TRENDS IN CYCLISTIC BIKE SHARE RIDERSHIP

Examining Differences Among Members and Casual Riders

Prepared by Drew Brinkley

Junior Data Analyst, Cyclistic Marketing Analyst Team

December 14, 2021

Contents

Overview

Background

The Cyclistic bike share program has historically based its marketing efforts on building general brand awareness to attract customers without targeted specific groups. Pricing plans for riding passes in the Chicago service area have included three options: single ride, full-day, and annual memberships.

Cyclistic riders can be divided into two primary categories: casual riders (those who purchase single ride or full-day passes) and members with annual passes. Each rental includes a record of variables such as rental start and end times, location of rental origin and return, and type of bike rented.

Business Objective

In an effort to maximize annual memberships, the Marketing Analytics Team will be designing a new marketing strategy that aims to convert casual riders into annual members. The purpose of this analysis is to compare differences in ridership between Cyclistic members and casual riders to identify differences in how these groups use the bike share that can be used to develop this new strategy.

About the Data

Source

This analysis gathered the historical data on rentals during the previous 12-month period spanning November 2020 to October 2021 for the purpose of identifying trends in ridership that can be used to inform marketing goals. The data was harvested from Cyclistic's online repository, in which each month's data was contained in its own .CSV file. The data was saved locally for use in analysis so as to not compromise the integrity of the main Cyclistic files. There are no expected issues with data reliability since it is maintained in-house.

Assumptions

- Ride durations of less than 1 minute were not considered for analysis - these may mean that a user was trying to return a bike and re-docked it to ensure it was properly secure or represent a "false start" rather than an actual ride.

- The analysis excluded ride types of "docked." This classification represents an open rental that is currently parked at one of the Cyclistic bike stations throughout the service area, though the user will be returning to the bike before completely ending the ride. The additional trip(s) will be counted as new rides.

Limitations

In order to protect the privacy and identity of Cyclistic's customers, the monthly data is scrubbed of all information that could be used to determine a customer's identity. This creates a few limiting factors for this analysis:

1. A proportion of casual riders likely are repeat customers, but there is no way to determine how often they use the bike share service.
2. It cannot be determined how many of Cyclistic's customers reside in the service area and how many are out-of-town visitors / tourists.
3. There is no way to determine where customers choose to access Cyclistic bikes when compared to where they live or work.

Tools

Several tools were used to aid in the analysis in this study. These included the following:

- Excel: Excel was used to initially download and review the monthly Cyclistic data, perform data cleaning actions, and save the cleaned data into .CSV files. A log of the changes made to these files is contained in *Appendix 1*.

- SQL: BigQuery was used to combine the monthly data generated from the previous step into a single database. At that point, there was an additional round of data cleaning on the consolidated data, and then a number of queries were executed to generate summary statistics for analysis. An additional data table containing dates of federal holidays was also uploaded to include in the analysis. The results of these SQL queries were saved locally as .CSV files. A record of executed queries can be found in the *Appendix 2*.
- Google Sheets: Google Sheets was used to view the tables generated from BigQuery, perform calculations to determine various summary statistics, format the results, and generate charts for data visualization. Reference *Appendix 3* for detail on these files.
- Tableau: Tableau was used to assist in data visualization of the query results. Specifically, several maps were generated using the coordinates contained in the bike share data to show geographic trends.

Data Cleaning

Once the monthly data for Cyclistic's bike rentals was downloaded, several steps had to be taken to clean the data before it could be analyzed. The following actions were taken to verify the data before analysis:

- Duplicate records were deleted
- Starting and ending dates for each month were reviewed to verify the data's accuracy
- Minimum and maximum values for columns were calculated to check for outlier values
- Columns pertaining to station names of starting and ending locations were deleted as a large portion of them contained no information (latitudes and longitudes were available and provided more accurate detail on location data)
- Ride length (ending time - staring time) was calculated
    - Data that resulted in a negative ride time (ride ended before it started) was excluded from analysis
- The day of the week for each rental was determined based upon the rental start date
- Monthly data was consolidated into a single dataset via SQL
- Rider types of "docked" and ride durations of less than one minute were excluded from analysis
- Ride totals of the original dataset and cleaned dataset were compared to ensure the expected number of records were excluded
- Data was sorted by date where appropriate

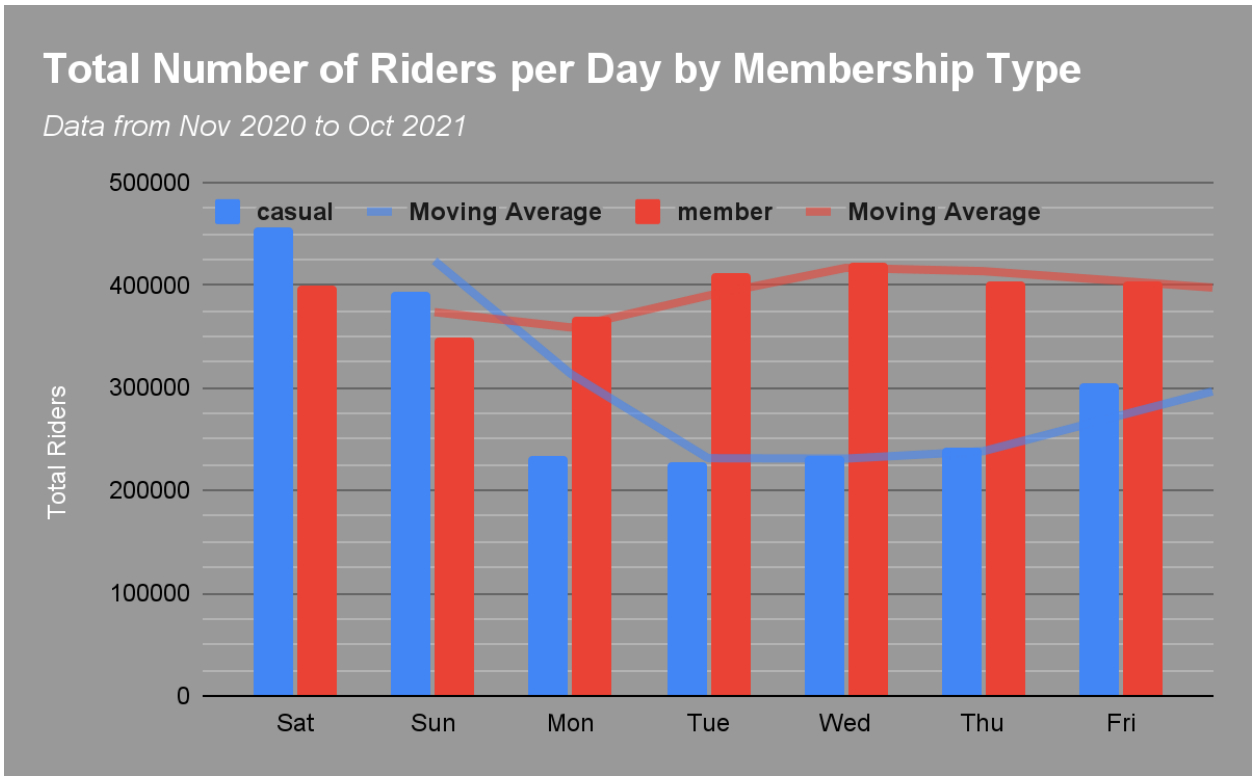
Data Findings

Average Riders per Day

The Cyclistic data revealed a difference in the distribution of riders during the course of a week when comparing the membership type. In order to examine this data, the day of the week each ride began was calculated based upon the rental date. The totals for each day are represented on the following bar chart, sorted by membership type. A line for a 2-period moving average for each member type was added to illustrate the trend.

The average number of members who ride each day equates to 394,077. As shown by the following bar graph, there is not a significant amount of deviation from the average during any day of the week.

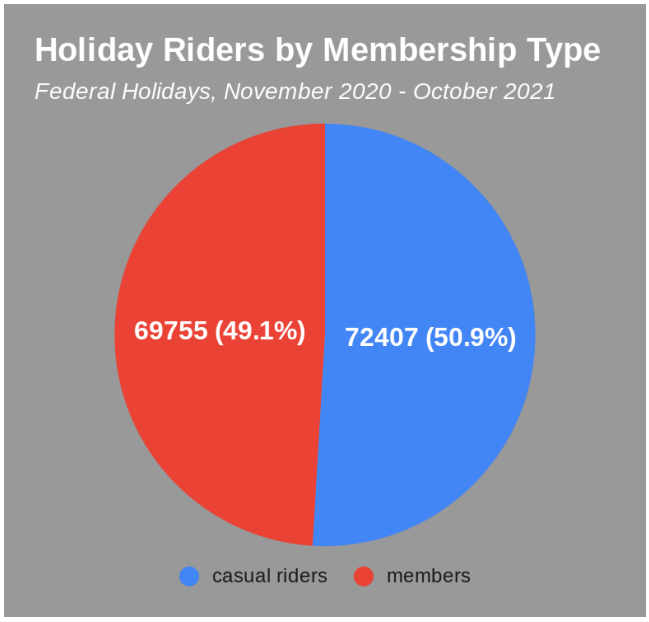
By comparison, casual riders favor using Cyclistic's bike share on weekends, with the peak ridership occurring on Friday, Saturday, and Sunday. These 3 days comprise 55 percent of all casual ridership. There is a significant drop in ridership for casual users during the work week.



	MInimum	**Maximum**	**Average**
Members	348,831	422,857	394,077
Casual Riders	228,379	456,934	299,007

Holidays

The data on riders per day showed that casual riders rode more than members on weekends. Based on this result, the data was queried further to determine the number of riders of each membership group. The federal holidays during the period covered by the Cyclistic data are listed below.



