---

title: “Cyclistic Case Study”

author: “Andy Zheng”

output: pdf_document

Cyclistic Case Study

Introduction:

In 2016, Cyclistic launched a successful bike-share offering. The bikes can be unlocked from one station and returned to any other station in the system anytime. Since then, the program has grown to a fleet of 5,824 bicycles that are geo-tracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

Scenario:

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, the company believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, the company believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs. Marketing team needs to design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ.

Objectives:

To discover how casual riders and Cyclistic members use their rental bikes differently. Finance analysts have concluded that annual members are more profitable.

- How do annual members and casual riders use Cyclistic bikes differently?
- Why would casual riders buy Cyclistic annual memberships?
- How can Cyclistic use digital media to influence casual riders to become members?

Business Task:

The results of this analysis will be used to design a new marketing strategy to aqcuire more subscribers / to convert casual riders to annual members.

Data Source:

- Motivate International Inc.

Data Integrity & Credibility:

- Data is company internal data, publicly available, and does not contain personal information.

Tools:

- Data cleaning and preparation done in *R*. (too large for excel)
- Data visualizations made in *R* and *Tableau*.

PREPARATION

Loading packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.4     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(dplyr)
library(ggplot2)
library(lubridate)
```

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

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

```
library(readxl)
library(janitor)
```

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

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

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(tidyr)
library(modeest)
```

```
## Registered S3 method overwritten by 'rmutil':
##   method         from
##   print.response httr
```

Importing 12 .xlsx datasets to R

```
rides2020_04 <- read_xlsx("202004-divvy-tripdata.xlsx")
rides2020_05 <- read_xlsx("202005-divvy-tripdata.xlsx")
rides2020_06 <- read_xlsx("202006-divvy-tripdata.xlsx")
rides2020_07 <- read_xlsx("202007-divvy-tripdata.xlsx")
rides2020_08 <- read_xlsx("202008-divvy-tripdata.xlsx")
rides2020_09 <- read_xlsx("202009-divvy-tripdata.xlsx")
rides2020_10 <- read_xlsx("202010-divvy-tripdata.xlsx")
rides2020_11 <- read_xlsx("202011-divvy-tripdata.xlsx")
rides2020_12 <- read_xlsx("202012-divvy-tripdata.xlsx")
rides2021_01 <- read_xlsx("202101-divvy-tripdata.xlsx")
rides2021_02 <- read_xlsx("202102-divvy-tripdata.xlsx")
rides2021_03 <- read_xlsx("202103-divvy-tripdata.xlsx")
```

Checking for column name inconsistencies

```
colnames(rides2020_04)
```

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

```
colnames(rides2020_05)
```

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

```
colnames(rides2020_06)
```

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

```
colnames(rides2020_07)
```

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

```
colnames(rides2020_08)
```

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

```
colnames(rides2020_09)
```

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

```
colnames(rides2020_10)
```

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

```
colnames(rides2020_11)
```

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

```
colnames(rides2020_12)
```

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

```
colnames(rides2021_01)
```

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

```
colnames(rides2021_02)
```

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

```
colnames(rides2021_03)
```

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

```
- All columns have same names.
```

