Data Analysis: NYC CitiBike Rentals - June 2023

When looking at this data, we're looking to answer the following questions from a Business perspective:

- Who are our most passionate users; who rides the longest between them?
- Are there specific ride preferences?
- Are there popular start and endpoints, which stations are these?
- Where do people cycle the most?
- Which days of the week are most popular for cycling; which type of user is making use of our bikes on these days?

For more information regarding my collection, inspection, scanning, processing and storage of data, please check my website or github repository for more details to learn how I work.

Let's explore the Data!

```
#to query the data, we need to use pandas, so we import it here
import pandas as pd
#we've imported our CSV file into datalore and access it at this path
file_path = "/data/notebook_files/JC-202306-citibike-tripdata.csv"
#we want to see the first few lines of data from each column of data in a tab
df = pd.read_csv(file_path)
#we request the first 10 lines of data to be rendered wo we can see it
print("Head Data-------------------------------------------------------------
print(df.head(10))
print("Tail Data-------------------------------------------------------------
#we also want to see the last few lines of data
print(df.tail(10))
```

```
Head Data--------------------------------------------------------------------
            ride_id  rideable_type           started_at             ended_
0  37BE5FCB1A385CDA    classic_bike  2023-06-27 16:06:27  2023-06-27 16:10:
1  9C96B4C6CBBB31AD    classic_bike  2023-06-24 10:46:58  2023-06-24 10:55:
2  C91293605D4BEC07   electric_bike  2023-06-04 20:30:13  2023-06-04 20:34:
3  2920063442116A46    classic_bike  2023-06-08 19:10:06  2023-06-08 19:13:
4  E205FD8C18BA263A    classic_bike  2023-06-23 18:53:51  2023-06-23 19:17:
5  47D0CEE9AD0F0F5F    classic_bike  2023-06-04 14:19:40  2023-06-04 14:24:
6  FB14B5FB30B1205C   electric_bike  2023-06-27 08:19:07  2023-06-27 08:23:
7  033A18A93A11FE3F    classic_bike  2023-06-08 20:17:25  2023-06-08 20:19:
8  48E9626FE5418C84    classic_bike  2023-06-11 19:57:03  2023-06-11 20:35:
9  2FA0ECB0D11C2E82    classic_bike  2023-06-11 19:12:02  2023-06-11 19:55:

      start_station_name start_station_id       end_station_name  \
0                Hilltop            JC019        Christ Hospital
1  Baldwin at Montgomery            JC020          Hamilton Park
2                Hilltop            JC019           Brunswick St
3  Baldwin at Montgomery            JC020            Astor Place
```

