Curso 8 Google CIEE - Estudo de caso 1: Como um compartilhamento de bicicletas possibilita o sucesso rápido?

INTRODUÇÃO

Em 2016, a Cyclistic (empresa fictícia) lançou uma oferta bem-sucedida de compartilhamento de bicicletas. Desde então, o programa cresceu para uma frota de 5.824 bicicletas com rastreamento geográfico e bloqueio dentro de uma rede de 692 estações (dockers) em Chicago.

Até agora, a estratégia de marketing da Cyclistic baseava-se na conscientização geral e no apelo a amplos insights do consumidor, uma abordagem que foi ajudada pela flexibilidade dos planos de preços da empresa.Neste aspecto, a diretora de marketing da empresa, Lily Moreno, acredita que o sucesso futuro está na maximização do número de associações anuais.

Os analistas financeiros da Cyclistic concluíram que os membros anuais são muito mais lucrativos do que os passageiros casuais. Embora a flexibilidade de preços ajude a Cyclistic a atrair mais clientes, Lily Moreno acredita que maximizar o número de membros anuais será a chave para o crescimento futuro. Em vez de criar uma campanha de marketing voltada para novos clientes, ela acredita que há uma boa chance de converter passageiros casuais em membros. Ela observa que os ciclistas casuais já estão cientes do programa Cyclistic e escolheram a Cyclistic para suas necessidades de mobilidade.

A Lily estabeleceu um objetivo claro: criar estratégias de marketing destinadas a converter passageiros casuais em membros anuais. Para fazer isso, no entanto, a equipe de analistas de marketing precisa entender melhor como os membros anuais e os passageiros casuais diferem, por que os passageiros casuais iriam querer adquirir um plano e como a mídia digital poderia afetar suas táticas de marketing. A Lily e sua equipe estão interessados em analisar os dados históricos de trajetos de bicicleta da Cyclistic para identificar tendências.

Regras do negócio

1. A empresa oferece três planos de preços a seus clientes: passes de viagem única, passes de dia inteiro e planos anuais.
2. As bicicletas podem ser desbloqueadas de uma estação e devolvidas a qualquer outra estação do sistema 24 horas por dia, 7 dias por semana.
3. Os clientes que adquirem passes de viagem única ou de dia inteiro são chamados de passageiros (ciclistas) casuais.
4. Os clientes que adquirem planos anuais são membros Cyclistic.
5. Os dados são parte do programa de compartilhamento de bicicletas da cidade de Chicago/EUA que são coletados por sensores nas estações de acoplamento.
6. Oferece bicicletas reclináveis, triciclos manuais e bicicletas de carga, tornando o compartilhamento de bicicletas mais inclusivo para pessoas com deficiência e ciclistas que não podem usar uma bicicleta padrão de duas rodas.
7. A maioria dos ciclistas opta por bicicletas tradicionais; cerca de 8% dos motociclistas usam as opções assistivas.
8. Os usuários da Cyclistic são mais propensos a pedalar por lazer, mas cerca de 30% utilizam as bicicletas para se deslocarem ao trabalho diariamente.

FASE 1 - PERGUNTAR

Objetivo:

Este projeto visa analisar os dados coletados ao longo de 12 meses (2T2019, 3T2019, 4T2019 e 1T2020) tendo como objetivo responder a pergunta-chave: “Como os membros anuais e os ciclistas casuais usam as bicicletas da Cyclistic de forma diferente?”

Entregáveis

Criar um relatório com as entregas a seguir:

1. Uma declaração clara da tarefa de negócios;
2. Uma descrição de todas as fontes de dados usadas;
3. Documentação de qualquer limpeza ou manipulação de dados;
4. Um resumo da sua análise;
5. Como justificar visualizações e descobertas-chave;
6. Suas três principais recomendações com base em sua análise;

Ferramenta/software/linguagem

Os dados serão saneados e processados com R, usando o RStudio.

FASE 2 - PREPARAÇÃO

1. Instalando e carregando pacotes R e bibliotecas

a. Os pacotes necessários para importação de dados e análises:

- **tidyverse** (conjunto de pacotes R ‘tidyverse’)
- **sqldf** (pacote R para executar instruções SQL em dataframes R)
- **lubridate** (pacote R para datas e horas e intervalos de tempo)
- **ggplot2** (pacote R para gráficos e visualizações)

```
install.packages('lubridate', repos = "http://cran.us.r-project.org")
```

```
## package 'lubridate' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'lubridate'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying D:
## \Program Files\R\R-4.2.2\library\00LOCK\lubridate\libs\x64\lubridate.dll to D:
## \Program Files\R\R-4.2.2\library\lubridate\libs\x64\lubridate.dll: Permission
## denied
```

```
## Warning: restored 'lubridate'
```

```
##
## The downloaded binary packages are in
##  C:\Users\XXXXX\AppData\Local\Temp\RtmpCMKGBT\downloaded_packages
```

```
library(lubridate)
```

```
## Carregando pacotes exigidos: timechange
```

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

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

```
install.packages('tidyverse', repos = "http://cran.us.r-project.org")
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\XXXXX\AppData\Local\Temp\RtmpCMKGBT\downloaded_packages
```

```
library(tidyverse)
```

```
## ── Attaching packages
## ──────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ─────────────────────────────────────── tidyverse_conflicts() ──
## ✕ lubridate::as.difftime() masks base::as.difftime()
## ✕ lubridate::date()        masks base::date()
## ✕ dplyr::filter()          masks stats::filter()
## ✕ lubridate::intersect()   masks base::intersect()
## ✕ dplyr::lag()             masks stats::lag()
## ✕ lubridate::setdiff()     masks base::setdiff()
## ✕ lubridate::union()       masks base::union()
```

```
install.packages('sqldf', repos = "http://cran.us.r-project.org")
```

```
## package 'sqldf' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\XXXXX\AppData\Local\Temp\RtmpCMKGBT\downloaded_packages
```

```
library(sqldf)
```

```
## Carregando pacotes exigidos: gsubfn
## Carregando pacotes exigidos: proto
## Carregando pacotes exigidos: RSQLite
```

```
library(ggplot2)
```

2. Definindo o diretório de trabalho para simplificar as chamadas para dados.

```
getwd() #Exibe o diretório de trabalho
```

```
## [1] "D:/Coursera_CIEE/Cenario1"
```

```
setwd("D:/Coursera_CIEE/Cenario1/Datasets")
```

#FASE 3 - PROCESSAR

1. Coleta dos dados

A coleta dos dados históricos de trajetos da Cyclistic dos últimos 12 meses (2T2019, 3T2019, 4T2019 e 1T2020) serão usados para analisar e identificar tendências.

Nota: Os dados são publicados como arquivos CSV e podem ser encontrados em Divvy Trip Data, um programa do Departamento de Transportes de Chicago, é um sistema de compartilhamento de bicicletas em Chicago e Evanston

Fonte dos dados: https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html)

Observação: os conjuntos de dados têm um nome diferente porque a Cyclistic é uma empresa fictícia. Para os propósitos deste estudo de caso, os conjuntos de dados são adequados e permitem que se responda às perguntas de negócios. Os dados foram disponibilizados pela Motivate International Inc. sob esta licença - https://www.divvybikes.com/data-license-agreement (https://www.divvybikes.com/data-license-agreement)).

