# Case Study Bike Share Company

Travis Smothermon

2025-07-15

## Step 1: Ask

### Business Task

The goal is to analyze how annual members and casual riders use Cyclistic bikes differently and provide recommendations to convert casual riders into annual members.

### Key Stakeholders

- Cyclistic marketing team
- Director of marketing
- Cyclistic executive team

### Key Questions

- How do casual riders and annual members use Cyclistic bikes differently?
- When do they ride (day of week, time of day)?
- What types of bikes do they prefer?
- How long are their rides?

## Step 2: Prepare

### Data Source

The data used in this analysis comes from Cyclistic's publicly available bike-share trip data. I downloaded datasets from 2019 and 2020, which are stored in the `rawdata` folder of this project.

### File Organization

- `rawdata/` contains the original CSV files
- `clean data/` will contain cleaned and merged datasets for analysis
- `visualizations/` will store plots generated from the analysis

### Data Description

Each CSV file contains ride-level data, including: - `ride_id` (unique ID for each trip) - `rideable_type` (type of bike used) - `started_at` and `ended_at` (timestamps) - `start_station_name`, `end_station_name` - `member_casual` (user type)

I will load the data, inspect the structure, and prepare it for cleaning in the next step.

**Data Credibility**

This dataset is provided by Motivate International Inc., considered reliable for internal business decisions. However, I will check for: - Missing values - Inconsistent or incorrect timestamps - Duplicates

**Loading and previewing the data**

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
df_2019_q1 <- read_csv("rawdata/bike_data_2019.csv")
```

```
## Rows: 365069 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): start_time, end_time, from_station_name, to_station_name, usertype,...
## dbl (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num (1): tripduration
##
## 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.
```

```
df_2020_q1 <- read_csv("rawdata/bike_data_2020.csv")
```

```
## Rows: 426887 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, en...
##
## 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.
```

```
glimpse(df_2019_q1)
```

```
## Rows: 365,069
## Columns: 12
## $ trip_id           <dbl> 21742443, 21742444, 21742445, 21742446, 21742447, 21~
## $ start_time        <chr> "2019-01-01 0:04:37", "2019-01-01 0:08:13", "2019-01~
```

```
## $ end_time          <chr> "2019-01-01 0:11:07", "2019-01-01 0:15:34", "2019-01-~
## $ bikeid            <dbl> 2167, 4386, 1524, 252, 1170, 2437, 2708, 2796, 6205,~
## $ tripduration      <dbl> 390, 441, 829, 1783, 364, 216, 177, 100, 1727, 336, ~
## $ from_station_id   <dbl> 199, 44, 15, 123, 173, 98, 98, 211, 150, 268, 299, 2~
## $ from_station_name <chr> "Wabash Ave & Grand Ave", "State St & Randolph St", ~
## $ to_station_id     <dbl> 84, 624, 644, 176, 35, 49, 49, 142, 148, 141, 295, 4~
## $ to_station_name   <chr> "Milwaukee Ave & Grand Ave", "Dearborn St & Van Bure~
## $ usertype          <chr> "Subscriber", "Subscriber", "Subscriber", "Subscribe~
## $ gender            <chr> "Male", "Female", "Female", "Male", "Male", "Female"~
## $ birthyear         <dbl> 1989, 1990, 1994, 1993, 1994, 1983, 1984, 1990, 1995~
```

```r
head(df_2019_q1)
```

```
## # A tibble: 6 x 12
##    trip_id start_time         end_time       bikeid tripduration from_station_id
##      <dbl> <chr>              <chr>           <dbl>        <dbl>           <dbl>
## 1 21742443 2019-01-01 0:04:37 2019-01-01 0:~   2167          390             199
## 2 21742444 2019-01-01 0:08:13 2019-01-01 0:~   4386          441              44
## 3 21742445 2019-01-01 0:13:23 2019-01-01 0:~   1524          829              15
## 4 21742446 2019-01-01 0:13:45 2019-01-01 0:~    252         1783             123
## 5 21742447 2019-01-01 0:14:52 2019-01-01 0:~   1170          364             173
## 6 21742448 2019-01-01 0:15:33 2019-01-01 0:~   2437          216              98
## # i 6 more variables: from_station_name <chr>, to_station_id <dbl>,
## #   to_station_name <chr>, usertype <chr>, gender <chr>, birthyear <dbl>
```