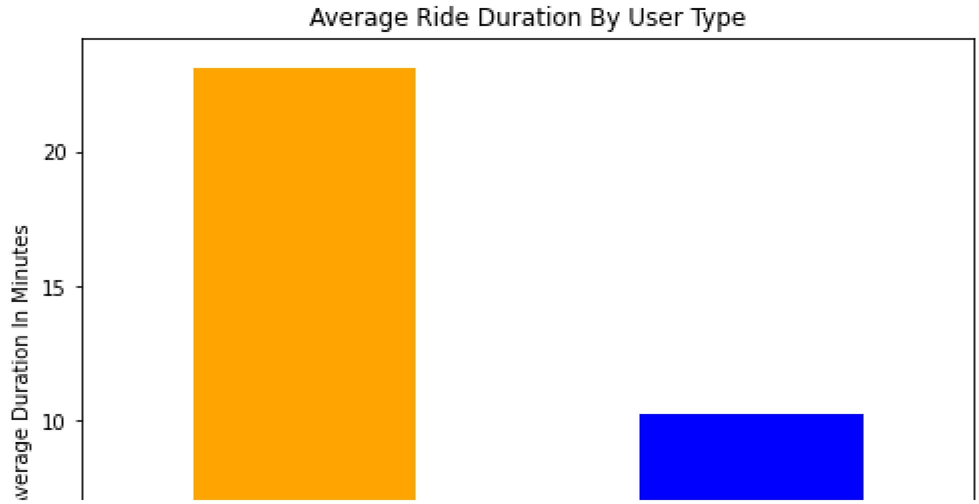
```
4  Baldwin at Montgomery           JC020       Adams St & 11 St
```

```
#we're going to be looking for some data to create a visualization, so we're
df['ride_durations'] = df['start_lat'].astype(str) + ',' + df['start_lng'].as
df['end_lat'].astype(str) + ',' + df['end_lng'].astype(str)
#we performed a calculation in the column subrtacting end from start latitude
print(df[['start_lat', 'start_lng', 'end_lat', 'end_lng', 'ride_durations']].
#the content we requested has a new column with the ride duration calculated
```

```
   start_lat  start_lng    end_lat    end_lng               ride_durations
0  40.731115 -74.057468  40.734786 -74.050444  40.731114864,-74.057468176|
1  40.723473 -74.064338  40.727596 -74.044247  40.723472595,-74.064338207|
2  40.731169 -74.057574  40.724176 -74.050656      40.7311689,-74.0575736|
3  40.723499 -74.064335  40.719282 -74.071262  40.723499179,-74.064335108|
4  40.723511 -74.064277  40.750916 -74.033541  40.723511338,-74.064276576|
```

```
#we need to create our vusualization and for this we use matplotlib
import matplotlib.pyplot as plt
#we first convert our start and ending distances into a time format and creat
df['started_at'] = pd.to_datetime(df['started_at'])
df['ended_at'] = pd.to_datetime(df['ended_at'])
#this is where we take the data from our previous "ride_duration" column to p
df['ride_duration'] = (df['ended_at'] - df['started_at']).dt.total_seconds()
#now we group our cycle time by type of user (subscriber, or casual)
average_duration_by_usertype = df.groupby('member_casual')['ride_duration'].m
#first we decide how big our graphic will be( for each number shown, add "00"
plt.figure(figsize=( 8, 6))
#now we tell the program what color the bars should be
average_duration_by_usertype.plot(kind='bar', color=['orange', 'blue'])
#what do we want to label our graph/image as
plt.title('Average Ride Duration By User Type')
#what are we going to show on the bottom (the "X Axis") - this is normally wh
plt.xlabel('User Type')
#what number is each bar going to reach (the "Y Axis") - holds the value of a
plt.ylabel('Average Duration In Minutes')
#X-ticxks, just labels each column individually - 0 = member, 1 = casual
plt.xticks(rotation=0, ticks=[0, 1], labels=['Member', 'Casual'])
#we tell the program to show the graphic
plt.show()
```

Download



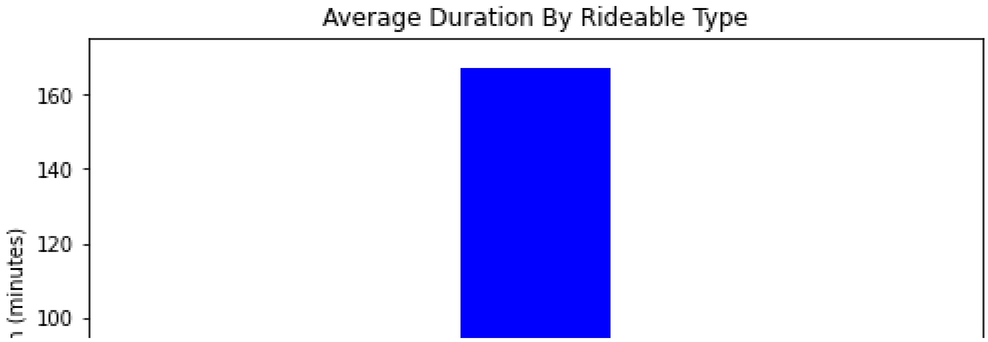
Answering The First Question: Who are our most passionate users?

From what we can see of the data, the subscribers make up the vast majority of passionate users, virtually doubling the number of casual users by the amount of time spent cycling.

Subscribers cycle the longest.

```
#we're creating a new query to show how often people cycle based on the type
average_duration_by_rideable_type = df.groupby('rideable_type')['ride_duratio
#here we create a bar chart, similar to the previous one
plt.figure(figsize=(8, 6))
average_duration_by_rideable_type.plot(kind='bar', color='blue')
plt.title('Average Duration By Rideable Type')
plt.xlabel('Rideable Type')
plt.ylabel('Average Duration (minutes)')
plt.xticks(rotation=0)
plt.show()
```

