Citi Bike Trip Data

Citi Bike is a New York based bike share program that enables short-term bicycle rentals. The infrastructure includes a network of over 800 docking stations and a fleet of over 14,000 bicycles across Manhattan, Brooklyn, Queens, and Jersey City. It is easy to observe the ubiquity of Citi Bike infrastructure across the city. What may not be evident is that Citi Bike is the 25th largest transit system in the United States by volume. That makes Citi Bike larger than BART in the San Francisco Bay Area and almost as large as the PATH system in New York.

https://www.bloomberg.com/news/articles/2022-10-04/when-public-transit-stumbles-bikesharing-can-step-up

Citi Bike has experienced tremendous growth for a company launched in 2013 with 332 stations and 6,000 bikes. In February of 2023, Citi Bike reported over 1,750,000 trips. However, as Citi Bike continues expanding its footprint and bicycle fleet, it faces new obstacles.

The COVID-19 pandemic fueled the popularity of Citi Bike, but it has also caused more complicated usage patterns. With hybrid work schedules, a shift from public transportation. Usage has increased; however, the usage patterns have become more chaotic. Citi Bike actively uses machine learning to help optimize its fleet and balance bike availability with demand.

Proposal

This project will use the monthly Citi Bike rider data to visualize ridership patterns across the network. Highlighting volume at each station and aggregate trip information. The goal is to identify high-traffic stations and routes across the Citi Bike network. The Citi Bike data is available through a non-exclusive, royalty-free, limited, perpetual license from CityBike directly or through a creative commons license from Google's Big Query. It is the same data set available through 2 different channels.

https://creativecommons.org/licenses/by/4.0/ https://ride.citibikenyc.com/data-sharing-policy

For this project, I will explore a map view of the trip data. The geographic representation of data will enable a better understanding of relationships between trips in physical space.

Data Source

I will download February 2023 data for the amazon web services website.

https://s3.amazonaws.com/tripdata/index.html

The 02302-citibike-tripdata.csv file contains 1752148 rows representing individual trips and 13 columns using the following schema:

- Ride ID
- Rideable type
- Started at
- Ended at
- Start station name
- Start station ID
- End station name
- End station ID
- Start latitude
- Start longitude
- End latitude
- End Longitude
- Member or casual ride

Data Access

In [243…

```
import numpy as np
import pandas as pd
import json
import folium
```

In [244…

```
graph_factor = 1
hurdle_rate = 2
```

In [245…

```
df = pd.read_csv('./data/202302-citibike-tripdata.csv', parse_dates=['starte
df.head()
```

Out[245]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_stat
0	16991A7C313082EB	classic_bike	2023-02-16 18:20:42	2023-02-16 18:38:06	Kosciuszko St & Nostrand Ave	4
1	856FFB566BEEB824	classic_bike	2023-02-09 17:29:36	2023-02-09 17:33:07	Riverside Dr & W 138 St	7
2	B1FE28D50B493430	classic_bike	2023-02-16 15:33:51	2023-02-16 15:35:01	Clinton St & Tillary St	4
3	870EA3D724EA6162	classic_bike	2023-02-23 17:11:39	2023-02-23 17:12:56	Clinton St & Tillary St	4
4	7DE8FA9EAAE8C4ED	electric_bike	2023-02-18 19:29:17	2023-02-18 19:50:52	Audubon Ave & W 192 St	84

In [246…

```
df.shape
```

Out[246]:

```
(1752148, 13)
```

In [247…

```
df.columns
```

Out[247]:

```
Index(['ride_id', 'rideable_type', 'started_at', 'ended_at',
       'start_station_name', 'start_station_id', 'end_station_name',
       'end_station_id', 'start_lat', 'start_lng', 'end_lat', 'end_lng',
       'member_casual'],
      dtype='object')
```

