Cyclistic Case Study

2022-08-24

Summary

Cyclistic, a successful bike-sharing company launched in 2016 in Chicago, has a fleet of over 5,800 bikes that can be unlocked from one station and returned to any other station in the system at anytime. The fleet consists of traditional bikes, reclining bikes, hand tricycles and cargo bikes. The majority of riders opt for traditional bikes (8% of riders use the assistive options). Cyclistic users are more likely to ride for leisure, though approximately 30% use them to commute to work each day.

Until now, the company's marketing strategy has relied on building general awareness and appealing to broad consumer segments. The company offers different pricing plans, including single ride passes, full day passes, and annual memberships. Customers are classified as casual riders, those purchasing single ride or full day passes, and annual members, those who purchase annual memberships. Cyclistic's finance analysts have determined that annual members are more profitable than casual riders. The company believes maximizing the number of annual members will be key to future growth and that there is opportunity to convert causal riders into members. Cyclistic will look to design marketing strategies aimed at converting casual riders into annual members. In order to design a strategy, the company seeks to better understand a) how annual members and casual riders use Cyclistic bikes differently, b) why casual riders would buy Cyclistic annual memberships, and c) how Cyclistic can use digital media to influence casual riders to become members.

The scope of this report will be focused on the first question, how do annual members and casual riders use Cyclistic bikes differently?

Data Source

Data for analysis includes six months of Cyclistic trip data through July 2022, which can be found here (https://divvy-tripdata.s3.amazonaws.com/index.html). Permission for use of data can be found here (https://www.divvybikes.com/data-license-agreement).

As internally sourced data from Cyclistic, the data is deemed credible and is current. While a full 12 months of data was preferred for analysis, data processing capacity was limited and attempts at using 12 months of data was not feasible due to system constraints. The six months of data from February - July 2022 still aims to capture seasonality with a duration covering winter - summer.

The data set includes ride observation data as follows: ride ID, bike type, start date/time, end date/time, start station name, start station ID, end station name, end station ID, start latitude, start longitude, end latitude, end longitude, and member type. Each month of data consistently includes all the same variables. Total observations aggregate to over 2.99 mm for the six month period.

Review of the data shows there is missing start station name and id and end station name and id for some observations in each month of data, although the observations include start and end latitude and longitude data. Approximately 14%-15% of observations are missing start or end station name from aggregate six month data. While an attempt was made to compare the number of distinct latitude and longitude pairs to the distinct station names, the disparity was significantly large, making pairing of station names to latitude-longitude coordinates difficult to determine in order to confidently fill the missing data accurately. Given the specificity of the latitude and longitude data and the significant difference in identified stations and latitude-longitude pairs, the data may indicate multiple latitude-longitude pairs align to a single station in the network if the docking stations are large enough, though that is speculative. Analysis for this project includes these rides except where analysis relates to start and end station data.

Due to data privacy issues, riders' personally identifiable information is not disclosed and brings about limitations, such as inability to connect pass purchases to credit card numbers to determine if a casual rider has purchased multiple single passes or if a rider lives in the company's service area. Thus, the data does not reveal patterns or frequency of use, if any, of individual casual riders.

Data Processing

Data processing began in Excel with a review of the six months of data files from February - July 2022 and the variables included in the data which was consistent across the six months.

Review of the data showed 14%-15% of the rides missing start station or end station data. In order to further study this issue and to determine if a match between station names and latitude-longitude coordinates could be made to fill in the missing data, analysis was completed on a single month of data. A concatenate function was used to create latitude-longitude coordinates for rides in the data set and uploaded to Big Query. A simple COUNT DISTINCT function was used to determine the number of latitude-longitude coordinates, start station names and end station names. There were significantly more distinct latitude-longitude coordinates than there were station names, indicating it would be difficult to try to accurately assign station names to latitude-longitude coordinates for observations that were missing this data and attempts to do this were abandoned.

To prepare data for further analysis in RStudio, the monthly files were further processed in excel. The concatenated data in the one month of data used for analysis noted above was deleted. Data related to latitude and longitude and start station ID and end station ID were deleted;data for day of week (1=Sunday to 7=Saturday) and month (1-12) were added. All files included the same remaining columns of data in the same format to be further processed in RStudio.

