Case Study: How does a Bike-Share Navigate Speedy Success

**1.Ask-Clear Statement of the Business Task**

Cyclistic, a bike-sharing program, believes that the key for long term economic growth in their company, is by converting the casual riders to annual members. Because of this, Lily Moreno, the marketing director, is set to design marketing strategies to convert the usual riders to annual members. For these reasons, I, the data analyst, have been tasked to analyse the Cyclistic bike usage for annual members and casual riders and investigate why casual riders would convert to annual memberships. These findings would then allow me to help Cyclistic make data-driven decisions for their marketing campaign in attempt to convert more casual riders to annual members, and accomplish the business task.

To accomplish this goal, I will be analysing:

**2.Prepare- Data Sources Used**

For this analysis, the data was provided Cyclistic, and contains data about its riders 2013 to 2022. However, for this analysis, only 12 months of data from 2020 was used. The data can be accessed through a provided link by Cyclistic, where all the data is installed provided as .csv files in .zip folders, and is organized in ascending dates.

As the data was gathered by Cyclistic (for the purpose of this research), the data is internal data and therefore is safe, reliable, unbiased, and credible. The data has been made available by Cyclistic and Motivate International Inc., where customers privacy is respected since the rider identifiable information cant be accessed. This prohibits anyone connecting pass purchases to determine the credit card numbers of riders, which makes the data reliable, original, comprehensive, current, and cited which follows the “ROCCC” recommendation for a good quality data source, and meets all the data integrity requirements.

I first had a look at columns making up the dataset, and figured out what each column referred to as shown in the table below.

|  |  |
| --- | --- |
| ride\_id | Unique ID given for every ride |
| rideable\_type | Type of bicycle used during the ride |
| started\_at | Date and time ride started |
| ended\_at | Date and time ride finished |
| start\_station\_name | Station name at the start of the ride |
| start\_station\_id | Unique Start Station ID |
| end\_station\_name | Station name at the end of the ride |
| end\_station\_id | Unique End Station ID |
| start\_lat | Latitude of start station |
| start\_lng | Longitude of start station |
| end\_lat | Latitude of end station |
| end\_lng | Longitude of end station |
| member\_casual | Field indicating if the rider is a casual or an annual member |

At this initial stage I was just checking to see if the data was making sense, if there were any obvious mistakes, and if there is any large error in the unclean and disorganized data.

**3. Process - Cleaning the Data**

To clean the data, the tools used included Google Sheets, SQL, and R but mainly R for the deep cleaning and statistical analysis. For the initial cleaning, Google Sheets was used to sort and filter the data. The table below shows what columns made up the dataset.

**Google Sheets: Cleaning Phase**

The steps I took in **Google Sheets** for these initial cleaning steps are shown below.

1. Imported the csv file to a Google Sheets Workbook.
2. Remove Duplicates.
3. Trimmed Whitespace.
4. Spaced out the columns to check for spelling mistakes in the headers.
5. Checked columns for empty data and deleted rows of missing data.
6. Formatted the Date to a dd/mm/yyyy - hh/mm/ss
7. Added ride\_duration column by subtracting the end\_at from the started\_at.
8. Sorted the ride duration column in ascending order to find the shortest journeys
9. Deleted journeys with negative durations

These edits were made to ensure the integrity of the data. Primarily, I wanted to see if there was any missing data, or if there were missing columns, or spelling mistakes in the column names. Then, I wanted to edit the format of the columns and rows in a way where the data looks more aesthetic, clean, and organized. After altering the appearance of the data, I then wanted to determine if there was any data that didn’t make sense. To test this, I decided to add a column called ride\_duration that subtracted the “ended\_at” from the “started\_at” and study this column. To further study this column, I decided to sort the sheet by ascending order, and see the ride to determine if there any rides were negative or zero. This would mean that ended\_at is before the started\_at which means the data is wrong. After this I decided to delete all the negative ride\_durations as this is doesn’t make sense, but I kept the ride\_durations with a zero as it may represent the rides where users quickly changed to another bike to use, or glitches in the system.

By using the WEEKDAY function in Google Sheet, I proceeded to make a column called the “day\_of\_week” column using this function, to be able to have data on when the bicycles were checked out during every day of the week. This would allow me to be able to analyse trends on how bicycle usage would change throughout the week.