Download

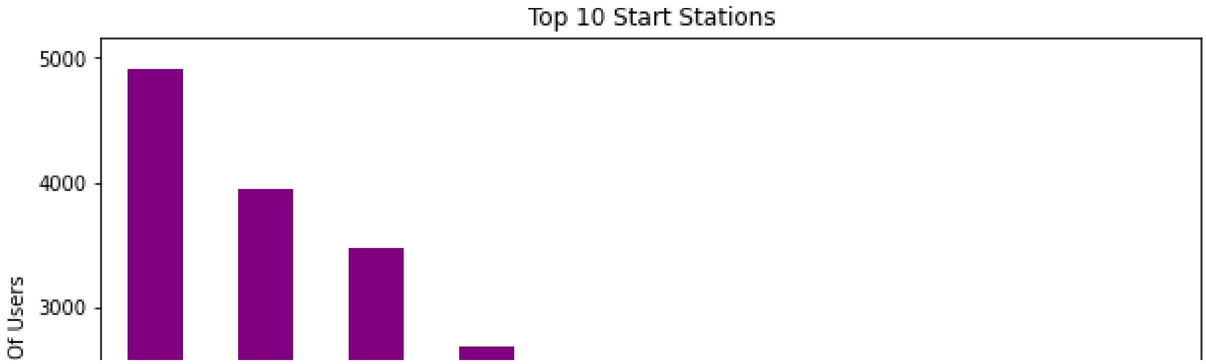


Answering the Second Question: Are there specific Bike Preferences?

From the data, we can see that the vast majority of bikes used are the docked rideable type.

```
#here we are listing the top 10 start stations
top_start_stations = df['start_station_name'].value_counts().head(10)
#again, we're creating a bar chart, following a similar process as before
plt.figure(figsize=(10, 6))
top_start_stations.plot(kind='bar', color='purple')
plt.title('Top 10 Start Stations')
plt.xlabel('Start Station Name')
plt.ylabel('Number Of Users')
plt.xticks(rotation=45, ha='right')
plt.show()
```

Download



```
# Create a DataFrame to count the number of rides ending at each station
top_10_end_stations = df['end_station_name'].value_counts().head(10)

# Create a bar chart
plt.figure(figsize=(10, 6))
top_10_end_stations.plot(kind='bar', color='grey')
plt.title('Top 10 Most Popular End Stations')
plt.xlabel('Number of Rides')
plt.ylabel('End Station Name')
plt.xticks(rotation=45, ha='right')
plt.show()
```

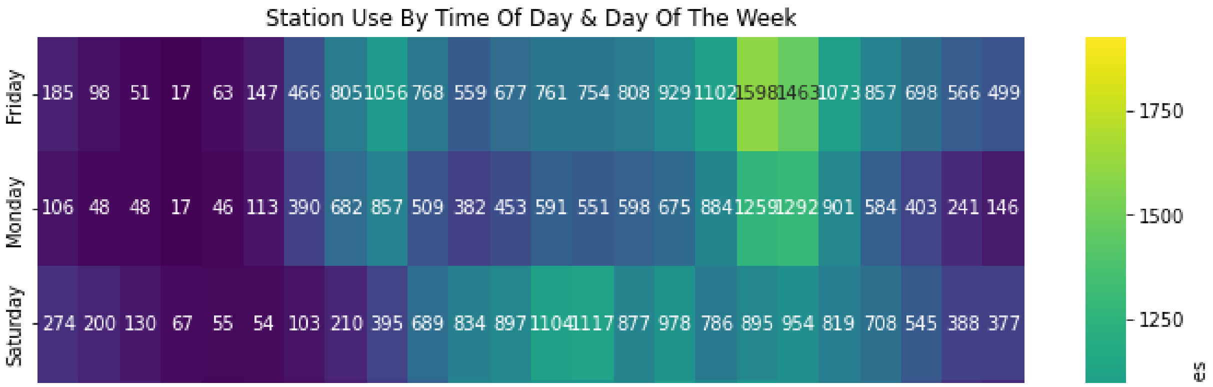
Download

Answering the Third Question: Are there popular start and end stations?

We've listed the top 10 starting and top 10 end stations. Below, we'll provide a graphic heatmap to show the most popular, and busiesst stations by day of week and time of day.

```
#we're importing seaborn for the heatmap visual
import seaborn as sns

#write code to show where most rides start from and at what time of day and w
df['started_at'] = pd.to_datetime(df['started_at'])

#which day of the week is busiest?
df['day_of_week'] = df['started_at'].dt.day_name()

#what hour of the day is busiest?
df['hour_of_day'] = df['started_at'].dt.hour

#let's prepare our heatmap data
heatmap_data = df.pivot_table(index='day_of_week', columns='hour_of_day',valu

#time to create the visual
plt.figure(figsize=(12,8))
sns.heatmap(heatmap_data, cmap='viridis', annot=True, fmt='g', cbar_kws={'lab
plt.title('Station Use By Time Of Day & Day Of The Week')
plt.show()
```