Data processing in RStudio

Install packages that may be used for cleaning and analysis if not already installed

```
install.packages("tidyverse")
```

Call associated libraries

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

Read/import data files

```r
c202202 <- read_csv("C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/Excel Files/c202202.csv"
)
```

```
## Rows: 115609 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (2): day_of_week, month
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
c202203 <- read_csv("C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/Excel Files/c202203.csv"
)
```

```
## Rows: 284042 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (2): day_of_week, month
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
c202204 <- read_csv("C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/Excel Files/c202204.csv"
)
```

```
## Rows: 371249 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (2): day_of_week, month
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
c202205 <- read_csv("C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/Excel Files/c202205.csv"
)
```

```
## Rows: 634858 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (2): day_of_week, month
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
c202206 <- read_csv("C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/Excel Files/c202206.csv"
)
```

```
## Rows: 769204 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (2): day_of_week, month
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
c202207 <- read_csv("C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/Excel Files/c202207.csv"
)
```

```
## Rows: 823488 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (2): day_of_week, month
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Review imported data to see columns and data types

```
glimpse(c202202)
```

```
## Rows: 115,609
## Columns: 9
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <chr> "2/19/2022 18:08", "2/20/2022 17:41", "2/25/2022 18…
## $ ended_at           <chr> "2/19/2022 18:23", "2/20/2022 17:45", "2/25/2022 19…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 7, 1, 6, 2, 4, 2, 2, 3, 6, 1, 2, 3, 7, 7, 2, 2, 2, …
## $ month              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …
```

```
glimpse(c202203)
```

```
## Rows: 284,042
## Columns: 9
## $ ride_id            <chr> "47EC0A7F82E65D52", "8494861979B0F477", "EFE527AF80…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "classic_bike", "c…
## $ started_at         <chr> "3/21/2022 13:45", "3/16/2022 9:37", "3/23/2022 19:…
## $ ended_at           <chr> "3/21/2022 13:51", "3/16/2022 9:43", "3/23/2022 19:…
## $ start_station_name <chr> "Wabash Ave & Wacker Pl", "Michigan Ave & Oak St", …
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Orleans St & Chestnut …
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 2, 4, 4, 3, 2, 2, 5, 7, 5, 6, 1, 4, 2, 2, 4, 4, 4, …
## $ month              <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, …
```

```
glimpse(c202204)
```

```
## Rows: 371,249
## Columns: 9
## $ ride_id            <chr> "3564070EEFD12711", "0B820C7FCF22F489", "89EEEE3229…
## $ rideable_type      <chr> "electric_bike", "classic_bike", "classic_bike", "c…
## $ started_at         <chr> "4/6/2022 17:42", "4/24/2022 19:23", "4/20/2022 19:…
## $ ended_at           <chr> "4/6/2022 17:54", "4/24/2022 19:43", "4/20/2022 19:…
## $ start_station_name <chr> "Paulina St & Howard St", "Wentworth Ave & Cermak R…
## $ end_station_name   <chr> "University Library (NU)", "Green St & Madison St",…
## $ member_casual      <chr> "member", "member", "member", "casual", "member", "…
## $ day_of_week        <dbl> 4, 1, 4, 6, 7, 5, 2, 3, 6, 6, 7, 4, 4, 7, 4, 2, 2, …
## $ month              <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, …
```

```
glimpse(c202205)
```

```
## Rows: 634,858
## Columns: 9
## $ ride_id            <chr> "EC2DE40644C6B0F4", "1C31AD03897EE385", "1542FBEC83…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <chr> "5/23/2022 23:06", "5/11/2022 8:53", "5/26/2022 18:…
## $ ended_at           <chr> "5/23/2022 23:40", "5/11/2022 9:31", "5/26/2022 18:…
## $ start_station_name <chr> "Wabash Ave & Grand Ave", "DuSable Lake Shore Dr & …
## $ end_station_name   <chr> "Halsted St & Roscoe St", "Field Blvd & South Water…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 2, 4, 5, 3, 3, 4, 6, 1, 2, 4, 5, 3, 2, 4, 5, 3, 1, …
## $ month              <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, …
```

