# Google Data Analytics Capstone Project: "Cyclistic"

Milos Zubac

## Introduction

This is my approach and work on how I would solve the Google Data Analytics Certificate Capstone Project: Case Study 1. The scenario is that *Cyclistic* is a bike-share company operating in Chicago. I am a junior data analyst for them and the director of marketing wants me to figure out how to maximize the number of annual memberships. Therfore they want to convert as many casual customers as possible into annual members. I will use the approach that was taught in the Google Data Analytics Certificate which is **Ask**, **Prepare**, **Process**, **Analyze**, **Share**, and **Act**.

## Ask

The main question we want to answer is how do casual members and annual members use the bike-share differently. The assumption I made was that casual members use cyclistic mainly for leisure while annual members use it as a means of transportation. More data collection and analysis will be needed to test this theory and to see marketing opportunities.

## Prepare

In this step I prepare the data and get it ready for use. The datasets are 12 csv files that have data on 12 months of the year. We download all 12 zip files and extract them. I don't need to mind or scrape the data since they are already in csv files.

## Process

I then cleaned the data and got it more usable for analysis. I did this by using R

```
#Download the neccesary packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
```

```
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ---------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.0.5
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#Import the data files
df1 <- read_csv('202004-divvy-tripdata.csv')
```

```
## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df2 <- read_csv('202005-divvy-tripdata.csv')
```

```
## Parsed with column specification:
## cols(
```

```
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df3 <- read_csv('202006-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df4 <- read_csv('202007-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
```

```
##   member_casual = col_character()
## )

df5 <- read_csv('202008-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df6 <- read_csv('202009-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df7 <- read_csv('202010-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
```

```
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df8 <- read_csv('202011-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df9 <- read_csv('202012-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df10 <- read_csv('202101-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
```

```
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df11 <- read_csv('202102-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )

df12 <- read_csv('202103-divvy-tripdata.csv')

## Parsed with column specification:
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
```

```
##   member_casual = col_character()
## )

#join the data files together into one
bike_rides <- rbind(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)
dim(bike_rides)

## [1] 3489748      13

#remove any rows and columns that don't have values in them
bike_rides <- janitor::remove_empty(bike_rides, which = c('cols'))
bike_rides <- janitor::remove_empty(bike_rides, which = c('rows'))
dim(bike_rides)

## [1] 3489748      13

#get the ride length in mins
bike_rides$ride_length <- difftime(bike_rides$ended_at,bike_rides$started_at,
units = c("mins"))
head(bike_rides)

## # A tibble: 6 x 14
##   ride_id rideable_type started_at          ended_at
start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 A847FA~ docked_bike   2020-04-26 17:45:14 2020-04-26 18:12:03 Eckhart
Park
## 2 5405B8~ docked_bike   2020-04-17 17:08:54 2020-04-17 17:17:03 Drake Ave
& Ful~
## 3 5DD24A~ docked_bike   2020-04-01 17:54:13 2020-04-01 18:08:36 McClurg Ct
& Er~
## 4 2A59BB~ docked_bike   2020-04-07 12:50:19 2020-04-07 13:02:31 California
Ave ~
## 5 27AD30~ docked_bike   2020-04-18 10:22:59 2020-04-18 11:15:54 Rush St &
Hubba~
## 6 356216~ docked_bike   2020-04-30 17:55:47 2020-04-30 18:01:11 Mies van
der Ro~
## # ... with 9 more variables: start_station_id <chr>, end_station_name
<chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>, ride_length <drtn>

#get the start time in hours
hour_ride <- hour(bike_rides$started_at)
bike_rides <- mutate(bike_rides, ride_length = hour_ride)

#what day of the week the start time was
day_ride <- wday(bike_rides$started_at, TRUE)
bike_rides <- mutate(bike_rides, start_day = day_ride)

#what month the ride was in
```

```
month_ride <- month(bike_rides$started_at)
bike_rides <- mutate(bike_rides, start_month = month_ride)
head(bike_rides)
```

```
## # A tibble: 6 x 16
##   ride_id rideable_type started_at          ended_at
start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 A847FA~ docked_bike   2020-04-26 17:45:14 2020-04-26 18:12:03 Eckhart
Park
## 2 5405B8~ docked_bike   2020-04-17 17:08:54 2020-04-17 17:17:03 Drake Ave
& Ful~
## 3 5DD24A~ docked_bike   2020-04-01 17:54:13 2020-04-01 18:08:36 McClurg Ct
& Er~
## 4 2A59BB~ docked_bike   2020-04-07 12:50:19 2020-04-07 13:02:31 California
Ave ~
## 5 27AD30~ docked_bike   2020-04-18 10:22:59 2020-04-18 11:15:54 Rush St &
Hubba~
## 6 356216~ docked_bike   2020-04-30 17:55:47 2020-04-30 18:01:11 Mies van
der Ro~
## # ... with 11 more variables: start_station_id <chr>, end_station_name
<chr>,
##   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
##   end_lng <dbl>, member_casual <chr>, ride_length <int>, start_day
<ord>,
##   start_month <dbl>
```

```
#get rid of values for which duration is negative
bike_rides <- filter(bike_rides, ride_length > 0) %>% drop_na()
dim(bike_rides)
```

