Case Study: Cyclistic

Ekaterina Kemenova

2025-11-15

Contents

1 Introduction

Background: Cyclistic is a Chicago-based bike-share program with over 5,800 bicycles and 600 docking stations. Its pricing structure includes single-ride passes, full-day passes, and annual memberships, with casual riders purchasing short-term passes and members committing to annual plans. Because annual members generate higher long-term revenue, the executive and marketing teams aim to increase membership by converting existing casual riders into annual members. To support this goal, they seek data-driven insights into how the two groups differ in their usage patterns.

Objective: To analyze how annual members and casual riders use the Cyclistic bike-share program differently, in order to generate insights that will inform Cyclistic's marketing strategy aimed at converting casual riders into annual members.

Tasks:

1. Prepare and merge the Q1 2019 and Q1 2020 Cyclistic datasets.
2. Clean the data by fixing formats, standardizing fields and removing invalid records.
3. Analyze usage patterns of members vs casual riders across trip duration, time, and geography.
4. Assess demographic differences between rider groups where data is available.
5. Visualize key findings to illustrate behavioral patterns.
6. Formulate actionable recommendations to support converting casual riders into annual members.

2 Data import and overview

Loading the packages necessary for the analysis:

```
library(ggplot2)
library(tidyverse)
library(skimr)
library(janitor)
library(lubridate)
```

This analysis uses Cyclistic's historical bike trip data from Q1 2019 and Q1 2020 containing detailed records of individual rides. Two separate datasets represent these time periods:

```
df_2019 <- read_csv("Trips_2019_Q1.csv")
df_2020 <- read_csv("Trips_2020_Q1.csv")
```

Inspecting the data:

```
head(df_2019)
```

```
## # 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
head(df_2020)
```

```
## # 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
glimpse(df_2019)
```

```
## 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
glimpse(df_2020)
```

```
## 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", "~
```

The datasets have similar structure but slightly different column names and formats.

- The 2019 Q1 dataset contains 12 columns and 365,069 rows, including trip identifiers, start and end timestamps (stored as character values), station IDs and names, trip duration in seconds, rider type (Subscriber or Customer), gender, and birth year.
- The 2020 Q1 dataset contains 13 columns and 426,887 rows, including ride IDs, bike types, start and end timestamps, station information, GPS coordinates for trip start and end locations, and a rider classification variable.

The column names accurately reflect the contents of the data and follow an appropriate naming convention. Most columns use correct data types; however, before analysis, timestamps must be converted from chr to appropriate datetime formats. The datasets will then be standardized and merged to allow consistent comparison of member and casual rider behavior across the two years.

3 Data preparation

Converting the timestamp fields:

```
df_2019 <- df_2019 %>%
  mutate(
    start_time = ymd_hms(start_time),
    end_time   = ymd_hms(end_time)
  )
```

```
df_2020 <- df_2020 %>%
  mutate(
    started_at = ymd_hms(started_at),
    ended_at   = ymd_hms(ended_at)
  )
```

Changing the column names for consistency:

```
df_2020 <- df_2020 %>%
  rename(
    start_time = started_at,
    end_time = ended_at,
    trip_id = ride_id,
    user_type = member_casual
    )
```

```
df_2019 <- df_2019 %>%
  rename(
    birth_year = birthyear,
    bike_id = bikeid,
    trip_duration = tripduration,
    user_type = usertype,
    start_station_name = from_station_name,
    end_station_name = to_station_name,
    start_station_id = from_station_id,
    end_station_id = to_station_id
  )
```

```
unique(df_2019$user_type)
```

```
## [1] "Subscriber" "Customer"
```

```
unique(df_2020$user_type)
```

```
## [1] "member" "casual"
```

For the `user_type` column, we assume that the values "Subscriber" and "Customer" correspond to "member" and "casual rider," respectively. Therefore, we will recode these values to ensure consistency across both datasets.

```
df_2019 <- df_2019 %>%
  mutate(user_type = case_when(
    user_type == "Subscriber" ~ "member",
    user_type == "Customer"   ~ "casual",
    TRUE ~ user_type
  ))
```

Based on the timestamp fields, we will create a `trip_duration` variable in the 2020 dataset so that it can be directly compared to the 2019 dataset.

```r
df_2020 <- df_2020 %>%
  mutate(trip_duration = as.numeric(end_time - start_time))
```

Checking for errors in the `trip_duration` field (zero or negative values):

```r
df_2019 %>% filter(trip_duration <= 0)
```

```
## # A tibble: 0 x 12
## # i 12 variables: trip_id <dbl>, start_time <dttm>, end_time <dttm>,
## #   bike_id <dbl>, trip_duration <dbl>, start_station_id <dbl>,
## #   start_station_name <chr>, end_station_id <dbl>, end_station_name <chr>,
## #   user_type <chr>, gender <chr>, birth_year <dbl>
```

```r
df_2020 %>% filter(trip_duration <= 0)
```

```
## # A tibble: 210 x 14
##    trip_id          rideable_type start_time          end_time
##    <chr>            <chr>         <dttm>              <dttm>
##  1 23EF1DCC9FCA40BA docked_bike   2020-02-28 11:34:40 2020-02-28 11:34:40
##  2 9461DFF13D8BA8AD docked_bike   2020-02-28 10:09:43 2020-02-28 10:09:42
##  3 86163D9676BBBE62 docked_bike   2020-02-26 14:41:16 2020-02-26 14:41:16
##  4 836931C569802344 docked_bike   2020-02-27 09:56:47 2020-02-27 09:56:47
##  5 07CD3CBC94106B37 docked_bike   2020-02-28 10:02:30 2020-02-28 10:02:30
##  6 83D849E5C5716FA3 docked_bike   2020-02-28 10:39:01 2020-02-28 10:39:01
##  7 4BF5C10795152574 docked_bike   2020-02-26 15:11:49 2020-02-26 15:11:49
##  8 6EB2E392C75D5246 docked_bike   2020-02-26 12:49:59 2020-02-26 12:49:59
##  9 8B167ABFC026622D docked_bike   2020-02-26 12:50:52 2020-02-26 12:50:52
## 10 4CCD45F6BA577FF3 docked_bike   2020-02-26 15:06:47 2020-02-26 15:06:47
## # i 200 more rows
## # i 10 more variables: start_station_name <chr>, start_station_id <dbl>,
## #   end_station_name <chr>, end_station_id <dbl>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, user_type <chr>,
## #   trip_duration <dbl>
```

We found 210 rows with zero or negative values in the `trip_duration` column in the 2020 dataset. These entries are invalid and must be filtered out:

```r
df_2020 <- df_2020 %>%
  filter(trip_duration > 0)
```