In [248…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1752148 entries, 0 to 1752147
Data columns (total 13 columns):
 #   Column              Dtype         
---  ------              -----         
 0   ride_id             object        
 1   rideable_type       object        
 2   started_at          datetime64[ns]
 3   ended_at            datetime64[ns]
 4   start_station_name  object        
 5   start_station_id    object        
 6   end_station_name    object        
 7   end_station_id      object        
 8   start_lat           float64       
 9   start_lng           float64       
 10  end_lat             float64       
 11  end_lng             float64       
 12  member_casual       object        
dtypes: datetime64[ns](2), float64(4), object(7)
memory usage: 173.8+ MB
```

Visualization

The map-based visualization will focus on a single weekend to simplify the view. The start stations for a specific trip is highlighted in green, and the ending station will be red. The size of the marker used will indicate traffic volume.

In [249…

```
df = df[(df['started_at'] > "2023-02-04") & (df['started_at'] <= "2023-02-05
```

In [250…

```
df.shape
```

Out[250]:

```
(25740, 13)
```

In [251…

```
import folium
m = folium.Map(location=[40.730610, -73.935242],zoom_start=13, tiles = 'Cart
```

In [252…

```
start_df = df.groupby(['start_station_id','start_station_name','start_lat','
start_df.rename(columns={'ride_id':'count'}, inplace=True)
start_df
```

Out[252]:

	start_station_id	start_station_name	start_lat	start_lng	count
0	2782.02	5 Ave & 66 St	40.635911	-74.019768	1
1	2832.03	4 Ave & Shore Road Dr	40.637033	-74.022141	2
2	2883.03	3 Ave & Wakeman Pl	40.638303	-74.024734	1
3	2912.08	6 Ave & 60 St	40.638226	-74.013803	1
4	2923.01	62 St & 4 Ave	40.639859	-74.019776	2
...	...	...	...	...	...
11983	8778.01	E Mosholu Pkwy & Van Cortlandt Ave E	40.876518	-73.883670	1
11984	8795.01	Jerome Ave & E Mosholu Parkway S	40.879447	-73.885350	1
11985	8795.01	Jerome Ave & E Mosholu Parkway S	40.879455	-73.885175	1
11986	8795.01	Jerome Ave & E Mosholu Parkway S	40.879497	-73.885213	1
11987	8841.03	W Mosholu Pkwy S & Sedgwick Ave	40.882260	-73.887020	1

11988 rows × 5 columns

In [253…

```
end_df = df.groupby(['end_station_id','end_station_name','end_lat','end_lng'
end_df.rename(columns={'ride_id':'count'}, inplace=True)
end_df
```

Out[253]:

	end_station_id	end_station_name	end_lat	end_lng	count
0	2821.05	7 Ave & 62 St	40.635560	-74.012980	1
1	2883.03	3 Ave & Wakeman Pl	40.638246	-74.024714	2
2	2932.03	Wakeman Pl & Ridge Blvd	40.639421	-74.026823	1
3	3011.03	59 St & 4 Ave	40.641269	-74.017651	7
4	3038.08	50 St & 7 Ave	40.642501	-74.006055	1
...	...	...	...	...	...
1796	8795.03	Grand Concourse & E Mosholu Pkwy S	40.877964	-73.884755	1
1797	8841.03	W Mosholu Pkwy S & Sedgwick Ave	40.882260	-73.887020	1
1798	JC072	Morris Canal	40.712419	-74.038526	1
1799	SYS035	Pier 40 Dock Station	40.728660	-74.011980	2
1800	SYS038	Morgan Loading Docks	40.709306	-73.931175	3

1801 rows × 5 columns

Start and End Stations

Map the start and end stations across New York

In [254…