Getting an overview of the data

```
tibble(rides2020_04)
```

```
## # A tibble: 84,776 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
```

```
##  1 A847FADBBC638E45 docked_bike   2020-04-26 17:45:14 2020-04-26 18:12:03
##  2 5405B80E996FF60D docked_bike   2020-04-17 17:08:54 2020-04-17 17:17:03
##  3 5DD24A79A4E006F4 docked_bike   2020-04-01 17:54:13 2020-04-01 18:08:36
##  4 2A59BBDF5CDBA725 docked_bike   2020-04-07 12:50:19 2020-04-07 13:02:31
##  5 27AD306C119C6158 docked_bike   2020-04-18 10:22:59 2020-04-18 11:15:54
##  6 356216E875132F61 docked_bike   2020-04-30 17:55:47 2020-04-30 18:01:11
##  7 A2759CB06A81F2BC docked_bike   2020-04-02 14:47:19 2020-04-02 14:52:32
##  8 FC8BC2E2D54F35ED docked_bike   2020-04-07 12:22:20 2020-04-07 13:38:09
##  9 9EC5648678DE06E6 docked_bike   2020-04-15 10:30:11 2020-04-15 10:35:55
## 10 A8FFF89140C33017 docked_bike   2020-04-04 15:02:28 2020-04-04 15:19:47
## # ... with 84,766 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_05)
```

```
## # A tibble: 200,274 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 02668AD35674B983 docked_bike   2020-05-27 10:03:52 2020-05-27 10:16:49
##  2 7A50CCAF1EDDB28F docked_bike   2020-05-25 10:47:11 2020-05-25 11:05:40
##  3 2FFCDFDB91FE9A52 docked_bike   2020-05-02 14:11:03 2020-05-02 15:48:21
##  4 58991CF1DB75BA84 docked_bike   2020-05-02 16:25:36 2020-05-02 16:39:28
##  5 A79651EFECC268CD docked_bike   2020-05-29 12:49:54 2020-05-29 13:27:11
##  6 1466C5B39F68F746 docked_bike   2020-05-29 13:27:24 2020-05-29 14:14:45
##  7 2500D7957D4D0A34 docked_bike   2020-05-20 12:51:41 2020-05-20 13:46:47
##  8 ED42D3E06AFB2F26 docked_bike   2020-05-06 18:21:42 2020-05-06 19:07:07
##  9 23AFBD962F9C8F14 docked_bike   2020-05-30 17:00:58 2020-05-30 17:19:52
## 10 52C0D13F6B81C5F8 docked_bike   2020-05-23 10:22:02 2020-05-23 10:52:02
## # ... with 200,264 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_06)
```

```
## # A tibble: 343,005 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 8CD5DE2C2B6C4CFC docked_bike   2020-06-13 23:24:48 2020-06-13 23:36:55
##  2 9A191EB2C751D85D docked_bike   2020-06-26 07:26:10 2020-06-26 07:31:58
##  3 F37D14B0B5659BCF docked_bike   2020-06-23 17:12:41 2020-06-23 17:21:14
##  4 C41237B506E85FA1 docked_bike   2020-06-20 01:09:35 2020-06-20 01:28:24
##  5 4B51B3B0BDA7787C docked_bike   2020-06-25 16:59:25 2020-06-25 17:08:48
##  6 D50DF288196B53BE docked_bike   2020-06-17 18:07:18 2020-06-17 18:18:14
##  7 165FA6D223E58600 docked_bike   2020-06-25 07:24:33 2020-06-25 07:31:11
##  8 D8236CFC050E591C docked_bike   2020-06-19 00:00:56 2020-06-19 00:09:15
##  9 9D82B9B53C37C55C docked_bike   2020-06-30 12:11:36 2020-06-30 12:32:43
## 10 3DFF4AB10A6895A3 docked_bike   2020-06-28 14:17:09 2020-06-28 14:27:51
## # ... with 342,995 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_07)
```

```
## # A tibble: 551,480 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 762198876D69004D docked_bike   2020-07-09 15:22:02 2020-07-09 15:25:52
##  2 BEC9C9FBA0D4CF1B docked_bike   2020-07-24 23:56:30 2020-07-25 00:20:17
##  3 D2FD8EA432C77EC1 docked_bike   2020-07-08 19:49:07 2020-07-08 19:56:22
##  4 54AE594E20B35881 docked_bike   2020-07-17 19:06:42 2020-07-17 19:27:38
##  5 54025FDC7440B56F docked_bike   2020-07-04 10:39:57 2020-07-04 10:45:05
##  6 65636B619E24257F docked_bike   2020-07-28 16:33:03 2020-07-28 16:49:10
##  7 22DB94283ECFDBD2 docked_bike   2020-07-30 11:58:12 2020-07-30 12:16:12
##  8 C9D789BEF899F4B8 docked_bike   2020-07-13 16:48:03 2020-07-13 16:57:00
##  9 FBE24D943CDE35A1 docked_bike   2020-07-30 11:00:12 2020-07-30 11:14:48
## 10 8A2BBE457325A2E7 docked_bike   2020-07-06 18:05:29 2020-07-06 18:15:38
## # ... with 551,470 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_08)
```

```
## # A tibble: 622,361 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 322BD23D287743ED docked_bike   2020-08-20 18:08:14 2020-08-20 18:17:51
##  2 2A3AEF1AB9054D8B electric_bike 2020-08-27 18:46:04 2020-08-27 19:54:51
##  3 67DC1D133E8B5816 electric_bike 2020-08-26 19:44:14 2020-08-26 21:53:07
##  4 C79FBBD412E578A7 electric_bike 2020-08-27 12:05:41 2020-08-27 12:53:45
##  5 13814D3D661ECADB electric_bike 2020-08-27 16:49:02 2020-08-27 16:59:49
##  6 56349A5A42F0AE51 electric_bike 2020-08-27 17:26:23 2020-08-27 18:07:50
##  7 EB6ABC5570C29B22 electric_bike 2020-08-26 20:14:02 2020-08-26 20:34:00
##  8 B4ECE389A1DE922D electric_bike 2020-08-26 21:59:50 2020-08-26 22:12:35
##  9 0B355B0FE076D010 electric_bike 2020-08-26 19:17:42 2020-08-26 19:32:14
## 10 1ECE04F779E9FDF6 electric_bike 2020-08-27 15:13:57 2020-08-27 15:41:59
## # ... with 622,351 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_09)
```

```
## # A tibble: 532,958 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 2B22BD5F95FB2629 electric_bike 2020-09-17 14:27:11 2020-09-17 14:44:24
##  2 A7FB70B4AFC6CAF2 electric_bike 2020-09-17 15:07:31 2020-09-17 15:07:45
##  3 86057FA01BAC778E electric_bike 2020-09-17 15:09:04 2020-09-17 15:09:35
##  4 57F6DC9A153DB98C electric_bike 2020-09-17 18:10:46 2020-09-17 18:35:49
##  5 B9C4712F78C1AE68 electric_bike 2020-09-17 15:16:13 2020-09-17 15:52:55
##  6 378BBCE1E444EB80 electric_bike 2020-09-17 18:37:04 2020-09-17 19:23:28
##  7 0CB5E2496B6F1DF8 electric_bike 2020-09-16 21:39:57 2020-09-16 21:53:44
```

```
##  8 9673F5D39BDBA8BE electric_bike 2020-09-17 12:18:06 2020-09-17 12:18:19
##  9 54B91F5C95B20268 electric_bike 2020-09-17 17:09:17 2020-09-17 17:34:20
## 10 91CEBB66076D4713 electric_bike 2020-09-17 12:20:25 2020-09-17 12:29:47
## # ... with 532,948 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_10)
```

```
## # A tibble: 388,653 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 ACB6B40CF5B9044C electric_bike 2020-10-31 19:39:43 2020-10-31 19:57:12
##  2 DF450C72FD109C01 electric_bike 2020-10-31 23:50:08 2020-11-01 00:04:16
##  3 B6396B54A15AC0DF electric_bike 2020-10-31 23:00:01 2020-10-31 23:08:22
##  4 44A4AEE261B9E854 electric_bike 2020-10-31 22:16:43 2020-10-31 22:19:35
##  5 10B7DD76A6A2EB95 electric_bike 2020-10-31 19:38:19 2020-10-31 19:54:32
##  6 DA6C3759660133DA electric_bike 2020-10-29 17:38:04 2020-10-29 17:45:43
##  7 C2F3808FD56B4F84 electric_bike 2020-10-29 09:03:06 2020-10-29 09:17:56
##  8 15B13B5A508BA2B6 electric_bike 2020-10-29 16:37:21 2020-10-29 16:52:40
##  9 285D224410C101C5 electric_bike 2020-10-28 23:12:03 2020-10-28 23:24:32
## 10 E1FB79FFE6DB0117 electric_bike 2020-10-29 16:38:44 2020-10-29 16:50:17
## # ... with 388,643 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_11)
```

```
## # A tibble: 259,716 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 BD0A6FF6FFF9B921 electric_bike 2020-11-01 13:36:00 2020-11-01 13:45:40
##  2 96A7A7A4BDE4F82D electric_bike 2020-11-01 10:03:26 2020-11-01 10:14:45
##  3 C61526D06582BDC5 electric_bike 2020-11-01 00:34:05 2020-11-01 01:03:06
##  4 E533E89C32080B9E electric_bike 2020-11-01 00:45:16 2020-11-01 00:54:31
##  5 1C9F4EF18C168C60 electric_bike 2020-11-01 15:43:25 2020-11-01 16:16:52
##  6 7259585D8276D338 electric_bike 2020-11-14 15:55:17 2020-11-14 16:44:38
##  7 91FE5C8F8A676594 electric_bike 2020-11-14 16:47:29 2020-11-14 17:03:03
##  8 9E7A79ADA90C2695 electric_bike 2020-11-14 16:04:15 2020-11-14 16:19:33
##  9 A5B02C0D41DBCDAF electric_bike 2020-11-14 16:24:09 2020-11-14 16:51:34
## 10 8234407C29FE41DC electric_bike 2020-11-14 01:24:22 2020-11-14 01:31:42
## # ... with 259,706 more rows, and 9 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>,
## #   member_casual <chr>
```

```
tibble(rides2020_12)
```

```
## # A tibble: 131,573 x 13
##    ride_id          rideable_type started_at          ended_at
```

```
##    <chr>            <chr>         <dttm>              <dttm>
##  1 70B6A9A437D4C30D classic_bike  2020-12-27 12:44:29 2020-12-27 12:55:06
##  2 158A465D4E74C54A electric_bike 2020-12-18 17:37:15 2020-12-18 17:44:19
##  3 5262016E0F1F2F9A electric_bike 2020-12-15 15:04:33 2020-12-15 15:11:28
##  4 BE119628E44F871E electric_bike 2020-12-15 15:54:18 2020-12-15 16:00:11
##  5 69AF78D57854E110 electric_bike 2020-12-22 12:08:17 2020-12-22 12:10:59
##  6 C1DECC4AB488831C electric_bike 2020-12-22 13:26:37 2020-12-22 13:34:50
##  7 B014A60B856C02B1 electric_bike 2020-12-03 16:23:48 2020-12-03 16:33:39
##  8 1E127B1929C0A976 electric_bike 2020-12-03 15:03:38 2020-12-03 15:12:39
##  9 05F41F5137B5048E electric_bike 2020-12-12 09:26:17 2020-12-12 09:26:35
## 10 BB807646588DC5B1 electric_bike 2020-12-18 12:52:06 2020-12-18 12:52:23
## # ... with 131,563 more rows, and 9 more variables: start_station_name <chr>,
## #   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>
```

```
tibble(rides2021_01)
```

```
## # A tibble: 96,834 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 E19E6F1B8D4C42ED electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44
##  2 DC88F20C2C55F27F electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12
##  3 EC45C94683FE3F27 electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14
##  4 4FA453A75AE377DB electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55
##  5 BE5E8EB4E7263A0B electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45
##  6 5D8969F88C773979 electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54
##  7 09275CC10F854E9E electric_bike 2021-01-04 05:05:04 2021-01-04 05:10:39
##  8 DF7A32A217AEFB14 electric_bike 2021-01-14 15:07:00 2021-01-14 15:13:40
##  9 C2EFC62379EB716C electric_bike 2021-01-09 09:57:55 2021-01-09 10:00:26
## 10 B9F73448DFBE0D45 classic_bike  2021-01-24 19:15:38 2021-01-24 19:22:51
## # ... with 96,824 more rows, and 9 more variables: start_station_name <chr>,
## #   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>
```

```
tibble(rides2021_02)
```

```
## # A tibble: 49,622 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 89E7AA6C29227EFF classic_bike  2021-02-12 16:14:56 2021-02-12 16:21:43
##  2 0FEFDE2603568365 classic_bike  2021-02-14 17:52:38 2021-02-14 18:12:09
##  3 E6159D746B2DBB91 electric_bike 2021-02-09 19:10:18 2021-02-09 19:19:10
##  4 B32D3199F1C2E75B classic_bike  2021-02-02 17:49:41 2021-02-02 17:54:06
##  5 83E463F23575F4BF electric_bike 2021-02-23 15:07:23 2021-02-23 15:22:37
##  6 BDAA7E3494E8D545 electric_bike 2021-02-24 15:43:33 2021-02-24 15:49:05
##  7 A772742351171257 classic_bike  2021-02-01 17:47:42 2021-02-01 17:48:33
##  8 295476889D9B79F8 classic_bike  2021-02-11 18:33:53 2021-02-11 18:35:09
##  9 362087194BA4CC9A classic_bike  2021-02-27 15:13:39 2021-02-27 15:36:36
## 10 21630F715038CCB0 classic_bike  2021-02-20 08:59:42 2021-02-20 09:17:04
## # ... with 49,612 more rows, and 9 more variables: start_station_name <chr>,
## #   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>
```

```
tibble(rides2021_03)
```

```
## # A tibble: 228,496 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 CFA86D4455AA1030 classic_bike  2021-03-16 08:32:30 2021-03-16 08:36:34
##  2 30D9DC61227D1AF3 classic_bike  2021-03-28 01:26:28 2021-03-28 01:36:55
##  3 846D87A15682A284 classic_bike  2021-03-11 21:17:29 2021-03-11 21:33:53
##  4 994D05AA75A168F2 classic_bike  2021-03-11 13:26:42 2021-03-11 13:55:41
##  5 DF7464FBE92D8308 classic_bike  2021-03-21 09:09:37 2021-03-21 09:27:33
##  6 CEBA8516FD17F8D8 classic_bike  2021-03-20 11:08:47 2021-03-20 11:29:39
##  7 297268586B79588B classic_bike  2021-03-20 14:10:41 2021-03-20 14:22:13
##  8 F39301858B6077DD electric_bike 2021-03-23 07:56:51 2021-03-23 08:05:50
##  9 D297F199D875BABE electric_bike 2021-03-31 15:31:19 2021-03-31 15:35:58
## 10 36B877141175ED7E classic_bike  2021-03-11 17:37:37 2021-03-11 17:52:44
## # ... with 228,486 more rows, and 9 more variables: start_station_name <chr>,
## #   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>
```