2. Carregando os conjuntos de dados Divvy (arquivos csv), criaremos dataframes específicos para q(1..4):

```
q2_2019 <- read_csv("D:/Coursera_CIEE/Cenario1/Datasets/Divvy_Trips_2019_Q2.csv")
```

```
## Rows: 1108163 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): 03 - Rental Start Station Name, 02 - Rental End Station Name, User...
## dbl  (5): 01 - Rental Details Rental ID, 01 - Rental Details Bike ID, 03 - R...
## num  (1): 01 - Rental Details Duration In Seconds Uncapped
## dttm (2): 01 - Rental Details Local Start Time, 01 - Rental Details Local En...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q3_2019 <- read_csv("D:/Coursera_CIEE/Cenario1/Datasets/Divvy_Trips_2019_Q3.csv")
```

```
## Rows: 1640718 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q4_2019 <- read_csv("D:/Coursera_CIEE/Cenario1/Datasets/Divvy_Trips_2019_Q4.csv")
```

```
## Rows: 704054 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q1_2020 <- read_csv("D:/Coursera_CIEE/Cenario1/Datasets/Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 426887 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

FASE 4 - ANALISAR

1. Visualizando e comparando as colunas de cada um dos datasets:

```
colnames(q3_2019)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```
colnames(q4_2019)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```
colnames(q2_2019)
```

```
##  [1] "01 - Rental Details Rental ID"
##  [2] "01 - Rental Details Local Start Time"
##  [3] "01 - Rental Details Local End Time"
##  [4] "01 - Rental Details Bike ID"
##  [5] "01 - Rental Details Duration In Seconds Uncapped"
##  [6] "03 - Rental Start Station ID"
##  [7] "03 - Rental Start Station Name"
##  [8] "02 - Rental End Station ID"
##  [9] "02 - Rental End Station Name"
## [10] "User Type"
## [11] "Member Gender"
## [12] "05 - Member Details Member Birthday Year"
```

```
colnames(q1_2020)
```

```
##  [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"
```

Adotaremos os seguintes campos (design a partir de 1T2020) para consolidação em um único dataset:

- **ride_id**: id único de uma única viagem (string)
- **rideable_type**: tipo de bicicleta usada (string)
- **start_at**: data e hora em que a bicicleta foi desacoplada (datetime)
- **end_at**: data e hora em que a bicicleta foi encaixada (datetime)
- **start_station_name**: endereço de localização da estação de docking inicial (string)
- **start_station_id**: id da estação de docking inicial (string)
- **end_station_name**: endereço de localização da estação de docking final (string)
- **end_station_id**: id da estação de docking final (string)
- **start_lat**: latitude inicial da estação de acoplamento (numérico)
- **start_lng**: longitude da estação de acoplamento (numérico)
- **end_lat**: latitude final da estação de acoplamento (numérico)
- **end_lng**: longitude final da estação de acoplamento (numérico)
- **member_casual**: tipo de passageiro (string)

Nota: Embora os nomes das colunas nos datasets não precisem estar na mesma ordem, elas precisam ser equivalentes antes de serem combinado em um único dataset.

2. Renomeando os campos de q2_2019, q3_2019 e q4_2019 para torná-las consistentes com q1_2020.

```
(q4_2019 <- rename(q4_2019
                   ,ride_id = trip_id
                   ,rideable_type = bikeid
                   ,started_at = start_time
                   ,ended_at = end_time
                   ,start_station_name = from_station_name
                   ,start_station_id = from_station_id
                   ,end_station_name = to_station_name
                   ,end_station_id = to_station_id
                   ,member_casual = usertype))
```

```
## # A tibble: 704,054 × 12
##     ride_id started_at          ended_at            rideable_t…¹ tripd…² start…³
##       <dbl> <dttm>              <dttm>                     <dbl>   <dbl>   <dbl>
##  1 25223640 2019-10-01 00:01:39 2019-10-01 00:17:20         2215     940      20
##  2 25223641 2019-10-01 00:02:16 2019-10-01 00:06:34         6328     258      19
##  3 25223642 2019-10-01 00:04:32 2019-10-01 00:18:43         3003     850      84
##  4 25223643 2019-10-01 00:04:32 2019-10-01 00:43:43         3275    2350     313
##  5 25223644 2019-10-01 00:04:34 2019-10-01 00:35:42         5294    1867     210
##  6 25223645 2019-10-01 00:04:38 2019-10-01 00:10:51         1891     373     156
##  7 25223646 2019-10-01 00:04:52 2019-10-01 00:22:45         1061    1072      84
##  8 25223647 2019-10-01 00:04:57 2019-10-01 00:29:16         1274    1458     156
##  9 25223648 2019-10-01 00:05:20 2019-10-01 00:29:18         6011    1437     156
## 10 25223649 2019-10-01 00:05:20 2019-10-01 02:23:46         2957    8306     336
## # … with 704,044 more rows, 6 more variables: start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>, and abbreviated variable names
## #   ¹rideable_type, ²tripduration, ³start_station_id
```

```
(q3_2019 <- rename(q3_2019
                   ,ride_id = trip_id
                   ,rideable_type = bikeid
                   ,started_at = start_time
                   ,ended_at = end_time
                   ,start_station_name = from_station_name
                   ,start_station_id = from_station_id
                   ,end_station_name = to_station_name
                   ,end_station_id = to_station_id
                   ,member_casual = usertype))
```

```
## # A tibble: 1,640,718 × 12
##     ride_id started_at          ended_at            rideable_t…¹ tripd…² start…³
##       <dbl> <dttm>              <dttm>                     <dbl>   <dbl>   <dbl>
##  1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41         3591    1214     117
##  2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44         5353    1048     381
##  3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42         6180    1554     313
##  4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10         5540    1503     313
##  5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26         6014    1213     168
##  6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31         4941     310     300
##  7 23479394 2019-07-01 00:02:24 2019-07-01 00:23:12         3770    1248     168
##  8 23479395 2019-07-01 00:02:26 2019-07-01 00:28:16         5442    1550     313
##  9 23479396 2019-07-01 00:02:34 2019-07-01 00:28:57         2957    1583      43
## 10 23479397 2019-07-01 00:02:45 2019-07-01 00:29:14         6091    1589      43
## # … with 1,640,708 more rows, 6 more variables: start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>, and abbreviated variable names
## #   ¹rideable_type, ²tripduration, ³start_station_id
```

```
(q2_2019 <- rename(q2_2019
                  ,ride_id = "01 - Rental Details Rental ID"
                  ,started_at = "01 - Rental Details Local Start Time"
                  ,ended_at = "01 - Rental Details Local End Time"
                  ,rideable_type = "01 - Rental Details Bike ID"
                  ,tripduration = "01 - Rental Details Duration In Seconds Uncapped"
                  ,start_station_name = "03 - Rental Start Station Name"
                  ,start_station_id = "03 - Rental Start Station ID"
                  ,end_station_name = "02 - Rental End Station Name"
                  ,end_station_id = "02 - Rental End Station ID"
                  ,member_casual = "User Type"
                  ,gender = "Member Gender"
                  ,birthyear = "05 - Member Details Member Birthday Year"))
```

```
## # A tibble: 1,108,163 × 12
##     ride_id started_at          ended_at            rideable_t…¹ tripd…² start…³
##       <dbl> <dttm>              <dttm>                     <dbl>   <dbl>   <dbl>
##  1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48         6251     446      81
##  2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30         6226    1048     317
##  3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19         5649     252     283
##  4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58         4151     357      26
##  5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13         3270    1007     202
##  6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56         3123     257     420
##  7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41         6418     548     503
##  8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11         4513     383     260
##  9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44         3280    2137     211
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39         5534    2120     211
## # … with 1,108,153 more rows, 6 more variables: start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>, and abbreviated variable names
## #   ¹rideable_type, ²tripduration, ³start_station_id
```

3. Inspecionando os datasets a procura de incongruências.