We also identified rows where the start or end station name is listed as "HQ QR." We assume these entries represent internal bike movements by the company rather than customer trips. Therefore, these rows should be filtered out as well:

```r
df_2019 %>% filter(start_station_name == "HQ QR" | end_station_name == "HQ QR")
```

```
## # A tibble: 0 x 12
## # i 12 variables: trip_id <dbl>, start_time <dttm>, end_time <dttm>,
## #   bike_id <dbl>, trip_duration <dbl>, start_station_id <dbl>,
## #   start_station_name <chr>, end_station_id <dbl>, end_station_name <chr>,
## #   user_type <chr>, gender <chr>, birth_year <dbl>
```

```r
df_2020 %>% filter(start_station_name == "HQ QR" | end_station_name == "HQ QR")
```

```
## # A tibble: 3,558 x 14
##    trip_id          rideable_type start_time          end_time
##    <chr>            <chr>         <dttm>              <dttm>
##  1 83A921BEF3BE183B docked_bike   2020-02-27 11:20:16 2020-02-27 11:20:18
##  2 640B93AEBA2725D2 docked_bike   2020-02-27 11:20:39 2020-02-27 11:20:41
##  3 3485EA9EB52C8270 docked_bike   2020-02-27 10:04:20 2020-02-27 10:04:23
##  4 7926327328D7C62F docked_bike   2020-02-27 10:04:57 2020-02-27 10:05:00
##  5 7FACDA7C9B5863DE docked_bike   2020-02-27 10:04:36 2020-02-27 10:04:39
##  6 9258A6281AFF4107 docked_bike   2020-02-26 15:29:24 2020-02-26 15:29:26
##  7 64A2FE6DA75AEB68 docked_bike   2020-02-27 10:49:41 2020-02-27 10:49:43
##  8 4D95E87C66E0275E docked_bike   2020-02-27 10:49:11 2020-02-27 10:49:15
##  9 6EE351862E5A5EEB docked_bike   2020-02-27 10:48:42 2020-02-27 10:48:45
## 10 CF7AAF783C578ED6 docked_bike   2020-02-26 13:00:24 2020-02-26 13:00:26
## # i 3,548 more rows
## # i 10 more variables: start_station_name <chr>, start_station_id <dbl>,
## #   end_station_name <chr>, end_station_id <dbl>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, user_type <chr>,
## #   trip_duration <dbl>
```

```r
df_2020 <- df_2020 %>%
  filter(start_station_name != "HQ QR" & end_station_name != "HQ QR")
```

In this step, the datasets were checked for errors and standardized to ensure consistency. Timestamp fields were converted to proper datetime formats, column names were aligned, rider types were recoded to a common format. A `trip_duration` variable was also created for the 2020 data to match the 2019 dataset, and rows with zero or negative trip durations were filtered out. Additionally, records associated with internal bike movements ("HQ QR") were filtered out to ensure that only customer trips remained. These transformations prepare the data for accurate comparison and analysis.

4 Data cleaning

Checking for missing values:

```r
colSums(is.na(df_2019))
```

```
##            trip_id         start_time           end_time            bike_id
##                  0                  0                  0                  0
##      trip_duration   start_station_id start_station_name     end_station_id
##                  0                  0                  0                  0
##   end_station_name          user_type             gender         birth_year
##                  0                  0              19711              18023
```

```r
colSums(is.na(df_2020))
```

```
##            trip_id      rideable_type         start_time           end_time
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0                  0                  0
##          user_type      trip_duration
##                  0                  0
```

```r
colMeans(is.na(df_2019))
```

```
##            trip_id         start_time           end_time            bike_id
##         0.00000000         0.00000000         0.00000000         0.00000000
##      trip_duration   start_station_id start_station_name     end_station_id
##         0.00000000         0.00000000         0.00000000         0.00000000
##   end_station_name          user_type             gender         birth_year
##         0.00000000         0.00000000         0.05399253         0.04936875
```

```r
colMeans(is.na(df_2020))
```

```
##            trip_id      rideable_type         start_time           end_time
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0                  0                  0
##          user_type      trip_duration
##                  0                  0
```

In the 2019 dataset, there are missing values in two columns: `gender` (about 5%) and `birth_year` (about 5%). We cannot reliably replace these missing values without additional information from the dataset creators, so we will leave them as they are.

Checking for duplicates:

```
sum(duplicated(df_2019))
```

```
## [1] 0
```

```
sum(duplicated(df_2020))
```

```
## [1] 0
```

No exact duplicates were found in the data. Now let's check for non-exact duplicates:

```
df_2019 %>% filter(duplicated(trip_id))
```

```
## # A tibble: 0 x 12
## # i 12 variables: trip_id <dbl>, start_time <dttm>, end_time <dttm>,
## #   bike_id <dbl>, trip_duration <dbl>, start_station_id <dbl>,
## #   start_station_name <chr>, end_station_id <dbl>, end_station_name <chr>,
## #   user_type <chr>, gender <chr>, birth_year <dbl>
```

```
df_2020 %>% filter(duplicated(trip_id))
```

```
## # A tibble: 0 x 14
## # i 14 variables: trip_id <chr>, rideable_type <chr>, start_time <dttm>,
## #   end_time <dttm>, start_station_name <chr>, start_station_id <dbl>,
## #   end_station_name <chr>, end_station_id <dbl>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, user_type <chr>,
## #   trip_duration <dbl>
```

We confirmed that trip identifiers are unique in both datasets.

Normalizing the station name fields by converting them to lowercase and removing extra whitespace to ensure that formatting differences do not produce false duplicates:

```
df_2019 <- df_2019 %>%
  mutate(across(c(start_station_name, end_station_name),
                ~ tolower(trimws(.))))
df_2020 <- df_2020 %>%
  mutate(across(c(start_station_name, end_station_name),
                ~ tolower(trimws(.))))
```

```r
sum(duplicated(df_2019))
```

```
## [1] 0
```

```r
sum(duplicated(df_2020))
```

```
## [1] 0
```

After normalizing the station name columns, no duplicates were found.

In this step, the datasets were examined for missing values and duplicates to ensure data quality. The 2019 dataset contained missing values only in the gender and birth_year fields, which could not be reliably imputed and were therefore left unchanged. No exact or non-exact duplicates were found, and station name fields were standardized to prevent false duplicates. The cleaned datasets are now ready for reliable analysis.

5 Data analysis

Merging the datasets for analysis:

```r
df_2019 <- df_2019 %>% mutate(trip_id = as.character(trip_id))
df_2020 <- df_2020 %>% mutate(trip_id = as.character(trip_id))
```

```r
df <- bind_rows(df_2019, df_2020)
```

```r
glimpse(df)
```

```
## Rows: 788,188
## Columns: 17
## $ trip_id            <chr> "21742443", "21742444", "21742445", "21742446", "21~
## $ start_time         <dttm> 2019-01-01 00:04:37, 2019-01-01 00:08:13, 2019-01-~
## $ end_time           <dttm> 2019-01-01 00:11:07, 2019-01-01 00:15:34, 2019-01-~
## $ bike_id            <dbl> 2167, 4386, 1524, 252, 1170, 2437, 2708, 2796, 6205~
## $ trip_duration      <dbl> 390, 441, 829, 1783, 364, 216, 177, 100, 1727, 336,~
## $ start_station_id   <dbl> 199, 44, 15, 123, 173, 98, 98, 211, 150, 268, 299, ~
## $ start_station_name <chr> "wabash ave & grand ave", "state st & randolph st",~
## $ end_station_id     <dbl> 84, 624, 644, 176, 35, 49, 49, 142, 148, 141, 295, ~
## $ end_station_name   <chr> "milwaukee ave & grand ave", "dearborn st & van bur~
## $ user_type          <chr> "member", "member", "member", "member", "member", "~
## $ gender             <chr> "Male", "Female", "Female", "Male", "Male", "Female~
## $ birth_year         <dbl> 1989, 1990, 1994, 1993, 1994, 1983, 1984, 1990, 199~
```

```
## $ rideable_type      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lat          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lng          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_lat            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_lng            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