Federal Holidays:

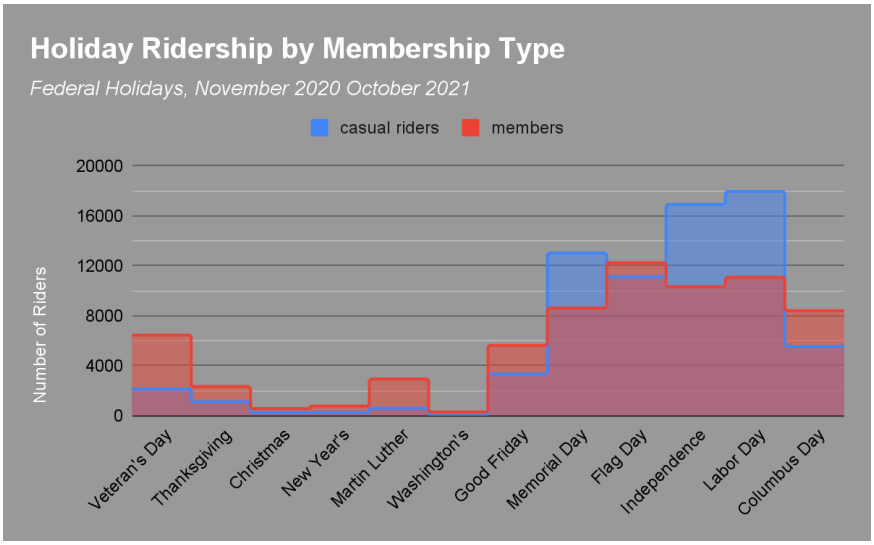
2020

- Veteran's Day: Wed, Nov 11
- Thanksgiving: Thu, Nov 26
- Christmas: Fri, Dec 25

2021

- New Year's Day: Fri, Jan 1
- Martin Luther King Jr. Day: Mon, Jan 18
- Washington's Birthday: Mon, Feb 15
- Good Friday: Fri, Apr 2
- Memorial Day: Mon, May 31
- Flag Day: Mon, Jun 14
- Independence Day: Mon, Jul 5
- Labor Day: Mon, Sep 6
- Columbus Day: Mon, Oct 11

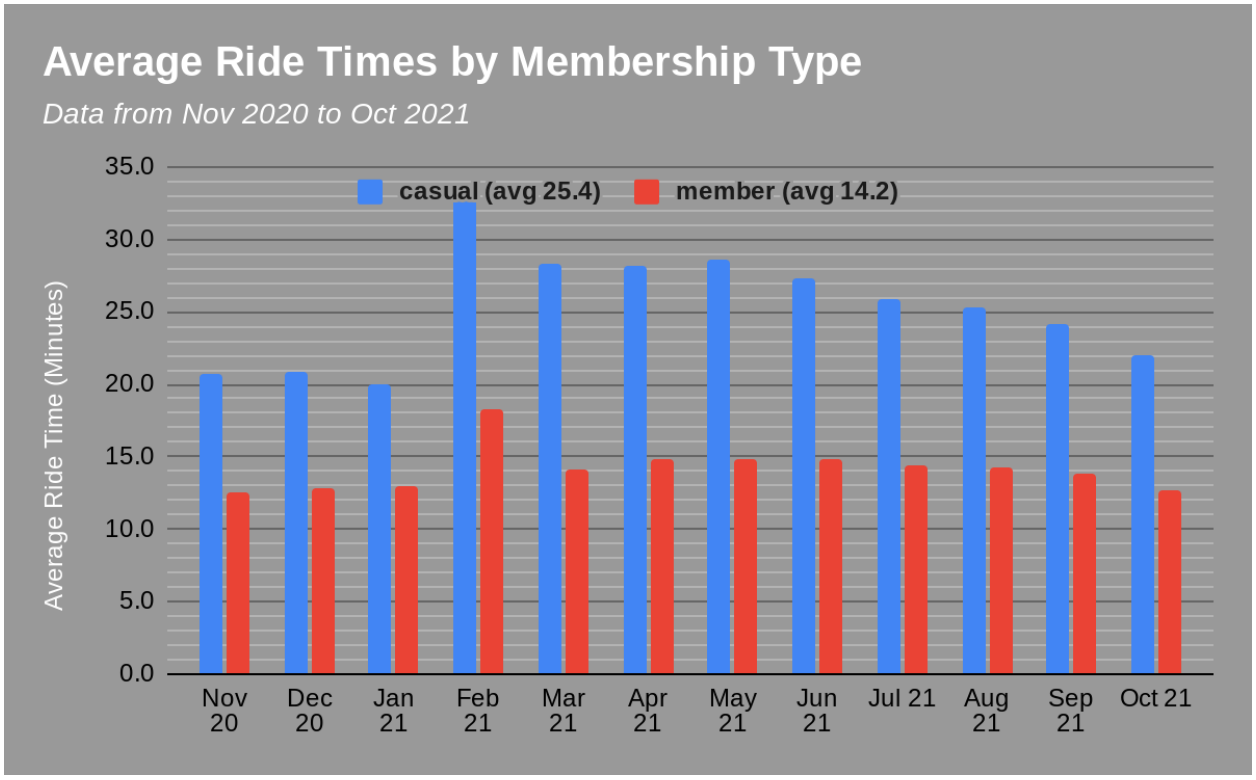
As reflected in the pie chart above, the data shows a nearly even split for the total number of casual riders (50.1 percent) and members (49.1 percent) on federal holidays.



With respect to the ridership for these groups on each individual holiday, members used Cyclistic bikes more than casual riders on every holiday except for three: Memorial Day, Independence Day, and Labor Day. Casual ridership exceeded that of members by an average of 5,957 riders on those three warm-weather holidays, and these holidays comprised 66 percent of all casual holiday ridership. The bar chart above shows a graphical comparison of the number of riders from both membership groups that rode on each holiday.

Average Ride Times

Cyclistic bike share members have an average ride time of 14.2 minutes. The ride average ride time for members is 11.2 minutes fewer than casual riders, who typically bike for 25.4 minutes. In addition, the average monthly ride time for members has less variation from month-to-month than the average for casual riders, ranging from a low of 12.5 minutes in November to a high of 18.3 minutes in February 2021. By comparison, the average ride times for casual riders went from a low of 20.0 minutes in January 2021 to a peak of 33.6 minutes the following month. There is seemingly a seasonal low in ride times for casual riders during the months of November, December, and January.



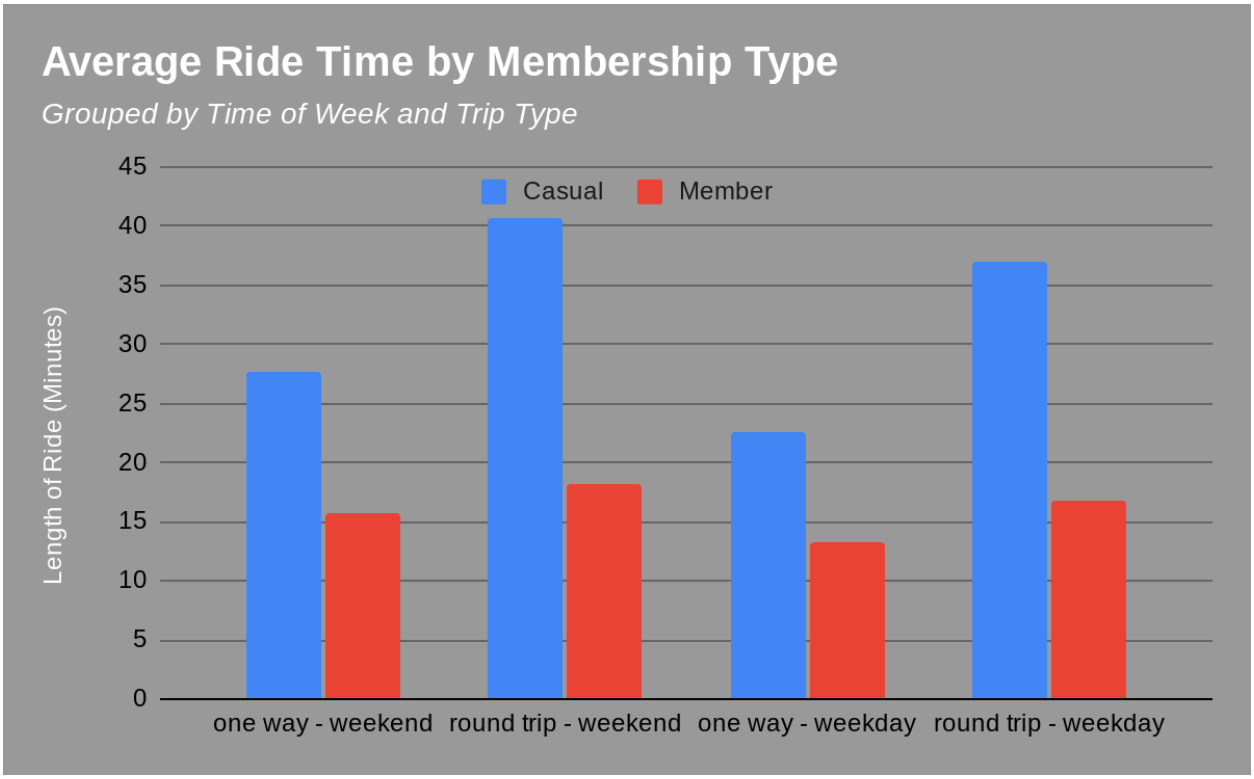
Interestingly, the highest ride times for both groups occurred in the same month, February 2021. More analysis would be needed to determine if this is a typical pattern or if there were unique circumstances (e.g. special events in the area, easing of COVID-19 restrictions that caused people to go out in unseasonably large numbers) during this month that explain the sharp increase from months prior.

Trip Destination

The data for bike rentals was investigated for any patterns based upon the destination of the trip. To conduct this analysis, information was gathered about the starting and ending

coordinates of each bike trip. Trips that started and ended at the same location were considered “Round Trips” and all others were classified as “One Way.”