```
str(q1_2020)
```

```
## spc_tbl_ [426,887 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472C
A96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike" "docked_bike" "docked_bi
ke" ...
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2020-01-30 14:22:
39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2020-01-30 14:26:
22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St & Montrose Ave"
"Broadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport Ave & Irving Park
Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ...
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. 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()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q4_2019)
```

```
## spc_tbl_ [704,054 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:704054] 25223640 25223641 25223642 25223643 25223644 ...
##  $ started_at        : POSIXct[1:704054], format: "2019-10-01 00:01:39" "2019-10-01 00:02:
16" ...
##  $ ended_at          : POSIXct[1:704054], format: "2019-10-01 00:17:20" "2019-10-01 00:06:
34" ...
##  $ rideable_type     : num [1:704054] 2215 6328 3003 3275 5294 ...
##  $ tripduration      : num [1:704054] 940 258 850 2350 1867 ...
##  $ start_station_id  : num [1:704054] 20 19 84 313 210 156 84 156 156 336 ...
##  $ start_station_name: chr [1:704054] "Sheffield Ave & Kingsbury St" "Throop (Loomis) St &
Taylor St" "Milwaukee Ave & Grand Ave" "Lakeview Ave & Fullerton Pkwy" ...
##  $ end_station_id    : num [1:704054] 309 241 199 290 382 226 142 463 463 336 ...
##  $ end_station_name  : chr [1:704054] "Leavitt St & Armitage Ave" "Morgan St & Polk St" "W
abash Ave & Grand Ave" "Kedzie Ave & Palmer Ct" ...
##  $ member_casual     : chr [1:704054] "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
##  $ gender            : chr [1:704054] "Male" "Male" "Female" "Male" ...
##  $ birthyear         : num [1:704054] 1987 1998 1991 1990 1987 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_datetime(format = ""),
##   ..   end_time = col_datetime(format = ""),
##   ..   bikeid = col_double(),
##   ..   tripduration = col_number(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   usertype = col_character(),
##   ..   gender = col_character(),
##   ..   birthyear = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q3_2019)
```

```
## spc_tbl_ [1,640,718 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:1640718] 23479388 23479389 23479390 23479391 23479392 ...
##  $ started_at        : POSIXct[1:1640718], format: "2019-07-01 00:00:27" "2019-07-01 00:0
1:16" ...
##  $ ended_at          : POSIXct[1:1640718], format: "2019-07-01 00:20:41" "2019-07-01 00:1
8:44" ...
##  $ rideable_type     : num [1:1640718] 3591 5353 6180 5540 6014 ...
##  $ tripduration      : num [1:1640718] 1214 1048 1554 1503 1213 ...
##  $ start_station_id  : num [1:1640718] 117 381 313 313 168 300 168 313 43 43 ...
##  $ start_station_name: chr [1:1640718] "Wilton Ave & Belmont Ave" "Western Ave & Monroe S
t" "Lakeview Ave & Fullerton Pkwy" "Lakeview Ave & Fullerton Pkwy" ...
##  $ end_station_id    : num [1:1640718] 497 203 144 144 62 232 62 144 195 195 ...
##  $ end_station_name  : chr [1:1640718] "Kimball Ave & Belmont Ave" "Western Ave & 21st St"
"Larrabee St & Webster Ave" "Larrabee St & Webster Ave" ...
##  $ member_casual     : chr [1:1640718] "Subscriber" "Customer" "Customer" "Customer" ...
##  $ gender            : chr [1:1640718] "Male" NA NA NA ...
##  $ birthyear         : num [1:1640718] 1992 NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_datetime(format = ""),
##   ..   end_time = col_datetime(format = ""),
##   ..   bikeid = col_double(),
##   ..   tripduration = col_number(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   usertype = col_character(),
##   ..   gender = col_character(),
##   ..   birthyear = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q2_2019)
```

```
## spc_tbl_ [1,108,163 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:1108163] 22178529 22178530 22178531 22178532 22178533 ...
##  $ started_at        : POSIXct[1:1108163], format: "2019-04-01 00:02:22" "2019-04-01 00:0
3:02" ...
##  $ ended_at          : POSIXct[1:1108163], format: "2019-04-01 00:09:48" "2019-04-01 00:2
0:30" ...
##  $ rideable_type     : num [1:1108163] 6251 6226 5649 4151 3270 ...
##  $ tripduration      : num [1:1108163] 446 1048 252 357 1007 ...
##  $ start_station_id  : num [1:1108163] 81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr [1:1108163] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle
St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:1108163] 56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr [1:1108163] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt
Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:1108163] "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
##  $ gender            : chr [1:1108163] "Male" "Female" "Male" "Male" ...
##  $ birthyear         : num [1:1108163] 1975 1984 1990 1993 1992 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `01 - Rental Details Rental ID` = col_double(),
##   ..   `01 - Rental Details Local Start Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Local End Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Bike ID` = col_double(),
##   ..   `01 - Rental Details Duration In Seconds Uncapped` = col_number(),
##   ..   `03 - Rental Start Station ID` = col_double(),
##   ..   `03 - Rental Start Station Name` = col_character(),
##   ..   `02 - Rental End Station ID` = col_double(),
##   ..   `02 - Rental End Station Name` = col_character(),
##   ..   `User Type` = col_character(),
##   ..   `Member Gender` = col_character(),
##   ..   `05 - Member Details Member Birthday Year` = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

3.1 Como ride_id e rideable_type nestes conjuntos de dados se apresentam como numéricos eles serão convertidos em caracteres para que eles possam ser agrupados corretamente:

```
q4_2019 <-  mutate(q4_2019, ride_id = as.character(ride_id),rideable_type = as.character(ride
able_type))
```

```
q3_2019 <-  mutate(q3_2019, ride_id = as.character(ride_id),rideable_type = as.character(ride
able_type))
```

```
q2_2019 <-  mutate(q2_2019, ride_id = as.character(ride_id),rideable_type = as.character(ride
able_type))
```

3.2 Combinando os datasets de trimestres individuais em um único dataset.

```
all_trips <- bind_rows(q2_2019, q3_2019, q4_2019, q1_2020)
str(all_trips)
```

```
## tibble [3,879,822 × 16] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...
##  $ started_at        : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:0
3:02" ...
##  $ ended_at          : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:2
0:30" ...
##  $ rideable_type     : chr [1:3879822] "6251" "6226" "5649" "4151" ...
##  $ tripduration      : num [1:3879822] 446 1048 252 357 1007 ...
##  $ start_station_id  : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle
St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt
Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:3879822] "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
##  $ gender            : chr [1:3879822] "Male" "Female" "Male" "Male" ...
##  $ birthyear         : num [1:3879822] 1975 1984 1990 1993 1992 ...
##  $ start_lat         : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
##  $ start_lng         : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
##  $ end_lat           : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
##  $ end_lng           : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
```

LIMPEZA E TRATAMENTO DOS DADOS

1. As seguintes ações serão realizadas para inspecionar o novo dataset.

```
head(all_trips) # Visualizar as primeiras 6 linhas do dataset.
```

```
## # A tibble: 6 × 16
##   ride_id  started_at          ended_at            rideable_type tripd…¹ start…²
##   <chr>    <dttm>              <dttm>              <chr>           <dbl>   <dbl>
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251              446      81
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226             1048     317
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649              252     283
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151              357      26
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270             1007     202
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123              257     420
## # … with 10 more variables: start_station_name <chr>, end_station_id <dbl>,
## #   end_station_name <chr>, member_casual <chr>, gender <chr>, birthyear <dbl>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, and
## #   abbreviated variable names ¹tripduration, ²start_station_id
```

```
colnames(all_trips) # Listar os nomes de cada coluna.
```