```
summary(df)
```

```
##    trip_id            start_time                        end_time                    
##  Length:788188      Min.   :2019-01-01 00:04:37   Min.   :2019-01-01 00:11:07  
##  Class :character   1st Qu.:2019-02-28 13:39:55   1st Qu.:2019-02-28 13:51:43  
##  Mode  :character   Median :2020-01-07 07:59:49   Median :2020-01-07 08:10:57  
##                     Mean   :2019-08-31 14:14:22   Mean   :2019-08-31 14:34:11  
##                     3rd Qu.:2020-02-19 12:38:45   3rd Qu.:2020-02-19 12:57:02  
##                     Max.   :2020-03-31 23:51:34   Max.   :2020-05-19 20:10:34  
##                                                                                 
##     bike_id       trip_duration      start_station_id start_station_name
##  Min.   :   1     Min.   :       1   Min.   :  2.0    Length:788188     
##  1st Qu.:1777     1st Qu.:     331   1st Qu.: 77.0    Class :character  
##  Median :3489     Median :     539   Median :174.0    Mode  :character  
##  Mean   :3429     Mean   :    1189   Mean   :202.2                      
##  3rd Qu.:5157     3rd Qu.:     912   3rd Qu.:289.0                      
##  Max.   :6471     Max.   :10628400   Max.   :673.0                      
##  NA's   :423119                                                         
##  end_station_id  end_station_name    user_type            gender         
##  Min.   :  2.0   Length:788188      Length:788188      Length:788188     
##  1st Qu.: 77.0   Class :character   Class :character   Class :character  
##  Median :173.0   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :202.1                                                           
##  3rd Qu.:289.0                                                           
##  Max.   :673.0                                                           
##                                                                          
##    birth_year     rideable_type        start_lat        start_lng     
##  Min.   :1900     Length:788188      Min.   :41.74    Min.   :-87.77  
##  1st Qu.:1975     Class :character   1st Qu.:41.88    1st Qu.:-87.65  
##  Median :1985     Mode  :character   Median :41.89    Median :-87.64  
##  Mean   :1982                        Mean   :41.90    Mean   :-87.64  
##  3rd Qu.:1990                        3rd Qu.:41.92    3rd Qu.:-87.63  
##  Max.   :2003                        Max.   :42.06    Max.   :-87.55  
##  NA's   :441142                      NA's   :365069   NA's   :365069  
##     end_lat          end_lng      
##  Min.   :41.74    Min.   :-87.77  
##  1st Qu.:41.88    1st Qu.:-87.65  
##  Median :41.89    Median :-87.64  
```

```
##   Mean   :41.90    Mean   :-87.64
##   3rd Qu.:41.92    3rd Qu.:-87.63
##   Max.   :42.06    Max.   :-87.55
##   NA's   :365069   NA's   :365069
```

5.1 Trip duration and temporal patterns

We can see that the `trip_duration` column contains anomalous values. The median (539 seconds) is less than half of the mean (1189 seconds), which indicates the presence of outliers. The maximum value (10,628,400 seconds) appears to be an error.

```
p99 <- quantile(df$trip_duration, 0.99, na.rm = TRUE)
p99
```

```
##  99%
## 4641
```

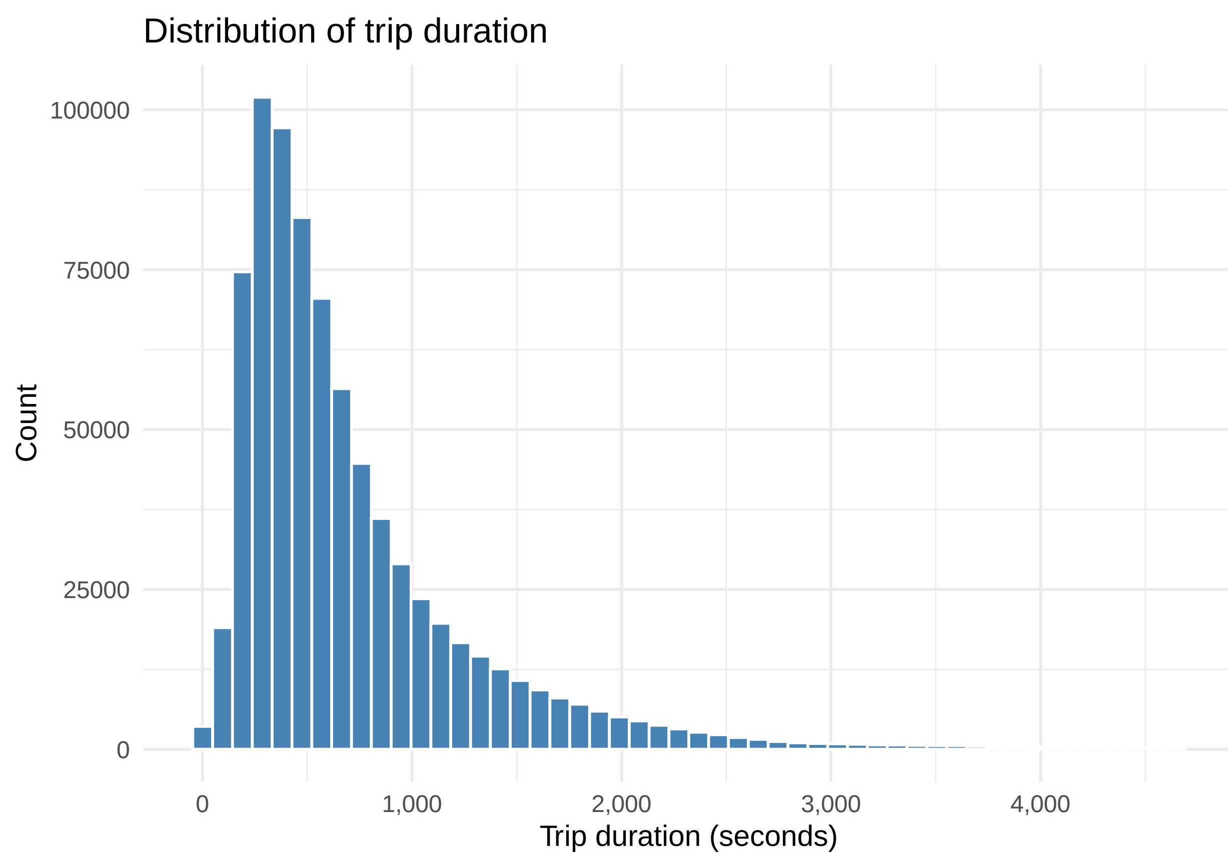
```
df %>%
  filter(trip_duration > p99)
```

```
## # A tibble: 7,874 x 17
##    trip_id  start_time          end_time            bike_id trip_duration
##    <chr>    <dttm>              <dttm>                <dbl>         <dbl>
##  1 21742549 2019-01-01 02:21:04 2019-01-02 09:35:30    2048        112466
##  2 21742597 2019-01-01 04:07:10 2019-01-02 06:37:40    3500         95430
##  3 21742765 2019-01-01 10:11:08 2019-01-01 12:29:19    1076          8291
##  4 21742783 2019-01-01 10:22:26 2019-01-02 10:08:20    1164         85554
##  5 21742906 2019-01-01 11:22:38 2019-01-01 13:28:00    3703          7522
##  6 21742908 2019-01-01 11:23:15 2019-01-01 13:18:12    2732          6897
##  7 21743016 2019-01-01 12:02:53 2019-01-01 14:39:24     441          9391
##  8 21743073 2019-01-01 12:23:28 2019-01-01 14:22:17     287          7129
##  9 21743130 2019-01-01 12:44:46 2019-01-02 09:57:16    4676         76350
## 10 21743133 2019-01-01 12:45:14 2019-01-02 07:15:36    4750         66622
## # i 7,864 more rows
## # i 12 more variables: start_station_id <dbl>, start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, user_type <chr>,
## #   gender <chr>, birth_year <dbl>, rideable_type <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>
```

We identified 7,874 rows with `trip_duration` values above the 0.99 quantile. These extreme values represent outliers and should be removed from the dataset.