The Cyclistic data revealed that a staggering majority of 95 percent of rides were “One Way” trips. Casual riders were responsible for 58 percent of the total rides (249,157) that were classified as “Round Trips.”



This data on trip destinations revealed several notable differences in average ride times when it was filtered by the time of the week (weekday vs. weekends).

- Ride times for “Round Trips” are longer than “One-way” trips on average for both casual riders and members.
- As was the case when reviewing average ride times by month, there is much less variation across the variables being considered for members.
- Average ride times for casual riders were more than double that of members during “Round Trips”
    - Weekend: 40.7 vs. 18.1 minutes for casual riders and members, respectively
    - Weekday: 37.1 vs. 16.8 minutes for casual riders vs. members

Ridership by Time of Day

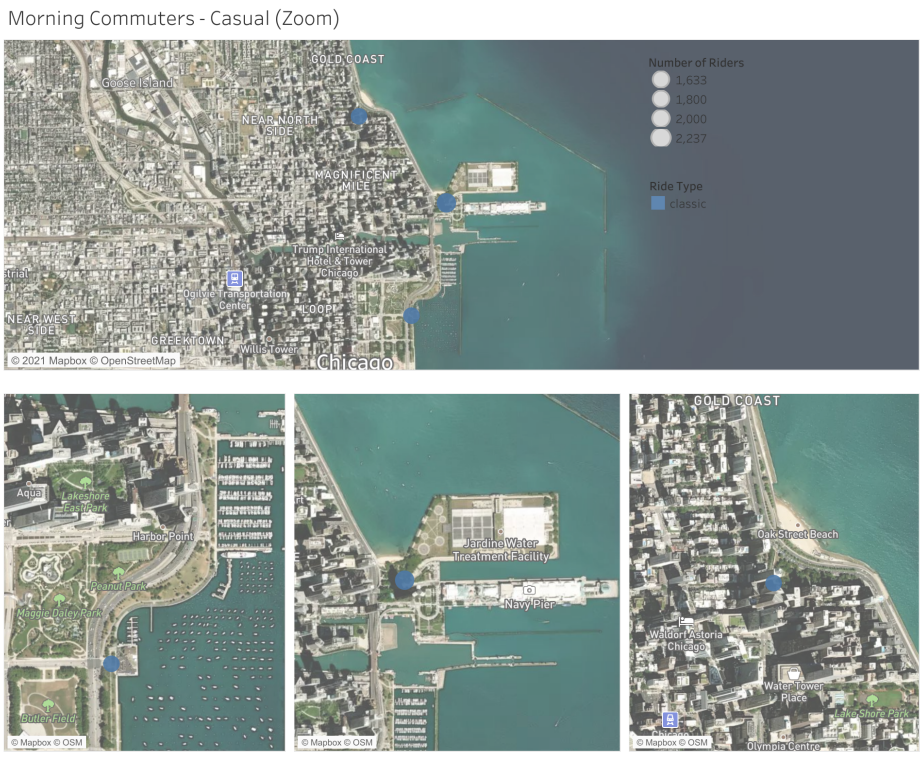
There are myriad reasons such as commuting to/from work, running errands, leisure riding, site seeing, and exercising that may drive someone to use Cyclistic’s bike share program. Although

the use of a bike during a specific window of time does not give detail on the reason for rental, comparing patterns in rental times between groups may offer insight into subsets of members and casual riders that are using the bikes in the same way. The hypothesis is that a large concentration of members and casual riders using the bikes around the same time at the same location(s) may have the same usage purpose.

The starting time for rentals of Cyclistic bikes was used to examine how time of day factored into the usage for members and casual riders. For the purposes of this analysis, the rental was placed into one of the following categories:

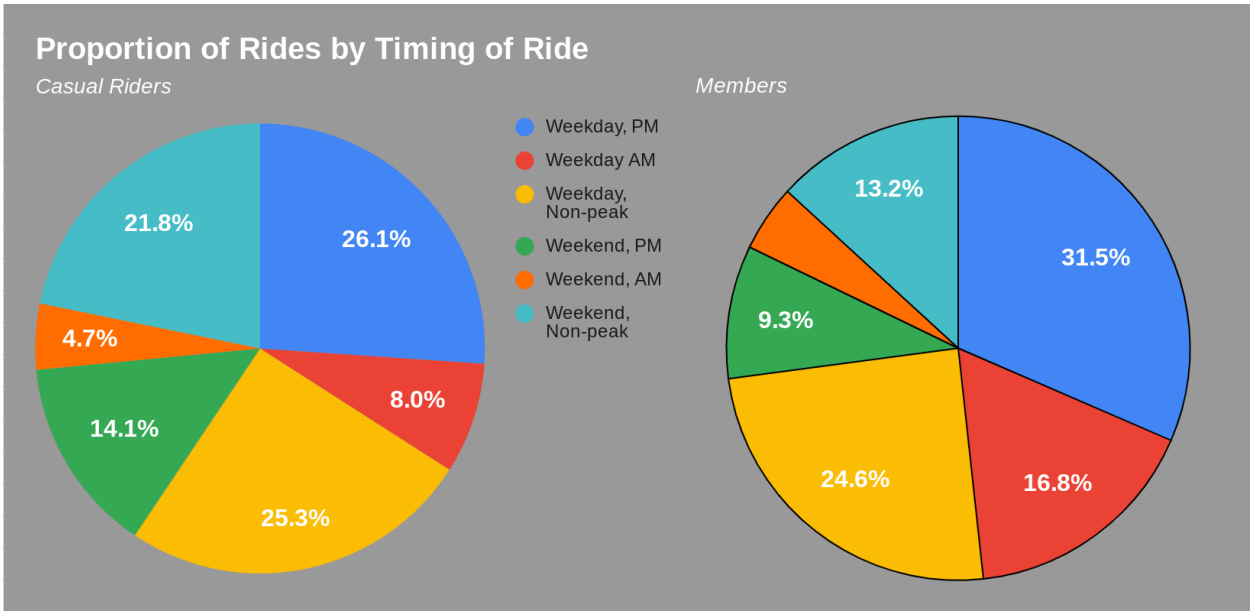
- Morning Commute: rentals that started between the hours of 6:00 a.m. and 10:00 a.m.
- Evening Commute: rentals that started between the hours of 3:00 p.m. and 7:00 p.m.
- Non-peak: rentals that did not occur during the commuting hours defined above

The latitude and longitude coordinates contained in each rental record allowed each rental location to be plotted on a map of the service area. In order to take a deeper dive into the geographic trends in ridership for casual riders and members, the mapping data was filtered by time of rental for each member group.



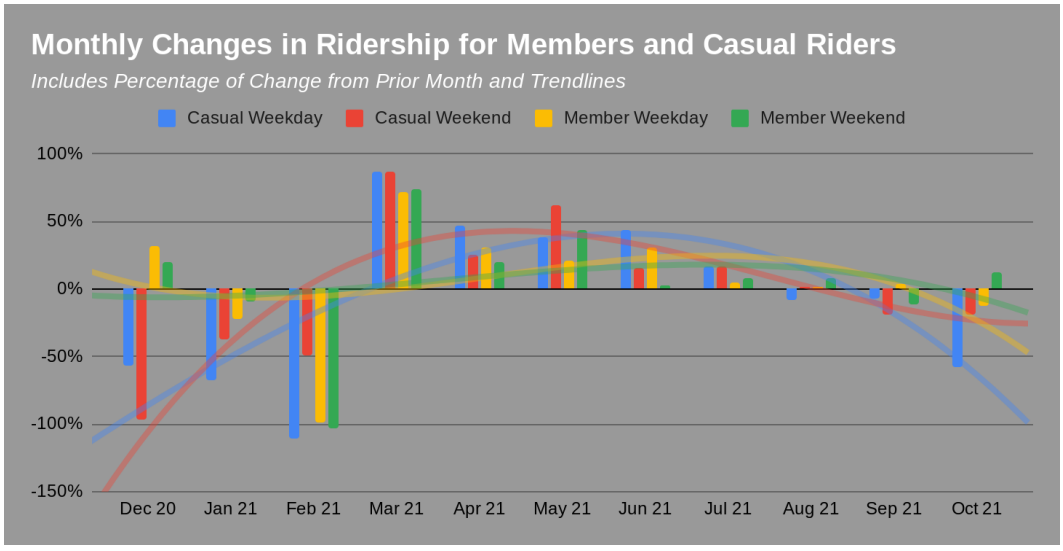
There were only 3 of the top 250 rental locations for casual riders that appeared during the morning commute time. These clustered around parks, beaches, the Navy Pier, waterfronts, and major hotels. It stands to reason that a significant number of these riders may not reside in the Cyclistic service area.

The pie charts below show the proportion of rides taken by casual riders and Cyclistiic members, broken down by the categories for time of day previously noted and by day (weekday vs. weekend). For both groups, the top two categories are represented by times during weekday evening commutes and weekday non-peak hours. These categories make up over half the ridership for both members and casual riders.