```r
summary(df_2019_q1)
```

```
##     trip_id           start_time          end_time             bikeid    
##  Min.   :21742443   Length:365069      Length:365069      Min.   :   1  
##  1st Qu.:21848765   Class :character   Class :character   1st Qu.:1777  
##  Median :21961829   Mode  :character   Mode  :character   Median :3489  
##  Mean   :21960872                                         Mean   :3429  
##  3rd Qu.:22071823                                         3rd Qu.:5157  
##  Max.   :22178528                                         Max.   :6471  
##                                                                         
##   tripduration      from_station_id from_station_name  to_station_id  
##  Min.   :      61   Min.   :  2.0   Length:365069      Min.   :  2.0  
##  1st Qu.:     326   1st Qu.: 76.0   Class :character   1st Qu.: 76.0  
##  Median :     524   Median :170.0   Mode  :character   Median :168.0  
##  Mean   :    1016   Mean   :198.1                      Mean   :198.6  
##  3rd Qu.:     866   3rd Qu.:287.0                      3rd Qu.:287.0  
##  Max.   :10628400   Max.   :665.0                      Max.   :665.0  
##                                                                       
##  to_station_name      usertype            gender            birthyear    
##  Length:365069      Length:365069      Length:365069      Min.   :1900   
##  Class :character   Class :character   Class :character   1st Qu.:1975   
##  Mode  :character   Mode  :character   Mode  :character   Median :1985   
##                                                           Mean   :1982   
##                                                           3rd Qu.:1990   
##                                                           Max.   :2003   
##                                                           NA's   :18023  
```

```r
glimpse(df_2020_q1)
```

```
## Rows: 426,887
## Columns: 13
## $ ride_id            <chr> "EACB19130B0CDA4A", "8FED874C809DC021", "789F3C21E4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <chr> "2020-01-21 20:06:59", "2020-01-30 14:22:39", "2020~
## $ ended_at           <chr> "2020-01-21 20:14:30", "2020-01-30 14:26:22", "2020~
## $ start_station_name <chr> "Western Ave & Leland Ave", "Clark St & Montrose Av~
## $ start_station_id   <dbl> 239, 234, 296, 51, 66, 212, 96, 96, 212, 38, 117, 1~
## $ end_station_name   <chr> "Clark St & Leland Ave", "Southport Ave & Irving Pa~
## $ end_station_id     <dbl> 326, 318, 117, 24, 212, 96, 212, 212, 96, 100, 632,~
## $ start_lat          <dbl> 41.9665, 41.9616, 41.9401, 41.8846, 41.8856, 41.889~
## $ start_lng          <dbl> -87.6884, -87.6660, -87.6455, -87.6319, -87.6418, -~
## $ end_lat            <dbl> 41.9671, 41.9542, 41.9402, 41.8918, 41.8899, 41.884~
## $ end_lng            <dbl> -87.6674, -87.6644, -87.6530, -87.6206, -87.6343, -~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```r
head(df_2020_q1)
```

```
## # A tibble: 6 x 13
##   ride_id  rideable_type started_at ended_at start_station_name start_station_id
##   <chr>    <chr>         <chr>      <chr>    <chr>                         <dbl>
## 1 EACB191~ docked_bike   2020-01-2~ 2020-01~ Western Ave & Lel~              239
## 2 8FED874~ docked_bike   2020-01-3~ 2020-01~ Clark St & Montro~              234
## 3 789F3C2~ docked_bike   2020-01-0~ 2020-01~ Broadway & Belmon~              296
## 4 C9A388D~ docked_bike   2020-01-0~ 2020-01~ Clark St & Randol~               51
## 5 943BC3C~ docked_bike   2020-01-3~ 2020-01~ Clinton St & Lake~               66
## 6 6D9C8A6~ docked_bike   2020-01-1~ 2020-01~ Wells St & Hubbar~              212
## # i 7 more variables: end_station_name <chr>, end_station_id <dbl>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>
```

```r
summary(df_2020_q1)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:426887      Length:426887      Length:426887      Length:426887
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_name start_station_id end_station_name   end_station_id
##  Length:426887      Min.   :  2.0    Length:426887      Min.   :  2.0
##  Class :character   1st Qu.: 77.0    Class :character   1st Qu.: 77.0
##  Mode  :character   Median :176.0    Mode  :character   Median :175.0
##                     Mean   :209.8                       Mean   :209.3
##                     3rd Qu.:298.0                       3rd Qu.:297.0
##                     Max.   :675.0                       Max.   :675.0
##                                                         NA's   :1
```

```
##     start_lat        start_lng          end_lat          end_lng
##  Min.   :41.74   Min.   :-87.77   Min.   :41.74   Min.   :-87.77
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.89   Median :-87.64   Median :41.89   Median :-87.64
##  Mean   :41.90   Mean   :-87.64   Mean   :41.90   Mean   :-87.64
##  3rd Qu.:41.92   3rd Qu.:-87.63   3rd Qu.:41.92   3rd Qu.:-87.63
##  Max.   :42.06   Max.   :-87.55   Max.   :42.06   Max.   :-87.55
##                                   NA's   :1       NA's   :1
##  member_casual
##  Length:426887
##  Class :character
##  Mode  :character
##
##
##
##
```

**Issues found**

- In df_2019_q1 the `start_time` and `end_time` are set as characters instead of date type
- In df_2020_q1 the `started_at` and `ended_at` are set as characters as well instead of date type

## Step 3: Process (Data Cleaning)

**We are trying to solve a business problem for our stakeholders, that is "How do annual members and casual riders use Cyclistic bikes differently?" being able to track the average trip length between the two customer types will be critical. In order to do that, we must convert `start_time`, `end_time`, `started_at`, and `ended_at` into datetime format.**