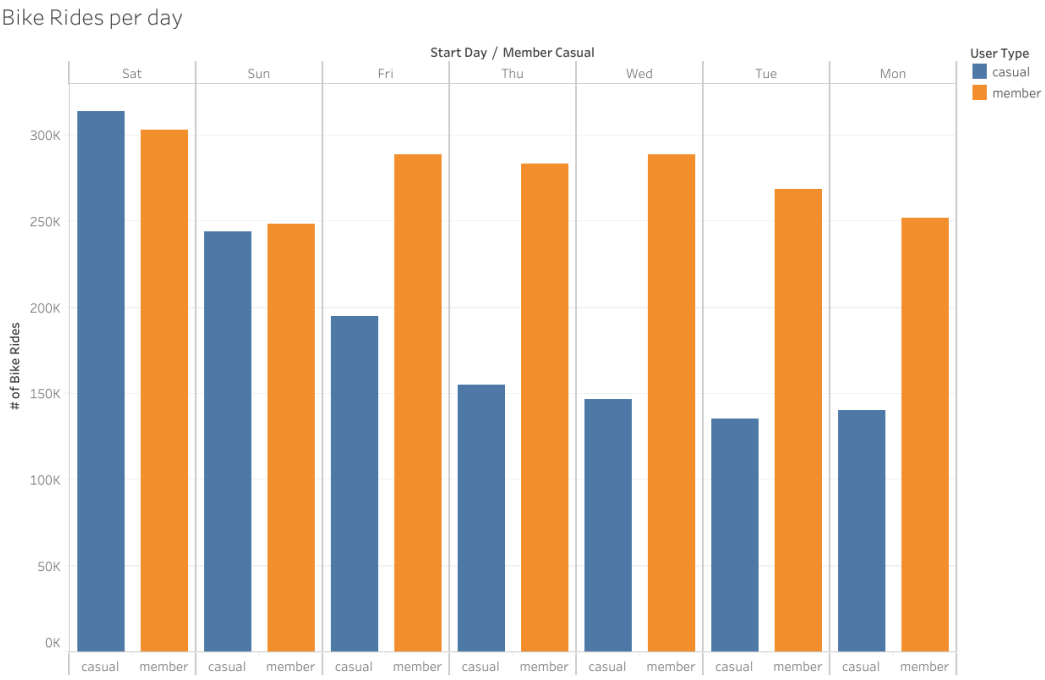
```
## [1] 3262879      16
```

```
#export the data
write.csv(bike_rides, "bike_rides.csv")
```

## Analyze

For the analysis I made my graphs using Tableau.