Download



```
#Let's count the number of rides that start
start_station_counts = df['start_station_name'].value_counts().reset_index()
start_station_counts.columns = ['Start Station Name', 'Ride Count']

# Next we count the number of ending rides
end_station_counts = df['end_station_name'].value_counts().reset_index()
end_station_counts.columns = ['End Station Name', 'Ride Count']

# Let's have a look at the data to get our totals
total_station_counts = pd.concat([start_station_counts, end_station_counts]).

# Here we sort the locations in descending order
top_25_locations = total_station_counts.sort_values(by='Ride Count', ascendin

# View the 25 locations listed in order of popularity
print(top_25_locations)
```

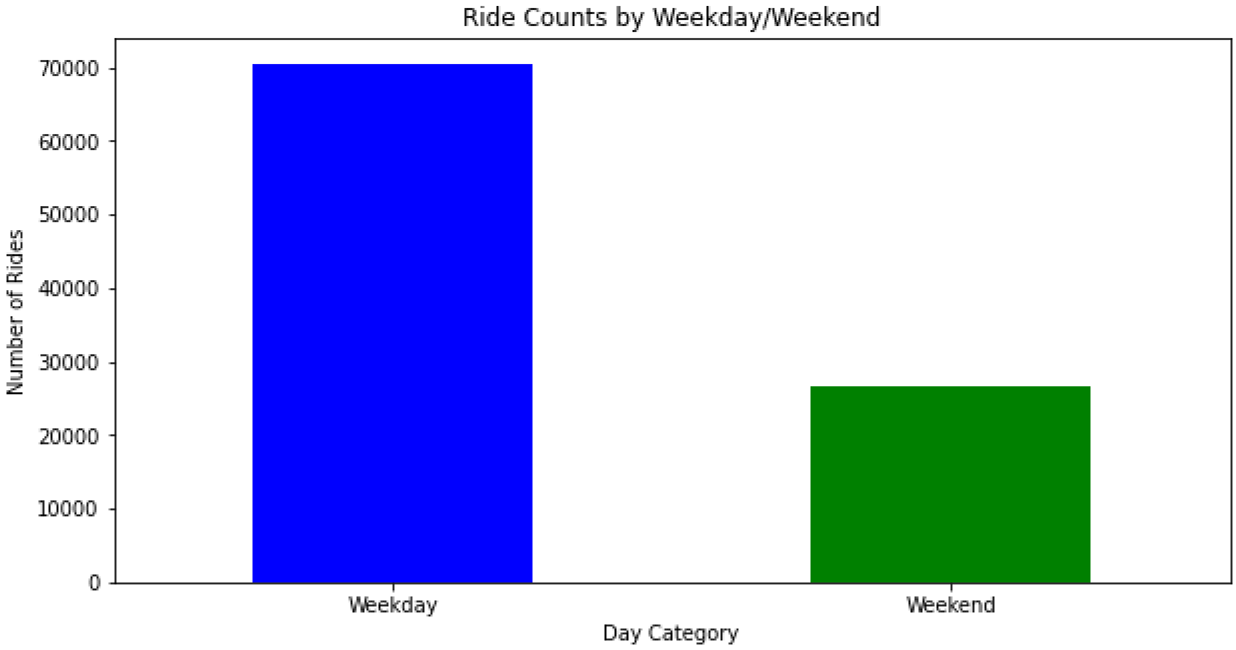
```
                              Start Station Name  Ride Count
73                                  Grove St PATH        4918
81         Hoboken Terminal - River St & Hudson Pl        3942
122   South Waterfront Walkway - Sinatra Dr & 1 St        3474
80        Hoboken Terminal - Hudson St & Hudson Pl        2686
110                                   Newport PATH        2504
42               City Hall - Washington St & 1 St        2394
74                                  Hamilton Park        2232
111                                   Newport Pkwy        2177
3                            12 St & Sinatra Dr N        2164
4                  14 St Ferry - 14 St & Shipyard Ln        1931
101                               Marin Light Rail        1897
79                       Hoboken Ave at Monmouth St        1843
41                                      City Hall        1823
65                                    Exchange Pl        1786
2                            11 St & Washington St        1782
82                                 Hudson St & 4 St        1670
75                                     Harborside        1619
47              Columbus Park - Clinton St & 9 St        1590
92                              Liberty Light Rail        1582
```

Answering Question Four: Where Do People Cycle The Most?