Following this step, I used the Pivot Tables on Google Sheets to be able to quickly distinguish the number of rides for different riders, and the average monthly riders and casuals. Using google sheets I was also able to observe the bicycle usage throughout the different seasons which helped identify trends during the analysis. However, this was done only for one dataset, and since I wanted a more efficient and quicker way to analyse a combination of 12 datasets, I opted for the use of RStudio.

**RStudio: Cleaning Phase**

As I was using my own computer to processes all these data sets, using large datasets on Google Sheets would freeze my computer, and cause it to overheat. So to combine, analyse, and manipulate the large amount of data sets after they were merged together, I opted for the use of RStudio as opposed to SQL.

Below is the summary of the steps taken in **R Studio** for the deep cleaning phase before entering the data manipulation and analysis phase.

1. Install and loading the necessary packages for cleaning: ‘tidyverse’, ‘janitor’, ‘lubridate’, and ‘ggplot2’.
2. Import all of the CSV files into RStudio, assigning them to corresponding variable names, and examining them to make sure everything is compatible.
3. Changed the datatype of “start\_station\_id” and “end\_station\_id” from character to integer in the “tripdata\_202012” dataset, for it to match with the other datasets.
4. Merged all of the individual datasets into one large data set called “tripdata\_2020”.
5. Checked the total number of rows of the combined dataset to the sum of all the rows in the individual datasets to ensure all of the datasets were included.
6. Examined the combined data set and decided to create 5 new columns including; a new date column in “YYYY-MM-DD” format, an abbreviated month column, a day column, a year column, and a day\_of\_week column.
7. Removed any NA cells in the data set and any duplicated rows which led to the removal of 44 duplicated rows.

**4 Analysis- Analysing the Data**

**RStudio: Data Manipulation and Analysis Phase**

After the data was cleaned and organized, using the various tools in RStudio, I was able to manipulate the data and statistically analyse the data.

Below is the summary of the steps taken in **R Studio** for the data manipulation and analysis phase.

1. Transformed the “started\_at” and “ended\_at” columns to time stamps as opposed to character string to be able to create a “ride\_length” column determining the ride duration, as well as filtered out all of the ride lengths less than 0 mins.
2. Determined the number of casual and annual members.
3. Determined the average, standard deviation, median, minimum, and maximum of the average ride length.
4. Determined the mode of the day of the week (As RStudio doesn’t have a function for finding the mode, I created one using help from tutorialspoint.com.)[1]
5. Determined the most popular months by analysing the number\_of\_rides and sorting them in a descending manner.
6. Determined the most popular station for all riders, and then for casual riders only and annual members only.

These steps allowed me to obtain the information I needed to make my analysis, perform calculations, and find the necessary trends and information to help make data driven decisions.

**Summary of the Analysis -**Casual Rider Vs Annual Member

In 2020, it was found that were fewer casual members then annual members as shown in Table 1. The casual members were shown to have longer average ride lengths as well having the longest ride lengths. For both riders the most popular day for any type of Cyclist user was Saturday, however, the annual member’s most popular day for bike usage was Wednesday. For both users, although there are other days where the average ride length is greater, the days with the most users is Saturday for the casual riders, and Wednesday for the annual members.

**Table 1** Analysis for Casual Riders vs Annual Members

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of Member** | **Number of Members** | **Average Ride Length (mins)** | **Longest Ride Length (mins)** | **Most Popular Day** | **Average Ride length on Popular Day (mins)** |
| Casual | 1278811 | 48.27 | 156450 | Saturday | 49.184 |
| Annual | 2009756 | 15.75 | 93793 | Wednesday | 14.734 |

**Analysis of the Popular Months and Stations**

In 2020, the data shows that the most popular month for all the Cylistic bike users was August with December being the least popular month having the lowest number of rides as well as the lowest ride durations as seen in Table 2.

**Table 2** Analysis for the Popular Months

|  |  |  |
| --- | --- | --- |
| **Months** | **Number of Rides** | **Length of Rides (mins)** |
| August | 605746 | 29.4548 |
| December | 23295 | 17.8963 |

In 2020, the most popular station was “Streeter Dr & Grand Ave” however, the most popular start stations for casual riders and annual members were different as seen in Table 3.

