# Ask

There are two segments of users belonging to the bike share company, Cyclist, which is based in Chicago.

1. Casual riders purchase single-ride or full-day passes.
2. Annual riders purchase annual mermberships.

According to Cyclistic’s finance analysts annual membership is more profitable. That is why the Lily Moreno, my manager and the director of marketing, would like to convert casual riders into members. In order to achieve that I am given a task by Mrs. Moreno to understand how casual riders and annual members use Cyclistic bikes differently. If Cyclistic executives approve my recommendations they will design a new marketing strategy using these insights to convert casual members into annual members. However my recommendations must be backed up with compelling data insights and professional data visualizations in order for them to use it.

# Prepare

As the data source I will be using Cyclistic’s historical trip data from the last 12 months to analyze and identify trends. The data has been made available by Bikeshare.

The data has been organized as a wide format because every observed cycling operation corresponds to each single line. Wide format is more commonly used than long format in data analysis.

After downloading the CSV file and uploading it to the Postit Cloud I entered these commands in order to import this CSV file and assign a data frame variable to it:

library(readr)

Divvy\_Trips\_2020\_Q1 <- read\_csv("Divvy\_Trips\_2020\_Q1.csv")

In the CSV file of “Divvy\_Trips\_2020\_Q1.csv” there are 13 column names:

[1] "ride\_id", [2] "rideable\_type", [3] "started\_at", [4] "ended\_at", [5] "start\_station\_name", [6] "start\_station\_id", [7] "end\_station\_name", [8] "end\_station\_id", [9] "start\_lat", [10] "start\_lng", [11] "end\_lat", [12] "end\_lng", [13] "member\_casual"

The types of these columns belonging to the Divvy\_Trips\_2020\_Q1 datasets appear as follows:

.. 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()

.. )

By using the summary() function we see the summary statistics such as min, max, median, mean and quarter for each of these columns. There are 426,887 entries (rides) in total. The time range occurs from January to March 2020. Actually the max value for ended\_at column happened in May 2020, but that is likely to be an outlier or data error. There are two rider types as can be seen by using the unique() function, which are “member” and “casual” users. There is also one missing value in the variable end\_station\_id which is stated as “NA”.

In order to clean data I used the clean\_names() function on Posit Cloud. It automatically standardized the column names and ensured that they are only made of letters, numbers and underscores.

ROCCC principles are known as data being reliable, original, comprehensive, current and cited: Lyft Bikes and Scooters, LLC (“Bikeshare”) acquired Motivate International Inc. in 2018 and operates the City of Chicago’s (“City”) Divvy bicycle sharing service currently**.** Since this dataset is made available by BikeshareI can call it reliable. However because of the disclaimer commanding us to use the data at our own risk this limits the reliability of the data. The data is owned by the City of Chicago, and Bikeshare is authorized to distribute it, which ensures the originality of the data and not being a third-party, while also being properly cited. However the Data License Agreement does not define how complete or comprehensive the dataset is. Also there is no commitment to provide updated data. Users are allowed to include the data in analyses of non-commercial purposes.

Data-privacy issues prohibit me from using riders’ personally identifiable information which means that I won’t be able to connect pass purchases to credit card numbers to

determine if casual riders live in the Cyclistic service area or if they have purchased multiplesingle passes. This limits some details but is enough to find trends.

In order to export the CSV file on R I entered the command:

write.csv(Divvy\_Trips\_2020\_Q1, "Divvy\_Trips\_2020\_Q1.csv", row.names = FALSE).

Next I downloaded the CSV file which appeared on the files pane at the bottom right corner of Postit Cloud.

# Process

Here are the tools I will be using and the reasons why I would choose them:

* Google Sheets: It’s simple and works well for this task. At the first stage I will be using it to add new columns to the dataset easily.
* Posit Cloud: It is the defacto tool to work with big datasets and I’ll be using it to to clean and analyze datasets.

My data was obtained by Bikeshare. However I will be transforming and checking the information to ensure integrity.

Then, I opened the CSV file (which I had exported on Postit Cloud) on Google Sheets and created a column named ride\_length. I Calculated the length of each ride by subtracting the column started\_at from the column ended\_at and formatted the output as HH:MM:SS using Format > Number > Duration. Then I created another column called day\_of\_week, and calculated the day of the week that each ride started using the WEEKDAY command (e.g., =WEEKDAY(C2,1)) in each file. The numbers represent the days of the week starting from 1 which is Sunday to 7 which is Saturday.