```r
library(tidyverse)
library(lubridate)

#This converts start_time and end_time to the appropriate datetime format
df_2019_cleaned <- df_2019_q1 %>%
  mutate(
    start_time = ymd_hms(start_time),
    end_time =ymd_hms(end_time),
    ride_length = as.numeric(end_time - start_time, units = 'mins')
   ) %>%
  filter(ride_length > 0)
```

```r
df_2020_cleaned <- df_2020_q1 %>%
  mutate(
    started_at = ymd_hms(started_at),
    ended_at = ymd_hms(ended_at),
    ride_length = as.numeric(ended_at - started_at, units = "mins")
  ) %>%
  filter(ride_length > 0)
```

**It is time to combine the two cleaned dataframes. This is what will allow us to easily compare the datasets from quarter 1 in both 2019 and 2020. In order to do this, we need to change the colnames in one dataset in order to combine them with the other.**

```
#Checking column names

colnames(df_2019_cleaned)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
## [13] "ride_length"
```

```
colnames(df_2020_cleaned)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"      "ride_length"
```

**We need to rename quite a few columns and also only keep columns relevant to the business question to combine them. First we need to rename the member types to be uniform across both datasets.**

```
df_2019_cleaned <- df_2019_cleaned %>%
  mutate(member_type = case_when(
    usertype == "Subscriber" ~ "member",
    usertype == "Customer" ~ "casual",
    TRUE ~ as.character(usertype)
  ))
```

```
df_2020_cleaned <- df_2020_cleaned %>%
  rename(member_type = member_casual)
```

```
df_2019_cleaned <- df_2019_cleaned %>%
  # Rename columns to match 2020 names
  rename(
    ride_id = trip_id,
    start_time = start_time,
    end_time = end_time,
    bike_id = bikeid,
    start_station_id = from_station_id,
    start_station_name = from_station_name,
    end_station_id = to_station_id,
    end_station_name = to_station_name
  ) %>%
  # Select only columns present in 2020 + member_type (standardized earlier)
```

```r
  select(
    ride_id,
    start_time,
    end_time,
    ride_length,
    start_station_id,
    start_station_name,
    end_station_id,
    end_station_name,
    member_type
  )
```

```r
df_2020_cleaned <- df_2020_cleaned %>%
  rename(
    start_time = started_at,
    end_time = ended_at,

  ) %>%
  select(
    ride_id,
    start_time,
    end_time,
    ride_length,
    start_station_id,
    start_station_name,
    end_station_id,
    end_station_name,
    member_type
  )
```

**ride_id is different in the two datasets, we are going to change the datatype to be characters for both**

```r
df_2019_cleaned <- df_2019_cleaned %>%
  mutate(ride_id = as.character(ride_id))
```

**Now the data we need from both datasets is uniform, we can combine them and move on to analyzing**

```r
combined_df <- bind_rows(df_2019_cleaned, df_2020_cleaned)
```

```r
View(combined_df)
```

**Saving the clean combined dataset to a new CSV so others could use it if needed**

```r
write.csv(combined_df, "Cleaned_Data_Final_Version/combined_bike_data.csv", row.names = FALSE)
```

## Step 4: Analyze

**We have a cleaned combined dataset, now we can analyze the data**

```
combined_df %>%
  count(member_type)
```

```
## # A tibble: 2 x 2
##   member_type      n
##   <chr>        <int>
## 1 casual       71433
## 2 member      720313
```

**There are significantly more members than casual customers**

```
combined_df %>%
  group_by(member_type) %>%
  summarize(
    average_ride_length = mean(ride_length, na.rm = TRUE),
    median_ride_length = median(ride_length, na.rm = TRUE),
    max_ride_length = max(ride_length, na.rm = TRUE),
    min_ride_length = min(ride_length, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 5
##   member_type average_ride_length median_ride_length max_ride_length
##   <chr>                     <dbl>              <dbl>           <dbl>
## 1 casual                     85.1              22.1          177200.
## 2 member                     13.3               8.47         101607.
## # i 1 more variable: min_ride_length <dbl>
```

**It appears that casual riders take longer trips on average**

```
combined_df <- combined_df %>%
  mutate(day_of_week = weekdays(as.Date(start_time)))
```