```
m = folium.Map(location=[40.730610, -73.935242],zoom_start=13, tiles = 'Cart

for i in range(0,len(start_df)):
   folium.Circle(
        location=[start_df.iloc[i]['start_lat'], start_df.iloc[i]['start_lng
        radius=float(start_df.iloc[i]['count'])*graph_factor,
        popup=start_df.iloc[i]['start_station_name'],
        color="green",
        fill=True,
        fill_color="green"
   ).add_to(m)

m
```

Out[254]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [255…

```
# reset graph
m = folium.Map(location=[40.730610, -73.935242],zoom_start=13, tiles = 'Cart
for i in range(0,len(end_df)):
   folium.Circle(
        location=[end_df.iloc[i]['end_lat'], end_df.iloc[i]['end_lng']],
        radius=float(end_df.iloc[i]['count'])*graph_factor,
        popup=end_df.iloc[i]['end_station_name'],
        color="red",
        fill=True,
        fill_color="red"
   ).add_to(m)

m
```

Out[255]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [256…

```
m = folium.Map(location=[40.730610, -73.935242],zoom_start=13, tiles = 'Cart

for i in range(0,len(end_df)):
   folium.Circle(
        location=[end_df.iloc[i]['end_lat'], end_df.iloc[i]['end_lng']],
        radius=float(end_df.iloc[i]['count'])*graph_factor,
        popup=end_df.iloc[i]['end_station_name'],
        color="red",
        fill=True,
        fill_color="red"
   ).add_to(m)

for i in range(0,len(start_df)):
   folium.Circle(
        location=[start_df.iloc[i]['start_lat'], start_df.iloc[i]['start_lng
        radius=float(start_df.iloc[i]['count'])*graph_factor,
        popup=start_df.iloc[i]['start_station_name'],
        color="green",
        fill=True,
        fill_color="green"
   ).add_to(m)



m
```

Out[256]: Make this Notebook Trusted to load map: File -> Trust Notebook

Trip Data

Map the trips across New York

In [257…

```
trip_df = df.groupby(['start_station_id','start_station_name','start_lat','s
                      'end_station_id','end_station_name','end_lat','end_lng'
trip_df.rename(columns={'ride_id':'count'}, inplace=True)
trip_df = trip_df[trip_df['count'] > hurdle_rate]
trip_df.shape
```

Out[257]:

```
(599, 9)
```

In [258…

```
trip_df.sort_values('count', ascending=True)[:10]
```

Out[258]:

	start_station_id	start_station_name	start_lat	start_lng	end_station_id	end_s
35	3169.07	53 St & 4 Ave	40.644862	-74.014531	3220.01	5
10317	6233.05	W 16 St & The High Line	40.743349	-74.006818	6233.05	W 16
10290	6230.04	FDR Drive & E 35 St	40.744219	-73.971212	6322.01	E
10289	6230.04	FDR Drive & E 35 St	40.744219	-73.971212	6230.04	FDR D
10267	6224.06	8 Ave & W 24 St	40.745911	-73.998071	6382.05	W 2
18268	5382.07	Forsyth St & Grand St	40.717798	-73.993161	5262.09	M
18288	5406.02	Rivington St & Ridge St	40.718502	-73.983299	5406.02	F
18289	5406.02	Rivington St & Ridge St	40.718502	-73.983299	5453.01	F
10235	6224.05	W 20 St & 8 Ave	40.743453	-74.000040	6022.04	E
10205	6224.03	W 22 St & 8 Ave	40.744751	-73.999154	6072.11	8

In [259…

```
for i in range(0,len(trip_df)):
    lat_lng_points = list()
    lat_lng_points.append(trip_df.iloc[i][['start_lat','start_lng']].values.
    lat_lng_points.append(trip_df.iloc[i][['end_lat','end_lng']].values.toli
    #lat_lng_points

    folium.PolyLine(lat_lng_points,
                    color='gray',
                    tooltip=trip_df.iloc[i]['count'],
                    weight=float(trip_df.iloc[i]['count'])*graph_factor,  #
                    opacity=0.2  # transparency
                    ).add_to(m)

m
```

```
Out[259]: Make this Notebook Trusted to load map: File -> Trust Notebook
```

Initial Observations

- High Volume Stations have net inflows of traffic. The red circles are larger than the green circles indicating that these stations are more often the end of trips vs. the start of trips.
- High Traffic End Stations are in congested areas. There seems to be a band of high-volume stations between Canal Street and Columbus Circle.
- There is a higher frequency of Short Trips. The most frequent trip between stations is shorter in distance less than ten blocks.