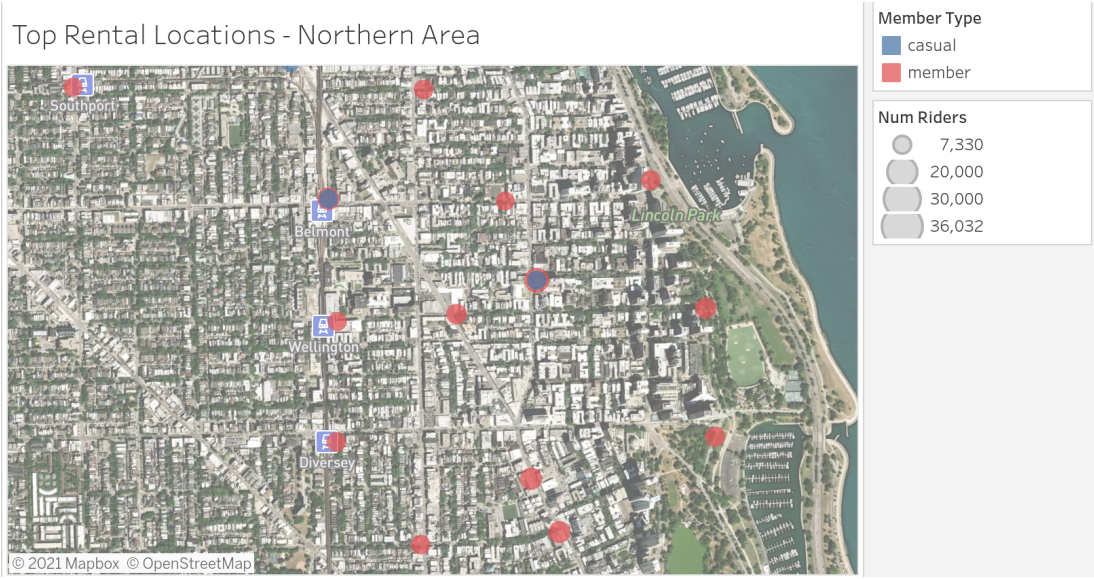
It has already been seen that casual riders use the bike share more on weekend days. This is supported by the third-largest category, weekend non-peak hours, for that rider group; this category comprises nearly 22 percent of casual riders. In contrast, the third-largest group for members is the morning commute segment.

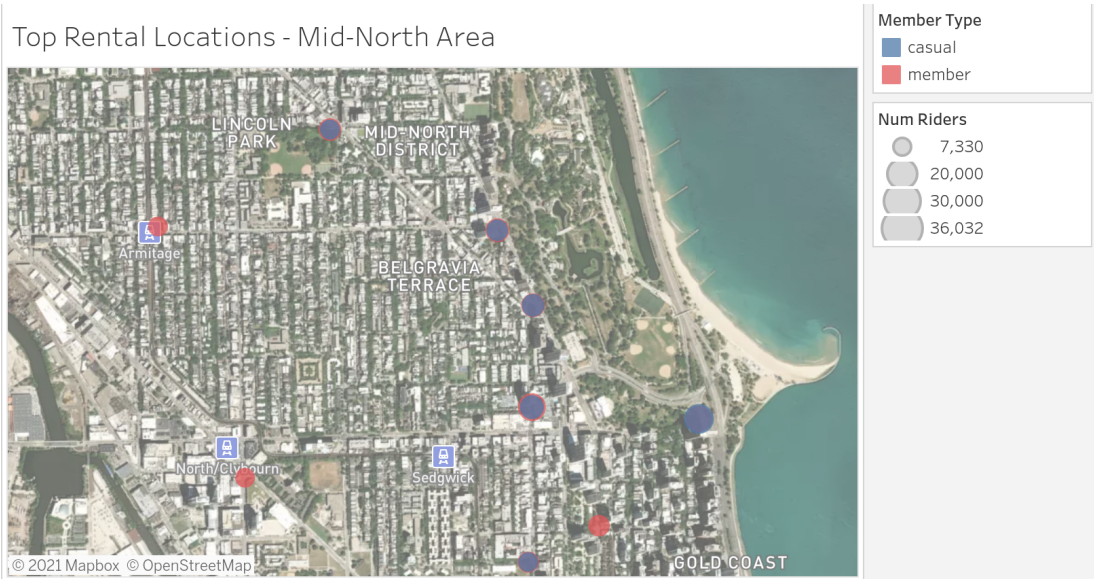
The chart to the right shows us the monthly changes in the number of riders for members and casual groups. The trendlines illustrate more seasonal variation in casual ridership, which increases more than that of members during the warmer months of the year relative to other months.

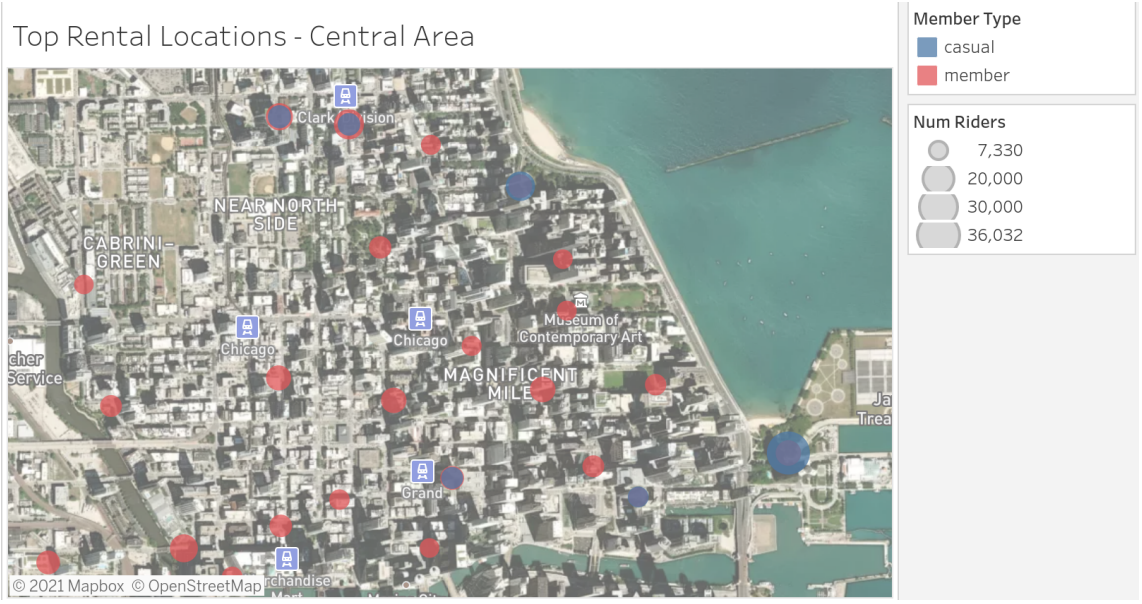


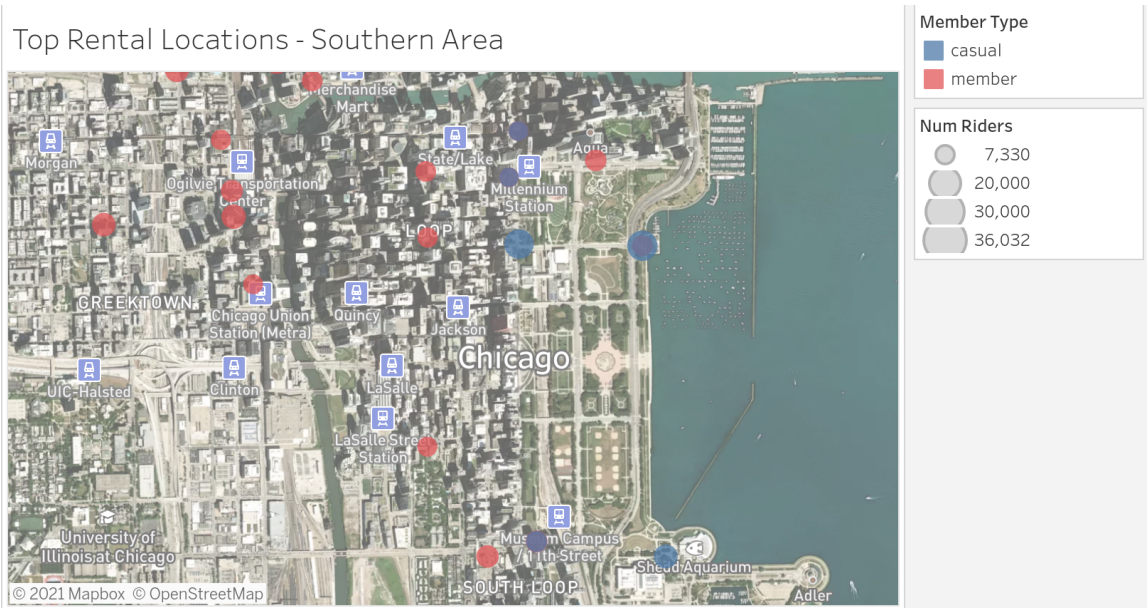
Rental Location

In order to look for patterns in the rental locations for members and casual riders, the geographical coordinates in the rider data were used to generate maps plotting the starting location of rides. The data was filtered to generate a list of the top 100 locations for rental to find hotspots throughout the service area. These data points were overlaid on satellite maps of the Cyclistic service area which included labels for neighborhoods, major points of interest, and public transportation stations. The following graphics show those maps, moving from north to south through the service area:







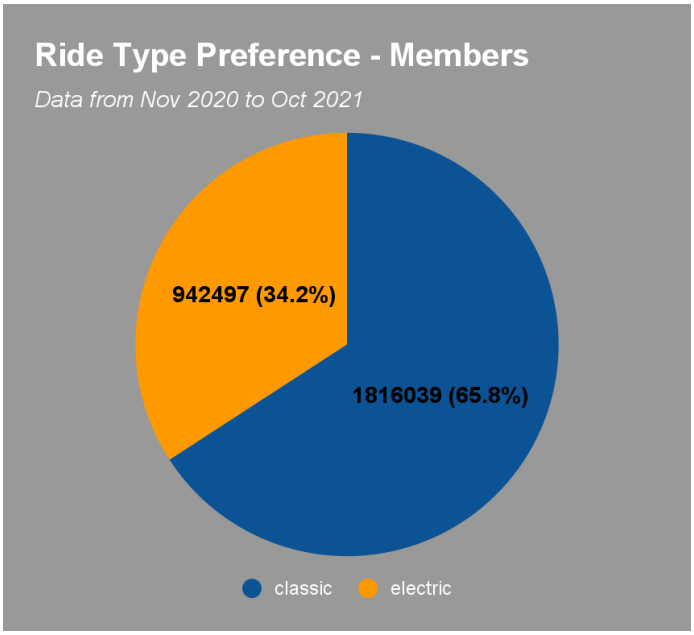


In each of the maps of rental top rental locations, there are hotspots for bike rentals by members located within close proximity to public transportation. There are also multiple rental locations (as shown in the north, mid-north, and central maps) that are significant rental locations for both membership groups. These locations are also ones that are near transportation stations.