Mutate (rides2020_12), (rides2021_01), (rides2021_02), (rides2021_03) to be consistent with other datesets; Changing [end_station_id] & [start_station_id] from *chr* changed to *dbl*

```
rides2020_12 <- mutate(rides2020_12,start_station_id = as.double(start_station_id), end_station_id = as
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
rides2021_01 <- mutate(rides2021_01,start_station_id = as.double(start_station_id), end_station_id = as
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
rides2021_02 <- mutate(rides2021_02,start_station_id = as.double(start_station_id), end_station_id = as
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
rides2021_03 <- mutate(rides2021_03,start_station_id = as.double(start_station_id), end_station_id = as
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

Checking if change took effect

```
is.double(rides2020_12$start_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2020_12$end_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2021_01$start_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2021_01$end_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2021_02$start_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2021_02$end_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2021_03$start_station_id)
```

```
## [1] TRUE
```

```
is.double(rides2021_03$end_station_id)
```

```
## [1] TRUE
```

- all outputs true

Merging all data sets April, 2020 - May, 2021 into one dataset (rides_202004_202103)

```
rides_202004_202103 <- bind_rows(rides2020_04, rides2020_05,rides2020_06, rides2020_07, rides2020_08, r
```

Adding columns for [date], [month], [day], [day_of_week], [year], [ride_length_secs], [ride_length_mins] pulled from columns: [started_at], [ended_at]

