overall demand and supply at popular stations. Popularity of a station is indicative of the usage, i.e. the sum total of check-ins and check-outs at a station.

STEPS:

Below is a snippet for reference (python code file (LyftBikeSharingQ3Q4) for complete stepwise analysis).

1. Add a column with the total number of check-ins and check-outs for each station and arrange in descending order.

```
# Find the number of check-outs per station
station_start=sf_stations['start_station_name'].value_counts()
# Find the number of check-ins per station
station_end=sf_stations['end_station_name'].value_counts()
# Create a DataFrame with the check-outs and check-ins
# Create a column that sums check-outs and check-ins
# Create a column to get difference of check in - check out
station_counts = pd.concat([station_start, station_end], axis=1)
station_counts.rename(columns={'start_station_name':'Check_out', 'end_station_name':'Check_in'}, inplace=True)
station_counts['Total'] = station_counts['Check_out'] + station_counts['Check_in']
station_counts['CheckIn-CheckOut'] = station_counts['Check_in']- station_counts['Check_out']
# arrange in descending order to get top stations
station_counts = station_counts.sort_values('Total', ascending=False)
```

Figure 15 Code to get stations in descending order

2. Get top stations where the total number of rides > 25000 (arbitrary).

```
# top stations = total number of rides > 25000
top_stations= station_counts[station_counts['Total']>25000]
top_stations
```

	Check_out	Check_in	Total	CheckIn-CheckOut
San Francisco Caltrain Station 2 (Townsend St at 4th St)	19961	29032	48993	9071
Market St at 10th St	19226	19235	38461	9
San Francisco Ferry Building (Harry Bridges Plaza)	15727	18805	34532	3078
Montgomery St BART Station (Market St at 2nd St)	15006	18967	33973	3961
Berry St at 4th St	15985	15346	31331	-639

Figure 16 Code to get top stations

3. The figure below shows the 10 most popular stations by total usage with over 25,000 total rides (check-ins and check-outs).

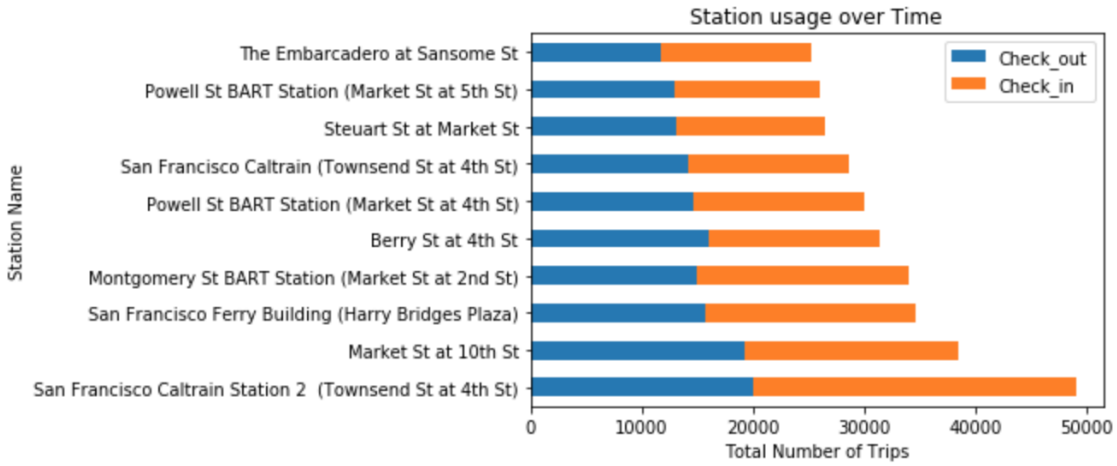


Figure 17 Horizontal Bar graph: 10 most population stations based on frequency of trips

RESULT

Further investigation on the characteristics of the area where these stops are located may show more clarity on the reasons for their popularity. However, timestamp data would give more accurate results.

STRATEGY #2

1. **Analyzing popular routes based on frequency of check-ins & check-outs**

Diving deeper into the topic, another possible way of assessing the demand-supply is to analyze which stations had more bikes either checked into or out of each station (i.e. net activity). For example, if the number of check-ins minus check-outs is negative, this would indicate more bikes were leaving station than entering. This suggests there is a high demand at that station for bikes.

STEPS:

Below is a snippet for reference (python code file (LyftBikeSharingQ3Q4) for complete stepwise analysis).