Based upon the data here, it seems like the area surrounding Grove St Path is the most popular cycling area, beating out other areas by almost 1000 additional rides.

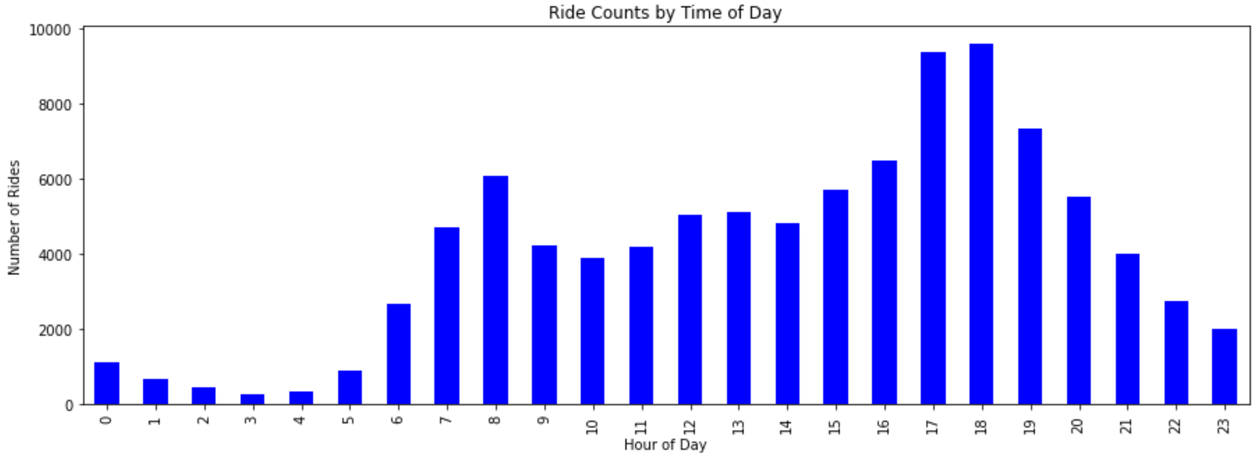
```
# Convert 'started_at' column to datetime object
df['started_at'] = pd.to_datetime(df['started_at'])

# Categorize days into weekdays and weekends
df['day_category'] = df['started_at'].dt.dayofweek.map(lambda x: 'weekend' if

# Plot ride counts for weekdays and weekends
plt.figure(figsize=(10, 5))
df.groupby('day_category')['ride_id'].count().plot(kind='bar', color=['blue',
plt.title('Ride Counts by Weekday/Weekend')
plt.xlabel('Day Category')
plt.ylabel('Number of Rides')
plt.xticks(rotation=0, ticks=[0, 1], labels=['Weekday', 'Weekend'])
plt.show()
```