```
rides_202004_202103$date <- as.Date(rides_202004_202103$started_at)
rides_202004_202103$month <- format(as.Date(rides_202004_202103$date),"%m")
rides_202004_202103$day <- format(as.Date(rides_202004_202103$date),"%d")
rides_202004_202103$year <- format(as.Date(rides_202004_202103$date),"%Y")
rides_202004_202103$day_of_week <- format(as.Date(rides_202004_202103$date),"%A")
rides_202004_202103$ride_length_secs <- as.numeric(difftime(rides_202004_202103$ended_at,rides_202004_2
rides_202004_202103$ride_length_mins <-as.numeric(rides_202004_202103$ride_length_secs / 60)
```

Removing rows with N/A, negative values, test rides into new dataset (rides_202004_202103_v2)

```
# rides_202004_202103_v2 <- drop_na(rides_202004_202103)

rides_202004_202103_v2 <- rides_202004_202103[!(rides_202004_202103$ride_length_secs < 0),]
rides_202004_202103_v2 <- rides_202004_202103_v2 [!((rides_202004_202103_v2$start_station_name %like% "
```

- Rows with N/A make up a significant portion of data \~540,000 rides. Not going to remove these assumi

- 10552 rows removed (3489748-3479196), negative values make up .30% of (rides_202004_202103_v2)

- 3352 rows removed (3479196-3475844), test & TEST rides make up .39% of (rides_202004_202103_v2)

Checking for distinct values in column member_casual, Counting trips by type of rider

```
unique(rides_202004_202103_v2[c("member_casual")])
```

```
## # A tibble: 2 x 1
##   member_casual
##   <chr>
## 1 member
## 2 casual
```

```
table(rides_202004_202103_v2$member_casual)
```

```
##
##  casual  member
## 1423876 2051968
```

- Only two values found: member (2051968), casual (1423876)

ANALYZE

Summary of fully cleaned dataset (rides_202004_202103_v2)

```
summary(rides_202004_202103_v2)
```

```
##    ride_id          rideable_type        started_at
##  Length:3475844     Length:3475844     Min.   :2020-04-01 00:00:30
##  Class :character   Class :character   1st Qu.:2020-07-14 18:32:50
##  Mode  :character   Mode  :character   Median :2020-08-29 14:41:34
##                                        Mean   :2020-09-10 01:58:04
##                                        3rd Qu.:2020-10-21 05:54:26
##                                        Max.   :2021-03-31 23:59:08
##
##     ended_at                   start_station_name start_station_id
##  Min.   :2020-04-01 00:10:45   Length:3475844     Min.   :    2
##  1st Qu.:2020-07-14 19:02:40   Class :character   1st Qu.:  109
```

```
##  Median :2020-08-29 15:13:19   Mode  :character   Median :  212
##  Mean   :2020-09-10 02:26:03                      Mean   : 1018
##  3rd Qu.:2020-10-21 06:09:55                      3rd Qu.:  332
##  Max.   :2021-04-06 11:00:11                      Max.   :20258
##                                                   NA's   :388180
##  end_station_name   end_station_id     start_lat       start_lng
##  Length:3475844     Min.   :    2    Min.   :41.64   Min.   :-87.87
##  Class :character   1st Qu.:  110    1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Median :  213    Median :41.90   Median :-87.64
##                     Mean   : 1018    Mean   :41.90   Mean   :-87.64
##                     3rd Qu.:  332    3rd Qu.:41.93   3rd Qu.:-87.63
##                     Max.   :20258    Max.   :42.08   Max.   :-87.52
##                     NA's   :403898
##     end_lat          end_lng       member_casual          date
##  Min.   :41.54   Min.   :-88.07   Length:3475844     Min.   :2020-04-01
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   1st Qu.:2020-07-14
##  Median :41.90   Median :-87.64   Mode  :character   Median :2020-08-29
##  Mean   :41.90   Mean   :-87.64                      Mean   :2020-09-09
##  3rd Qu.:41.93   3rd Qu.:-87.63                      3rd Qu.:2020-10-21
##  Max.   :42.16   Max.   :-87.44                      Max.   :2021-03-31
##  NA's   :4712    NA's   :4712
##     month               day                year           day_of_week
##  Length:3475844     Length:3475844     Length:3475844     Length:3475844
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  ride_length_secs  ride_length_mins
##  Min.   :      0   Min.   :    0.00
##  1st Qu.:    477   1st Qu.:    7.95
##  Median :    875   Median :   14.58
##  Mean   :   1679   Mean   :   27.98
##  3rd Qu.:   1602   3rd Qu.:   26.70
##  Max.   :3523202   Max.   :58720.03
##
```

- Average ride is around 28 minutes

Summary by member type

```
aggregate(rides_202004_202103_v2$ride_length_mins ~ rides_202004_202103_v2$member_casual, FUN = mean)
```

```
##   rides_202004_202103_v2$member_casual rides_202004_202103_v2$ride_length_mins
## 1                               casual                                45.07176
## 2                               member                                16.11562
```

```
aggregate(rides_202004_202103_v2$ride_length_mins ~ rides_202004_202103_v2$member_casual, FUN = median)
```

```
##   rides_202004_202103_v2$member_casual rides_202004_202103_v2$ride_length_mins
## 1                               casual                                21.26667
## 2                               member                                11.48333
```

```
rides_202004_202103_v2 %>%
  group_by(member_casual) %>%
  summarise(min_ride_length_mins= min(ride_length_mins),max_ride_length_mins = max(ride_length_mins),
            median_ride_length_mins = median(ride_length_mins), mean_ride_length_mins = mean(ride_lengt
```

```
## # A tibble: 2 x 5
##   member_casual min_ride_length_mins max_ride_length_mins median_ride_length_mi~
##   <chr>                        <dbl>                 <dbl>                  <dbl>
## 1 casual                           0                55684.                   21.3
## 2 member                           0                58720.                   11.5
## # ... with 1 more variable: mean_ride_length_mins <dbl>
```

```
- Average casual ride is ~45 minutes. Average member ride is ~16 minutes.
```

```
- Casual riders spend more than double the time per ride when compared to member riders. Casual riders a
```

Determining mode, what days are the most busy?