1. Add a column with the difference of check-ins and check-outs for each station and arrange in descending order.
2. Get top and bottom 10 stations.

```
# BIKE DEMAND
# arrange in descending order to get difference
station_counts = station_counts.sort_values('CheckIn-CheckOut', ascending=False)
# 1. If more bikes check-in than check-out, implies bikes in excess
excess= station_counts.head(10)
# 2. If more bikes check-out than check-in, implies bikes in demand
demand= station_counts.tail(10)
# concatenating the two dataframes (excess and demand)
bike_demand = pd.concat([excess, demand])
bike_demand['Stations']= bike_demand.index
bike_demand.index = range(1,21)
bike_demand[['Stations','Check_out','Check_in','CheckIn-CheckOut']]
```

```
# table for data visualization
bike_demand_chart = bike_demand[['Stations','CheckIn-CheckOut']]
bike_demand_chart
```

	Stations	CheckIn-CheckOut
1	San Francisco Caltrain Station 2 (Townsend St...	9071
2	Montgomery St BART Station (Market St at 2nd St)	3961
3	San Francisco Ferry Building (Harry Bridges Pl...	3078
4	The Embarcadero at Sansome St	1858
5	8th St at Brannan St	1214
6	Civic Center/UN Plaza BART Station (Market St ...	992
7	11th St at Bryant St	736
8	Folsom St at 3rd St	676
9	4th St at Mission Bay Blvd S	640
10	Powell St BART Station (Market St at 4th St)	618
11	24th St at Chattanooga St	-822

Figure 18 Code snippet for Bike demand calculation

The figure below shows a data visualization of the top 10 stations (with more check-ins than check-outs) and the bottom 10 stations (with more check-outs than check-ins).

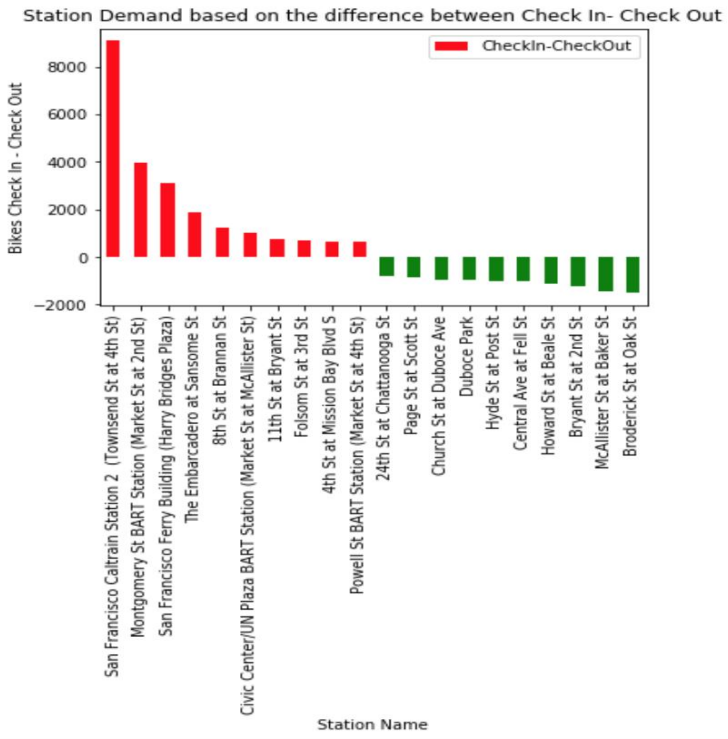


Figure 19 Bar graph: 10 stations representing excess and in demand bikes based on bike Check-in- Check-out

INTERPRETATION

This suggests that the top 10 stations are the ones that had excess bikes at the station, while the bottom stations are the ones that should be monitored for reloading.

QUESTION 4: Percentage contribution of top 9 routes in terms of Lyft revenue based on day user type

Revenue generation is another metric to assess the success of the program. Studying the revenue contribution of top routes might give an insight to the pattern of usage and inform specific marketing ideas for Lyft to utilize in the future.

STRATEGY

Cost of a trip

https://www.lyft.com/bikes/bay-wheels/pricing

- **Subscriber** pays $15 or $149 respectively to become a monthly or annual member and receives unlimited 45-minute trips.
- **Customer** (or a day user) pays $2 and receives a 30-minute trip.
- Any time over the allotted amount costs $3 per additional 15 minutes.

These bikes can be convenient, affordable options for tourists and office-goers to explore and commute within the city. However, we can only find and analyze the revenue contribution for a day user and not a subscriber, even though almost 88% of bikes are used by subscribers. This is due to lack of unique identification of a subscriber (or lack of ballpark number of subscribers) for the dataset.