**Table 3** Analysis for the Popular Start Stations

|  |  |  |
| --- | --- | --- |
| **Type of Member** | **Popular Start Station** | **Number of Trips** |
| Casual | Clark St & Elm St | 19195 |
| Annual | Streeter Dr & Grand Ave | 25790 |

**5 Share – Sharing the Analysis**

**Analysing the Number of Members**

Firstly, looking at the analysis, it is clear that there are more annual members making up the total Cyclistic riders as seen in the pie chart below.

**Chart, bar chart

Description automatically generated**

**Analysing the Ride Durations**

Looking at the data, it is clear that the annual members, making up 61 percent of total yearly members, form a larger part of the total riders that the casual riders, which make up 39 percent of the total riders. Even if this true, however, from the table below it can be seen that the average ride lengths for the casual members is greater than the ride length for annual members which is a point that can be capitalised on by the marketing team. Since casual members are having longer average ride lengths, the marketing team can mention how people are missing out on not having the annual membership for their longer durations, and the benefits they are missing out on.

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedLooking at the bar chart below, the average ride duration throughout the week is also showing how casual riders have longer rides than annual members.

This might be because annual members may only be using the bikes to commute to work unlike the casual riders which may be using the bikes for leisure purposes, and hence having longer ride durations. This is another point which can be used by the marketing team to persuade the user using the Cyclist bikers that the annual membership is more cost effective. The marketing team can also mention in the offer how, if a user is using the bikes sharing services for more than a certain duration per week, it is beneficial to have the annual membership.

**Analysing the Number of Rides**

To further analyse this point a bar chart on the number of trips for both casual riders and members was made.

Chart, bar chart

Description automatically generated

It is clear that throughout the week, the rides for the casual riders not as high as it is during the weekends, whereas for the annual members, the rides were fairly consistent throughout the week with the lowest number of rides being on Sunday. This is probably because the causal riders are probably using the Cyclitic bikes for leisure activities throughout the weekend whereas, the annual members are using it for work. For this reason, to convert more causal riders to annual members, the marketing team can promote an offer where the casual riders can benefit by having cheaper fares on the weekend if they were to convert to the annual membership.

**Analysing the Monthly Bike Usage**

Analysing the month usage demonstrates the casual and annual member usage throughout the year as seen in the bar chart below.

Chart, bar chart, histogram

Description automatically generated

It is clear that as the months get colder, there is more of a change in the casual rider monthly average duration rather than the monthly member usage. The annual member Cyclitic average ride duration varies throughout the year yet is still fairly consistent month to month. This might be due to the fact that the average commutes to work don’t vary that much, whereas the casual rider average duration drastically changes may be occurring due to the weather.

Chart, bar chart

Description automatically generatedTo study this a further, barchart was made comparing the number of rides for casual and annual members throughout the months as shown below.

There are drastic changes in the number of rides in the warmer months as opposed to the colder months for the casual riders. For the annual members however, the number of rides is higher throughout the year but also has lower number of rides during the colder months. This means that to successfully be able to persuade more casual riders to convert to annual members, the marketing team needs to convince the users that even if the users do not like riding their bike in the colder months, a campaign showing the benefits of riding bike as opposed to using motor vehicles can be successful.

**Act –Recommendations based on my analysis**

Based on the analysis conducted the top three main recommendations are:

1. A marketing campaign lowering the fare for annual members during the weekend, to get more casual members to convert to annual members since the casual riders have longer ride durations and more rides during the weekend than annual members.
2. A marketing campaign lowering the fare for annual members with ride durations more than a certain limit. The campaign can show that if a user, is using the bikes sharing services for more than a certain duration per week, it is beneficial to have the annual membership. This will cause more casual members to become annual members as they have longer average ride durations than annual members.
3. A marketing campaign showing the benefits of cycling as opposed to using other carbon emitting modes of transport for both the user’s health, and the environment. This can also be done by including some benefits to users who use a bike during the colder months. Another additional option could be a marketing campaign that also shows the safety of cycling during the colder months alongside the health and environmental benefits.