Download

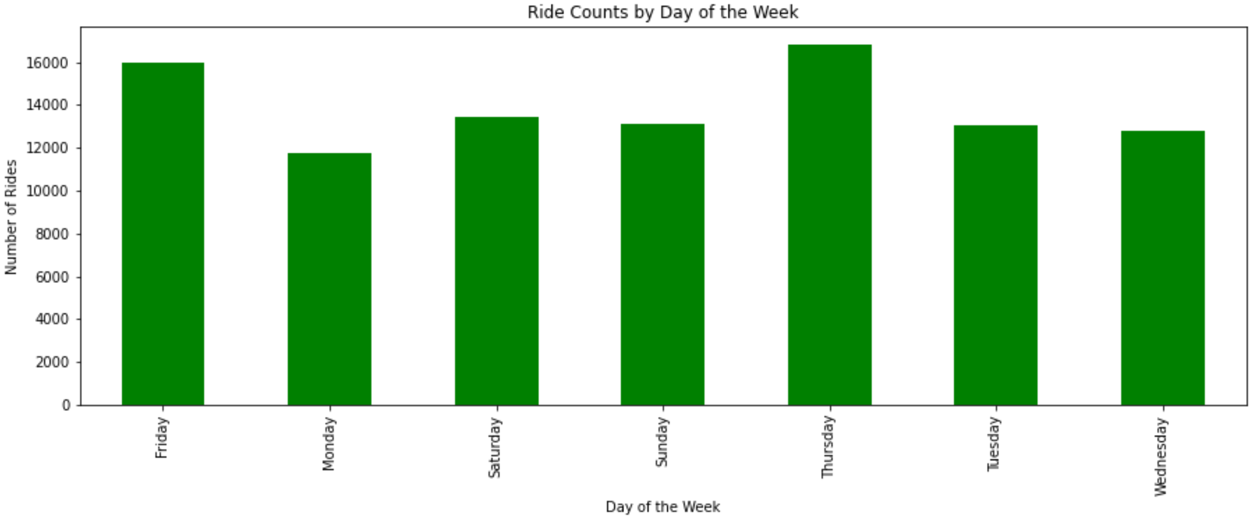


```
# Convert 'started_at' column to datetime object
df['started_at'] = pd.to_datetime(df['started_at'])

# Extract time of day, day of the week, and month
df['hour_of_day'] = df['started_at'].dt.hour
df['day_of_week'] = df['started_at'].dt.day_name()
df['month'] = df['started_at'].dt.month_name()

# Plot ride counts by time of day
plt.figure(figsize=(15, 5))
df.groupby('hour_of_day')['ride_id'].count().plot(kind='bar', color='blue')
plt.title('Ride Counts by Time of Day')
plt.xlabel('Hour of Day')
plt.ylabel('Number of Rides')
plt.show()

# Plot ride counts by day of the week
plt.figure(figsize=(15, 5))
df.groupby('day_of_week')['ride_id'].count().plot(kind='bar', color='green')
plt.title('Ride Counts by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Rides')
plt.show()

# Plot ride counts for each month separately
for month in df['month'].unique():
    plt.figure(figsize=(15, 5))
    df[df['month'] == month].groupby('day_of_week')['ride_id'].count().plot(k
    plt.title(f'Ride Counts by Day of the Week - {month}')
    plt.xlabel('Day of the Week')
    plt.ylabel('Number of Rides')
    plt.show()
```