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

```
nrow(all_trips) # Quantas linhas existem no dataset?
```

```
## [1] 3879822
```

```
dim(all_trips)  # Dimensões do dataset?
```

```
## [1] 3879822      16
```

```
str(all_trips) # Visualizar a lista de colunas e tipos de dados (numéricos, caracteres, et
c.).
```

```
## tibble [3,879,822 × 16] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...
##  $ started_at        : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:0
3:02" ...
##  $ ended_at          : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:2
0:30" ...
##  $ rideable_type     : chr [1:3879822] "6251" "6226" "5649" "4151" ...
##  $ tripduration      : num [1:3879822] 446 1048 252 357 1007 ...
##  $ start_station_id  : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle
St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt
Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:3879822] "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
##  $ gender            : chr [1:3879822] "Male" "Female" "Male" "Male" ...
##  $ birthyear         : num [1:3879822] 1975 1984 1990 1993 1992 ...
##  $ start_lat         : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
##  $ start_lng         : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
##  $ end_lat           : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
##  $ end_lng           : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...
```

```
summary(all_trips) # Resumo estatístico dos dados.
```

```
##    ride_id            started_at
##  Length:3879822     Min.   :2019-04-01 00:02:22.00
##  Class :character   1st Qu.:2019-06-23 07:49:09.25
##  Mode  :character   Median :2019-08-14 17:43:38.00
##                     Mean   :2019-08-26 00:49:59.38
##                     3rd Qu.:2019-10-12 12:10:21.00
##                     Max.   :2020-03-31 23:51:34.00
##
##     ended_at                      rideable_type       tripduration
##  Min.   :2019-04-01 00:09:48.00   Length:3879822     Min.   :     61
##  1st Qu.:2019-06-23 08:20:27.75   Class :character   1st Qu.:    424
##  Median :2019-08-14 18:02:04.00   Mode  :character   Median :    735
##  Mean   :2019-08-26 01:14:37.06                      Mean   :   1496
##  3rd Qu.:2019-10-12 12:36:16.75                      3rd Qu.:   1330
##  Max.   :2020-05-19 20:10:34.00                      Max.   :9056633
##                                                      NA's   :426887
##  start_station_id start_station_name end_station_id  end_station_name
##  Min.   :  1.0    Length:3879822     Min.   :  1.0   Length:3879822
##  1st Qu.: 77.0    Class :character   1st Qu.: 77.0   Class :character
##  Median :174.0    Mode  :character   Median :174.0   Mode  :character
##  Mean   :202.9                       Mean   :203.8
##  3rd Qu.:291.0                       3rd Qu.:291.0
##  Max.   :675.0                       Max.   :675.0
##                                      NA's   :1
##  member_casual         gender            birthyear        start_lat
##  Length:3879822     Length:3879822     Min.   :1759     Min.   :42
##  Class :character   Class :character   1st Qu.:1980     1st Qu.:42
##  Mode  :character   Mode  :character   Median :1988     Median :42
##                                        Mean   :1984     Mean   :42
##                                        3rd Qu.:1992     3rd Qu.:42
##                                        Max.   :2014     Max.   :42
##                                        NA's   :947615   NA's   :3452935
##    start_lng          end_lat           end_lng
##  Min.   :-88       Min.   :42        Min.   :-88
##  1st Qu.:-88       1st Qu.:42        1st Qu.:-88
##  Median :-88       Median :42        Median :-88
##  Mean   :-88       Mean   :42        Mean   :-88
##  3rd Qu.:-88       3rd Qu.:42        3rd Qu.:-88
##  Max.   :-88       Max.   :42        Max.   :-88
##  NA's   :3452935   NA's   :3452936   NA's   :3452936
```

```
sum(is.na(all_trips)) # Identificar a quantidade de valores nulos no dataset.
```

```
## [1] 16152628
```

Existem alguns problemas que precisaremos corrigir:

1. Remover os campos start_lat, start_lng, end_lat, end_lng, pois, esses dados foram incluídos a partir de 2020.
2. Remover os campos tripduration, birthyear e gender por apresentarem campos nulls (NA). Alternativamente - all_trips_clean <- drop_na(all_trips)

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender, tripduration))
```

Inspecionando o conjunto de dados após descartar colunas indesejadas:

```
head(all_trips)
```

```
## # A tibble: 6 × 9
##   ride_id  started_at          ended_at            rideable_type start…¹ start…²
##   <chr>    <dttm>              <dttm>              <chr>           <dbl> <chr>
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251               81 Daley …
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226              317 Wood S…
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649              283 LaSall…
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151               26 McClur…
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270              202 Halste…
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123              420 Ellis …
## # … with 3 more variables: end_station_id <dbl>, end_station_name <chr>,
## #   member_casual <chr>, and abbreviated variable names ¹start_station_id,
## #   ²start_station_name
```

```
str(all_trips)
```

```
## tibble [3,879,822 × 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...
##  $ started_at        : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:0
3:02" ...
##  $ ended_at          : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:2
0:30" ...
##  $ rideable_type     : chr [1:3879822] "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle
St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt
Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:3879822] "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
```

```
dim(all_trips)
```

```
## [1] 3879822       9
```

3. Na coluna “member_casual”, há dois nomes para membros (“member” e “Subscriber”) e dois nomes para passageiros casuais (“Customer” e “casual”).

```
table(all_trips$member_casual)
```

```
##
##     casual   Customer     member Subscriber
##      48480     857474     378407    2595461
```

Nota: Antes de 2020, a Divvy usava rótulos diferentes para esses dois tipos de dados, para termos o dataset consistente com a atual nomenclatura (rótulo) atual (2020), precisaremos consolidá-los de quatro para dois rótulos.

a. Na coluna “member_casual”, substituiremos “Subscriber” por “member” e “Customer” por “casual”.

```
all_trips <-  all_trips %>%
  mutate(member_casual = recode(member_casual,"Customer" = "casual","Subscriber" = "member"))
```

b. Verificando se o número adequado de observações foi reatribuída corretamente.

```
table(all_trips$member_casual)
```

```
##
##  casual  member
##  905954 2973868
```

4. As configurações de localidade são dependentes do sistema operacional. A fim de mantermos o padrão americano (“en_US”) nas análises precisamos verificar através do comando a seguir:

```
Sys.getlocale("LC_TIME")
```

```
## [1] "Portuguese_Brazil.utf8"
```

a.1 Como a saída foi “Portuguese_Brazil.utf8” (day_of_week = domingo segunda-feira terça-feira quarta-feira quinta-feira sexta-feira sábado), executaremos o comando a seguir para adotar o padrão (US) English para os dias da semana.

```
Sys.setlocale("LC_TIME", "en_US")
```

```
## [1] "en_US"
```

5. Os dados só podem ser agregados no ride-level, o que também é granular. Com esta finalidade adicionaremos as colunas data (date), mês (month), dia (day), ano (year) e hora de início (started_hour) de cada viagem - o formato padrão adotado será o yyyy-mm-dd (EN_US). Isso permitirá agregar dados de viagem para cada mês, dia ou ano.

```
all_trips$date <- as.Date(all_trips$started_at)
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
all_trips$started_hour <- format(all_trips$started_at, "%H")
```

a. Visualizando os dados relacionados aos dias da semana.

```
table(all_trips$day_of_week)
```

```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##    575723    576648    497501    449271    587524    599636    593519
```

6. Adicionado um campo adicional para calcular a duração de cada passeio, uma vez que os dados do arquivo referente ao 1T2020 (Divvy_Trips_2020_Q1.csv) não possui a coluna “tripduration”. Vamos adicionar “ride_length” a todo o dataset para alcançar a consistência dos dados.

a. Adicionando um cálculo de “ride_duration” ao all_trips (valores em segundos)

```
all_trips$ride_duration <- difftime(all_trips$ended_at,all_trips$started_at)
```

b. Verificando o dataset após a adição da nova coluna (ride_duration).