The initial average value for the ride\_length column that I just created was 00:22:07 and I learnt this value from the bottom right dropdown menu on Google Sheets bar after clicking the ride\_length column. But this column also had some unexpected numbers exceeding more than one day. That is why I filtered out the rides with more than 24 hours if the rides belonged to the casual users. I found 227 records as such and deleted them all. There was also another issue: Some rides took negative time which did not make sense at all. I filtered them and deleted 210 further records

(*Warning*: Here I should also have erased the 6 empty cells all belonging to members, that is why the average ride length for members on pivot table that I added the Analyze section is slightly greater than the average value that I calculated on R Posit Cloud in the Share section).

The missing value for the end\_station\_id variable was gone when I filtered the variable.

There was also another issue with the table: The maximum value for ended\_at column happened in May 2020, while the max value for started\_at column was March 2020. To further investigate this issue I sorted the started\_at column after which I saw that the latest record was on 2020-03-31. That is why I filtered out ended\_at variable which had 3 values later than 2020-04-02. found and deleted these 3 records.

The final average value for the ride\_length column does now stand at 0:15:00.

All in all I am now left with 426447 records.

# Analyze

Column headings were standardized and all data in the same row column was checked to be of the same type. I also tried to find possible duplicates by data > data cleanup > remove duplicates, but there was no record like this.

The average time for each ride was found to be 15 minutes (0:15:00). The maximum value for the ride\_length was 715:56:14 belonging to members. This number may seem quite high but since I deleted casual riders with ride\_length values greater than 24 hours (24:00:00), the remaining greater time periods belong to members. And I did not delete these numbers as I think members might just leave the bikes at their homes for a long while instead of bringing them back to the docking stations. Finally, the most frequent travel day was found to be Wednesday (number 4).

Here are 3 pivot tables:

| casual | 0:37:02 |
| --- | --- |
| member | 0:12:13 |
| **Grand Total** | **0:15:00** |

***Fİgure 1.1.*** *The average ride\_length for members and casual riders*

| *AVERAGE of ride\_length* | *day\_of\_week* |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *member\_casual* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Grand Total |
| casual | 0:42:17 | 0:27:31 | 0:30:38 | 0:39:07 | 0:31:27 | 0:35:43 | 0:40:03 | 0:37:02 |
| member | 0:15:49 | 0:11:28 | 0:11:29 | 0:11:40 | 0:11:33 | 0:12:37 | 0:12:52 | 0:12:13 |
| **Grand Total** | **0:23:33** | **0:12:37** | **0:12:50** | **0:13:57** | **0:13:00** | **0:14:34** | **0:18:15** | **0:15:00** |

***Fİgure 1.2.*** *The average ride\_length for users by day\_of\_week*

| *COUNTA of ride\_id* | *day\_of\_week* |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *member\_casual* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Grand Total |
| casual | 14820 | 4785 | 5225 | 5846 | 4819 | 5103 | 7445 | 48043 |
| member | 35964 | 61922 | 69696 | 63978 | 61245 | 55496 | 30103 | 378404 |
| **Grand Total** | **50784** | **66707** | **74921** | **69824** | **66064** | **60599** | **37548** | **426447** |

***Fİgure 1.3.*** *The number of rides for users by day\_of\_week*

In order to visualize data comfortably on Postit Cloud I downloaded the Google Sheets file as a CSV file on Google Sheets. I again imported this CSV file and assigned a data frame variable to it:

Divvy\_Trips\_2020\_Q1 <- read\_csv("Divvy\_Trips\_2020\_Q1.csv")

# Share

First of all I imported the ggplot2 package for visualization and the dplyr package for calculating the average ride length:

library(ggplot2)

library(dplyr)

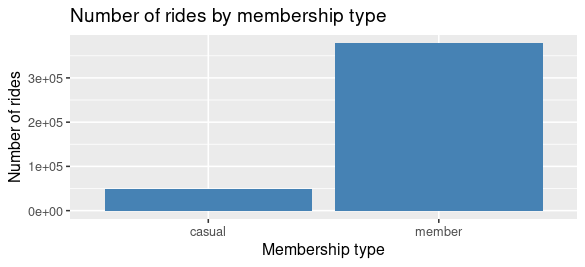
1) Number of rides by membership type:

ggplot(data\_trips, aes(x = member\_casual)) +

geom\_bar(fill = "steelblue") +

labs(title = "Number of rides by membership type",

x = "Membership type", y = "Number of rides")



Insight: As we see members have more number of rides but since the number of members (378404) heavily outweighs the number of casual riders (48043) I can not make a clear deduction out of this graph. However I can still infer that an average member drives at least 1.3 times more frequently than an average casual rider.