```
glimpse(c202206)
```

```
## Rows: 769,204
## Columns: 9
## $ ride_id            <chr> "600CFD130D0FD2A4", "F5E6B5C1682C6464", "B6EB6D27BA…
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", …
## $ started_at         <chr> "6/30/2022 17:27", "6/30/2022 18:39", "6/30/2022 11…
## $ ended_at           <chr> "6/30/2022 17:35", "6/30/2022 18:47", "6/30/2022 12…
## $ start_station_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "…
## $ day_of_week        <dbl> 5, 5, 5, 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, …
## $ month              <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, …
```

```
glimpse(c202207)
```

```
## Rows: 823,488
## Columns: 9
## $ ride_id            <chr> "954144C2F67B1932", "292E027607D218B6", "5776585258…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <chr> "7/5/2022 8:12", "7/26/2022 12:53", "7/3/2022 13:58…
## $ ended_at           <chr> "7/5/2022 8:24", "7/26/2022 12:55", "7/3/2022 14:06…
## $ start_station_name <chr> "Ashland Ave & Blackhawk St", "Buckingham Fountain …
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th St"…
## $ member_casual      <chr> "member", "casual", "casual", "casual", "member", "…
## $ day_of_week        <dbl> 3, 3, 1, 1, 4, 6, 2, 5, 1, 1, 6, 7, 3, 4, 2, 1, 5, …
## $ month              <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, …
```

Combine monthly data into one data frame

```
t6202207 <- rbind(c202202, c202203, c202204, c202205, c202206, c202207)
```

Review consolidated data

```
glimpse(t6202207)
```

```
## Rows: 2,998,450
## Columns: 9
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <chr> "2/19/2022 18:08", "2/20/2022 17:41", "2/25/2022 18…
## $ ended_at           <chr> "2/19/2022 18:23", "2/20/2022 17:45", "2/25/2022 19…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 7, 1, 6, 2, 4, 2, 2, 3, 6, 1, 2, 3, 7, 7, 2, 2, 2, …
## $ month              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …
```

From reviewing the data, start and end date and time is noted as being character data type and not in the correct data type for R calculations, so convert data type and review data to check change in data type

```
t6202207$started_at = strptime(t6202207$started_at, format = "%m/%d/%Y %H:%M")
t6202207$ended_at = strptime(t6202207$ended_at, format = "%m/%d/%Y %H:%M")
```

```
glimpse(t6202207)
```

```
## Rows: 2,998,450
## Columns: 9
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <dttm> 2022-02-19 18:08:00, 2022-02-20 17:41:00, 2022-02-…
## $ ended_at           <dttm> 2022-02-19 18:23:00, 2022-02-20 17:45:00, 2022-02-…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 7, 1, 6, 2, 4, 2, 2, 3, 6, 1, 2, 3, 7, 7, 2, 2, 2, …
## $ month              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …
```

Now that start and end time data type is corrected, add trip duration to data frame and review new column of data added

```
t6202207 <- mutate(t6202207, tripduration = difftime(ended_at, started_at, units = "mins"))
```

```
glimpse(t6202207)
```

```
## Rows: 2,998,450
## Columns: 10
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <dttm> 2022-02-19 18:08:00, 2022-02-20 17:41:00, 2022-02-…
## $ ended_at           <dttm> 2022-02-19 18:23:00, 2022-02-20 17:45:00, 2022-02-…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 7, 1, 6, 2, 4, 2, 2, 3, 6, 1, 2, 3, 7, 7, 2, 2, 2, …
## $ month              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …
## $ tripduration       <drtn> 15 mins, 4 mins, 14 mins, 7 mins, 3 mins, 16 mins,…
```

Check to make sure all trips are allocated to casual or member user classifications correctly

```
summarize(t6202207, tot_trips = n()) ## total trips
```

```
## # A tibble: 1 × 1
##   tot_trips
##       <int>
## 1   2998450
```

```
t6202207 %>%
  filter(member_casual == "casual" | member_casual == "member") %>%
  summarize(total = n()) ##total trips allocated to member or casual user type should equal total trips
```

```
## # A tibble: 1 × 1
##     total
##     <int>
## 1 2998450
```

Check to make sure all trips are allocated to February (2) through July (7)

```
t6202207 %>%
  filter(month >=2 & month <= 7) %>%
  summarize(total = n()) ##total trips classified between 2-7 should equal total trips
```