```
combined_df %>%
  group_by(member_type, day_of_week) %>%
  summarize(number_of_rides = n(), .groups = "drop")
```

```
## # A tibble: 14 x 3
##    member_type day_of_week number_of_rides
##    <chr>       <chr>                 <int>
##  1 casual      Friday                 8508
##  2 casual      Monday                 6694
##  3 casual      Saturday              13473
##  4 casual      Sunday                18652
```

```
##   5 casual      Thursday               7771
##   6 casual      Tuesday                7972
##   7 casual      Wednesday              8363
##   8 member      Friday               115168
##   9 member      Monday               110430
##  10 member      Saturday              59413
##  11 member      Sunday                60197
##  12 member      Thursday             125228
##  13 member      Tuesday              127974
##  14 member      Wednesday            121903
```

**Added a new column to show the day of week**

```
combined_df %>%
  group_by(member_type) %>%
  summarize(
    avg_ride_length = mean(ride_length, na.rm = TRUE),
    sd_ride_length = sd(ride_length, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 3
##   member_type avg_ride_length sd_ride_length
##   <chr>                 <dbl>          <dbl>
## 1 casual                 85.1          1625.
## 2 member                 13.3           273.
```

**The standard deviation of ride lengths for casual riders is notably higher than for members, indicating a wider spread in how long casual users ride. This suggests casual riders' trips vary greatly, while members tend to have more regular trip durations.**

## Top 10 start locations by member type

```
top_start_stations <- combined_df %>%
  group_by(member_type, start_station_name) %>%
  summarize(ride_count = n(), .groups = "drop") %>%
  arrange(member_type, desc(ride_count)) %>%
  group_by(member_type) %>%
  slice_head(n = 10)

knitr::kable(top_start_stations, caption = "Top 10 Start Stations by Member Type")
```

Table 1: Top 10 Start Stations by Member Type

| member_type | start_station_name | ride_count |
|---|---|---|
| casual | HQ QR | 3556 |
| casual | Streeter Dr & Grand Ave | 2749 |
| casual | Lake Shore Dr & Monroe St | 2732 |
| casual | Shedd Aquarium | 1832 |

| member_type | start_station_name | ride_count |
|---|---|---|
| casual | Millennium Park | 1406 |
| casual | Michigan Ave & Oak St | 1017 |
| casual | Michigan Ave & Washington St | 839 |
| casual | Dusable Harbor | 832 |
| casual | Adler Planetarium | 827 |
| casual | Theater on the Lake | 795 |
| member | Canal St & Adams St | 13799 |
| member | Clinton St & Washington Blvd | 13434 |
| member | Clinton St & Madison St | 12891 |
| member | Kingsbury St & Kinzie St | 8720 |
| member | Columbus Dr & Randolph St | 8515 |
| member | Canal St & Madison St | 7957 |
| member | Franklin St & Monroe St | 7010 |
| member | Michigan Ave & Washington St | 6686 |
| member | Larrabee St & Kingsbury St | 6467 |
| member | Clinton St & Lake St | 6439 |

### Top 10 end locations by member type

```r
top_end_stations <- combined_df %>%
  group_by(member_type, end_station_name) %>%
  summarize(ride_count = n(), .groups = "drop") %>%
  arrange(member_type, desc(ride_count)) %>%
  group_by(member_type) %>%
  slice_head(n = 10)