```
str(all_trips)
```

```
## tibble [3,879,822 × 16] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...
##  $ started_at        : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:0
3:02" ...
##  $ ended_at          : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:2
0:30" ...
##  $ rideable_type     : chr [1:3879822] "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle
St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt
Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:3879822] "member" "member" "member" "member" ...
##  $ date              : Date[1:3879822], format: "2019-04-01" "2019-04-01" ...
##  $ month             : chr [1:3879822] "04" "04" "04" "04" ...
##  $ day               : chr [1:3879822] "01" "01" "01" "01" ...
##  $ year              : chr [1:3879822] "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr [1:3879822] "Monday" "Monday" "Monday" "Monday" ...
##  $ started_hour      : chr [1:3879822] "00" "00" "00" "00" ...
##  $ ride_duration     : 'difftime' num [1:3879822] 446 1048 252 357 ...
##   ..- attr(*, "units")= chr "secs"
```

c. Conversão “ride_length” de ‘difftime’ para ‘number’ para que seja possível executar cálculos sobre os dados.

```
is.numeric(all_trips$ride_duration)
```

```
## [1] FALSE
```

```
all_trips$ride_duration <- as.numeric(as.character(all_trips$ride_duration))
```

```
is.numeric(all_trips$ride_duration)
```

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

7. Existem alguns passeios em que a duração da viagem (ride_length) aparece como negativa, incluindo várias centenas de passeios onde Divvy tirou bicicletas de circulação por razões de Controle de Qualidade. E a coluna start_station_name’ contém entradas inválidas de HQ QR.

a. Removendo dados “ruins”

Criaremos uma nova versão do dataset (v2), uma vez que os dados relacionados as bicicletas divvy retiradas para controle de qualidade e ’ride_duration que é negativo serão sendo removidas.

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_duration
<0),]
```

b. Pesquisando pela existência de valores nulls.

```
sum(is.na(all_trips_v2))
```

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

```
dim(all_trips_v2)
```

```
## [1] 3876042      16
```

Nota: A remoção dos valores nulos e negativos não afetam a análise, pois estamos lidando apenas com ride_duration, dia, mês e ano.

ANÁLISE DESCRITIVA

1. Análise descritiva sobre ride_length (tempo em segundos).

- média (duração total do percurso / passeios);
- número de ponto médio na matriz ascendente de duração do passeio;
- passeio mais longo;
- passeio mais curto;

```
summary(all_trips_v2$ride_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     412     712    1479    1289 9387024
```

2. Comparando membros (members) e usuários casuais (casual).

```
aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_duration
## 1                     casual                  3552.7502
## 2                     member                   850.0662
```

```
aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_duration
## 1                     casual                       1546
## 2                     member                        589
```

```
aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_duration
## 1                     casual                    9387024
## 2                     member                    9056634
```

```
aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_duration
## 1                     casual                          2
## 2                     member                          1
```

3. Visualizando o tempo médio de viagem por dia para membros vs usuários casuais.

```
aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual + all_trips_v2$day_of_week,
FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week
## 1                      casual                   Friday
## 2                      member                   Friday
## 3                      casual                   Monday
## 4                      member                   Monday
## 5                      casual                 Saturday
## 6                      member                 Saturday
## 7                      casual                   Sunday
## 8                      member                   Sunday
## 9                      casual                 Thursday
## 10                     member                 Thursday
## 11                     casual                  Tuesday
## 12                     member                  Tuesday
## 13                     casual                Wednesday
## 14                     member                Wednesday
##    all_trips_v2$ride_duration
## 1                   3773.8351
## 2                    824.5305
## 3                   3372.2869
## 4                    842.5726
## 5                   3331.9138
## 6                    968.9337
## 7                   3581.4054
## 8                    919.9746
## 9                   3682.9847
## 10                   823.9278
## 11                  3596.3599
## 12                   826.1427
## 13                  3718.6619
## 14                   823.9996
```

Observe que os dias da semana estão fora de ordem. Vamos corrigir isso.

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "T
uesday", "Wednesday", "Thursday", "Friday", "Saturday"))
```

4. Agora, vamos executar o tempo médio de viagem por cada dia para membros vs usuários casuais.

```
aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual + all_trips_v2$day_of_week,
FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week
## 1                      casual                   Sunday
## 2                      member                   Sunday
## 3                      casual                   Monday
## 4                      member                   Monday
## 5                      casual                  Tuesday
## 6                      member                  Tuesday
## 7                      casual                Wednesday
## 8                      member                Wednesday
## 9                      casual                 Thursday
## 10                     member                 Thursday
## 11                     casual                   Friday
## 12                     member                   Friday
## 13                     casual                 Saturday
## 14                     member                 Saturday
##    all_trips_v2$ride_duration
## 1                   3581.4054
## 2                    919.9746
## 3                   3372.2869
## 4                    842.5726
## 5                   3596.3599
## 6                    826.1427
## 7                   3718.6619
## 8                    823.9996
## 9                   3682.9847
## 10                   823.9278
## 11                  3773.8351
## 12                   824.5305
## 13                  3331.9138
## 14                   968.9337
```

5. Analisando os dados de passageiros por tipo e dia da semana.

a. criação do campo adicional para o dia da semana usando weekday();
b. agrupando por tipo de usuário e dia da semana;
c. calculando o número de viagens e a duração média

```
all_trips_weekday <- all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),average_duration = mean(ride_duration)) %>%
  arrange(member_casual, weekday)
```

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

```
(all_trips_weekday)
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              181293            3581.
##  2 casual        Mon              103296            3372.
##  3 casual        Tue               90510            3596.
##  4 casual        Wed               92457            3719.
##  5 casual        Thu              102679            3683.
##  6 casual        Fri              122404            3774.
##  7 casual        Sat              209543            3332.
##  8 member        Sun              267965             920.
##  9 member        Mon              472196             843.
## 10 member        Tue              508445             826.
## 11 member        Wed              500329             824.
## 12 member        Thu              484177             824.
## 13 member        Fri              452790             825.
## 14 member        Sat              287958             969.
```

6. Calculando o número de viagens e a duração média.

a. criação de campo adicional para o mês usando month();
b. agrupando por tipo de usuário e mês;
c. calculando o número de viagens e a duração média

```
all_trips_month <- all_trips_v2 %>%
  mutate(month=month(started_at, label = TRUE)) %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides=n(), average_duration=mean(ride_duration)) %>%
  arrange(member_casual, month)
```

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

```
(all_trips_month)
```

```
## # A tibble: 24 × 4
## # Groups:   member_casual [2]
##    member_casual month number_of_rides average_duration
##    <chr>         <ord>           <int>            <dbl>
##  1 casual        Jan              7785            9699.
##  2 casual        Feb             12314            7997.
##  3 casual        Mar             24615            4250.
##  4 casual        Apr             47744            3057.
##  5 casual        May             81624            3074.
##  6 casual        Jun            130218            2755.
##  7 casual        Jul            175632            3587.
##  8 casual        Aug            186889            4020.
##  9 casual        Sep            129173            3100.
## 10 casual        Oct             71035            3540.
## # … with 14 more rows
```

7. Analisando o uso horário de bikers por tipo de usuário

a. Quantidade de viagens que começaram em um determinado horário por membros casuais.