```
## # A tibble: 1 × 1
##     total
##     <int>
## 1 2998450
```

Check trip durations to see if values make sense (none < 0)

```
t6202207 %>%
  filter(tripduration < 0) %>%
  summarize(total_invalid = n()) ##total trips with duration <0 should be 0
```

```
## # A tibble: 1 × 1
##   total_invalid
##           <int>
## 1            28
```

Since some trip durations seem to be <0 and invalid, exclude them from the data set to be analyzed (which will reduce the data set by 28 trips) and review results

```
t6202207 <- t6202207 %>%
  filter(tripduration >= 0)
```

```
glimpse(t6202207)
```

```
## Rows: 2,998,422
## Columns: 10
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <dttm> 2022-02-19 18:08:00, 2022-02-20 17:41:00, 2022-02-…
## $ ended_at           <dttm> 2022-02-19 18:23:00, 2022-02-20 17:45:00, 2022-02-…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 7, 1, 6, 2, 4, 2, 2, 3, 6, 1, 2, 3, 7, 7, 2, 2, 2, …
## $ month              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …
## $ tripduration       <drtn> 15 mins, 4 mins, 14 mins, 7 mins, 3 mins, 16 mins,…
```

Change month variables from 1-12 to Jan-Dec and review new data frame

```
t6202207 <- mutate(t6202207, month = month.abb[as.numeric(month)])
```

```
glimpse(t6202207)
```

```
## Rows: 2,998,422
## Columns: 10
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <dttm> 2022-02-19 18:08:00, 2022-02-20 17:41:00, 2022-02-…
## $ ended_at           <dttm> 2022-02-19 18:23:00, 2022-02-20 17:45:00, 2022-02-…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <dbl> 7, 1, 6, 2, 4, 2, 2, 3, 6, 1, 2, 3, 7, 7, 2, 2, 2, …
## $ month              <chr> "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "F…
## $ tripduration       <drtn> 15 mins, 4 mins, 14 mins, 7 mins, 3 mins, 16 mins,…
```

Change day of week variables from 1-7 to Sun-Sat and review new data frame

```
t6202207$day_of_week <- factor(t6202207$day_of_week, labels = c("Sun", "Mon","Tue","Wed", "Thu", "Fri", "Sat"), levels = c(1
,2,3,4,5,6,7))
```

```
glimpse(t6202207)
```

```
## Rows: 2,998,422
## Columns: 10
## $ ride_id            <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl…
## $ started_at         <dttm> 2022-02-19 18:08:00, 2022-02-20 17:41:00, 2022-02-…
## $ ended_at           <dttm> 2022-02-19 18:23:00, 2022-02-20 17:45:00, 2022-02-…
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood …
## $ end_station_name   <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ day_of_week        <fct> Sat, Sun, Fri, Mon, Wed, Mon, Mon, Tue, Fri, Sun, M…
## $ month              <chr> "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "F…
## $ tripduration       <drtn> 15 mins, 4 mins, 14 mins, 7 mins, 3 mins, 16 mins,…
```

```
write.csv(t6202207, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/t6202207v4.csv")
```

Analysis

Organize data by user type

```
cas_rides <- t6202207 %>%
  filter(member_casual == "casual")

member_rides <- t6202207 %>%
  filter(member_casual == "member")
```

Summarize data by user type for comparison

Total trips and average trip length

```
summarize(cas_rides, total_trips = n(), avg_trip_len = mean(tripduration))
```

```
## # A tibble: 1 × 2
##   total_trips avg_trip_len
##         <int> <drtn>
## 1     1293220 30.64258 mins
```

```
summarize(member_rides, total_trips = n(), avg_trip_len = mean(tripduration))
```

```
## # A tibble: 1 × 2
##   total_trips avg_trip_len
##         <int> <drtn>
## 1     1705202 13.06311 mins
```

Trips by month

```
cas_rides_monthly <- cas_rides %>%
  group_by(month) %>%
  summarize(total = n()) %>%
  arrange(desc(total))

member_rides_monthly <- member_rides %>%
  group_by(month) %>%
  summarize(total = n()) %>%
  arrange(desc(total))
```

```
glimpse(cas_rides_monthly)
```

```
## Rows: 6
## Columns: 2
## $ month <chr> "Jul", "Jun", "May", "Apr", "Mar", "Feb"
## $ total <int> 406047, 369045, 280415, 126417, 89880, 21416
```