STEPS

Below is a snippet for reference (python code file (LyftBikeSharingQ3Q4) for complete stepwise analysis).

1. From the dataset, only day user data is extracted.
2. Made and applied cost function to calculate the cost of each day user ride based on duration.

```
# extract only day user data
du_routes= sf_stations[sf_stations['user_type']=='Customer']
du_routes
# percentage contribution of day user to total ridership over 5 months
print("Total Number of users (Subscriber + day user): ",sf_stations.shape[0])
print("Number of day users: ",du_routes.shape[0])
print ("Percentage of day user riders out of total riders for 5 months is: ",
       (du_routes.shape[0]/sf_stations.shape[0])*100, "%" )

# function to calculate cost paid by day user
# day user pays $2 for first 30mins and $3 for every subsequent 15 minutes as per LYFT website
import math
def Calculate_Cost(x):

    #first 30 mins free every subsequent 15mins = $3
    if x>1800:
        total_time = x-1800
        if (total_time%900) == 0:
            total_cost = ((total_time/900)*3)+2
        else:
            total_cost = ((math.ceil(total_time/900))*3) + 2
    else:
        total_cost = 2
    return (total_cost)
# apply cost function to dataframe
du_routes['Cost_of_trip']= du_routes['trip_duration_sec'].apply(Calculate_Cost)
```

Figure 20 Code snippet of cost function to find day user ride trip cost

3. Made a dataframe with top routes based on frequency of that route (more than 250- arbitrary).
4. Calculated revenue contribution of those top routes (9 routes have more than 250 frequency) towards total day user revenue generation of 5 months.

```
# top routes in SF based on frequency of rides (more than 200) for day user
dutrips_df = du_routes.groupby(['start_station_id','end_station_id']).size().reset_index(name = 'number of trips')
top_dutrips = dutrips_df.sort_values('number of trips', ascending=False)
top_9_dutrips = top_dutrips[top_dutrips['number of trips']>200]
top_9_dutrips
# extract data based on the 9 top routes in the day user model. Below one example:
du_route1 = du_routes[(du_routes['start_station_id']== 15) & (sf_stations['end_station_id']==6)]
# concatenate the 9 dataframes
du_bike_revenue = pd.concat([du_route1, du_route2,du_route3,du_route4,du_route5,du_route6,du_route7,du_route8,du_route9])

# calculate the revenue from total day users in 5 months
du_routes['Cost_of_trip'].sum()
# calculate the revenue from day users of top 9 routes in 5 months
top9_routes= du_bike_revenue['Cost_of_trip'].sum()
# percentage contribution wrt revenue generation of day user to total day user ridership over 5 months
print("Total revenue generation by day user for 5 months): ",total_du_revenue)
print("Total revenue generation by day user for top 9 routes in 5 months):",top9_routes)
print ("Percentage of revenue contribution of day user riders to total for 5 months is: ",(top9_routes/total_du_revenue)*100, "%
# Percentage of revenue contribution of day user riders to total for 5 months is:  6.62%
```

Figure 21 Code snippet for calculating day user revenue generation and percentage contribution of top 9 rides towards it

RESULT

The total revenue generated by day users over the 5 months was $283,160.

We conducted further analysis to identify the top routes in SF and understand how these top routes (percentage-wise) are contributing to Lyft's revenue. This informed our assessment of if these routes are substantially impacting revenue or not. Therefore, we calculated top routes in SF based on frequency of rides (more than 250- arbitrary choice) for a day user.

The figure below demonstrates the top 9 routes contributed 6.62% of the total revenue generation by the day users over 5 months. Further study on duration could reveal more insight into other revenue contributions.

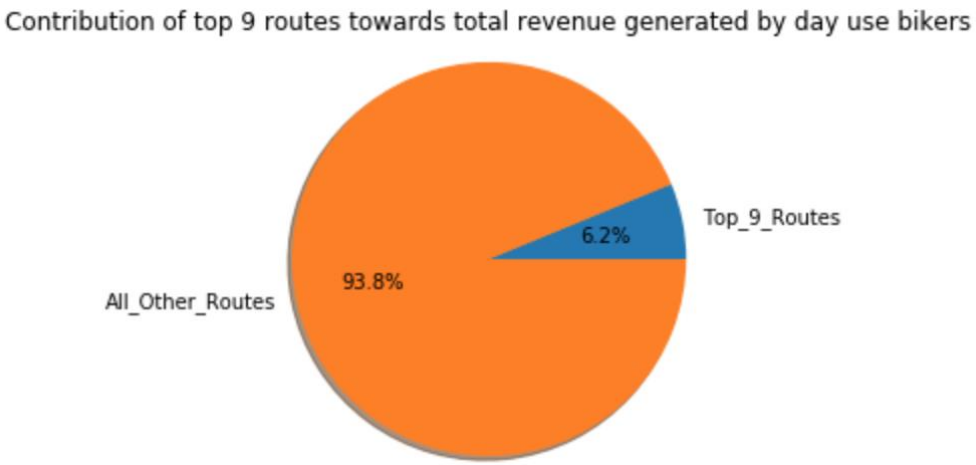


Figure 22 Pie Chart: Contribution of top 9 popular routes towards total day user revenue generation