These findings suggest that many Cyclistic riders may use the bike share as a means of intermediate transport to get from their locations to hubs of public transportation. This could be consistent with people who are commuting throughout the Chicago metropolitan area. When compared to walking, the bike share may help them with convenience and speed in their travels to and from transit stops.

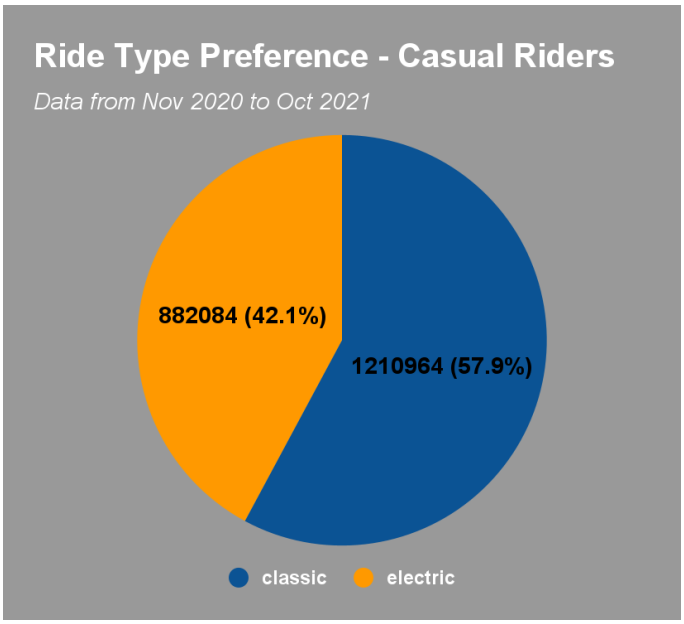
Bike Type

When viewing the aggregate of the data for Cyclisitc bike riders during the 12-month period from November 2020 to October 2021, it appears that riders have a preference for classic bikes over their electric counterparts.



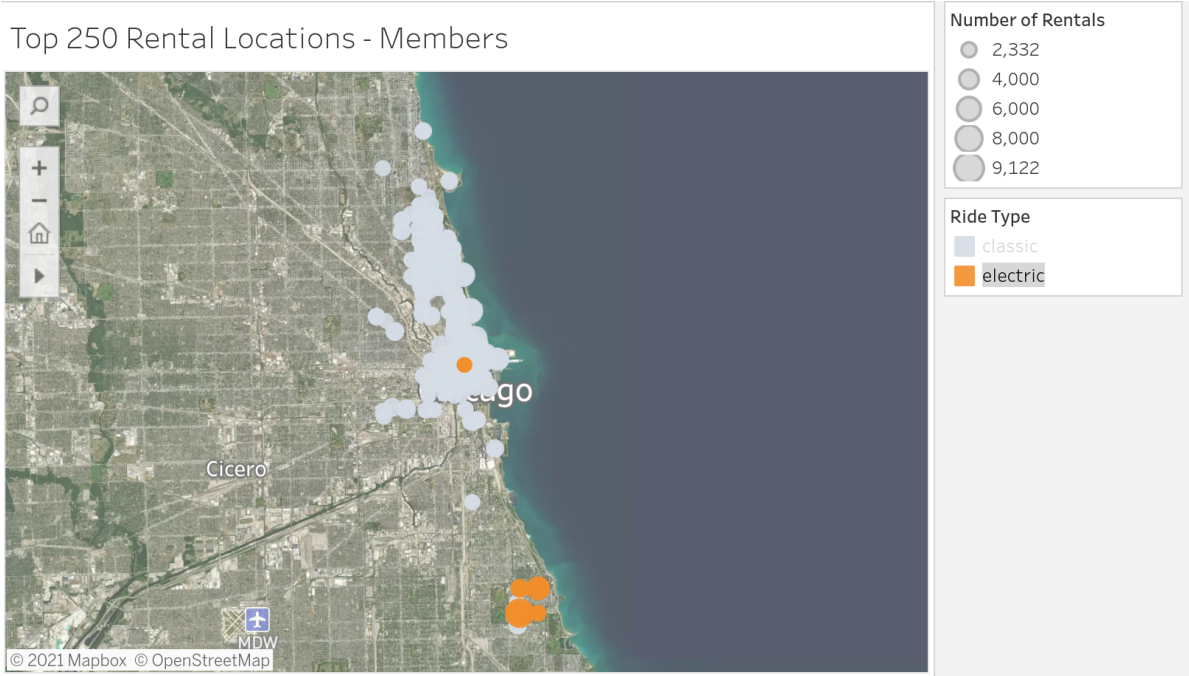
The data shows us that 66 percent of members and 58 percent of casual riders prefer classic bikes when compared to electric bikes.

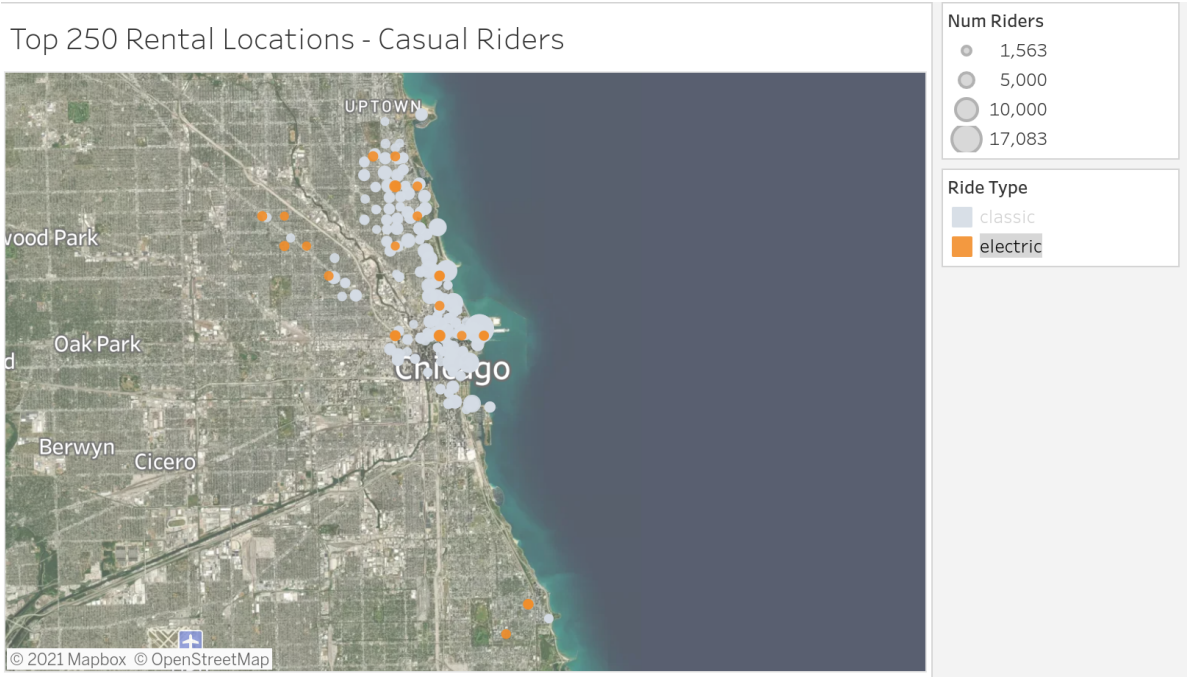
However, a deeper dive into the data revealed a greater disparity between the ride type preference than would be inferred by only considering the proportion of classic to electric bikes for the full dataset. The top 250 trip rental locations for each membership group were analyzed on the dimensions of rental type (electric vs. classic) and rental time (morning commute, evening commute, non-peak) to look for trends in bike type based upon when the ride took place. These subsets included 32 percent (874,855) of members and 33 percent (700,935) of casual riders.



A review of the data for the top rental locations for each membership group shows that an overwhelming majority of the riders frequenting those locations preferred classic bikes. In fact, only 25 of the top 250 rental locations for casual riders and 11 of the top 250 rental locations for

members pertain to riders who selected electric bikes. Of these top rental locations, only 7 percent (49,436) and 5 percent (46,205) of casual riders and members used electric bikes. The maps below show the geographic distribution of these throughout the Cyclistic service area. Note that the data does not provide information on whether the lack of prevalence for electric bikes is due to rider choice or bike availability in these locations.





Summary of Analysis

Based upon the analysis of the bike share usage data for members and casual riders, the following table presents observations that can be made about Cyclistic users:

	Casual Rider	**Member**	**Key Observations**
Number of Riders Total Ridership Daily Average	2,093,048 (**43%**) 299,077	2,758,536 (**57%**) 394,077	Casual riders use bikes in greater numbers on weekends (55% of group on Fri-Sun), whereas member ridership is evenly dispersed across each day
Average Ride Times Monthly Average Seasonality	25.4 minutes Ride times increase during warm-weather	14.2 minutes Ride times have less variation from month-to-month	Casual riders ride longer than members on average. The ride duration for members is consistent from month-to-month. Ride times for casual riders appear positively correlated with warm weather.
Holidays Number of Riders	72,407 (**50.9%**)	69,755 (**49.1%**)	Casual ridership was significantly higher on 3 holidays: Memorial Day, Independence Day, and Labor Day. Members lead on all other holidays.
Trip Destination One Way vs. Roundtrip	Round trip times exceed one-way trip times	Round trip times exceed one-way trip times	Casual riders took 58% of all roundtrips, and those trips were over twice as long as those of members. 95% of all trips are One Way
Time of Day Morning Commute	Top 3 categories: 1 - PM Commute, weekday	Top 3 categories: 1 - PM Commute, weekday	Casual riders and members have a similar distribution

Evening Commute Non-peak Times	2 - Non-peak, weekday 3 - Non-peak, weekend	2 - Non-peak, weekday 3 - AM Commute, weekday	during PM commute and non-peak times on weekdays.
Rental Location	Several hotspots in proximity to public transport	Significant number of hotspots in close proximity to public transport	Positive correlation between top rental locations and proximity to public transit
Bike Type Classic vs. Electric	Classic bikes are preferred by 58% of riders	Classic bikes are preferred by 66% of riders	Classic bikes were overwhelmingly used by riders of both groups at the top rental locations. Electric bike preference does not seem to be a factor.