```
glimpse(member_rides_monthly)
```

```
## Rows: 6
## Columns: 2
## $ month <chr> "Jul", "Jun", "May", "Apr", "Mar", "Feb"
## $ total <int> 417426, 400148, 354443, 244832, 194160, 94193
```

```
write.csv(cas_rides_monthly, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/cas_rides
_monthlyv4.csv")
write.csv(member_rides_monthly, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/memb_r
ides_monthlyv4.csv")
```

Trips by day and average trip length

```
cas_rides_day_avg_time <- cas_rides %>%
  group_by(day_of_week) %>%
  summarize(total = n(), avg_time = mean(tripduration)) %>%
  arrange(day_of_week)

member_rides_day_avg_time <- member_rides %>%
  group_by(day_of_week) %>%
  summarize(total = n(), avg_time = mean(tripduration)) %>%
  arrange(day_of_week)
```

```
glimpse(cas_rides_day_avg_time)
```

```
## Rows: 7
## Columns: 3
## $ day_of_week <fct> Sun, Mon, Tue, Wed, Thu, Fri, Sat
## $ total       <int> 239592, 159362, 142825, 142923, 169852, 171314, 267352
## $ avg_time    <drtn> 34.61540 mins, 31.65869 mins, 27.76225 mins, 26.42417 mins…
```

```
glimpse(member_rides_day_avg_time)
```

```
## Rows: 7
## Columns: 3
## $ day_of_week <fct> Sun, Mon, Tue, Wed, Thu, Fri, Sat
## $ total       <int> 215827, 240457, 263142, 256458, 268689, 229690, 230939
## $ avg_time    <drtn> 14.48239 mins, 12.78121 mins, 12.36779 mins, 12.30887 mins…
```

```
write.csv(cas_rides_day_avg_time, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/cas_
days_wk_durationv4.csv")
write.csv(member_rides_day_avg_time, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/m
emb_days_wk_durationv4.csv")
```

Start Station Popularity

```
top_cas_start_stn <- cas_rides %>%
  filter(!is.na(start_station_name)) %>% ##exclude rides with no start station name
  group_by(start_station_name) %>%
  summarize(total = n()) %>%
  arrange(desc(total))

top_memb_start_stn <- member_rides %>%
  filter(!is.na(start_station_name)) %>% ##exclude rides with no start station name
  group_by(start_station_name) %>%
  summarize(total = n()) %>%
  arrange(desc(total))
```

```
glimpse(top_cas_start_stn)
```

```
## Rows: 1,341
## Columns: 2
## $ start_station_name <chr> "Streeter Dr & Grand Ave", "DuSable Lake Shore Dr &…
## $ total              <int> 34775, 19134, 15418, 15080, 14861, 11708, 11225, 92…
```

```
glimpse(top_memb_start_stn)
```

```
## Rows: 1,236
## Columns: 2
## $ start_station_name <chr> "Kingsbury St & Kinzie St", "Wells St & Concord Ln"…
## $ total              <int> 13257, 11294, 11151, 10117, 10096, 10077, 9843, 979…
```

```
write.csv(top_cas_start_stn, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/cas_ride_
top_start_stationv4.csv")
write.csv(top_memb_start_stn, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/memb_rid
e_top_start_stationv4.csv")
```

End Station Popularity

```
top_cas_end_stn <- cas_rides %>%
  filter(!is.na(end_station_name)) %>% ##exclude rides with no end station name
  group_by(end_station_name) %>%
  summarize(total = n()) %>%
  arrange(desc(total))

top_memb_end_stn <- member_rides %>%
  filter(!is.na(end_station_name)) %>% ##exclude rides with no end station name
  group_by(end_station_name) %>%
  summarize(total = n()) %>%
  arrange(desc(total))
```

```
glimpse(top_cas_start_stn)
```

```
## Rows: 1,341
## Columns: 2
## $ start_station_name <chr> "Streeter Dr & Grand Ave", "DuSable Lake Shore Dr &…
## $ total              <int> 34775, 19134, 15418, 15080, 14861, 11708, 11225, 92…
```

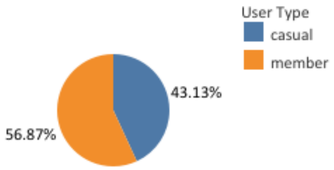
```
glimpse(top_memb_start_stn)
```