INTERPRETATION

Upon review, we observed another interesting insight: out of the top 9 routes, 5 routes were cyclic trips, i.e. start and end station were the same. Even when not cyclic, the trips only permutated over 6 unique stations. This implied that these could be leisurely tourist trips. As expected, when investigated further, these were mostly stations around tourist attractions. As a suggestion, this data can be used to increase revenue by target marketing (like discount coupons at these stations) or partnerships with agencies that can provide guided tours along these routes. The results can be found in figure 23 below.

	start_station_id	end_station_id	number of trips
1113	15.0	6.0	617
15695	377.0	377.0	503
16478	400.0	400.0	360
430	6.0	15.0	353
1237	15.0	371.0	291
16400	399.0	399.0	248
1120	15.0	15.0	237
423	6.0	6.0	228
1249	15.0	400.0	207

Figure 23 Result of top 9 routes showing cyclic trips

V. DATA MODEL & INTERPRETATION

STRATEGY & STEPS

We created a data model to analyze how several variables in our dataset predict the duration of trips. The response variables we chose were the rider's age range, gender, and user type. For gender, the possibilities were male and female after we removed rows with an "Other" gender. The two possible user types were customers and subscribers. Finally, the age range was a new field that we derived by calculating a rider's age from his or her birth year and applying a function to group users into 3 categories: 18-35, 36-50 and Above 50. Using age bands helped us create a regression analysis that could help explain ridesharing behavior for people at different stages in life. We decided to use statsmodel for our multiple regression analysis because all our variables were categorical and this library effectively determines response variables that help interpret coefficient behavior. We ran the regression on our combined Kaggle and Lyft dataset filtered down to rides that began and ended in San Francisco. As stated previously, the research we conducted before this analysis indicated that San Francisco is the busiest city in the Lyft Bay Area bike sharing system. We wanted to analyze the system's most utilized city to best understand what predicts the overall success of the bike sharing program.

RESULT

After creating the equation and running the regression, we discovered that the variable that most predicts a ride's length was the user type. According to our results, a trip duration will decrease by 595 seconds (almost 10 minutes) as the user type changes from a customer to a subscriber. This is most likely due to the fact that subscribers are probably commuters who repeat short distance trips over time. Even though subscribers far outnumber customers when it comes to ride frequency, this analysis reveals how ride duration and frequency behave differently. Customers are probably users who ride Lyft bikes on a more leisurely, random basis. Ride duration also decreases by 104 seconds (over 1.5 minutes) when the user's gender changes from female to male. This relationship suggests that, although men ride Lyft bikes more often than women, the overall trip duration is not necessarily longer for men over women. Also, as users change from the 18-35 age range to 36-50, trip duration decreases by 22 seconds. Although the overall decrease is not very substantial, this relationship suggests that middle age riders are more likely to take shorter rides than younger users. Finally, as the age range changes from 18-35 to Above 50, the ride duration increases by 35 seconds. This behavior is rather unexpected because it suggests that older riders are actively using the system longer than younger groups. However, it is possible that older riders tend to be customers, rather than subscribers. The longer durations we see with subscriber rides could explain the behavior we are seeing with these older riders.