knitr::kable(top_end_stations, caption = "Top 10 End Stations by Member Type")
```

Table 2: Top 10 End Stations by Member Type

| member_type | end_station_name | ride_count |
|---|---|---|
| casual | Streeter Dr & Grand Ave | 3790 |
| casual | HQ QR | 3555 |
| casual | Lake Shore Dr & Monroe St | 2160 |
| casual | Millennium Park | 1934 |
| casual | Shedd Aquarium | 1461 |
| casual | Michigan Ave & Oak St | 1185 |
| casual | Theater on the Lake | 1080 |
| casual | Michigan Ave & Washington St | 965 |
| casual | Lake Shore Dr & North Blvd | 771 |
| casual | Wabash Ave & Grand Ave | 745 |
| member | Canal St & Adams St | 14807 |
| member | Clinton St & Washington Blvd | 14580 |
| member | Clinton St & Madison St | 13310 |
| member | Kingsbury St & Kinzie St | 8798 |
| member | Canal St & Madison St | 8265 |
| member | Michigan Ave & Washington St | 7674 |
| member | Clinton St & Lake St | 6709 |
| member | Franklin St & Monroe St | 6315 |

| member_type | end_station_name | ride_count |
|---|---|---|
| member | Daley Center Plaza | 6300 |
| member | LaSalle St & Jackson Blvd | 6238 |