```
## Rows: 1,236
## Columns: 2
## $ start_station_name <chr> "Kingsbury St & Kinzie St", "Wells St & Concord Ln"…
## $ total              <int> 13257, 11294, 11151, 10117, 10096, 10077, 9843, 979…
```

```
write.csv(top_cas_end_stn, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/cas_ride_to
p_end_stationv4.csv")
write.csv(top_memb_end_stn, file ="C:/Users/Steph/Desktop/Google Certificate/Bike Share Case Study/Divvy Tripdata/memb_ride_
top_end_stationv4.csv")
```

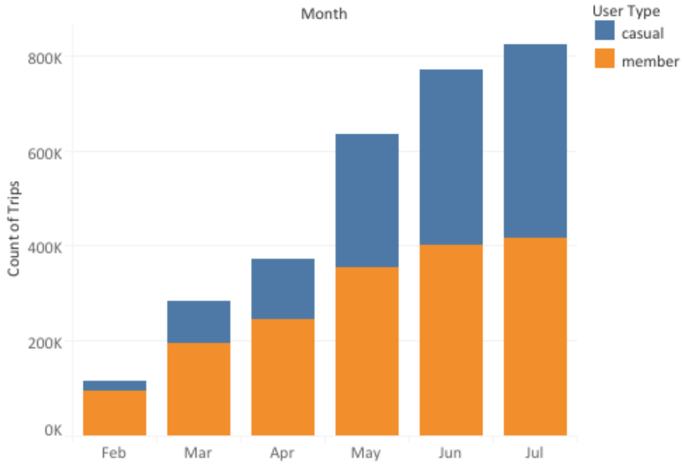
Trip Composition

Casual user trips account for a large portion of total trips



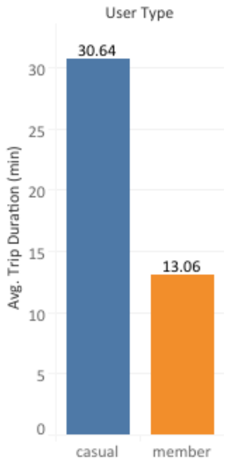
Seasonality

Bike usage is significantly higher in the Summer than Winter



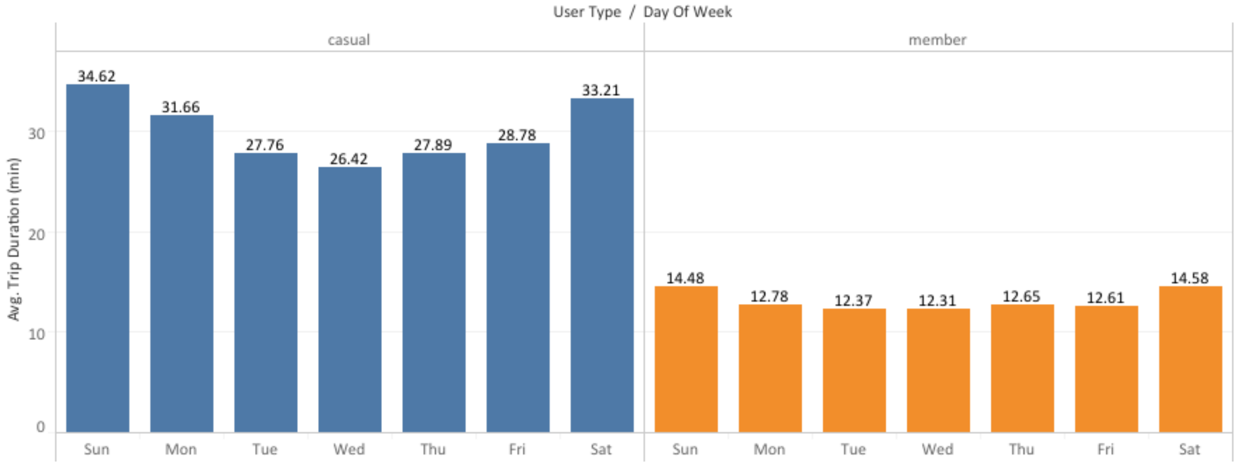
Average Trip Duration

Average trip duration for casual user trips is more than double that of member trips



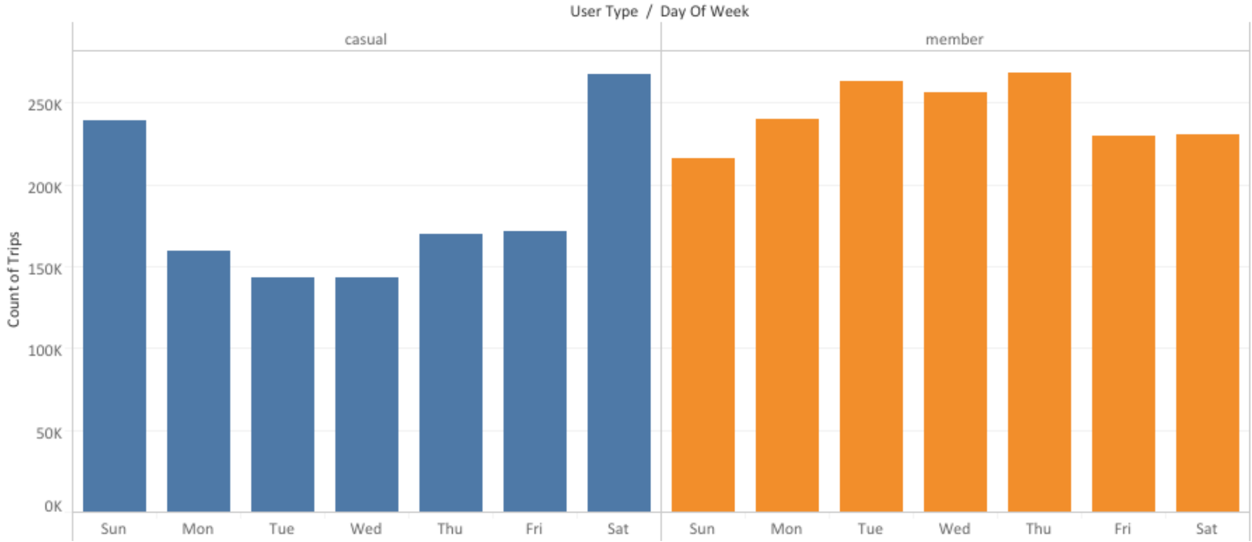
Daily Average Trip Duration

Average trip duration for members is shorter than casual users on all days



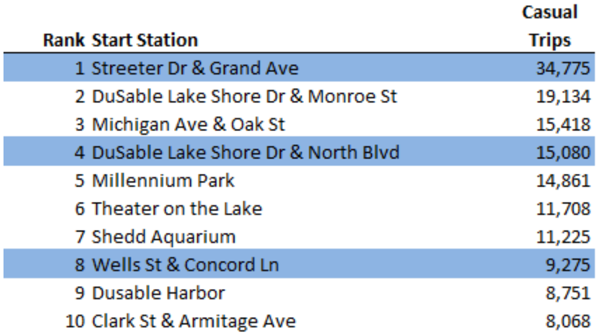
Daily Trips

Casual users take more trips on the weekend relative to members

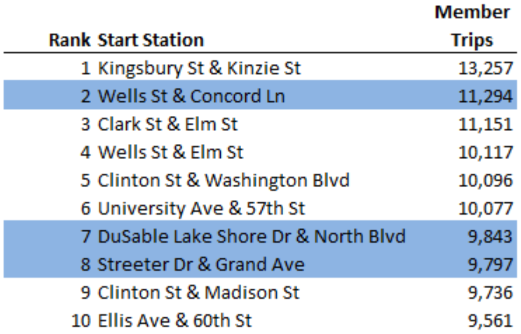


Top Start Stations

Start station popularity differs between casual users and members with three stations in the top ten overlapping



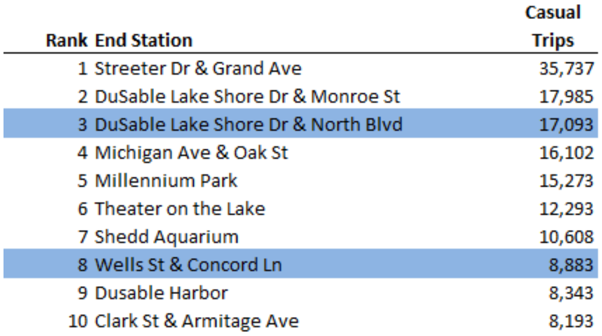
Rank	Start Station	Casual Trips
1	Streeter Dr & Grand Ave	34,775
2	DuSable Lake Shore Dr & Monroe St	19,134
3	Michigan Ave & Oak St	15,418
4	DuSable Lake Shore Dr & North Blvd	15,080
5	Millennium Park	14,861
6	Theater on the Lake	11,708