2) Average ride length by membership type:

When I enter the command str(data\_trips), the data type of ride\_length column already appears as col\_time() and mean() function automatically converts the time into seconds taking down the average. So the following command is not needed:

data\_trips$ride\_length <- as.difftime(data\_trips$ride\_length)

avg\_ride\_by\_usertype <- data\_trips %>%

group\_by(member\_casual) %>%

summarise(avg\_duration = mean(ride\_length, na.rm = TRUE))

View(avg\_ride\_by\_usertype)

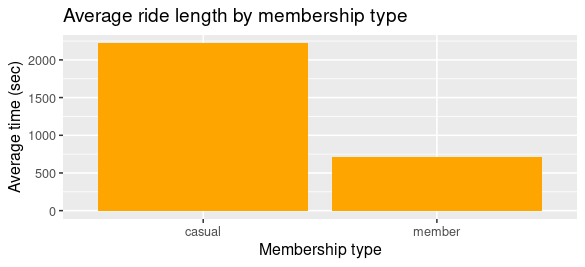
| **member\_casual** | **avg\_duration** |
| --- | --- |
| casual | 2221.6528 secs |
| member | 711.4855 secs |

ggplot(avg\_ride\_by\_usertype, aes(x=member\_casual, y=avg\_duration)) +

geom\_col(fill="orange") +

labs(title = "Average ride length by membership type",

x = "Membership type", y = "Average time (sec)")



Insight: Casual riders have a higher average time of riding their bikes than members. The ratio is more than 3 folds. Because customers who purchase full-day passes are also counted as casual riders, they probably would like to use their fares until the end of the day.

3) Number of rides by membership type and days:

Since day\_of\_week is numeric using factor() makes ggplot2 treat it as a categorical variable. Without it a numeric variable might be interpreted as continuous. Secondly factor() ensures that day\_of\_week is displayed ordered starting from 1 and ending at 7. It also allows us to control the order of levels although I am not using this feature here.

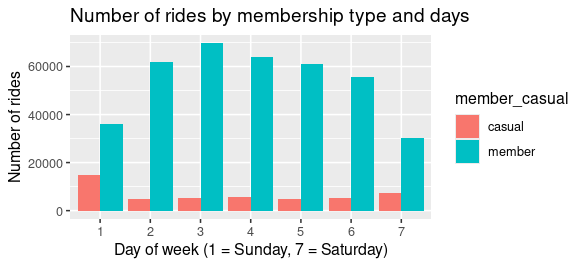
I use the position = "dodge" argument in order to show the groups side by side rather than stack them. This provides a more readable graph for comparative analyses.

ggplot(data\_trips, aes(x = factor(day\_of\_week), fill = member\_casual)) +

geom\_bar(position = "dodge") +

labs(title = "Number of rides by membership type and days",

x = "Day of week (1 = Sunday, 7 = Saturday)", y = "Number of rides")



Insight: Casual riders drive more often at the weekend whereas members drive more often during weekday. Members are probably using their bikes to serve them to their work, while casual riders use it more likely for fun.

4) Average ride length by days

avg\_ride\_by\_day <- data\_trips %>%

group\_by(day\_of\_week) %>%

summarise(avg\_duration = mean(ride\_length, na.rm = TRUE))

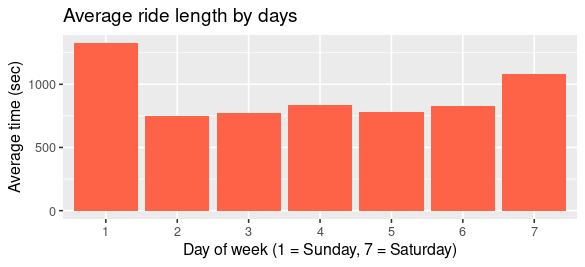
| **day\_of\_week** | **avg\_duration** |
| --- | --- |
| 1 | 1325.2592 secs |
| 2 | 747.4684 secs |
| 3 | 769.5373 secs |
| 4 | 837.4622 secs |
| 5 | 780.3368 secs |
| 6 | 831.4963 secs |
| 7 | 1084.8345 secs |

ggplot(avg\_ride\_by\_day, aes(x = factor(day\_of\_week), y = avg\_duration)) +

geom\_col(fill = "tomato") +

labs(title = "Average ride length by days",

x = "Day of week (1 = Sunday, 7 = Saturday)", y = "Average time (sec)")



Insight: Riders ride for longer periods during the weekend probably because it is their leisure time.