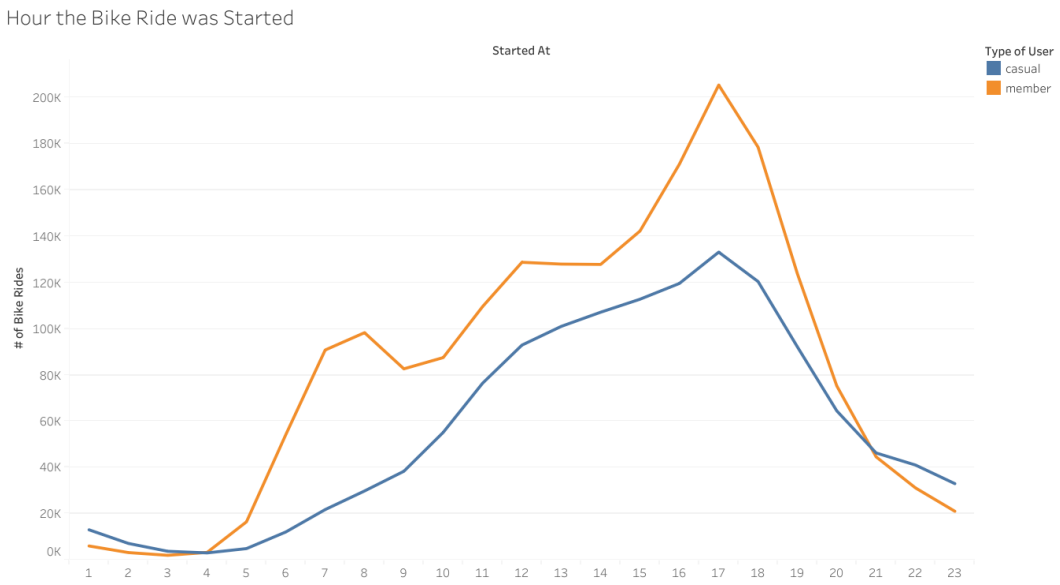
Now I want to see if I can try to conclude my theory that casual members use cyclistic for leisure while members use it for transportation First I looked at what days casual members and annual members used the bikes the most to see if there was a difference.




*Analyzed what day of the week each user used the bikes*

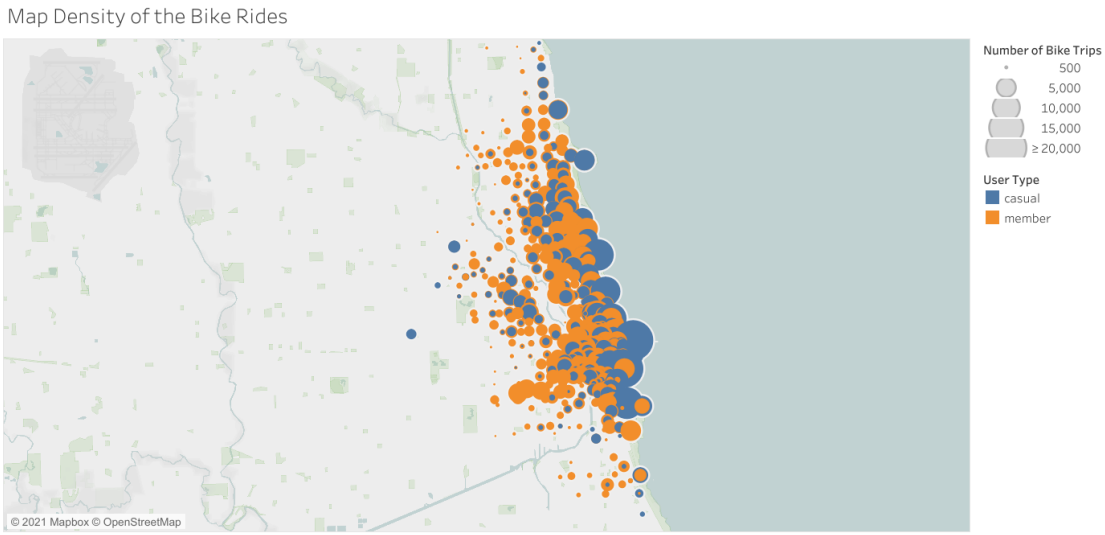
As you can see members use the bikes much more than casuals do during weekdays, but on weekends it is just about even. I think this can be attributed to the fact that most people have weekends off and thus use that time for leisure while members don't particularly favor any day over the others which can be attributed to the fact they use the bikes everyday for transportation.