## Summary

**Members are more consistent with their riding patterns as opposed to the casual customers. On average, casual customers ride roughly 6 times longer than members per trip, possibly due to less consistent/more one time rides. Canal St & Adams St is the most popular start and end spot, particularly among members.**

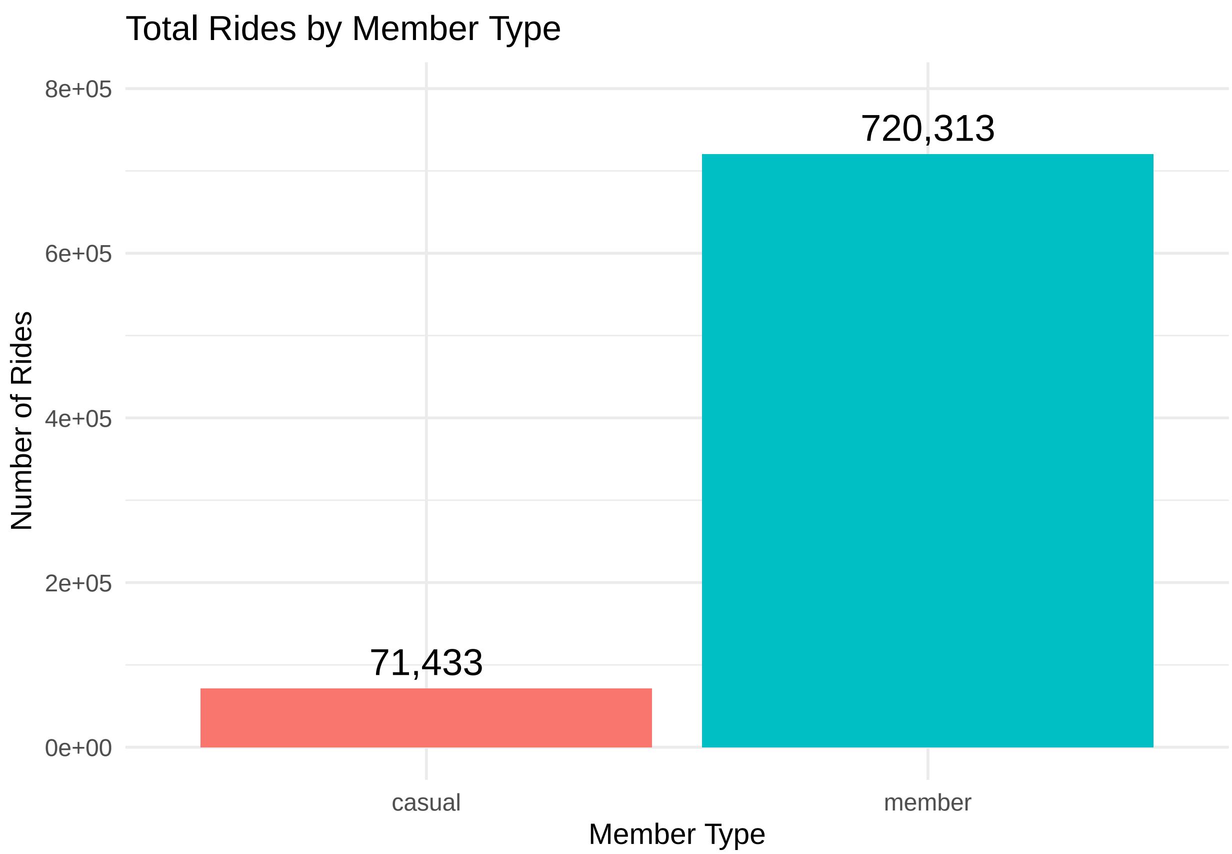
### Initial Suggestions

Since the goal is to convert casual riders to members, perhaps adopting a business model that incentivizes more consistent, shorter trips would be the best approach. Increasing the pricing of casual customer rides by how long the trip length is and giving members a flat monthly subscription fee would be effective. This would promote more customers to use the subscription service while also not impacting bike availability because of members shorter trip duration.

## Step 5: Share (Visualizations)

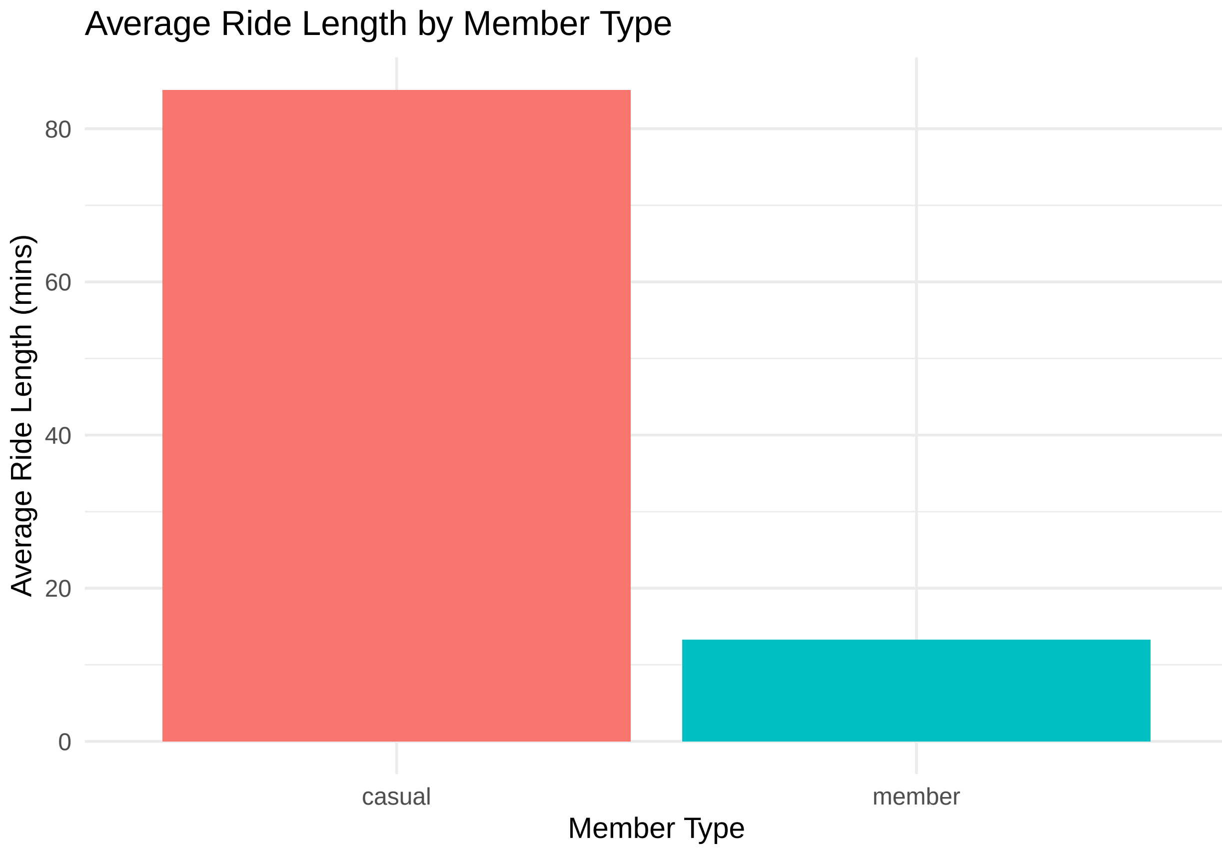
**We are going to make visualizations to show the differences between members and everyday customers to further add to our findings**