```
qtd_ride_casual_by_hour <- sqldf ("SELECT started_hour, COUNT(member_casual) AS number_of_cas
ual_riders
       FROM all_trips
       WHERE member_casual = 'casual'
       GROUP BY started_hour
       ORDER BY number_of_casual_riders DESC")
```

```
qtd_ride_casual_by_hour
```

```
##    started_hour number_of_casual_riders
## 1            17                   86809
## 2            16                   85432
## 3            15                   82647
## 4            14                   81357
## 5            13                   77710
## 6            12                   72035
## 7            18                   69299
## 8            11                   61902
## 9            19                   51181
## 10           10                   46430
## 11           20                   34930
## 12           09                   29831
## 13           21                   25433
## 14           08                   22479
## 15           22                   21421
## 16           23                   14339
## 17           07                   13335
## 18           00                    8363
## 19           06                    6291
## 20           01                    5501
## 21           02                    3361
## 22           05                    2690
## 23           03                    1982
## 24           04                    1196
```

b. Quantidade de viagens que começaram em um determinado horário por membros.

```
qtd_ride_member_by_hour <- sqldf ("SELECT started_hour, COUNT(member_casual) AS number_of_mem
ber_riders
      FROM all_trips
      WHERE member_casual = 'member'
      GROUP BY started_hour
      ORDER BY started_hour ASC")
```

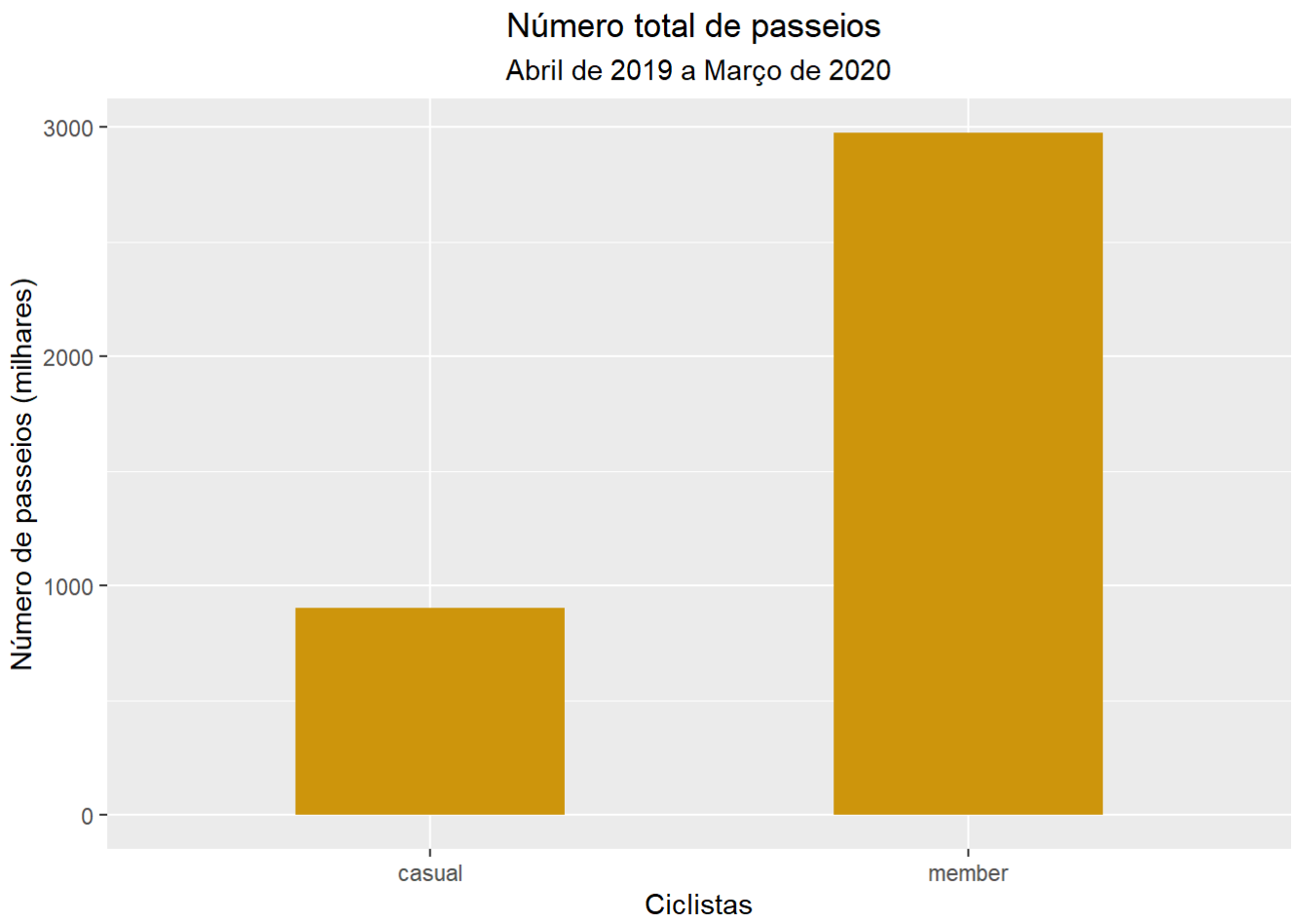
```
qtd_ride_member_by_hour
```

```
##    started_hour number_of_member_riders
## 1            00                  15749
## 2            01                   8974
## 3            02                   5230
## 4            03                   3546
## 5            04                   6686
## 6            05                  34443
## 7            06                 104094
## 8            07                 229602
## 9            08                 288164
## 10           09                 137069
## 11           10                 102288
## 12           11                 122331
## 13           12                 139549
## 14           13                 135236
## 15           14                 130495
## 16           15                 166568
## 17           16                 295834
## 18           17                 391278
## 19           18                 248234
## 20           19                 159256
## 21           20                 100127
## 22           21                  71863
## 23           22                  48800
## 24           23                  28452
```

FASE 5 - COMPARTILHAR

Gráfico 1. Total de passeios (12 meses) por tipo de usuário.