```r
df <- df %>%
  filter(trip_duration <= p99)
```

```r
ggplot(df, aes(x = trip_duration)) +
  geom_histogram(bins = 50, fill = "steelblue", color = "white") +
  scale_x_continuous(labels = scales::comma) +
  labs(
    title = "Distribution of trip duration",
    x = "Trip duration (seconds)",
    y = "Count"
  ) +
  theme_minimal()
```



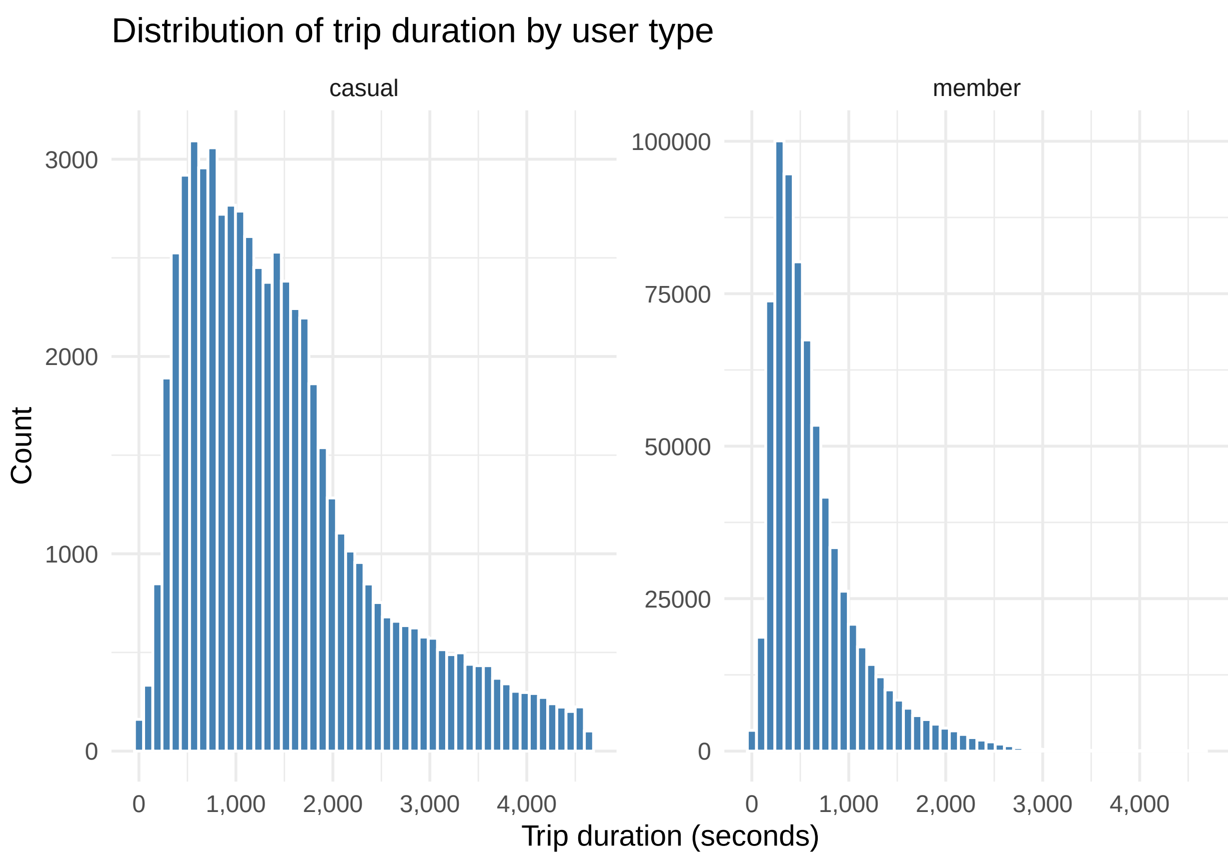

```r
summary(df$trip_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   329.0   535.0   716.4   895.0  4641.0
```

The distribution is right-skewed, indicating that most trips are relatively short (typically between 300 and 900 seconds 5 and 15 minutes), as shown by the median (535 seconds

8.9 minutes) and the third quartile (895 seconds   15 minutes). The mean (716 seconds) is higher than the median, confirming the presence of a long tail of less frequent, longer trips. Overall, the distribution shows that very long rides are uncommon and that the vast majority of trips fall well below the upper end of the range.

```
ggplot(df, aes(x = trip_duration)) +
  geom_histogram(bins = 50, fill = "steelblue", color = "white") +
  scale_x_continuous(labels = scales::comma) +
  facet_wrap(~ user_type, scales = "free_y") +
  labs(
    title = "Distribution of trip duration by user type",
    x = "Trip duration (seconds)",
    y = "Count"
  ) +
  theme_minimal()
```




```
df %>%
  group_by(user_type) %>%
  summarise(
    Min = min(trip_duration, na.rm = TRUE),
    Q1 = quantile(trip_duration, 0.25, na.rm = TRUE),
```