5) Average ride length by days and membership type

avg\_ride\_by\_day\_and\_usertype <- data\_trips %>%

group\_by(day\_of\_week, member\_casual) %>%

summarise(avg\_duration = mean(ride\_length, na.rm = TRUE))

View(avg\_ride\_by\_day\_and\_usertype)

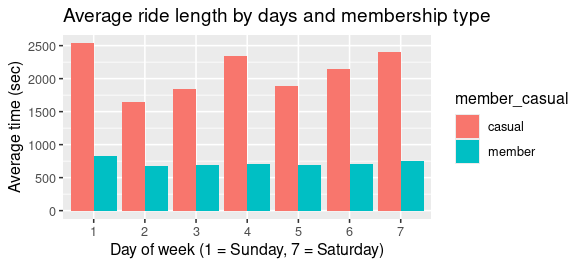
| **day\_of\_week** | **member\_casual** | **avg\_duration** |
| --- | --- | --- |
| 1 | casual | 2536.5793 secs |
| 1 | member | 826.0583 secs |
| 2 | casual | 1650.7158 secs |
| 2 | member | 677.6692 secs |
| 3 | casual | 1837.9064 secs |
| 3 | member | 689.4434 secs |
| 4 | casual | 2346.9066 secs |
| 4 | member | 699.5364 secs |
| 5 | casual | 1887.3507 secs |
| 5 | member | 693.2325 secs |
| 6 | casual | 2143.0388 secs |
| 6 | member | 710.8944 secs |
| 7 | casual | 2402.9469 secs |
| 7 | member | 758.8314 secs |

ggplot(avg\_ride\_by\_day\_and\_usertype, aes(x = factor(day\_of\_week), y = avg\_duration, fill = member\_casual)) +

geom\_col(position = "dodge") +

labs(title = "Average ride length by days and membership type",

x = "Day of week (1 = Sunday, 7 = Saturday)", y = "Average time (sec)")



Insight: Average ride length for members are at their maximum during the weekend and on Wednesday

6) Most popular top 10 stations by name of the starting station of each ride:

top10\_stations <- data\_trips %>%

filter(!is.na(start\_station\_name)) %>%

group\_by(start\_station\_name) %>%

summarise(ride\_count = n()) %>%

arrange(desc(ride\_count)) %>%

slice\_head(n = 10)

View(top10\_stations)

| **start\_station\_name** | **ride\_count** |
| --- | --- |
| Canal St & Adams St | 7813 |
| Clinton St & Madison St | 6796 |
| Clinton St & Washington Blvd | 5939 |
| Kingsbury St & Kinzie St | 4624 |
| Columbus Dr & Randolph St | 4424 |
| Franklin St & Monroe St | 3711 |
| Canal St & Madison St | 3636 |
| Clinton St & Lake St | 3579 |
| HQ QR | 3557 |
| Larrabee St & Kingsbury St | 3550 |

In the below ggplot2 visualization code, reorder() sorts the stations by number of rides in ascending order instead of alphabetical order of station names. Since I use reorder() it automatically turns stations into factor class and I do not need to call factor() again.

coord\_flip() f*lips the* x and y axesto make station names easier to read horizontally*.*theme\_minimal() applies a cleaner theme by removing unnecessary lines and background.

ggplot(top10\_stations, aes(x = reorder(start\_station\_name, ride\_count), y = ride\_count)) +

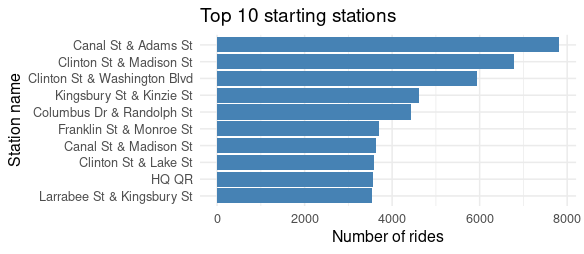
geom\_col(fill = "steelblue") +

coord\_flip() +

labs(title = "Top 10 starting stations",

x = "Station name", y = "Number of rides") +

theme\_minimal()



Insight: We see that the first 3 stations (i.e., Canal St & Adams St, Clinton St & Madison St, Clinton St & Washington Blvd) are clearly ahead regarding the number of rides, so we might focus on these stations to advertise.

My audience will be casual riders. Here are a few things we can do in order to convert them into members:

* We can target top 3 starting stations (i.e., Canal St & Adams St, Clinton St & Madison St, Clinton St & Washington Blvd) to advertise.
* In our advertisement we can offer 20% rate of discount during the 1st month in case casual riders become members.
* We can also offer loyalty programs like providing discounted membership to all members after a certain period of time in order to ensure they continue their membership.
* When pushing our campaign so we can especially focus on the days Sunday, Saturday and Wednesday, because casual riders drive most often in these 3 day both in terms of average ride length and number of rides.

# Act

I created a Github repository for the project “Case study: How does a bike-share navigate speedy success?” Then I uploaded into this repo the relevant R file and each of the PNG files that demonstrates my visualizations.