```
all_trips_v2 %>%
  count(member_casual, name = "number_of_trips") %>%
  ggplot(mapping = aes(x = member_casual, y = number_of_trips/1000, width = 0.5))+
  geom_bar(stat = "identity", fill = "darkgoldenrod3")+
  labs(title = "Número total de passeios ", subtitle = "Abril de 2019 a Março de 2020",
       x = "Ciclistas", y = "Número de passeios (milhares)")+
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5))
```

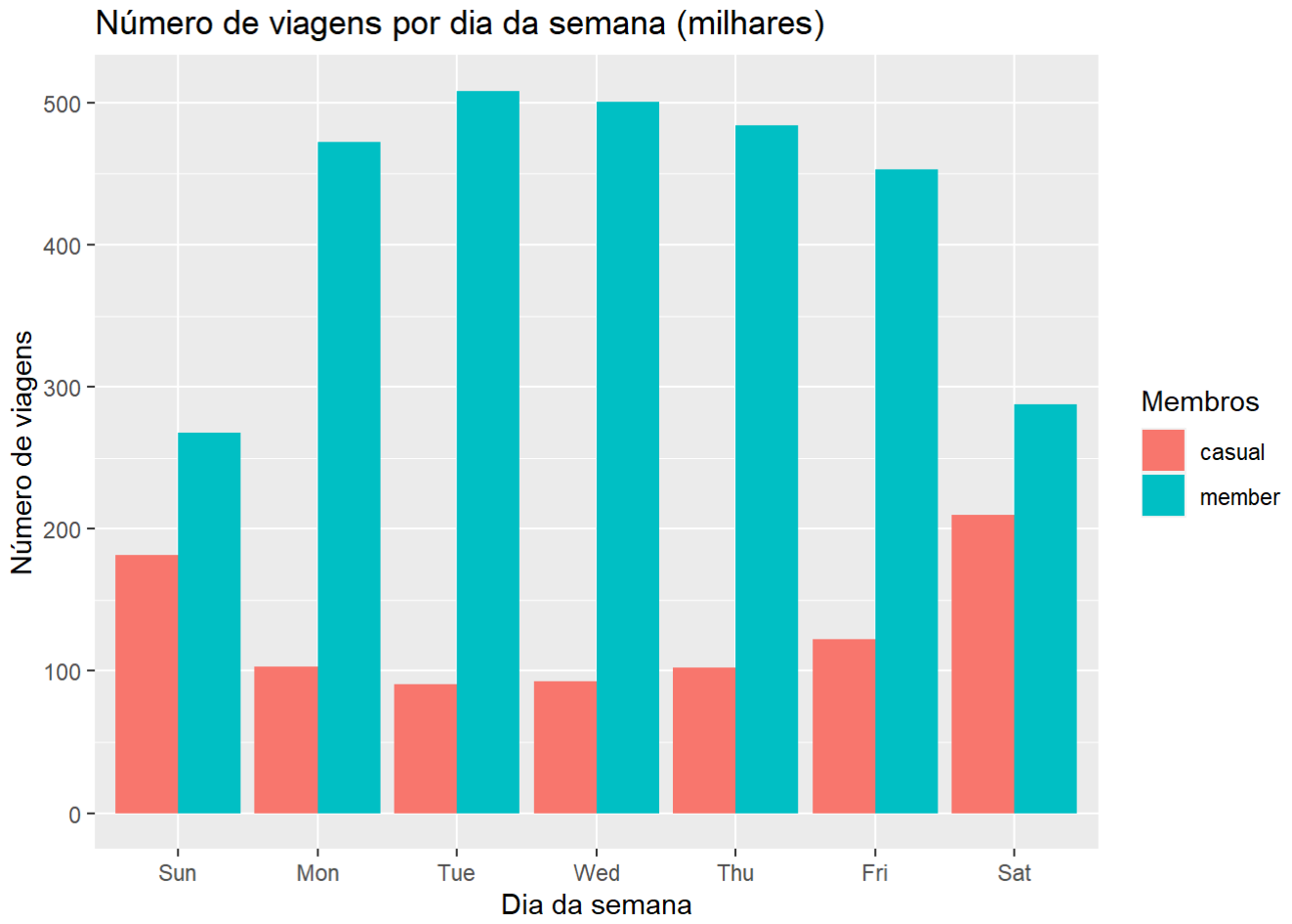


- A quantidade de passeios pelos membros é maior que os casuais no período de abril de 2019 a março de 2020.

Gráfico 2. Analisando o número de passeios por dia da semana e por tipo de membro.

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),average_duration = mean(ride_duration)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides/1000, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Número de viagens por dia da semana (milhares)", x="Dia da semana", y="Número
de viagens", fill="Membros")
```

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

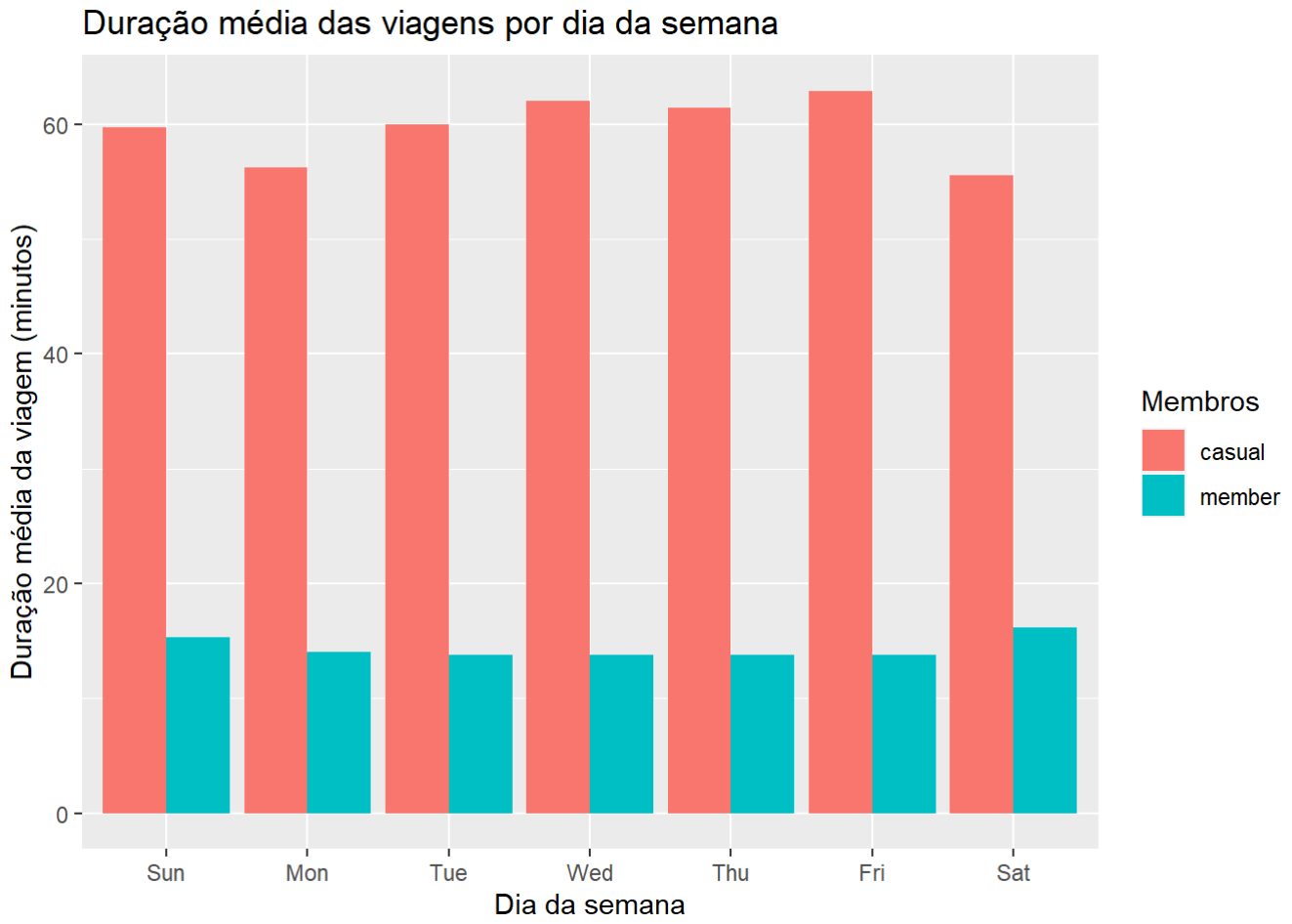


- Os ciclistas membros usam os serviços com mais frequência durante a semana do que os membros casuais.
- Há menos passeios por membros casuais durante a semana em comparação com fins de semana (sábado e domingo).

Gráfico 3. Analisando a duração média dos passeios por tipo de usuário e dias da semana