```
    Median = median(trip_duration, na.rm = TRUE),
    Mean = mean(trip_duration, na.rm = TRUE),
    Q3 = quantile(trip_duration, 0.75, na.rm = TRUE),
    Max = max(trip_duration, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 7
##   user_type   Min    Q1 Median  Mean    Q3   Max
##   <chr>     <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 casual        2   732   1268 1487.  1941  4641
## 2 member        1   317    507  650.   820  4641
```

The distributions show clear differences between user groups. Casual riders tend to take longer trips, while members have shorter and more consistent ride durations. The median trip duration for casual riders is 1268 seconds (~21 minutes), more than twice that of members (507 seconds ~8.5 minutes). Their upper range is also much higher: the 75th percentile for casual riders is 1941 seconds (~32 minutes), compared to 820 seconds (~13.7 minutes) for members. This suggests that casual users are more likely to ride for leisure or occasional outings, while members primarily use the service for short, routine travel.

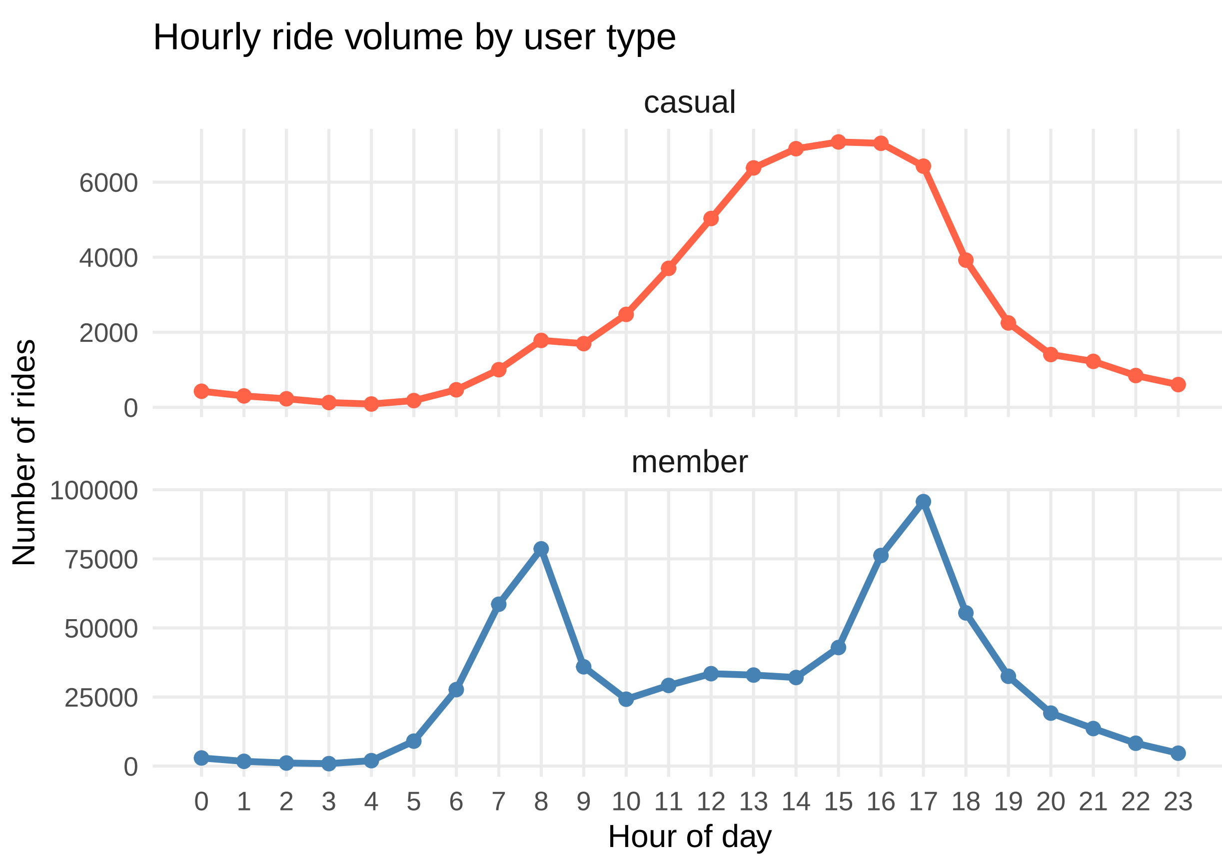
Let's create several additional columns to enable aggregation at the hourly, weekday and monthly levels. This will allow us to analyze how trip patterns change over time.

```
df <- df %>%
  mutate(
    hour = hour(start_time),
    weekday = wday(start_time, label = TRUE, abbr = TRUE),
    month = month(start_time, label = TRUE, abbr = TRUE),
    year = year(start_time),
  )
```

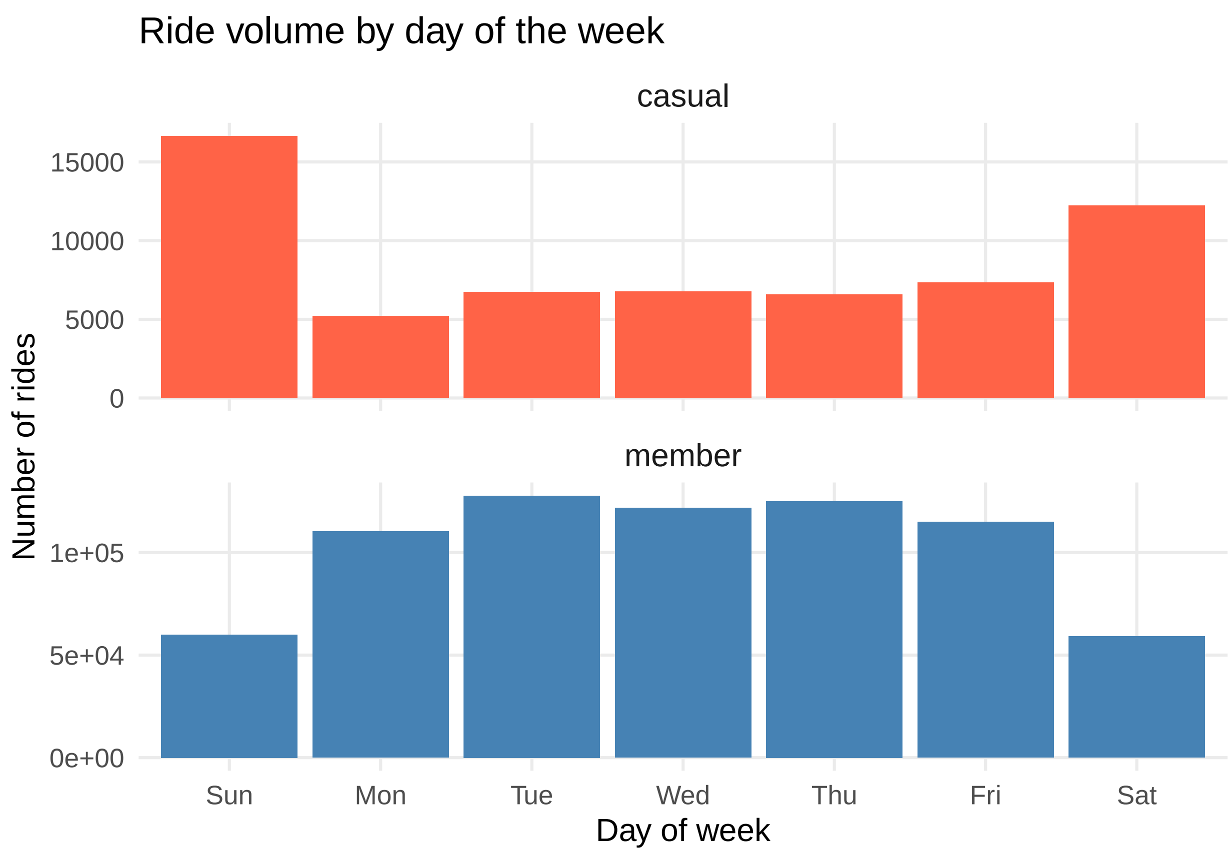
```
df %>%
  group_by(user_type, hour) %>%
  summarise(
    number_of_rides = n(),
    .groups = "drop"
  ) %>%
  ggplot(aes(x = hour, y = number_of_rides, color = user_type)) +

  geom_line(size = 1.2) +
  geom_point(size = 2) +

  facet_wrap(~ user_type, ncol = 1, scales = "free_y") +
```

```r
  scale_x_continuous(breaks = 0:23) +
  scale_color_manual(values = c("member" = "steelblue", "casual" = "tomato")) +

  labs(
    title = "Hourly ride volume by user type",
    x = "Hour of day",
    y = "Number of rides"
  ) +

  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    strip.text = element_text(size = 12),
    panel.grid.minor = element_blank()
  )
```