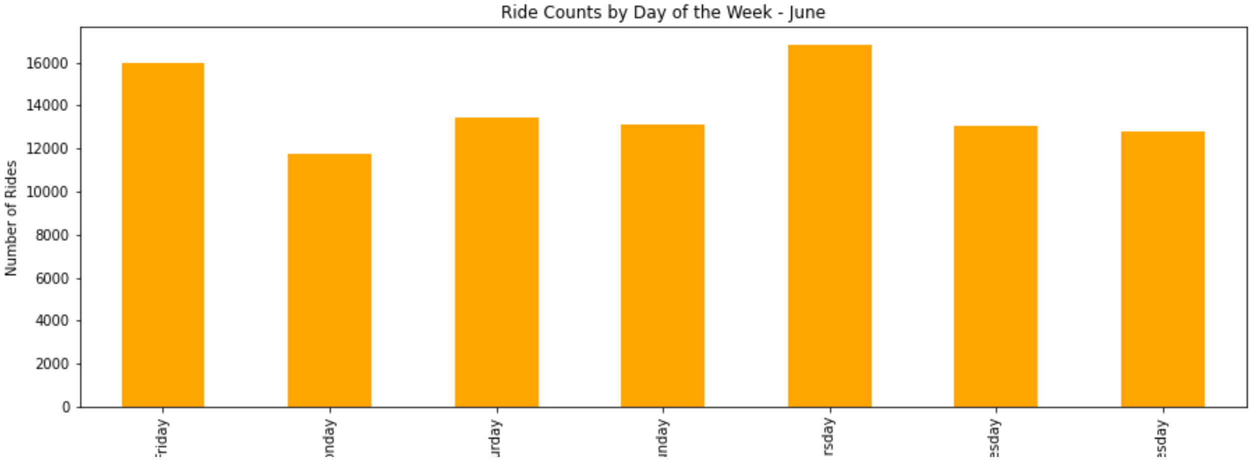
Download



Download

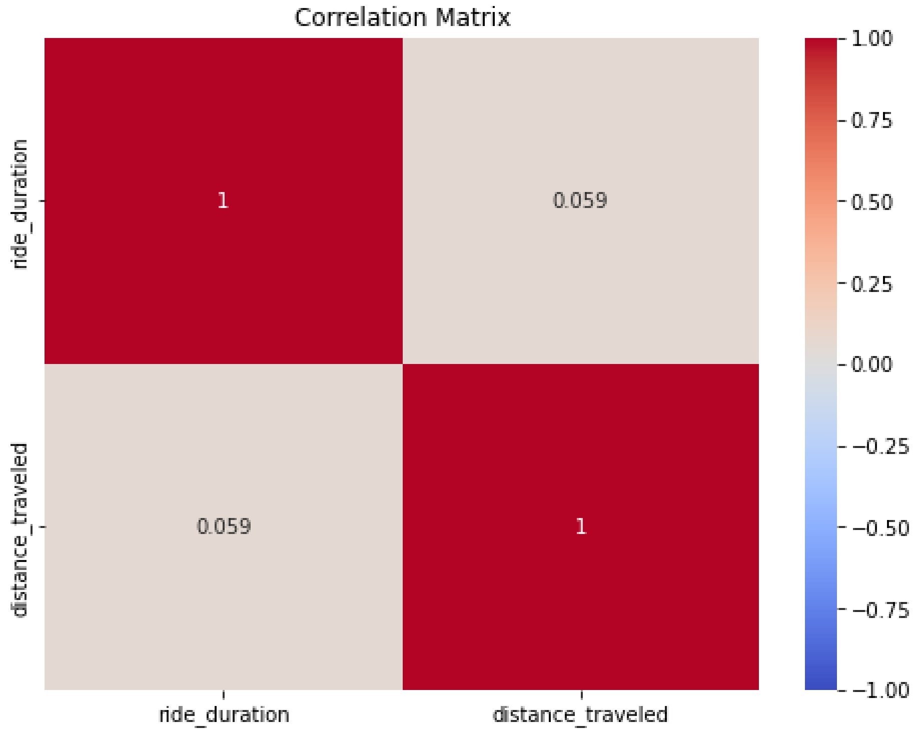

Ride Counts by Day of the Week
Number of Rides
0
2000
4000
6000
8000
10000
12000
14000
16000
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
Day of the Week

Download

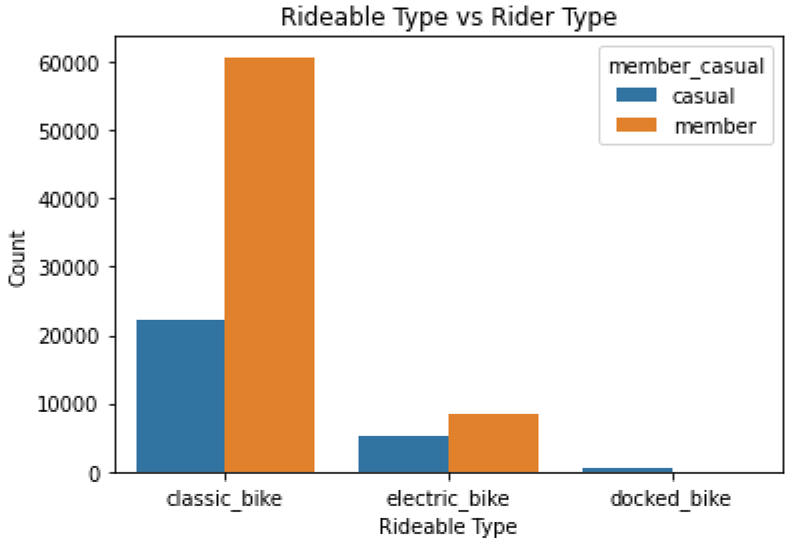

Ride Counts by Day of the Week - June
Number of Rides
0
2000
4000
6000
8000
10000
12000
14000
16000
Friday
onday
urday
unday
rsday
esday
esday

```
# Convert 'started_at' and 'ended_at' columns to datetime objects
df['started_at'] = pd.to_datetime(df['started_at'])
df['ended_at'] = pd.to_datetime(df['ended_at'])

# Calculate ride duration in minutes
df['ride_duration'] = (df['ended_at'] - df['started_at']).dt.total_seconds()

# Calculate distance traveled based on start and end locations (assuming stra
df['distance_traveled'] = ((df['end_lat'] - df['start_lat'])**2 + (df['end_ln

# Plot correlation matrix
correlation_matrix = df[['ride_duration', 'distance_traveled']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix')
plt.show()

# Plot categorical variables
sns.countplot(x='rideable_type', hue='member_casual', data=df)
plt.title('Rideable Type vs Rider Type')
plt.xlabel('Rideable Type')
plt.ylabel('Count')
plt.show()

# Plot starting and ending locations
plt.scatter(df['start_lng'], df['start_lat'], c='blue', label='Start Location
plt.scatter(df['end_lng'], df['end_lat'], c='red', label='End Location', alph
plt.title('Starting and Ending Locations')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.show()

# Plot bike type vs distance traveled
sns.boxplot(x='rideable_type', y='distance_traveled', data=df)
plt.title('Rideable Type vs Distance Traveled')
plt.xlabel('Rideable Type')
plt.ylabel('Distance Traveled')
plt.show()

# Plot member type frequency
sns.countplot(x='member_casual', data=df)
plt.title('Frequency of Member Type')
plt.xlabel('Member Type')
plt.ylabel('Count')
plt.show()
```