```
aggregate(rides_202004_202103_v2$day_of_week ~ rides_202004_202103_v2$member_casual, FUN = mfv)
```

```
##   rides_202004_202103_v2$member_casual rides_202004_202103_v2$day_of_week
## 1                              casual                           Saturday
## 2                              member                           Saturday
```

```
rides_202004_202103_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(),average_ride_length_mins = mean(ride_length_mins)) %>%
  arrange(member_casual, desc(number_of_rides))
```

```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_ride_length_mins
##    <chr>         <chr>                <int>                    <dbl>
##  1 casual        Saturday            334933                     47.0
##  2 casual        Sunday              262183                     50.8
##  3 casual        Friday              207852                     42.9
##  4 casual        Thursday            165806                     43.1
##  5 casual        Wednesday           157849                     40.5
##  6 casual        Monday              150590                     45.1
##  7 casual        Tuesday             144663                     40.7
##  8 member        Saturday            323109                     17.8
##  9 member        Friday              306388                     15.8
## 10 member        Wednesday           305049                     15.3
## 11 member        Thursday            300439                     15.2
## 12 member        Tuesday             284366                     15.1
## 13 member        Monday              267326                     15.3
## 14 member        Sunday              265291                     18.2
```

```
- Saturday has the most rides for both members and casuals. For both casual riders and member riders, th
```

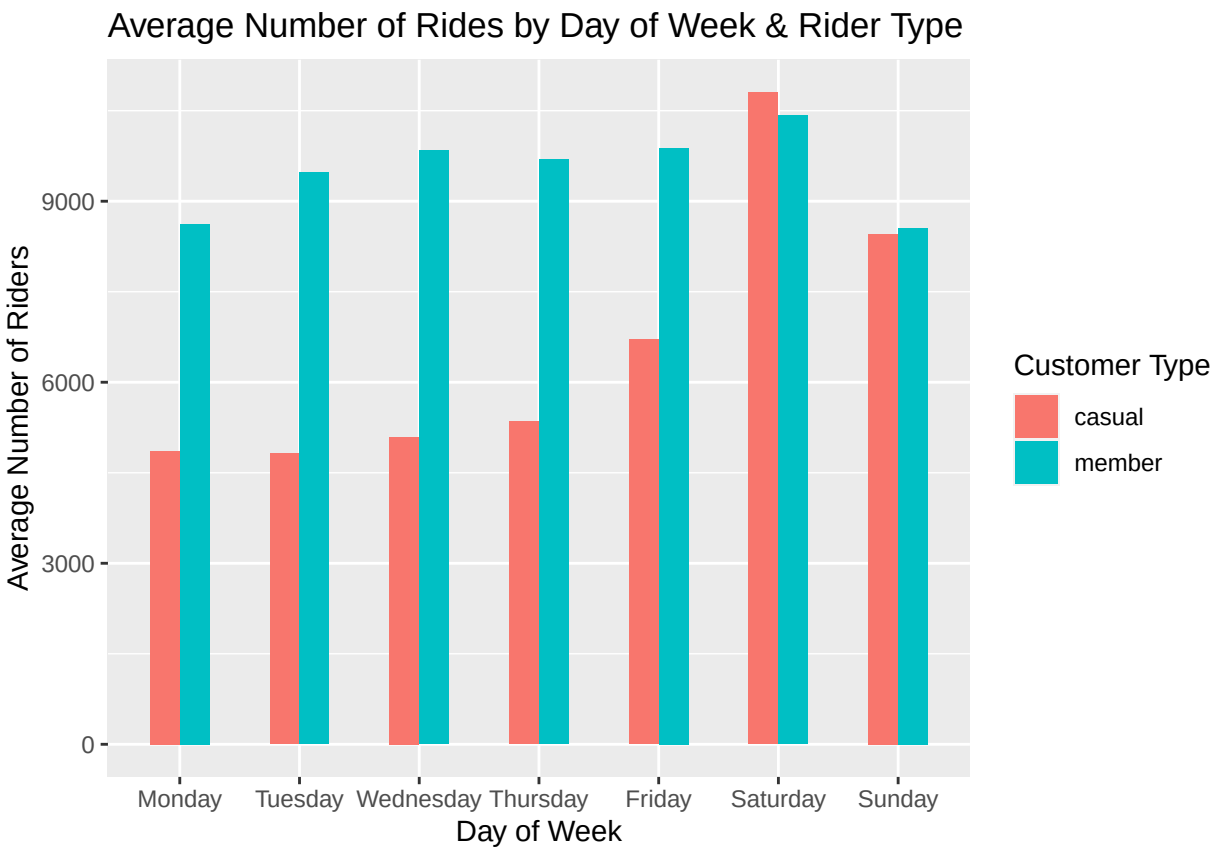
SHARE / VISUALIZATIONS

Average Number of Rides by Day of Week & Rider Type

```
avg_week <- rides_202004_202103_v2 %>%
  select(-day, -month) %>%
  group_by(day_of_week, member_casual) %>%
  summarise(number_of_riders = length(rideable_type))
```

```
## `summarise()` has grouped output by 'day_of_week'. You can override using the `.groups` argument.
```

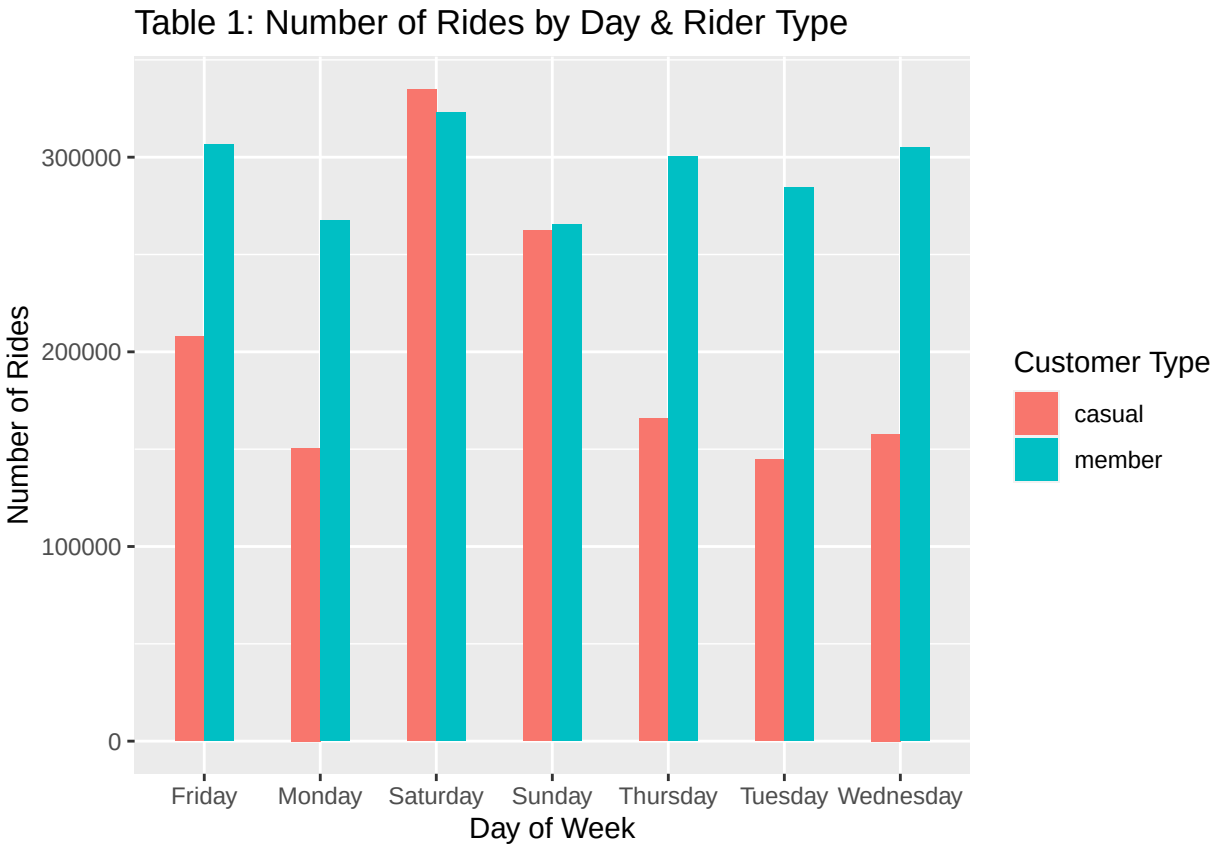
```
avg_day <- rides_202004_202103_v2 %>%
  distinct(day ,day_of_week)%>%
  group_by(day_of_week) %>%
  summarise(number_of_riders_day=length(day))