```
## 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.
```

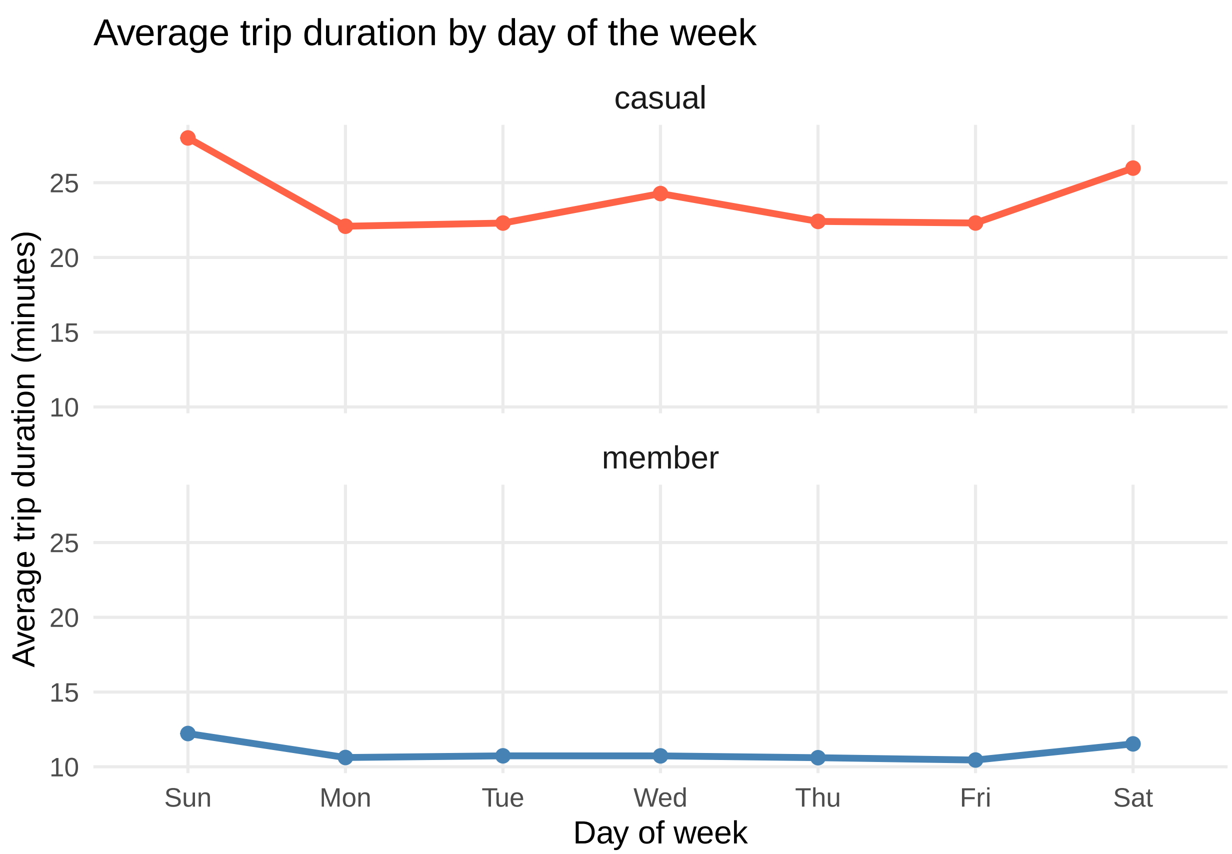


```r
df %>%
  group_by(user_type, weekday) %>%
  summarise(
    number_of_rides = n(),
    .groups = "drop"
  ) %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = user_type)) +

  geom_col() +

  facet_wrap(~ user_type, ncol = 1, scales = "free_y") +

  labs(
    title = "Ride volume by day of the week",
    x = "Day of week",
    y = "Number of rides"
  ) +

  scale_fill_manual(values = c("member" = "steelblue", "casual" = "tomato")) +

  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    strip.text = element_text(size = 12),
    panel.grid.minor = element_blank()
  )
```




```r
df %>%
  group_by(user_type, weekday) %>%
  summarise(
    avg_trip_duration_min = mean(trip_duration, na.rm = TRUE) / 60,
    .groups = "drop"
  ) %>%
  ggplot(aes(x = weekday, y = avg_trip_duration_min, group = user_type, color = user_typ

  geom_line(size = 1.2) +
  geom_point(size = 2) +

  facet_wrap(~ user_type, ncol = 1) +

  scale_color_manual(values = c("member" = "steelblue", "casual" = "tomato")) +

  labs(
    title = "Average trip duration by day of the week",
    x = "Day of week",
    y = "Average trip duration (minutes)"
  ) +
```

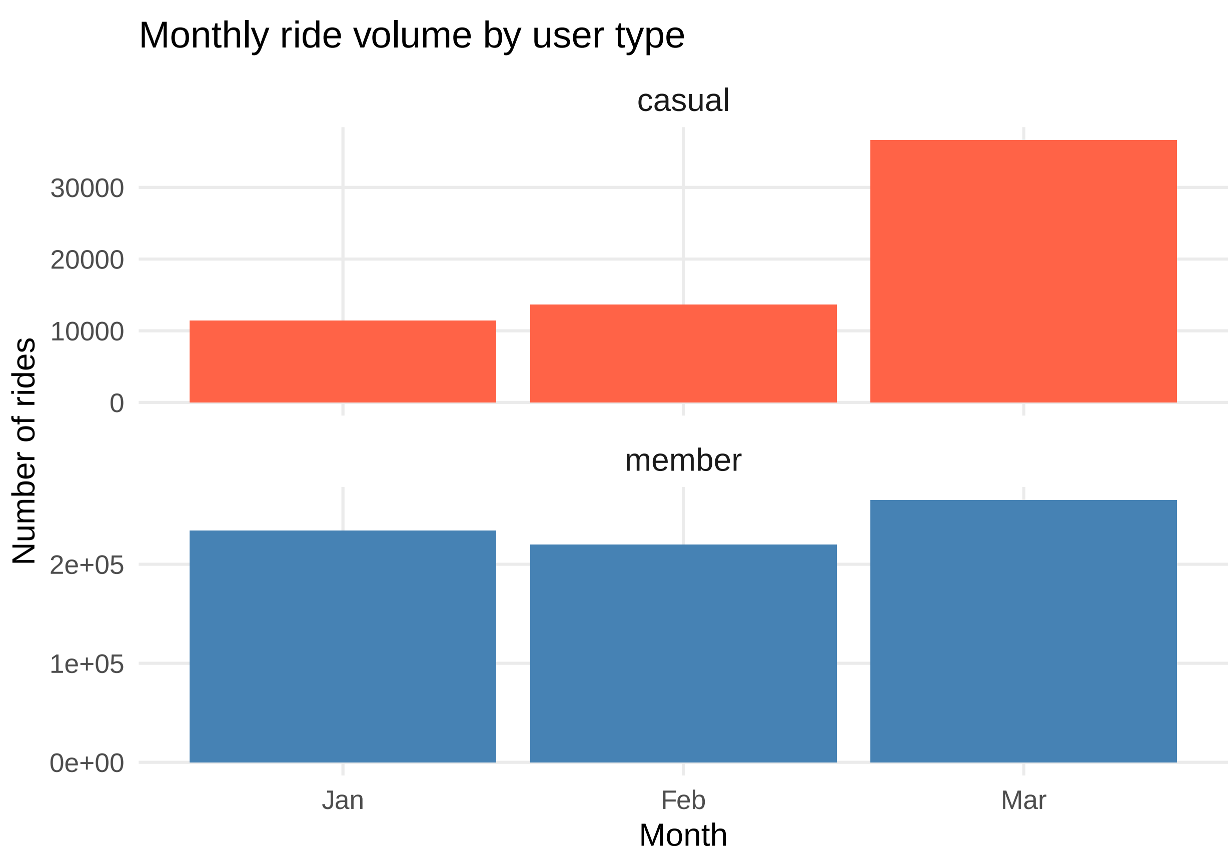
```
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    strip.text = element_text(size = 12),
    panel.grid.minor = element_blank()
  )
```



The hourly and weekday patterns show distinct usage behaviors between casual riders and members. Members show strong commuting trends, with ride peaks during typical rush hours (around 8 AM and 5 PM), the highest activity on weekdays, especially Tuesday–Thursday, and consistently short trips (about 10-13 minutes). Casual riders follow a leisure-oriented pattern: their activity gradually increases throughout the day, peaking in the afternoon (around 1 PM - 5 PM), and is highest on weekends. They also take longer trips (21–28 minutes), especially on Saturdays and Sundays.