Recommendations

1. Focus on commuters

Several observations about the data findings points towards member usage of Cyclistic bikes as an intermediary means of getting to public transportation hubs during commute:

- High volume rental locations are in close proximity to transit stations
- Overwhelming majority of rides taken as “one way”
- Average ride times are comparatively short
- Large percentage of rides occur during morning and evening commute times
- Even distribution of tides throughout the week

There is enough similarity between members and casual riders in several of these dimensions (rental location proximal to transit, trip taken as one-way, large chunk of rentals occur during evening commuting hours) to suggest that many casual riders use the bike share in the same way as members, but for whatever reason have not opted to become members.

One element of the marketing campaign should focus on how an annual membership can benefit commuters to encourage casual riders of this segment to purchase an annual pass.

2. Target casual riders in warm-weather months riding on weekends and holidays

The analysis of the ridership data identified seasonality in casual ridership, with the number of riders increasing as temperatures warm in spring, summer, and early fall months. Casual riders also turn out in large numbers during the holidays that fall during these months. Additionally, casual ridership is more heavily weighted to weekend days, with the majority of these riders renting bikes during Fridays, Saturdays, and Sundays.

In order to capitalize on the time periods when causal riders are most utilizing the Cyclistic bike share, the marketing campaign will get the most return on investment by focusing efforts to reach casual riders renting during these busy times.

3. Survey riders to learn more about their habits and preferences

In order to refine marketing efforts, a survey of Cyclistic riders should be considered to learn more about their habits and preferences. There are several important factors that could inform the marketing strategy that could not be determined from the existing data. A survey should be designed to gather information on the following questions:

- Where do riders live / work in relation to the Cyclisitic service area?
- Have they ever used the bike share before? If so, how often?
- How do they use the bike share (commuting, leisure, exercise, etc)?
- What factors made them decide to purchase the selected type of pass?

Appendix 1 - Data Cleaning Changelog

Date	Time	File Name	Description of Action	Change Made	Reason for Change	Changed By
12-03-21	14:17	202110_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 45 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-03-21	14:38	202110_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	14:42	202110_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-03-21	14:43	202110_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	14:46	202110_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-03-21	14:48	202110_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.6, max value 42.13	Drew Brinkley
12-03-21	14:50	202110_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.96, max value -87.52	Drew Brinkley
12-03-21	14:56	202110_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-03-21	14:58	202110_bike_data_cleaned.xlsx	calculate MIN of ride_length	none	no negative values (e.g. ride ended prior to start)	Drew Brinkley
12-03-21	15:15	202110_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-03-21	15:26	202109_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 75 rows with duplicate ride_id values	duplicate data	Drew Brinkley

12-03-21	15:40	202109_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	15:40	202109_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-03-21	15:41	202109_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	15:41	202109_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-03-21	15:42	202109_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.57, max value 42.16812	Drew Brinkley
12-03-21	15:43	202109_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.87, max value -87.50	Drew Brinkley
12-03-21	15:46	202109_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-03-21	15:50	202109_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 37 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-03-21	15:56	202109_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-03-21	16:29	202108_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 104 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-03-21	16:32	202108_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	16:32	202108_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley

12-03-21	16:35	202108_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	16:35	202108_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-03-21	16:36	202108_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.58, max value 42.15	Drew Brinkley
12-03-21	16:36	202108_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.85, max value -87.51	Drew Brinkley
12-03-21	16:37	202108_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-03-21	16:39	202108_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 29 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-03-21	16:42	202108_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-03-21	16:54	202107_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 75 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-03-21	16:54	202107_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	16:54	202107_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-03-21	16:55	202107_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	16:55	202107_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley

12-03-21	16:55	202107_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.63, max value 42.15	Drew Brinkley
12-03-21	16:56	202107_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.85, max value -87.49	Drew Brinkley
12-03-21	16:59	202107_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-03-21	17:00	202107_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 13 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-03-21	17:00	202107_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-03-21	17:28	202106_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 70 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-03-21	17:28	202106_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	17:28	202106_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-03-21	17:28	202106_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	17:29	202106_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-03-21	17:29	202106_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.51, max value 42.08	Drew Brinkley
12-03-21	17:29	202106_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.86, max value -87.49	Drew Brinkley

12-03-21	17:31	202106_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-03-21	17:32	202106_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 5 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-03-21	17:32	202106_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-03-21	17:42	202105_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 56 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-03-21	17:42	202105_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	17:43	202105_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-03-21	17:43	202105_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-03-21	17:43	202105_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-03-21	17:43	202105_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.56, max value 42.09	Drew Brinkley
12-03-21	17:44	202105_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.85, max value -87.52	Drew Brinkley
12-03-21	17:48	202105_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-03-21	17:48	202105_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 2 rows of data where	data was incorrect - ride cannot end before it starts	Drew Brinkley

				ride_length was negative		
12-03-21	17:48	202105_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-04-21	10:13	202104_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 11965 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-04-21	10:16	202104_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	10:16	202104_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-04-21	10:16	202104_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	10:16	202104_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-04-21	10:17	202104_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.59, max value 42.15	Drew Brinkley
12-04-21	10:17	202104_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.85, max value -87.52	Drew Brinkley
12-04-21	10:18	202104_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-04-21	10:18	202104_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 1 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-04-21	10:19	202104_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley

12-04-21	10:25	202103_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 7550 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-04-21	10:44	202103_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	10:44	202103_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-04-21	10:44	202103_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	10:44	202103_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-04-21	10:45	202103_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.64, max value 42.08	Drew Brinkley
12-04-21	10:45	202103_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.5282, max value -88.07	Drew Brinkley
12-04-21	10:46	202103_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-04-21	10:46	202103_bike_data_cleaned.xlsx	calculate MIN of ride_length	none	no negative values (e.g. ride ended prior to start)	Drew Brinkley
12-04-21	10:47	202103_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-04-21	10:55	202102_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 667 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-04-21	10:56	202102_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley

12-04-21	10:56	202102_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-04-21	10:56	202102_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	10:56	202102_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-04-21	10:56	202102_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.54, max value 42.07	Drew Brinkley
12-04-21	10:57	202102_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.5348, max value -87.7747	Drew Brinkley
12-04-21	10:58	202102_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-04-21	10:58	202102_bike_data_cleaned.xlsx	calculate MIN of ride_length	none	no negative values (e.g. ride ended prior to start)	Drew Brinkley
12-04-21	10:59	202102_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-04-21	11:03	202101_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 1070 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-04-21	11:03	202101_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	11:04	202101_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-04-21	11:04	202101_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley

12-04-21	11:04	202101_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-04-21	11:04	202101_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.64, max value 42.07	Drew Brinkley
12-04-21	11:04	202101_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.81, max value -87.51	Drew Brinkley
12-04-21	11:06	202101_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-04-21	11:07	202101_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 2 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-04-21	11:07	202101_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-04-21	11:12	202012_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 1841 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-04-21	11:13	202012_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	11:13	202012_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-04-21	11:13	202012_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	11:13	202012_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-04-21	11:13	202012_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.65, max value 42.07	Drew Brinkley

12-04-21	11:14	202012_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.79, max value -87.5282	Drew Brinkley
12-04-21	11:16	202012_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley
12-04-21	11:16	202012_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 399 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-04-21	11:17	202012_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-04-21	11:22	202011_bike_data_cleaned.xlsx	perform Remove Duplicates	removed 5786 rows with duplicate ride_id values	duplicate data	Drew Brinkley
12-04-21	11:23	202011_bike_data_cleaned.xlsx	calculate MIN of started_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	11:23	202011_bike_data_cleaned.xlsx	calculate MAX of started_at column to check for outliers	none	no values prior to end of current month	Drew Brinkley
12-04-21	11:23	202011_bike_data_cleaned.xlsx	calculate MIN of ended_at column to check for outliers	none	no values prior to start of current month	Drew Brinkley
12-04-21	11:23	202011_bike_data_cleaned.xlsx	calculate MAX of ended_at column to check for outliers	none	max value was a few days into following month	Drew Brinkley
12-04-21	11:23	202011_bike_data_cleaned.xlsx	calculate MIN and MAX of latitude columns to check for outliers	none	min value 41.54, max value 42.15	Drew Brinkley
12-04-21	11:23	202011_bike_data_cleaned.xlsx	calculate MIN and MAX of longitude columns to check for outliers	none	min value -87.87, max value -87.44	Drew Brinkley
12-04-21	11:25	202011_bike_data_cleaned.xlsx	calculate ride length	new column ride_length = ended_at - started_at	allow analysis on ride time, check for data errors	Drew Brinkley

12-04-21	11:25	202011_bike_data_cleaned.xlsx	calculate MIN of ride_length	removed 299 rows of data where ride_length was negative	data was incorrect - ride cannot end before it starts	Drew Brinkley
12-04-21	11:26	202011_bike_data_cleaned.xlsx	identify day bike was rented	new column day_of_week = WEEKDAY(started_at)	allow analysis on day of rental	Drew Brinkley
12-04-21	14:05	202011_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:08	202011_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	14:15	202012_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:15	202012_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	14:25	202101_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:25	202101_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley

12-04-21	14:28	202102_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:28	202102_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	14:32	202103_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:32	202103_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	14:38	202104_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:38	202104_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	14:44	202105_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:44	202105_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley

12-04-21	14:50	202106_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:50	202106_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	14:58	202107_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	14:58	202107_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	15:06	202108_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	15:06	202108_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-04-21	15:49	202109_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	15:49	202109_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley

12-04-21	16:00	202110_bike_data_cleaned_v1.xlsx	remove unneeded columns	delete columns pertaining to start and end location (lat and long will be used for location)	reduce file size for .csv export to other data tools	Drew Brinkley
12-04-21	16:00	202110_bike_data_cleaned_v1.xlsx	identify day of rental in text	new column day_text to convert number assigned from WEEKDAY to text	allow analysis on day of rental	Drew Brinkley
12-05-21	7:54		upload data to BigQuery	created cyclistic_trip_data dataset; uploaded prior 12 months of cleaned spreadsheet data via .CSV files	compile monthly data into one dataset for analysis; this set will be too large for analysis via spreadsheet	Drew Brinkley

Appendix 2 - SQL Queries

DATA CLEANING

/* UNION ALL statement to combine data from cyclistic_trip_data monthly tables of prior 12 months into one table of complete data. Results of query were saved as new table, cyclist_trip_data.bike_data_combined */

```
SELECT
	*
FROM
	cyclistic_trip_data.nov_2020_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.dec_2020_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.jan_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.feb_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.mar_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.apr_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.may_2021_data
UNION ALL
```

```
SELECT
	*
FROM
	cyclistic_trip_data.jun_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.jul_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.aug_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.sep_2021_data
UNION ALL
SELECT
	*
FROM
	cyclistic_trip_data.oct_2021_data
ORDER BY
started_at;


/* Query to determine total number of riders in the cyclistic_trip_data.bike_data_combined table
*/

SELECT
	COUNT(bike.ride_id) AS num_rides
FROM
	`cyclistic_trip_data.bike_data_combined` AS bike;
```

Row	num_rides
1	5348819

/* Query to determine total number of riders in the cyclistic_trip_data.bike_data_combined table, grouped by membership type */

```
SELECT
	COUNT(bike.ride_id) AS num_rides,
	bike.member_casual AS member_type
FROM
	`cyclistic_trip_data.bike_data_combined` AS bike
GROUP BY
	member_type;
```

Row	num_rides	member_type
1	2451580	casual
2	2897239	member

/* Query to determine number of rides in “docked” status, grouped by membership type */

```
SELECT
	COUNT(bike.ride_id) AS num_rides,
	bike.member_casual AS member_type
FROM
	`cyclistic_trip_data.bike_data_combined` AS bike
WHERE
	bike.rideable_type = “docked”
GROUP BY
	member_type;
```

Row	num_rides	member_type
1	110754	member
2	340315	casual

/* Query to determine number of rides with trip duration less than one minute, grouped by membership type */

```
SELECT
	COUNT(bike.ride_id) AS num_rides,
	bike.member_casual as member_type
FROM
```

```
	`cyclistic_trip_data.bike_data_combined` AS bike
WHERE
	TIMESTAMP_DIFF(bike.ended_at, bike.started_at, MINUTE) < 1
GROUP BY
	member_type;
```

Row	num_rides	member_type
1	19735	casual
2	29453	member

```
/* Query to select data from table cyclistic_trip_data.bike_data_combined where trip durations of less than one minute and ride types of “docked” will be excluded.  Results of query will be saved in a new table, cyclistic_trip_data.bike_data_clean and exported as .CSV file */

SELECT
	*
FROM
	`cyclistic_trip_data.bike_data_combined` AS bike
WHERE
	bike.rideable_type NOT IN (“docked“)
	AND
	TIMESTAMP_DIFF(bike.ended_at, bike.started_at, MINUTE) >= 1
ORDER BY
	bike.started_at;


/* Query to verify minimum trip duration in minutes of cleaned bike share data */

SELECT
	MIN(TIMESTAMP_DIFF(bike.ended_at, bike.started_at, MINUTE)) AS min_duration_clean
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike;
```

Row	min_duration_clean
1	1

```
/* Query to verify the exclusion of “docked” rides from cleaned bike share data */

SELECT
        COUNT(bike.ride_id) as num_rides,
        bike.member_casual AS member_type
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        member_type;
```

Row	num_rides	member_type
1	2758536	member
2	2093048	casual

DATA ANALYSIS

```
 /* Calculate total number of riders per day, by membership type */

SELECT
        COUNT(bike.ride_id) AS total_num_riders,
        bike.member_casual AS member_type,
        bike.day_text as day
FROM
        `cyclistic_trip_data.bike_data_clean` as bike
GROUP BY
        day,
        member_type
ORDER BY
        member_type;


/* Calculate total number of riders per month, by membership type */

SELECT
        FORMAT_DATETIME("%b %y", bike.started_at) AS rental_month,
        COUNT(bike.ride_id) AS num_riders,
```

```
        bike.member_casual AS member_type
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        rental_month,
        member_type
ORDER BY
        member_type;


/* Calculate number of riders per bike type, by membership type */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        bike.rideable_type AS bike_type,
        bike.member_casual AS member_type
FROM
        `cyclistic_trip_data.bike_data_clean` as bike
GROUP BY
        bike_type,
        member_type;


/* Calculate average ride duration (in minutes) per month, by membership type, and ranked in
descending order */

SELECT
        FORMAT_DATETIME("%b %y", bike.started_at) AS rental_month,
        ROUND(AVG(TIMESTAMP_DIFF(bike.ended_at, bike.started_at, MINUTE)),2) AS
ride_time_minutes,
        bike.member_casual AS member_type
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        rental_month,
        member_type
ORDER BY
        ride_time_minutes DESC;


/* Calculate average ride duration (minutes) per month, by membership type and ride type, and
ranked in descending order */

SELECT
```

```
        FORMAT_DATETIME("%b %y", bike.started_at) AS rental_month,
        ROUND(AVG(TIMESTAMP_DIFF(bike.ended_at, bike.started_at, MINUTE)),2) AS
ride_time_minutes,
        bike.member_casual AS member_type,
        bike.rideable_type AS ride_type
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        rental_month,
        member_type,
        ride_type
ORDER BY
        ride_time_minutes DESC;


/* Calculate number of riders, by membership type and day (weekend or weekday), during peak
vs non-peak commute times */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        bike.member_casual AS member_type,
        CASE
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
                THEN 'morning commute'
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
                THEN 'evening commute'
                ELSE 'non-peak'
        END AS time_of_day,
        CASE
                WHEN bike.day_text NOT IN (“Sat”, “Sun”)
                THEN ‘weekday’
                ELSE ‘weekend’
        END AS day
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        member_type,
        time_of_day,
        day;
```

```
/* Calculate number of riders per month, by membership type and day (weekend or weekday), during peak vs non-peak commute times */

SELECT
	COUNT(bike.ride_id) AS num_riders,
	FORMAT_DATETIME("%b %y", bike.started_at) AS rental_month,
	bike.member_casual AS member_type,
	CASE
		WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
		THEN 'morning commute'
		WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
		THEN 'evening commute'
		ELSE 'non-peak'
	END AS time_of_day,
	CASE
		WHEN bike.day_text NOT IN (“Sat”, “Sun”)
		THEN ‘weekday’
		ELSE ‘weekend’
	END AS day
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
	rental_month,
	member_type,
	time_of_day,
	day;


/* Calculate number of riders per month;  by membership type, ride type and day (weekday or weekend); during peak vs non-peak commute times */

SELECT
	COUNT(bike.ride_id) AS num_riders,
	FORMAT_DATETIME("%b %y", bike.started_at) AS rental_month,
	bike.member_casual AS member_type,
	bike.rideable_type AS ride_type,
	CASE
```

```
            WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
            THEN 'morning commute'
            WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
            THEN 'evening commute'
            ELSE 'non-peak'
        END AS time_of_day,
        CASE
            WHEN bike.day_text NOT IN (“Sat”, “Sun”)
            THEN ‘weekday’
            ELSE ‘weekend’
        END AS day
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        rental_month,
        member_type,
        ride_type,
        time_of_day,
        day;

/* Calculate average ride time per month, by membership and ride type and day (weekday or
weekend), during peak vs non-peak commute times */

SELECT
        ROUND(AVG(TIMESTAMP_DIFF(bike.ended_at,bike.started_at,MINUTE)),2) AS
avg_ride_time,
        FORMAT_DATETIME("%b %y", bike.started_at) AS rental_month,
        bike.member_casual AS member_type,
        bike.rideable_type AS ride_type,
        CASE
            WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
            THEN 'morning commute'
            WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
            THEN 'evening commute'
            ELSE 'non-peak'
        END AS time_of_day,
        CASE
            WHEN bike.day_text NOT IN ("Sat", "Sun")
            THEN 'weekday'
            ELSE 'weekend'
        END AS day
```

```
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
	rental_month,
	member_type,
	ride_type,
	time_of_day,
	day;


/* Rank the top 100 locations for rentals, grouped by membership type.  Latitude and longitude
coordinates were combined into single column to facilitate grouping and exportation */

SELECT
	COUNT(bike.ride_id) AS num_riders,
	bike.member_casual AS member_type,
	CONCAT(bike.start_lat, ",", bike.start_lng) AS start_location
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
	member_type,
	start_location
ORDER BY
	num_riders DESC
LIMIT
	100;


/* Rank the top 100 locations for rental returns, grouped by membership type.  Latitude and
longitude coordinates were combined into single column to facilitate grouping and exportation */

SELECT
	COUNT(bike.ride_id) AS num_riders,
	bike.member_casual AS member_type,
	CONCAT(bike.end_lat, ",", bike.end_lng) AS end_location
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
	member_type,
	end_location
ORDER BY
	num_riders DESC
LIMIT
	100;
```

```
/* Rank top 50 locations for rentals, grouped by commute type.  Latitude and longitude
coordinates were combined into single column to facilitate grouping and exportation */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        CASE
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
                THEN 'morning commute'
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
                THEN 'evening commute'
                ELSE 'non-peak'
        END AS time_of_day,
        CONCAT(bike.start_lat, ",", bike.start_lng) AS start_location
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        time_of_day,
        start_location
ORDER BY
        num_riders DESC
LIMIT
        50;


/* Rank top 50 locations for rentals, grouped by rental hour.  Latitude and longitude coordinates
were combined into single column to facilitate grouping and exportation */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        EXTRACT(HOUR FROM bike.started_at) AS hour_of_rental,
        CONCAT(bike.start_lat, ",", bike.start_lng) AS start_location
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        hour_of_rental,
        start_location
ORDER BY
        num_riders DESC
LIMIT
        50;
```

```
/* Rank top 250 locations for rentals, grouped by commute type and ride type (casual riders).
Latitude and longitude coordinates were combined into single column to facilitate grouping and
exportation */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        CASE
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
                THEN 'morning commute'
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
                THEN 'evening commute'
                ELSE 'non-peak'
        END AS time_of_day,
        bike.rideable_type AS ride_type,
        CONCAT(bike.start_lat, “,“, bike.start_lng) AS start_location
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
WHERE
        bike.member_casual = “casual”
GROUP BY
        time_of_day,
        ride_type,
        start_location
ORDER BY
        num_riders DESC
LIMIT
        250


/* Rank top 250 locations for rentals, grouped by commute type and ride type (members).
Latitude and longitude coordinates were combined into single column to facilitate grouping and
exportation */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        CASE
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 6 AND 10
                THEN 'morning commute'
                WHEN EXTRACT(HOUR FROM bike.started_at) BETWEEN 15 and 19
                THEN 'evening commute'
                ELSE 'non-peak'
        END AS time_of_day,
        bike.rideable_type AS ride_type,
        CONCAT(bike.start_lat, “,“, bike.start_lng) AS start_location
```

```
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike
WHERE
	bike.member_casual = “member”
GROUP BY
	time_of_day,
	ride_type,
	start_location
ORDER BY
	num_riders DESC
LIMIT
	250

/* Calculate number of riders, by membership type, time of week, and location of return */

SELECT
	COUNT(bike.ride_id) AS num_riders,
	bike.member_casual AS member_type,
	CASE
		WHEN bike.start_lat = bike.end_lat AND
		bike.start_lng = bike.end_lng
		THEN 'round trip'
		ELSE 'one way'
	END AS trip_type,
	CASE
		WHEN bike.day_text NOT IN (“Sat”, “Sun”)
		THEN ‘weekday’
		ELSE ‘weekend’
	END AS day
FROM
	`cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
	member_type,
	trip_type,
	day;


/* Calculate average ride time, by membership type, time of week, and location of return */

SELECT
```

```
        COUNT(bike.ride_id) AS num_riders,
        ROUND(AVG(TIMESTAMP_DIFF(bike.ended_at, bike.started_at, MINUTE)),2) AS
        ride_time_minutes,
        bike.member_casual AS member_type,
        CASE
                WHEN bike.start_lat = bike.end_lat AND
                bike.start_lng = bike.end_lng
                THEN 'round trip'
                ELSE 'one way'
        END AS trip_type,
        CASE
                WHEN bike.day_text NOT IN ("Sat", "Sun")
                THEN 'weekday'
                ELSE 'weekend'
        END AS day
FROM
        `cyclistic_trip_data.bike_data_clean` AS bike
GROUP BY
        member_type,
        trip_type,
        day;


/* Calculate holiday ridership by member type.  Bike data was joined with table containing
federal holidays during the time period of study to select the rider records */

SELECT
        COUNT(bike.ride_id) AS num_riders,
        holiday.holiday_date,
        holiday.holiday_name,
        bike.member_casual AS member_type
FROM
        `cyclistic_trip_data.federal_holidays` AS holiday
INNER JOIN
        `cyclistic_trip_data.bike_data_combined` AS bike
ON FORMAT_DATETIME("%b-%d-%Y", bike.started_at) = holiday.holiday_date
GROUP BY
        holiday_date,
        member_type,
        holiday_name
ORDER BY
        holiday_date;
```