avg_week <-merge(avg_week,avg_day,by="day_of_week") %>%
  mutate(ave_rider_count_week=number_of_riders/number_of_riders_day,.keep="unused")

avg_week_plot <- avg_week %>%
  mutate(day_of_week=factor(day_of_week, levels=c("Monday","Tuesday","Wednesday","Thursday","Friday","Sa
  ggplot(aes(x= day_of_week,y=ave_rider_count_week,fill=member_casual, width=0.5)) +
  geom_bar(position="dodge", stat="identity")

avg_week_plot + labs(title ="Average Number of Rides by Day of Week & Rider Type", x ="Day of Week", y =
```



Number of Ride by Day of Week & Rider Type

```
rides_202004_202103_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(width=0.5, position = position_dodge(width=0.5)) +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE)) +
  labs(title = "Table 1: Number of Rides by Day & Rider Type") +
  ylab("Number of Rides") +
  xlab("Day of Week") +
  labs(fill = 'Customer Type')
```

```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

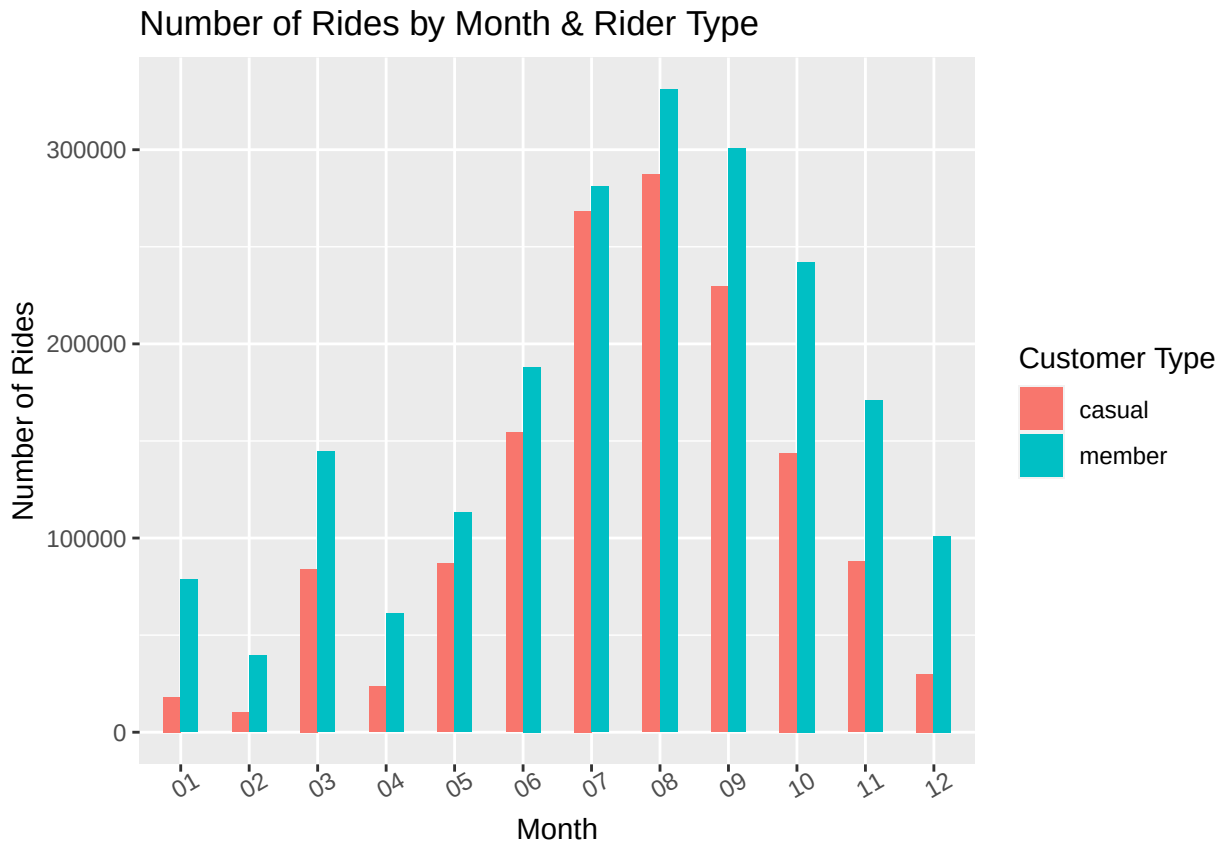


Number of Rides by Month & Rider Type

```
rides_202004_202103_v2 %>%
  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)) +
  labs(title ="Number of Rides by Month & Rider Type") +
```

```
  theme(axis.text.x = element_text(angle = 30)) +
  geom_col(width=0.5, position = position_dodge(width=0.5)) +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE)) +
  ylab("Number of Rides") +
  xlab("Month") +
  labs(fill = 'Customer Type')
```