```
df %>%
  group_by(user_type, month) %>%
  summarise(
    number_of_rides = n(),
    .groups = "drop"
  ) %>%
```

```r
ggplot(aes(x = month, y = number_of_rides, fill = user_type)) +

geom_col() +

facet_wrap(~ user_type, ncol = 1, scales = "free_y") +

labs(
  title = "Monthly ride volume by user type",
  x = "Month",
  y = "Number of rides"
) +

scale_fill_manual(values = c("member" = "steelblue", "casual" = "tomato")) +

theme_minimal(base_size = 12) +
theme(
  legend.position = "none",
  strip.text = element_text(size = 12),
  panel.grid.minor = element_blank()
)
```



We do not have enough data to analyze full seasonality, but based on the available months (January, February, and March), we can see that seasonality affects casual riders much more than members. Casual ridership rises sharply in March, more than doubling compared to January and February, while member ridership stays relatively stable across all three months. This indicates that casual users are more sensitive to weather and seasonal conditions, whereas members ride consistently. However, this pattern should be confirmed using data from the full year.

5.2 Station usage and geographic trends

```r
top_start_stations <- df %>%
  group_by(user_type, start_station_name) %>%
  summarise(rides = n(), .groups = "drop") %>%
  arrange(user_type, desc(rides))

top_start_stations %>%
  group_by(user_type) %>%
  slice_head(n = 10)
```

```
## # A tibble: 20 x 3
## # Groups:   user_type [2]
##    user_type start_station_name           rides
##    <chr>     <chr>                        <int>
##  1 casual    streeter dr & grand ave       2553
##  2 casual    lake shore dr & monroe st     2535
##  3 casual    shedd aquarium                1785
##  4 casual    millennium park               1245
##  5 casual    michigan ave & oak st          926
##  6 casual    adler planetarium              776
##  7 casual    dusable harbor                 773
##  8 casual    theater on the lake            749
##  9 casual    michigan ave & washington st   701
## 10 casual    field museum                   569
## 11 member    canal st & adams st          13787
## 12 member    clinton st & washington blvd 13417
## 13 member    clinton st & madison st      12864
## 14 member    kingsbury st & kinzie st      8707
## 15 member    columbus dr & randolph st     8499
## 16 member    canal st & madison st         7938
## 17 member    franklin st & monroe st       7004
## 18 member    michigan ave & washington st  6674
## 19 member    larrabee st & kingsbury st    6462
## 20 member    clinton st & lake st          6434
```

```r
top_end_stations <- df %>%
  group_by(user_type, end_station_name) %>%
  summarise(rides = n(), .groups = "drop") %>%
  arrange(user_type, desc(rides))

top_end_stations %>%
  group_by(user_type) %>%
  slice_head(n = 10)
```

```
## # A tibble: 20 x 3
## # Groups:   user_type [2]
##    user_type end_station_name            rides
##    <chr>     <chr>                       <int>
##  1 casual    streeter dr & grand ave      3534
##  2 casual    lake shore dr & monroe st    1957
##  3 casual    millennium park              1767
##  4 casual    shedd aquarium               1376
##  5 casual    michigan ave & oak st        1094
##  6 casual    theater on the lake           982
##  7 casual    michigan ave & washington st  810
##  8 casual    lake shore dr & north blvd    696
##  9 casual    adler planetarium             651
## 10 casual    michigan ave & lake st        527
## 11 member    canal st & adams st         14792
## 12 member    clinton st & washington blvd 14567
## 13 member    clinton st & madison st     13293
## 14 member    kingsbury st & kinzie st     8788
## 15 member    canal st & madison st        8253
## 16 member    michigan ave & washington st 7669
## 17 member    clinton st & lake st         6701
## 18 member    franklin st & monroe st      6307
## 19 member    daley center plaza           6288
## 20 member    lasalle st & jackson blvd    6231
```

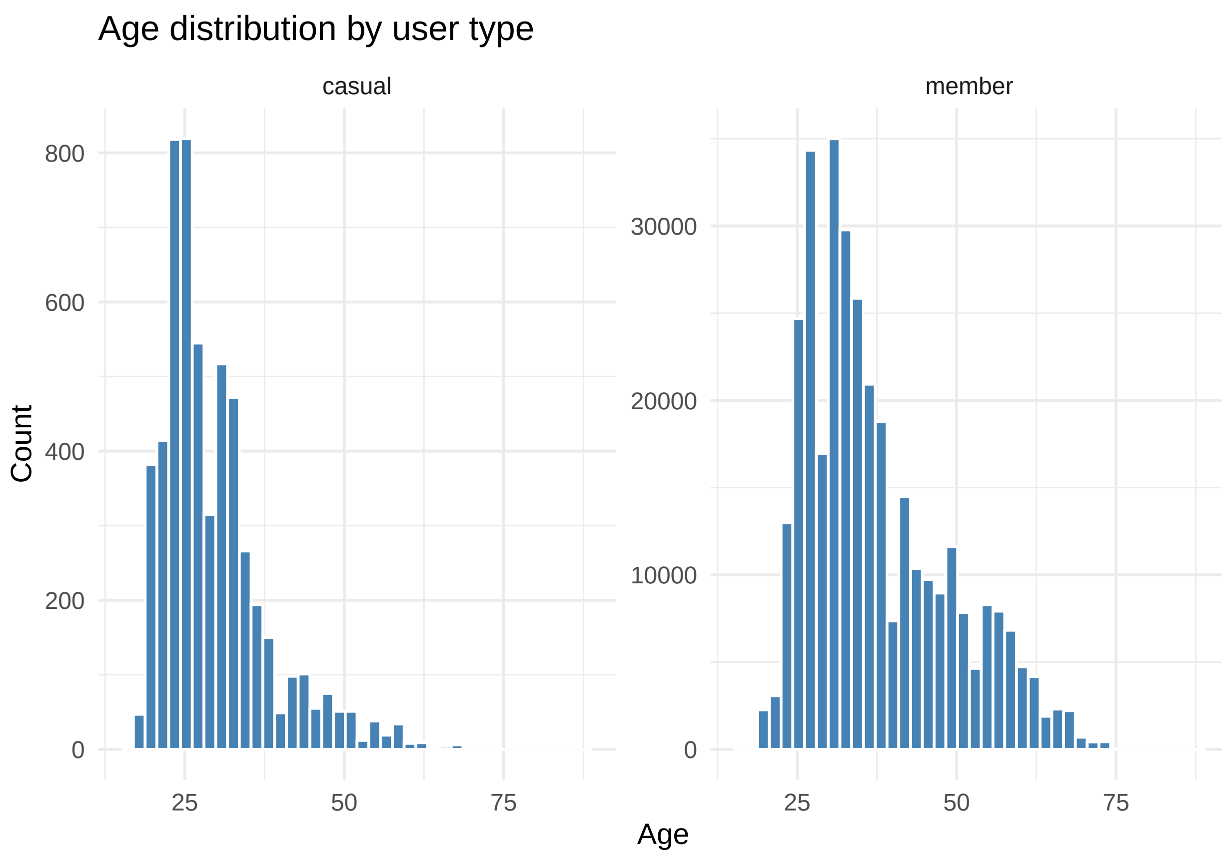
The station analysis shows a clear geographic split between user groups. Casual riders primarily start and end trips at tourist destinations such as Streeter Dr & Grand Ave, Lake Shore Dr & Monroe St, Shedd Aquarium, Millennium Park. Members, in contrast, concentrate around transit and business hubs near Union Station, such as Canal St & Adams St and Clinton St & Washington Blvd. This reinforces earlier findings: casual riders tend to use the service for leisure and recreation, while members use it for routine commuting.

5.3 Demographic profiles of riders