```
combined_df %>%
  count(member_type) %>%
  ggplot(aes(x = member_type, y = n, fill = member_type)) +
  geom_col() +
  geom_text(aes(label = scales::comma(n)), vjust = -0.5, size = 5) +
  expand_limits(y = max(combined_df %>% count(member_type) %>% pull(n)) * 1.1) +
  labs(
    title = "Total Rides by Member Type",
    x = "Member Type",
    y = "Number of Rides"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```




**Members take 10 times the amount of trips as casuals**

```
combined_df %>%
  group_by(member_type) %>%
  summarize(avg_ride_length = mean(ride_length, na.rm = TRUE)) %>%
  ggplot(aes(x = member_type, y = avg_ride_length, fill = member_type)) +
  geom_col() +
  labs(
    title = "Average Ride Length by Member Type",
    x = "Member Type",
    y = "Average Ride Length (mins)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```

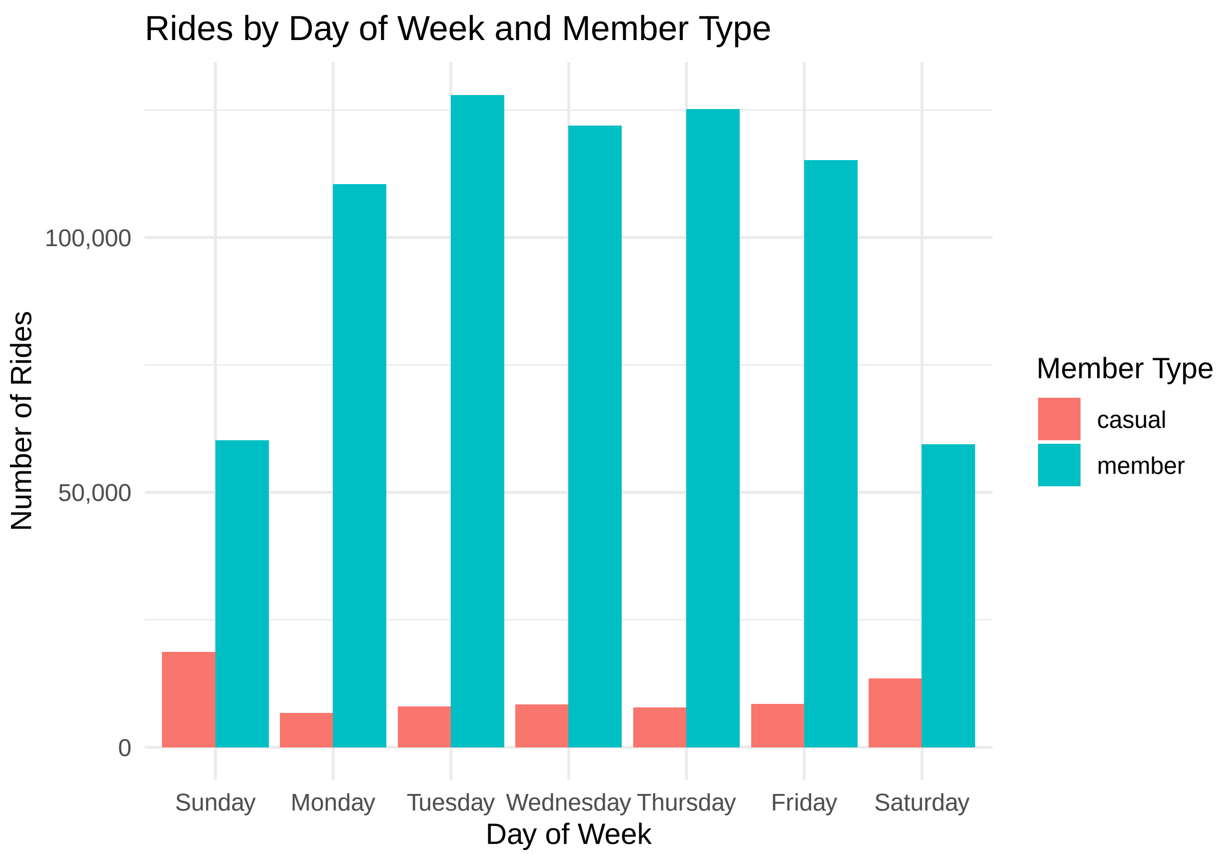



**The opposite is true for the average ride length**

**Lets see how these patterns change by the day of the week**

```
combined_df <- combined_df %>%
  mutate(day_of_week = factor(day_of_week,
                              levels = c("Sunday", "Monday", "Tuesday", "Wednesday","Thursday", "Friday
```

```
combined_df %>%
  group_by(member_type, day_of_week) %>%
  summarize(number_of_rides = n(), .groups = "drop") %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_type)) +
  geom_col(position = "dodge") +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Rides by Day of Week and Member Type",
    x = "Day of Week",
    y = "Number of Rides",
    fill = "Member Type"
  ) +
  theme_minimal()
```




### Members are significantly less likely to go biking on the weekends, while the opposite is true for casuals

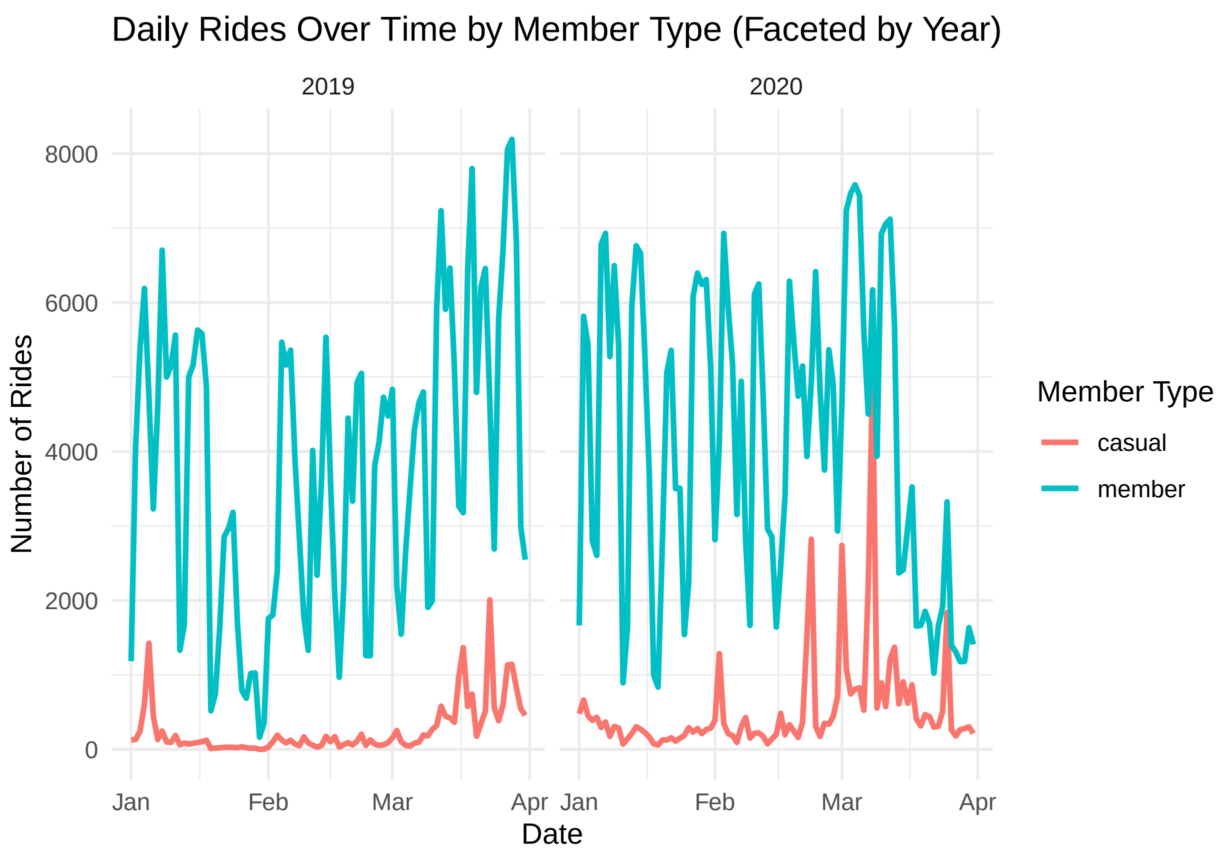
**Lets look at year over year trends**