INTERPRETATION

Overall, this regression reveals some interesting predicted behavior for the response variables. It appears that user type had the largest influence on trip length, although, in general, people tend to take far more frequent, shorter rides. This suggests that Lyft is experiencing a real benefit from offering two customer models. They appear to be accommodating the market for both leisure and commuter riders appropriately. It is also important to note that the r-squared for this analysis is .062, which is rather low. Unfortunately, this means that the input variables are only accounting for 6.2% of the variation we see in the data. A more robust dataset with additional variables to analyze could have helped improve the r-squared for this regression. However, we were still able to discern important insights from this analysis that helped answer the overall question for this project.

```
model = smf.ols(formula = 'trip_duration_sec ~ age_range + member_gender + user_type', data = san_francisco_rides)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      trip_duration_sec   R-squared:                       0.062
Model:                            OLS   Adj. R-squared:                  0.062
Method:                 Least Squares   F-statistic:                 1.182e+04
Date:                Wed, 11 Dec 2019   Prob (F-statistic):               0.00
Time:                        20:06:43   Log-Likelihood:            -5.8160e+06
No. Observations:              719010   AIC:                         1.163e+07
Df Residuals:                  719005   BIC:                         1.163e+07
Df Model:                           4
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                 1341.1971      3.116    430.408      0.000    1335.090    1347.305
age_range[T.36-50]         -21.6903      2.119    -10.236      0.000     -25.843     -17.537
age_range[T.Above 50]       35.3900      3.353     10.556      0.000      28.819      41.961
member_gender[T.Male]     -103.6008      2.203    -47.024      0.000    -107.919     -99.283
user_type[T.Subscriber]   -594.1069      2.842   -209.075      0.000    -599.676    -588.537
==============================================================================
Omnibus:                  1088086.896   Durbin-Watson:                   1.827
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        707810535.579
Skew:                           9.371   Prob(JB):                         0.00
Kurtosis:                     155.561   Cond. No.                         6.91
==============================================================================
```

Figure 24 Code and result of data model

VI. CONCLUSION, CHALLENGES & LEARNINGS

Overall, Lyft's Baywheels has had a rather successful first five months of 2019. The use of relevant open source data for research is a good strategy to gain a deeper understanding of the bike sharing initiative. Based on the data available, our data analysis demonstrates how the company can measure its success.

The metrics investigated here are not the only determinants of success. Other possible factors that can make bike sharing systems experience market failure may include bike vandalism, inappropriate bike maintenance, insufficient availability of bikes at certain times, or even the number of helmets. Further research is required to discover additional factors related to land use patterns, existing public transport networks, bike related facilities, and

bike infrastructure that may also affect the success of the shared bike system. Another avenue for research is to examine how bike sharing systems compare given a rapidly emerging dock-less bike system.

CHALLENGES

This project presented us with a few obstacles to navigate through. Some of these obstacles include data limitations, structuring good questions, and determining a way to geographically separate locations.

1. **Data Limitations:** The first challenge we discovered was that we were restricted by the fact that Lyft chose not to disclose individual identifiers for each user. If we had a unique identifier, we would have been able to determine which user rode with Baywheels most frequently. Many of the questions that arose in the beginning would have benefitted from a unique identifier, but we found other ways to classify groups that would still allow us to do a comparative analysis.

2. **Relevant Questions:** The second challenge that we encountered was structuring our questions in a way that was relevant, possible to answer through data modeling, and that flowed well together. We spent the first few meetings debating which questions fit the project and what would be the most effective way to represent the data.
3. **Geographical incorporation for comparative analysis and reduction:** Once we determined the questions, it became clear that we didn't have some necessary columns like "Start City" and "End City", so we had to figure out a way to represent cities to compare data between each. In order to overcome this challenge, we researched how to incorporate latitude and longitude, and proceeded to separate the cities based on this criteria. If we hadn't been able to incorporate the longitude and latitude, we wouldn't have been able to compare data between cities. This was a challenge, but beneficial in the long run. Each of these challenges helped us shape this project to be as efficient and effective as possible.

LEARNINGS

1. **Never assume, rely on data analysis**: Based on the data we had access to, we made some assumptions about how we thought certain variables would affect the duration of trips. While our predicted model was not as accurate as we had hoped, we learned that it is important to trust the power of data analysis
2. **Effect of data cleaning and outlier identification:** The data cleaning process made our data as accurate as possible and allowed us to identify and remove outliers that would have negatively impacted the interpretation of our data. For example, the effect of unusually long rides towards revenue generation.
3. **Benefits of data visualization:** We also learned that it is much easier to understand data through visualizations (as opposed to reading lines of code). An illustration of this learning is that the demand and supply analysis for stations was much better visualized graphically. It would have been quite tedious to analyze otherwise.

References

"Computational Urban Planning and Management for Smart Cities." *Google Books*, Google, https://books.google.com/books?id=ySiXDwAAQBAJ&printsec=frontcover#v=onepage&q&f=false.

Lyft, Inc. (n.d.). Bike share in the San Francisco Bay Area: Bay Wheels. Retrieved from https://www.lyft.com/bikes/bay-wheels.

Wikipedia contributors. (2019, December 13). San Francisco Bay Area. In *Wikipedia, The Free Encyclopedia*. Retrieved 22:52, December 13, 2019, from https://en.wikipedia.org/w/index.php?title=San_Francisco_Bay_Area&oldid=930617258