```
df <- df %>%
  mutate(age = year(start_time) - birth_year)

df_age <- df %>%
  filter(age <= 90)

age_dist <- df_age %>%
  group_by(user_type) %>%
  summarise(
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    count = n()
  )

age_dist
```

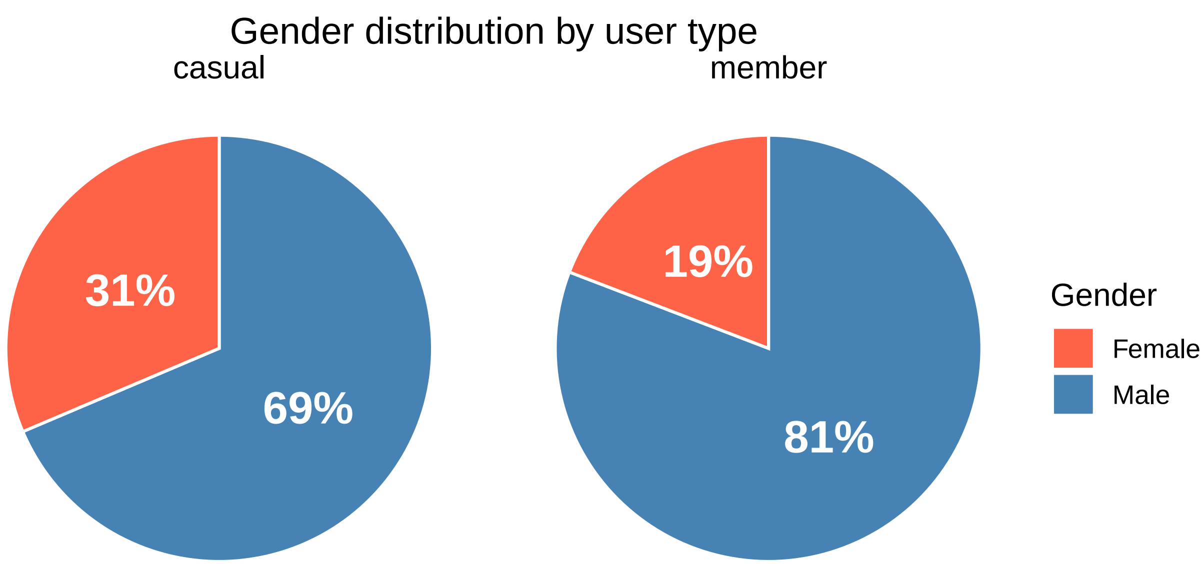
```
## # A tibble: 2 x 4
##   user_type mean_age median_age  count
##   <chr>        <dbl>      <dbl>  <int>
## 1 casual        29.5         28   5555
## 2 member        37.4         34 339895
```

```
df_age %>%
  ggplot(aes(x = age)) +
  geom_histogram(bins = 40, fill = "steelblue", color = "white") +
  facet_wrap(~ user_type, scales = "free_y") +
  labs(
    title = "Age distribution by user type",
    x = "Age",
    y = "Count"
  ) +
  theme_minimal()
```




```
gender_pie <- df %>%
  filter(!is.na(gender)) %>%
  group_by(user_type, gender) %>%
  summarise(count = n(), .groups = "drop") %>%
  group_by(user_type) %>%
  mutate(
    share = count / sum(count),
    percent_label = scales::percent(share, accuracy = 1)
  ) %>%
  arrange(user_type, desc(share)) %>%
  group_by(user_type) %>%
  mutate(ypos = cumsum(share) - 0.5 * share)

gender_pie %>%
  ggplot(aes(x = "", y = share, fill = gender)) +
  geom_col(width = 1, color = "white") +
  geom_text(aes(y = ypos, label = percent_label), color = "white", size = 6, fontface =
  coord_polar("y") +
  facet_wrap(~ user_type) +
  scale_fill_manual(values = c(
    "Male" = "steelblue",
```

```
    "Female" = "tomato"
  )) +
  labs(
    title = "Gender distribution by user type",
    fill = "Gender"
  ) +
  theme_void(base_size = 12) +
  theme(
    strip.text = element_text(size = 12),
    plot.title = element_text(hjust = 0.5)
  )
```



The demographic patterns show that casual riders are younger than members. The average casual rider is about 29 years old (median 28), while members are older, averaging 37 years (median 34). Gender distribution also differs: casual riders are more balanced (69% male, 31% female), whereas members are predominantly male (81% male, 19% female).

However, it is important to acknowledge that these findings are based solely on Q1 2019 data. Therefore, the conclusions should be validated against a more complete dataset before being considered definitive.

6 Key findings and recommendations

The analysis reveals consistent differences between casual riders and annual members across trip duration, usage patterns, geography and demographics:

- **Members** show strong commuter-driven behavior, with ride activity peaks around 8 AM and 5 PM, higher activity on weekdays, and short trips (average $\sim$ 9 minutes). Their most frequently used stations are concentrated around major transportation hubs and business districts.
- **Casual riders** follow leisure-oriented behavior, with usage peaking in the afternoon, highest activity on weekends, and longer rides (average $\sim$ 21 minutes). Their most popular stations include key tourist attractions.

Seasonality further highlights these differences: casual ridership more than doubles from winter to early spring, while member ridership remains stable. Demographically, casual riders are younger (average ~29) and more gender-balanced (69% male, 31% female), whereas members are older (average ~37) and predominantly male (81%).

It should be acknowledged that the seasonality and demographic analyses are based on limited data. Therefore, these findings should be interpreted with caution and validated using a full-year dataset before drawing definitive conclusions.

6.1 Recommendations:

Based on the analysis, Cyclistic should focus its conversion efforts on local casual riders with repeat usage while also broadening overall membership acquisition.

1. **Target frequent casual riders with flexible or seasonal membership offers** Casual riders typically take longer leisure trips, ride most often on weekends, and show seasonal growth. Cyclistic could introduce flexible or seasonal membership plans (e.g., a 3-month summer membership or weekend membership) and push in-app promotions when riders exceed a certain number of trips in a month.

2. **Promote commuting-related benefits to casual riders who ride on weekdays** Some casual riders still travel during peak weekday hours, indicating potential commuting habits. Cyclistic should identify these riders and offer commuter-focused incentives, such as discounted morning rides for the first month of membership or priority bike availability.

3. **Address the gender gap by tailoring marketing and safety-focused messaging to women** Women represent 31% of casual riders but only 19% of annual members, indicating an untapped conversion opportunity. Cyclistic could test campaigns focused on safety, well-lit stations, route recommendations, and partnerships with women-focused community groups.

4. **Expand acquisition efforts beyond the casual rider base** Since many casual riders are likely tourists and not viable membership prospects, Cyclistic should combine targeted conversions with broader outreach. This includes deploying signage, digital ads, and employer partnerships near major commuter hubs. Messaging should emphasize reliability, cost efficiency for daily travel, and exclusive member benefits.