**Conclusion**

Based on these findings, Cyclitic does have a lot of room for growth and many options for marketing campaign to increase the number of both their users and their riders. More data collected from surveys on why casual riders wouldn’t want to convert to members could be more helpful to make a more data driven decisions. However, with the recommendations stated above, the number of annual members would increase.

**References**

**[1]** https://www.tutorialspoint.com/r/r\_mean\_median\_mode.htm

Guide followed

https://d3c33hcgiwev3.cloudfront.net/aacF81H\_TsWnBfNR\_x7FIg\_36299b28fa0c4a5aba836111daad12f1\_DAC8-Case-Study-1.pdf?Expires=1666828800&Signature=aipr2YSXKZkzXkLhrGKpr0Q65Mome8-5ptMzYOTlBiePqdFFCCPIq6YW6M6GhsnvTgd6GgwJxhnCVpcKs6p2eRVZVOP06u75ZWLKr-LuKUPHZW8QDYCpb5fgb~Zr0LcQogPSZs~CxXZfEB-VEnYE1aujEwrs7vPsLV5QddZ1NSs\_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A

Links used for help

<https://quentinferreira.medium.com/applied-case-study-how-does-a-bike-share-navigate-speedy-success-769ce4938882>

https://www.kaggle.com/code/mmaguire/google-data-analytics-capstone-case-study-1

APPENDIX – Code used In R studio

## Installing the Packages

install.packages('tidyverse')

install.packages('janitor')

install.packages('lubridate')

install.packages("ggplot2")

install.packages("vctrs", type = "binary")

## Loading the installed Packages

library("tidyverse")

library("janitor")

library("lubridate")

library("ggplot2")

## Importing the CSV files

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

tripdata\_202004 <- read.csv("202004-divvy-tripdata.csv")

tripdata\_202005 <- read.csv("202005-divvy-tripdata.csv")

tripdata\_202006 <- read.csv("202006-divvy-tripdata.csv")

tripdata\_202007 <- read.csv("202007-divvy-tripdata.csv")

tripdata\_202008 <- read.csv("202008-divvy-tripdata.csv")

tripdata\_202009 <- read.csv("202009-divvy-tripdata.csv")

tripdata\_202010 <- read.csv("202010-divvy-tripdata.csv")

tripdata\_202011 <- read.csv("202011-divvy-tripdata.csv")

tripdata\_202012 <- read.csv("202012-divvy-tripdata.csv")

## Examining the datasets

str(tripdata\_2020\_Q1)

str(tripdata\_202004)

str(tripdata\_202005)

str(tripdata\_202006)

str(tripdata\_202007)

str(tripdata\_202008)

str(tripdata\_202009)

str(tripdata\_202010)

str(tripdata\_202011)

str(tripdata\_202012)

## The "start\_station\_id" and "end\_station\_id" in tripdata\_202012 is not an int unlike the other datasets.

## warn=-1: Ignores the warning message.

## mutate() adds new variables and preserves existing ones; transmute() adds new variables and drops existing ones. New variables overwrite existing variables of the same name. Variables can be removed by setting their value to NULL.

options(warn = -1)

tripdata\_202012 <- mutate(tripdata\_202012, start\_station\_id= as.integer(start\_station\_id))

tripdata\_202012 <- mutate(tripdata\_202012, end\_station\_id=as.integer(end\_station\_id))

str(tripdata\_202012)

## Combined all the individual datasets into one big dataset.

tripdata\_2020 <- rbind(tripdata\_2020\_Q1, tripdata\_202004, tripdata\_202005, tripdata\_202006, tripdata\_202007, tripdata\_202008, tripdata\_202009, tripdata\_202010, tripdata\_202011, tripdata\_202012)

## Finding the total number of rows for the individual data sets and confirming is it correct.

rowtotal <- sum(

nrow(tripdata\_2020\_Q1),

nrow(tripdata\_202004),

nrow(tripdata\_202005),

nrow(tripdata\_202006),

nrow(tripdata\_202007),

nrow(tripdata\_202008),

nrow(tripdata\_202009),

nrow(tripdata\_202010),

nrow(tripdata\_202011),

nrow(tripdata\_202012))

print(rowtotal)

print(nrow(tripdata\_2020))

## Examining Combined Dataset

str(tripdata\_2020)

head(tripdata\_2020)

## Cleaning the Data

## Formatting the data into individual columns, and adding a day of the week column.

## $ = To access one variable in a dataset, use the dollar sign “$”. For example, $vote1 returns the vote1 variable (the vote1 column).