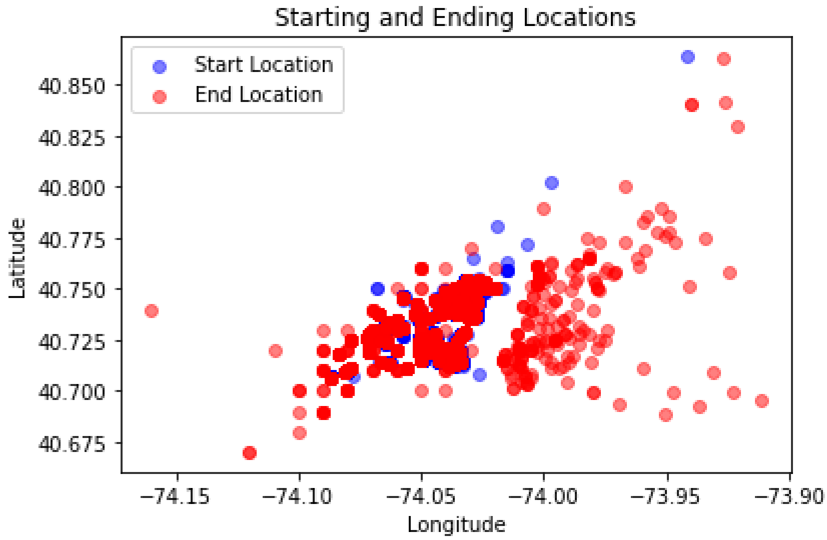
Download



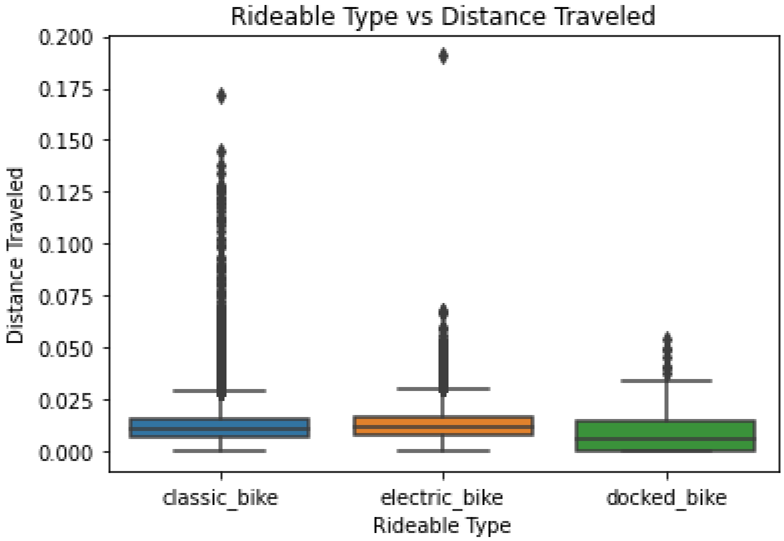
Download



Download



Download



Download

Answering Question Five: Which days of the week are most popular for cycling; which type of user is making use of our bikes on these days?

From what I can deduce from your data, weekdays (especially Thursday & Friday) are the most popular days of the week for cycling. Without a doubt, it seems like your members far outweigh casual users when it comes to the use of your Bikes on these days.

Conclusion

Looking at your data, summary:

- Members outnumber casual users
- Weekdays are the busiest
- The further people ride, the longer they use the bikes for
- Classic bikes are the favorite (by far)
- People seem to like cycling in the evening between 17:00-18:00
- The most popular cycling area is Grove St Path

I hope this information is insightful. There are many other ways of presenting the data that I have used, but for the platform on which I am working now, heatmaps and other interactive visualizations consume too much memory and cannot render.

Please contact me for further information, to ask questions, etc.