To further try to prove this I went and looked at what hours the bikes were being used.



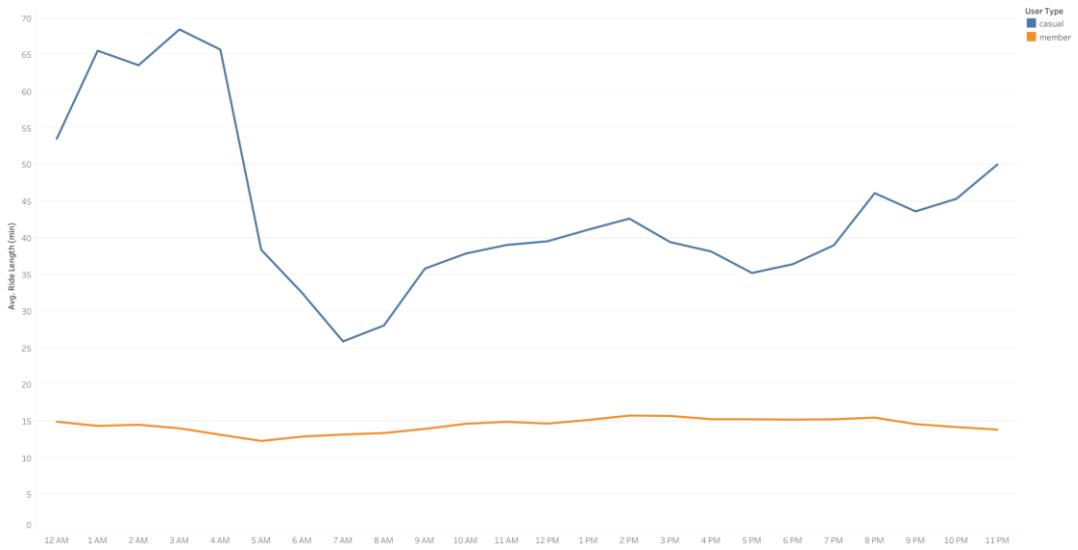

*Hours that Users Ride the Bikes*

Members seem to use the bikes a lot more than casuals during rush hours (6am-10am and 4pm-6pm), this is when most people are commuting to and from work thus further showing a correlation for members using the bikes for transportation. There also seems to be another spike for members at around the typical lunch time at noon which can also be attributed to people using the bikes as transportation to get lunch. This trend seems to be followed by casuals as well but to a lesser extent.