## abbreviated month %b, %d day as a number, %y two digit year, %A as an unabbreviated weekday

##

tripdata\_2020$date <- as.Date(tripdata\_2020$started\_at)

tripdata\_2020$month <- format(as.Date(tripdata\_2020$date), " %b ")

tripdata\_2020$day <- format(as.Date(tripdata\_2020$date), "%d")

tripdata\_2020$year <- format(as.Date(tripdata\_2020$date), "%Y")

tripdata\_2020$day\_of\_week <- format(as.Date(tripdata\_2020$date), "%A")

head(tripdata\_2020)

## Now to remove the Duplicate and all the NA cells.

## the duplicated() function, identifies duplicate elements in the database.

## paste converts its arguments (via as.character) to character strings, and concatenates them (separating them by the string given by sep). If the arguments are vectors, they are concatenated term-by-term to give a character vector result. Vector arguments are recycled as needed, with zero-length arguments being recycled to "".

tripdata\_2020 <- drop\_na(tripdata\_2020)

tripdata\_2020\_no\_duplicates <- tripdata\_2020[!duplicated(tripdata\_2020$ride\_id),]

print(paste("Removed", nrow(tripdata\_2020)-nrow(tripdata\_2020\_no\_duplicates), "duplicate rows"))

## Creating the ride\_length column and transforming to time stamps as opposed to character strings

## Used this for help: https://www.youtube.com/watch?v=rpPt0JCzL6Q

tripdata\_2020\_no\_duplicates$started\_at <- lubridate::ymd\_hms(tripdata\_2020\_no\_duplicates$started\_at)

tripdata\_2020\_no\_duplicates$ended\_at <- lubridate::ymd\_hms(tripdata\_2020\_no\_duplicates$ended\_at)

str(tripdata\_2020\_no\_duplicates)

tripdata\_2020\_v2 <- mutate(tripdata\_2020\_no\_duplicates, ride\_length = difftime(ended\_at, started\_at, units = "mins"))

str(tripdata\_2020\_v2)

## Filtering out the trips with a ride length less than 0 mins

nrow(tripdata\_2020\_v2[tripdata\_2020\_v2$ride\_length < 0 ,])

tripdata\_2020\_v3 <- tripdata\_2020\_v2[!tripdata\_2020\_v2$ride\_length < 0,]

glimpse(tripdata\_2020\_v3)

## Determining the number of casual vs annual members

rider\_type\_total <- table(tripdata\_2020\_v3$member\_casual)

view(rider\_type\_total)

## Statistical Analysis of the ride length column

## summarise() creates a new data frame. It will have one (or more) rows for each combination of grouping variables; if there are no grouping variables, the output will have a single row summarising all observations in the input. It will contain one column for each grouping variable and one column for each of the summary statistics that you have specified.

trip\_stats <- tripdata\_2020\_v3 %>%

group\_by(member\_casual) %>%

summarise(average\_ride\_length = mean(ride\_length), standard\_deviation = sd(ride\_length), median\_ride\_length= median(ride\_length), min\_ride\_length = min(ride\_length), max\_ride\_length = max(ride\_length))

head(trip\_stats)

## Determining the most frequent day when the bikes are used.

## Used help from code 1 earnt from tutorialspoint.com

##R does not have a standard in-built function to calculate mode. So we create a user function to calculate mode of a data set in R. This function takes the vector as input and gives the mode value as output.

## and help from this https://www.kaggle.com/code/mmaguire/google-data-analytics-capstone-case-study-1

getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

weekday\_mode <- getmode(tripdata\_2020\_v3$day\_of\_week)

print(weekday\_mode)

## Determining the most popular day based on rider type.

## use this to understand the levels: If the factor is ordered, then the specific order of the levels matters (small < medium < large).