Appendix 3 - Data Summary Changelog

Date	Time	File Name	Description of Action	Change Made	Reason for Change	Changed By
12/7/21	12:36	top_50_rental_locations_by_member_type_v1	split start_location	execute SPLIT() function to separate data into start_lat and start_long columns	format for mapping by coordinates	Drew Brinkley
12/7/21	12:47	top_50_rental_locations_by_commute_time_v1	split start_location	execute SPLIT() function to separate data into start_lat and start_long columns	format for mapping by coordinates	Drew Brinkley
12/8/21	7:52	daily_rides_by_member_type_v1	group data by membership type	insert rows between "casual" and "member" data, change column A heading to reflect member type	format data into different data series for chart	Drew Brinkley
12/8/21	8:01	monthly_rides_by_member_type_v1	group data by membership type	insert rows between "casual" and "member" data, change column B heading to reflect member type	format data into different data series for chart	Drew Brinkley
12/8/21	8:02	monthly_rides_by_member_type_v1	sort data by month	sort each data series by ascending month	order data for chart	Drew Brinkley
12/8/21	8:19	ride_types_by member_type_v1	group data by membership type	sort data by member type	format data into different data series for chart	Drew Brinkley
12/8/21	8:23	ride_types_by member_type_v1	group data by membership type	insert rows between "casual" and "member" data, change column A heading to reflect member type	format data into different data series for chart	Drew Brinkley
12/8/21	8:32	avg_monthly_ride_time_by_member_type_v1	group data by membership type	insert rows between "casual" and "member" data, change column B	format data into different data series for chart	Drew Brinkley

				heading to reflect member type		
12/8/21	8:33	avg_monthly_ride_time_by_member_type_v1	sort data by month	sort each data series by ascending month	order data for chart	Drew Brinkley
12/8/21	9:18	monthly_commuters_by_day_and_member_type_v1	summarize data by weekday vs weekend	create Pivot Table charts	analyze trends in ridership by member type	Drew Brinkley
12/8/21	9:29	monthly_commuters_by_day_and_member_type_v2	calculate percentages of each group	use CONCAT() function to combine columns for ease of labeling; calculate count of riders as percentage	format data for chart	Drew Brinkley
12/8/21	10:29	monthly_commuters_by_member_and_ride_type_and_day_v2	group data by membership type	sort data by member type	format data into different data series for chart	Drew Brinkley
12/8/21	10:33	monthly_commuters_by_member_and_ride_type_and_day_v2	format month	format cell values to show Mon YY. change year values for Nov and Dec from 2021 to 2020	format data for chart	Drew Brinkley
12/8/21	10:58	monthly_commuters_by_member_and_ride_type_and_day_v2	combine day and time columns	replaced cell values for time to shorten name, used CONCAT() function to combine day and time columns	format data for chart	Drew Brinkley
12/8/21	10:59	monthly_commuters_by_member_and_ride_type_and_day_v2	summarize data	create Pivot Table charts	analyze trends in ridership	Drew Brinkley
12/8/21	11:30	monthly_commuters_by_member_and_ride_type_and_day_v2	summarize data	add column ridership_percentage to calculate % ridership for each row	analyze trends in ridership	Drew Brinkley
12/8/21	11:42	monthly_commuters_by_day_and_member_type_v2	summarize data	add column ridership_percentage to calculate % ridership for each row	analyze trends in ridership	Drew Brinkley

12/8/21	11:43	monthly_commuters_by_day_and_member_type_v2	combine day and time columns	replaced cell values for time to shorten name, used CONCAT() function to combine day and time columns	format data for chart	Drew Brinkley
12/9/21	6:56	top_100_return_locations_by_member_type_v1	split start_location	execute SPLIT() function to separate data into start_lat and start_long columns; add new column for rank	format for mapping by coordinates	Drew Brinkley
12/9/21	7:24	top_100_rental_locations_by_member_type_v1	split start_location	execute SPLIT() function to separate data into start_lat and start_long columns; add new column for rank	format for mapping by coordinates	Drew Brinkley
12/9/21	8:53	avg_monthly_ride_time_by_member_and_ride_types_v1	sort data by type	sort data by member type and ride type	format data for chart	Drew Brinkley
12/9/21	8:55	avg_monthly_ride_time_by_member_and_ride_types_v1	compare monthly ride times to average	calculate average ride time for each group; create new column to show how monthly ride time compares to average	format data for chart	Drew Brinkley
12/9/21	10:06	top_250_start_locations_by_time_and_ride_type_casual_riders_v1	split start_location	execute SPLIT() function to separate data into start_lat and start_long columns; add new column for rank	format for mapping by coordinates	Drew Brinkley
12/9/21	13:59	top_250_start_locations_by_time_and_ride_type_casual_riders_v1	determine percentage of group represented	use COUNTIFS() and SUMIF() functions to calculate count and total of "electric" group	validate sample size, analyze data	Drew Brinkley

12/9/21	10:12	top_250_start_locations_by_time_and_ride_type_members_v1	split start_location	execute SPLIT() function to separate data into start_lat and start_long columns; add new column for rank	format for mapping by coordinates	Drew Brinkley
12/9/21	13:35	top_250_start_locations_by_time_and_ride_type_members_v1	determine percentage of group represented	use COUNTIFS() and SUMIF() functions to calculate count and total of "electric" group	validate sample size, analyze data	Drew Brinkley
12/11/21	13:51	riders_by_time_of_week_and_trip_type_v1	compare number of rides by group	generate pivot tables	analyze trends in ridership	Drew Brinkley
12/11/21	14:27	ride_time_by_rental_return_location_v1	compare ride time by group	use CONCAT() function to combine columns for ease of labeling; sort data into categories by ridership group	format data for chart	Drew Brinkley
12/12/21	9:40	holiday_riders_by_member_type_v1	compare holiday ride times of rider groups	use SUM() and AVERAGE() functions to calculate summary statistics. create charts	analyze trends in ridership	Drew Brinkley