```
combined_df <- combined_df %>%
  mutate(ride_date = as.Date(start_time))
```

```
combined_df %>%
  mutate(ride_date = as.Date(start_time),
         year = format(ride_date, "%Y")) %>%
  group_by(ride_date, member_type, year) %>%
  summarize(daily_rides = n(), .groups = "drop") %>%
  ggplot(aes(x = ride_date, y = daily_rides, color = member_type)) +
  geom_line(size = 1) +
  facet_wrap(~ year, scales = "free_x") +
  labs(
    title = "Daily Rides Over Time by Member Type (Faceted by Year)",
    x = "Date",
    y = "Number of Rides",
    color = "Member Type"
  ) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

```
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




**Casual customer traffic increases dramatically in the early spring and has increased year over year**

### Key Insights

- Members consistently take more rides than casual riders.
- Casual riders have a higher average ride length and more variability in trip duration.
- Casual riders tend to ride more on weekends, while members show more weekday usage — possibly indicating commuting.
- The most popular start and end stations differ slightly between rider types.
- Casual usage spikes suggest leisure activity, whereas member usage suggests routine travel.

### Step 6: Act

Based on the analysis of ride patterns, durations, and rider types, here are key recommendations to Lily Moreno (Marketing Manager) for Cyclistic to increase annual memberships:

#### 1. Promote Membership Benefits on Weekends

Casual riders are most active on weekends. Use this opportunity to promote: - Weekend membership discounts - Limited-time offers visible at popular stations - QR-code ads or app push notifications to sign up after rides - Possibly host community cycling events on Saturdays

#### 2. Target Popular Start/End Stations

Deploy marketing materials at the most-used casual rider stations. For example: - Station ambassadors offering membership flyers - Posters highlighting cost savings for frequent riders - Invest in additional amenities at the most popular stations (Coffee, Shops, etc)

#### 3. Emphasize Value Through Ride Duration

Casual riders often take longer rides. Cyclistic could: - Showcase how membership allows unlimited 45-minute rides (vs. costly casual fees) - Offer ride-time comparison calculators on the app - Showcase the health benefits from consistent cycling

#### 4. Offer Trial Memberships

To lower the barrier for casual riders: - Launch a **7-day free trial** - Offer a **first-month $1 promo** during peak seasons (spring/summer) - Market these promotions heavily on weekends

#### 5. Create Commuter Campaigns for Members

Members ride more during the week — likely commuters. Suggest: - Partnering with local employers for **commuter incentives** - Highlighting **reliable access**, **docking locations near offices**, etc. - Helps to retain current members while also increasing word of mouth

#### 6. Improve App-Based Engagement

Ensure that casual riders are nudged toward conversion: - In-app messages after X rides: "You could've saved $Y as a member!" - Gamify milestones (e.g., "You've ridden 5 times — unlock a membership bonus")

---

## Final Thoughts

By focusing on when, where, and how casual riders use the service, Cyclistic can design targeted campaigns that turn occasional users into loyal members.