7	Shedd Aquarium	11,225
8	Wells St & Concord Ln	9,275
9	Dusable Harbor	8,751
10	Clark St & Armitage Ave	8,068



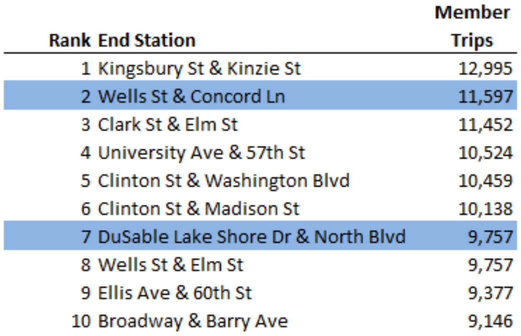
Rank	Start Station	Member Trips
1	Kingsbury St & Kinzie St	13,257
2	Wells St & Concord Ln	11,294
3	Clark St & Elm St	11,151
4	Wells St & Elm St	10,117
5	Clinton St & Washington Blvd	10,096
6	University Ave & 57th St	10,077
7	DuSable Lake Shore Dr & North Blvd	9,843
8	Streeter Dr & Grand Ave	9,797
9	Clinton St & Madison St	9,736
10	Ellis Ave & 60th St	9,561

Top End Stations

End station popularity differs between casual users and members with two stations in the top ten overlapping



Rank	End Station	Casual Trips
1	Streeter Dr & Grand Ave	35,737
2	DuSable Lake Shore Dr & Monroe St	17,985
3	DuSable Lake Shore Dr & North Blvd	17,093
4	Michigan Ave & Oak St	16,102
5	Millennium Park	15,273
6	Theater on the Lake	12,293
7	Shedd Aquarium	10,608
8	Wells St & Concord Ln	8,883
9	Dusable Harbor	8,343
10	Clark St & Armitage Ave	8,193



Rank	End Station	Member Trips
1	Kingsbury St & Kinzie St	12,995
2	Wells St & Concord Ln	11,597
3	Clark St & Elm St	11,452
4	University Ave & 57th St	10,524
5	Clinton St & Washington Blvd	10,459
6	Clinton St & Madison St	10,138
7	DuSable Lake Shore Dr & North Blvd	9,757
8	Wells St & Elm St	9,757
9	Ellis Ave & 60th St	9,377
10	Broadway & Barry Ave	9,146

Conclusions

Bike sharing is a seasonal business with significantly more trips taken in the summer than winter, regardless of user type. Casual users are an important segment of Cyclistic's user base, contributing 43% of total rides. There are differences in the usage patterns of casual users and members, with casual users taking longer trips on average and having higher weekend usage relative to members. Data also show differences in the top start and end stations for casual users and members.

Based on current analysis, broad recommendations for action include:
1) marketing during the summer or warmer months when usage is high and users are more likely to think they will use the service more frequently, making membership more attractive
2) targeting casual users who use the service on weekdays and for trip durations more similar to members (i.e. relatively shorter average trip durations)
3) targeting casual users who are using start and end stations that are popular among member users, as this may indicate the stations are used for routine, repeated activity, such as commuting for work

Limitations to the data however means additional information should be considered before action should be taken. Trends seen in the data suggest further research should be carried out, including gathering data at the user level and surveying users to better understand frequency of use, use case and motivations for use. Lack of individual user data limits the ability to specifically target users that may be more likely to convert to members. Additional research on station information, i.e. commercial, residential, population density, etc. may also be conducted to better understand popularity of stations and potential to target users of specific stations or stations that meet similar criteria that could indicate users that are good candidates for membership.