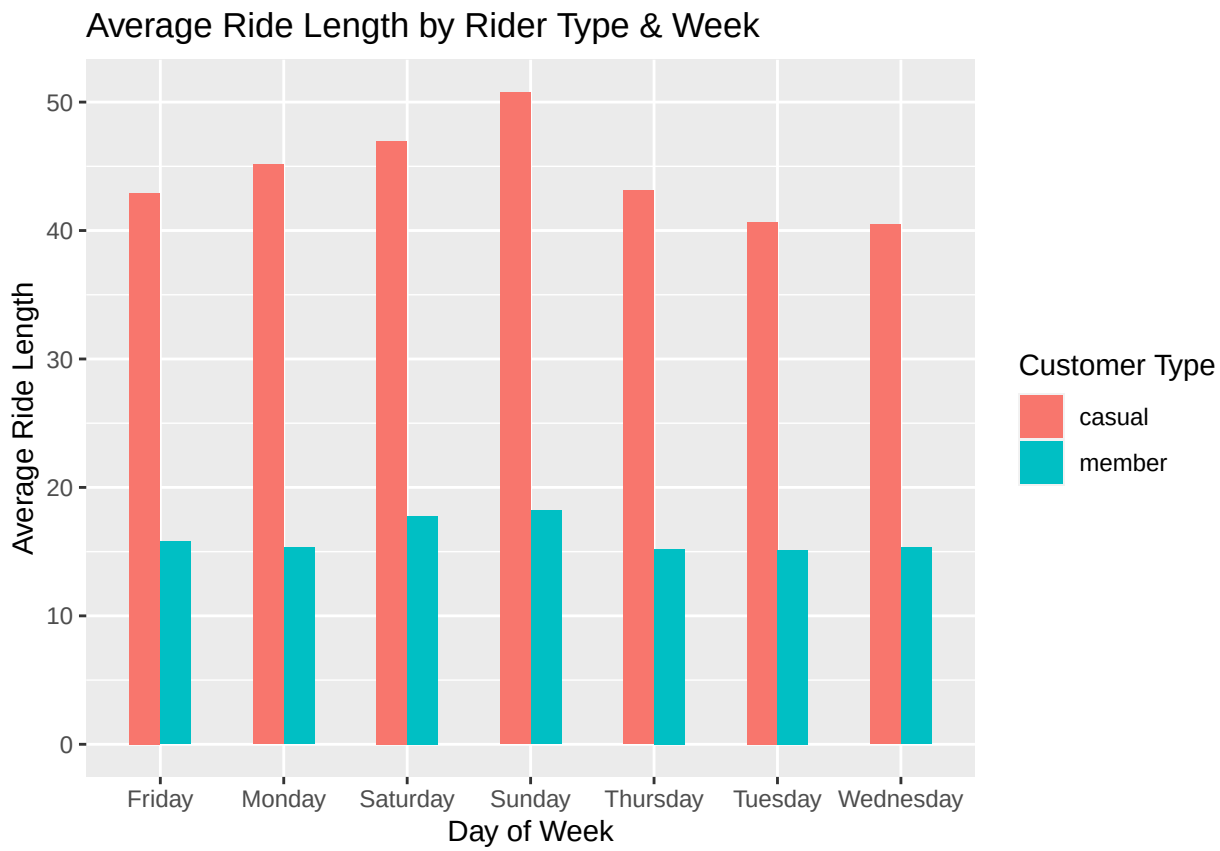
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



Average Ride Length by Rider Type & Week

```
rides_202004_202103_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(avg_ride_length = mean(ride_length_mins)) %>%
  ggplot(aes(x = day_of_week, y = avg_ride_length, fill = member_casual)) +
  geom_col(width=0.5, position = position_dodge(width=0.5)) +
  labs(title ="Average Ride Length by Rider Type & Week") +
  ylab("Average Ride Length") +
  xlab("Day of Week") +
  labs(fill = 'Customer Type')
```

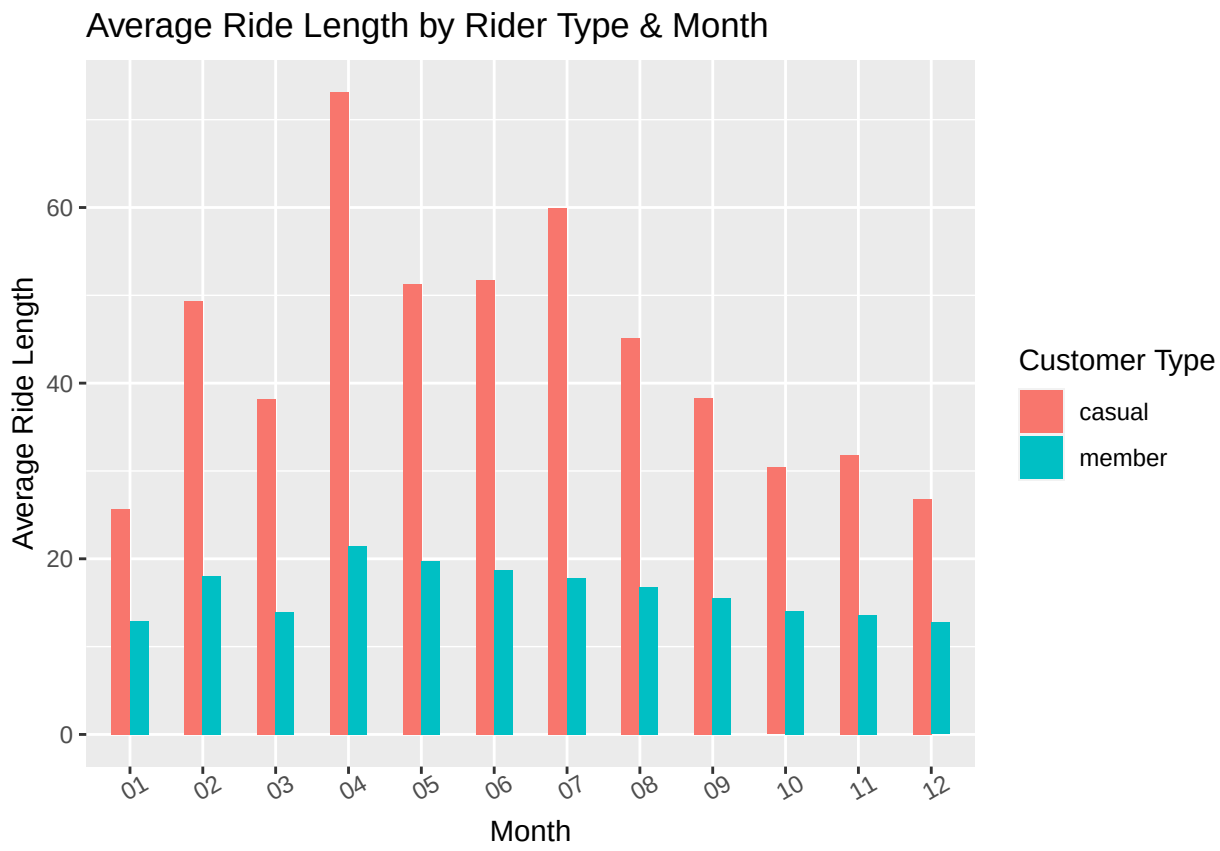
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