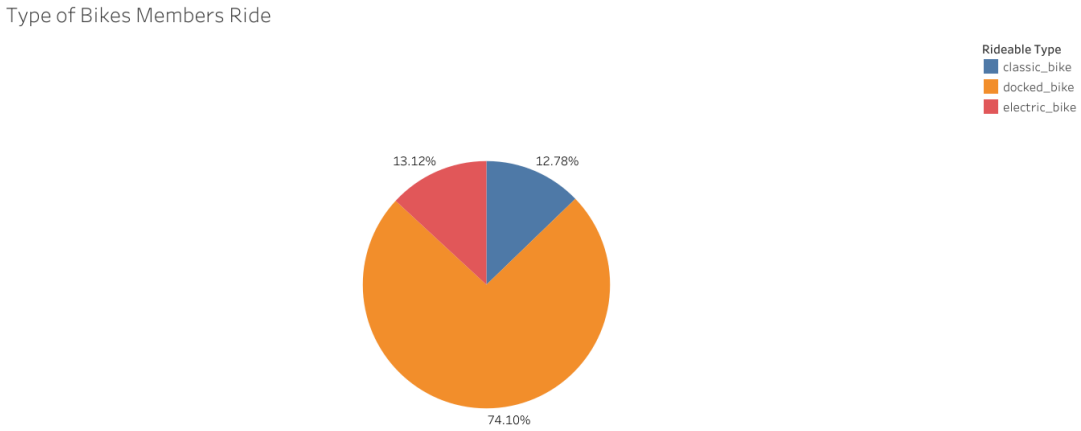

*Map Density for each type of User*

In this we examine where each user group uses the bikes. This is a map of a part of Chicago near a coastline. It seems that casuals are using the bikes more near the coastline where when you zoom in you can see is full of parks and bike trails among other kinds of leisurely attractions, while members are more spread out and in parts where there are lots of residential buildings. There isn't strong proof from this map since there are still lots of members around the coastline as well, but it does add to our theory.



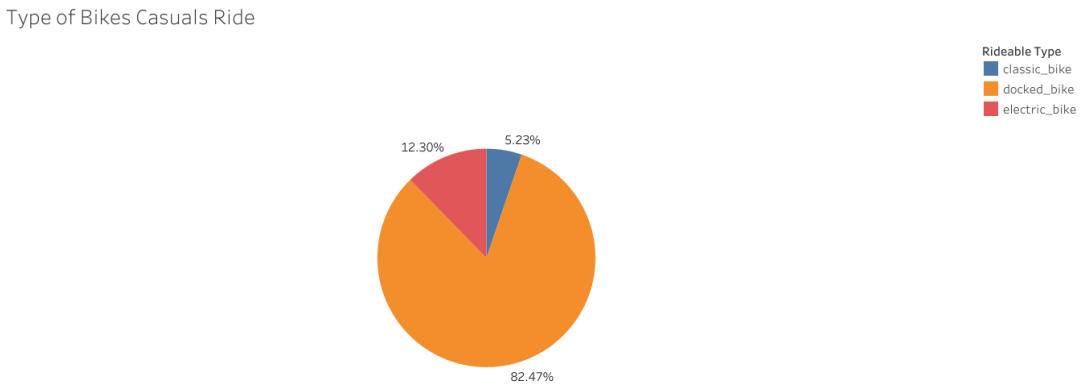

*Ride Length for each Type of User*

Casual members ride lengths are on average 2.7 times the length of a members. We aren't certain why it's taking them so long to go from point A to B I am assuming that since they are using a one time pass they want to make the most out of it, while members seem to just go from point A to B right away which is an indication they are using the bikes for transportation.




*Types of Bikes Members Use*

Members use docked bikes the most at 74.1% while electric bikes and classic bikes are around the same at 13.12% and 12.78% respectively.




*Types of Bikes Casuals Use*

Both casual users and members have the same preference when it comes to bike type. They both prefer docked bikes the most although members use electric bikes and classic bikes at almost the same rate whereas casuals use electric bikes a lot more than classic bikes at over double the percentage.

## Share

After looking at the findings from our analysis it's time to answer the question of how to convert casual users into members. Although our findings do seem to support our theory

that casuals use Cyclistic for leisure and there doesn't seem to be any evidence to support the contrary, the evidence is still inconclusive and we can't say with certainty that this is the case thus moving forward with the marketing strategy is risky. Having data on the average users' demographics and values like their age, annual income, and BMI would be helpful with making our findings conclusive.

Our recommendation for converting users would be to have a weekend member pass. Since our findings show that the typical casual user uses the bikes mostly on the weekends having a yearly membership that provides unlimited uses on the weekends could entice casual users who can't economically justify buying a membership that they won't get the full use of and helps solve the issue of repeatedly having to buy single use passes. We can also make having a membership come with privileges that casuals don't have. In the same way that first class has more privileges for flying than somebody flying in economy class having added bonuses for members could entice casuals to convert to having a membership. One idea is letting people with memberships have their choice of the type of bike they want to use whereas casuals get the leftover bikes that the members haven't taken.