## http://www.cookbook-r.com/Manipulating\_data/Changing\_the\_order\_of\_levels\_of\_a\_factor/

## use this to understand

## (n) <- Use with group\_by() or summarize . Counts the number of rows

tripdata\_2020\_v3$day\_of\_week <- ordered(tripdata\_2020\_v3$day\_of\_week, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

tripdata\_2020\_v3 %>%

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

summarise(rider\_type\_total = n(), average\_ride\_length = mean(ride\_length)) %>%

arrange(member\_casual, day\_of\_week)

## Finding the popular months during 2020

## - is for descending order

popular\_month <- tripdata\_2020\_v3 %>%

group\_by(month) %>%

summarise(number\_of\_rides = n(), average\_duration = mean(ride\_length)) %>%

arrange(-number\_of\_rides)

View(popular\_month)

## finding the most popular station by using the previous mode function we have

station\_mode <- getmode(tripdata\_2020\_v3$start\_station\_name)

print(station\_mode)

## finding the most popular start station for members

popular\_start\_stations\_member <- tripdata\_2020\_v3 %>%

filter(member\_casual == 'member') %>%

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

summarise(number\_of\_starts = n()) %>%

filter(start\_station\_name != "") %>%

arrange(-number\_of\_starts)

head(popular\_start\_stations\_member)

## finding the most popular start station for casual riders

popular\_start\_stations\_casual <- tripdata\_2020\_v3 %>%

filter(member\_casual == 'casual') %>%

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

summarise(number\_of\_starts = n()) %>%

filter(start\_station\_name != "") %>%

arrange(-number\_of\_starts)

head(popular\_start\_stations\_casual)

## Creating the Data Viz using GGplot

## Using the geom\_bar (position="dodge") places the two bars side by side.

## annual member vs casual rider

tripdata\_2020\_v3 %>%

group\_by(member\_casual) %>%

summarise (total\_rider\_type = n()) %>%

ggplot (aes(x= member\_casual, y = total\_rider\_type, fill = member\_casual)) + geom\_col(position = "dodge") + geom\_text(aes(label = total\_rider\_type, vjust = -0.25))

## rider types ride duration

rider\_type\_average\_duration <- tripdata\_2020\_v3 %>%

group\_by(member\_casual) %>%

summarize(average\_ride\_length = mean(ride\_length))

rider\_type\_average\_duration %>%

ggplot(aes(x = member\_casual, y = average\_ride\_length, fill = member\_casual)) +

geom\_col(position = "dodge") + geom\_text(aes(label = average\_ride\_length, vjust = -0.25))

## Weekday usage by casual riders and annual members

## The arrange() function in R programming is used to reorder the rows of a data frame/table by using column names. These columns are passed as the expression in the function.

tripdata\_2020\_v3 %>%

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

summarise(number\_of\_rides = n(),average\_duration = mean(ride\_length)) %>%

arrange(member\_casual, day\_of\_week) %>%

ggplot(aes(x = day\_of\_week, y = average\_duration, fill = member\_casual)) +

geom\_col(position = "dodge")

## Visualization of the number of trips by members and casual riders by the weekday

tripdata\_2020\_v3 %>%

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

summarise(number\_of\_rides = n(),average\_duration = mean(ride\_length)) %>%

arrange(member\_casual, day\_of\_week) %>%

ggplot(aes(x = day\_of\_week, y = number\_of\_rides, fill = member\_casual)) +

geom\_col(position = "dodge")

## Number of trips by both casual and annual members by month.

tripdata\_2020\_v3$month <- ordered(tripdata\_2020\_v3$month, levels=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))

head(tripdata\_2020\_v3$month)

tripdata\_2020\_v3 %>%

group\_by(member\_casual, month) %>%

summarise(number\_of\_rides = n(),average\_duration = mean(ride\_length) ) %>%

arrange(member\_casual, month) %>%

ggplot(aes(x = month, y = average\_duration, fill = member\_casual)) +

geom\_col(position = "dodge") +

geom\_text(aes(label = number\_of\_rides, angle = 90)) +

facet\_wrap(~member\_casual)

tripdata\_2020\_v3 %>%

group\_by(member\_casual, month) %>%

summarise(number\_of\_rides = n()) %>%

arrange(member\_casual, month) %>%

ggplot(aes(x = month, y = number\_of\_rides, fill = member\_casual)) +

geom\_col(position = "dodge") +

geom\_text(aes(label = number\_of\_rides, angle = 90)) +

facet\_wrap(~member\_casual)