Average Ride Length by Rider Type & Month

```
rides_202004_202103_v2 %>%
  group_by(member_casual, month) %>%
  summarise(avg_ride_length = mean(ride_length_mins)) %>%
  ggplot(aes(x = month, y = avg_ride_length, fill = member_casual)) +
  geom_col(width=0.5, position = position_dodge(width=0.5)) +
  labs(title ="Average Ride Length by Rider Type & Month") +
  theme(axis.text.x = element_text(angle = 30)) +
  ylab("Average Ride Length") +
  xlab("Month") +
  labs(fill = 'Customer Type')
```

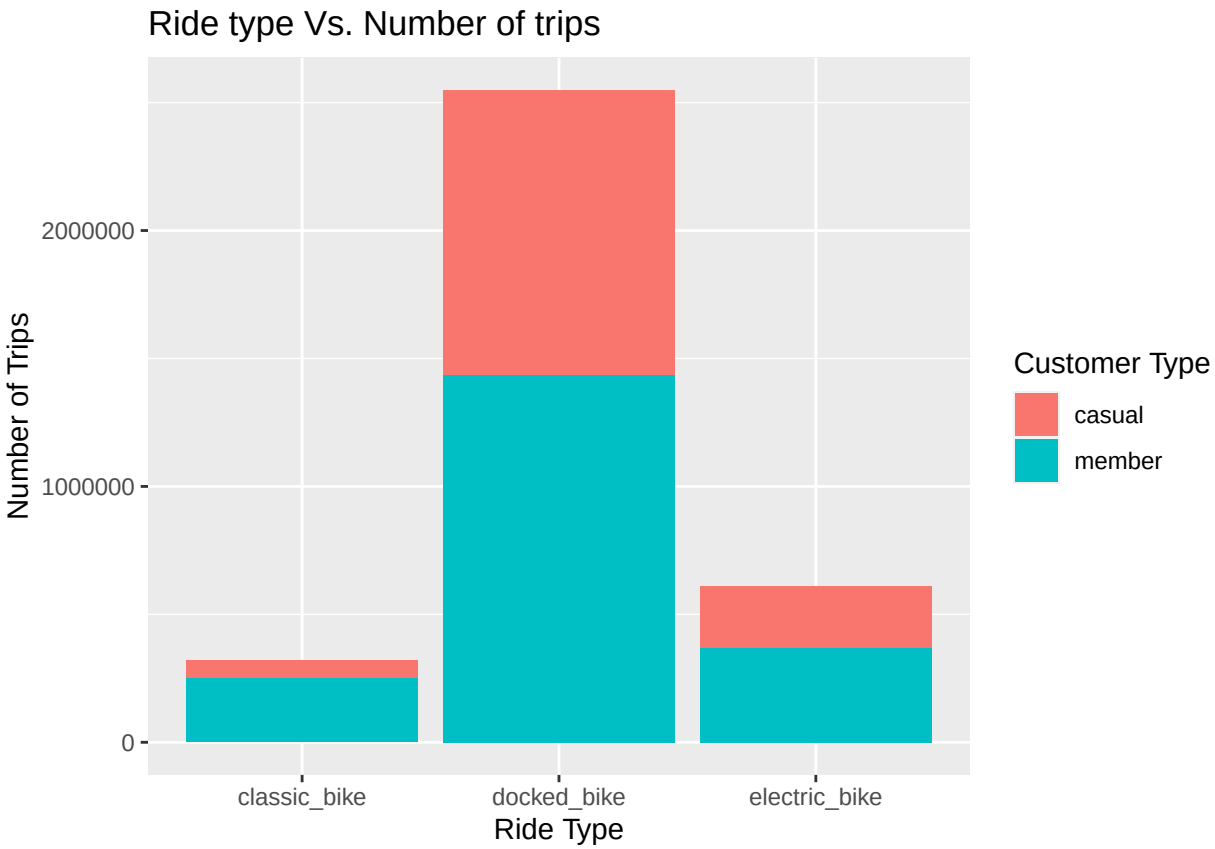
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



ride type vs number trips

```
rides_202004_202103_v2 %>%
  group_by(rideable_type, member_casual) %>%
  summarise(number_of_trips = n()) %>%
  ggplot(aes(x = rideable_type, y = number_of_trips, fill = member_casual))+
  geom_bar(stat ='identity') +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE)) +
  labs(title ="Ride type Vs. Number of trips") +
  ylab("Number of Trips") +
  xlab("Ride Type") +
  labs(fill = 'Customer Type')
```

```
## `summarise()` has grouped output by 'rideable_type'. You can override using the `.groups` argument.
```



FINDINGS

How do casual riders and Cyclistic members use their rental bikes differently?:

Based on 12 months of data from April, 2020 to March, 2021:

- Members make up ~60% of all rides while casual riders make up ~40% of all rides.
- The docked bike option is far more popular than both classic bikes and electric bikes for both casual riders and members.
- Throughout the year, there are always consistently more riders than casual riders with peak traffic from July to September.
- Members of Cyclistic are much more consistent with riding throughout the week, especially on weekdays based on trip duration and number of rides.
- Casual riders on the other hand prefer the weekend and have a large range for trip duration.
- On average, each bike trip takes 30 minutes. Casual members on average ride much longer(46 minutes) than members(16 minutes) per trip - nearly twice as much.
- It could be that members primarily use bikes for regular / scheduled commutes while casual riders are may use bikes for leisure and are more spontaneous.

RECOMMENDATIONS

Based on our findings, how can cyclistic aqcuire more subscribers / convert casual riders to annual members?:

- Create incentive for casual riders to ride more on weekdays such as a promotion or discount throughout work/school days(Monday - Friday).
- Consider alternative membership options for casual riders; perhaps a membership just for the weekend.
- Focus advertising for casual riders to expose them to more weekday riding.
- Run more campaigns during the summer when ridership is at it's highest.