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),average_duration = mean(ride_duration)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration/60, fill = member_casual)) +
  # facet_wrap(~member_casual)+
  geom_col(position = "dodge") +
  labs(title = "Duração média das viagens por dia da semana", x="Dia da semana", y="Duração m
édia da viagem (minutos)", fill="Membros")
```

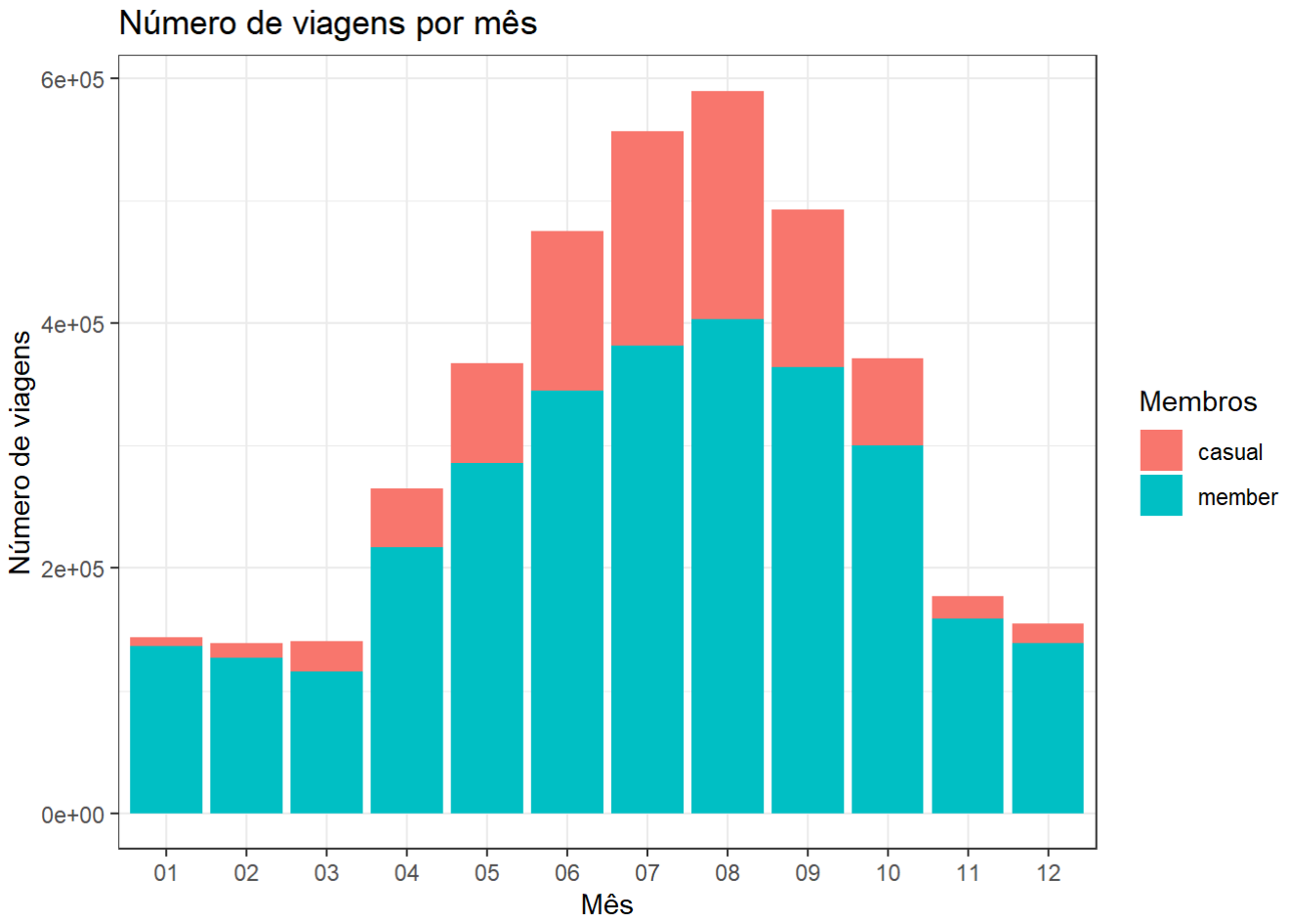
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



- A duração média dos passeios para membros casuais supera em muito a dos membros associados; isso pode ser atribuído ao uso de serviços de bicicleta para fins de lazer.
- A duração média do passeio dos membros permanece constante em torno de quase 15 minutos ao longo da semana.
- A duração média do passeio de membros casuais varia de mais de 50 minutos ao longo da semana.

Gráfico 4. Analisando o número de viagens por tipo de usuário e mês.

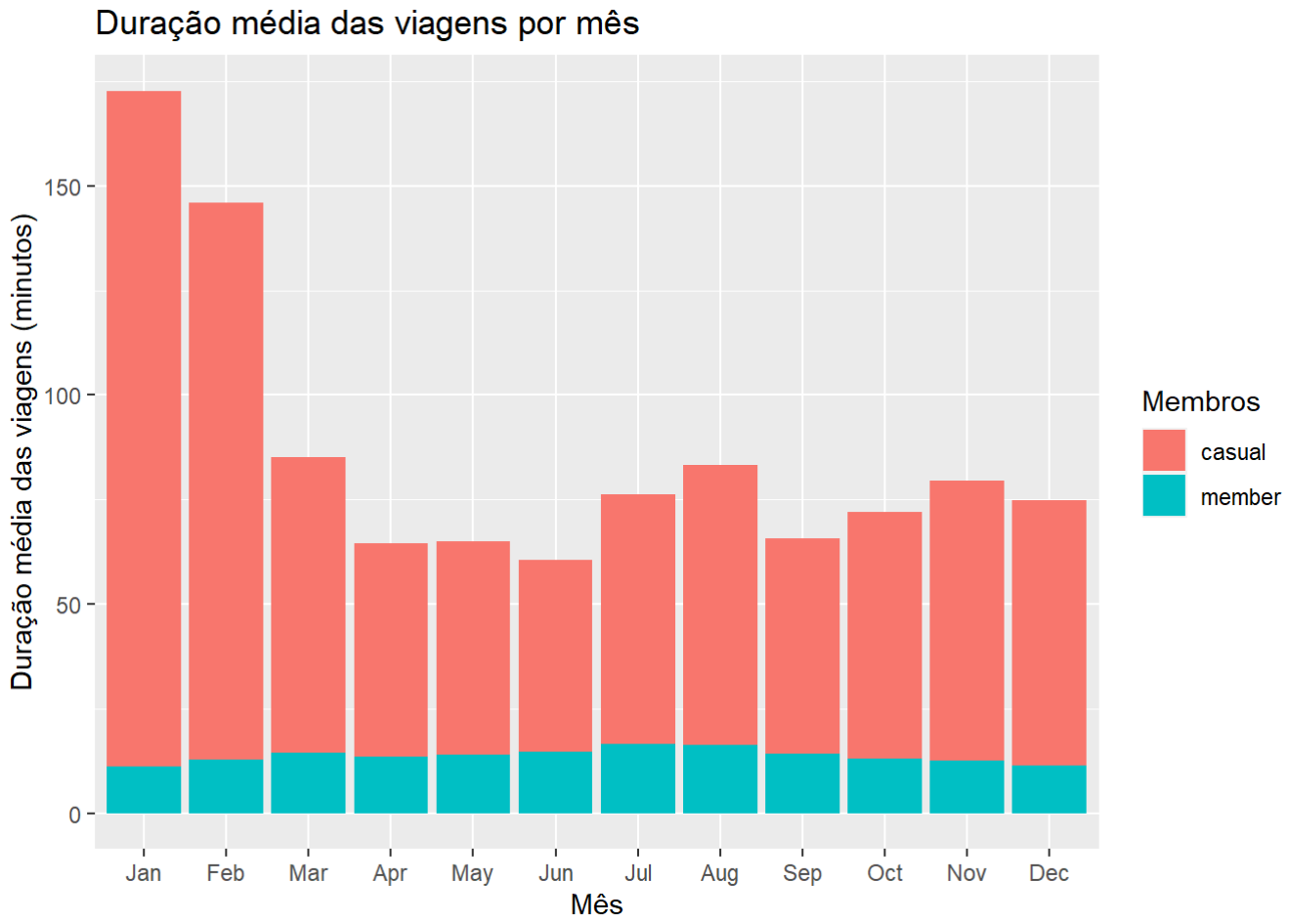
```
ggplot(all_trips_v2,aes(x=month, fill = member_casual))+
  theme_bw()+
  geom_bar()+
  # facet_wrap(~member_casual)+
  labs(title= "Número de viagens por mês", x = "Mês", y = "Número de viagens", fill="Membros"
)
```



- Tanto os ciclistas casuais quanto os membros associados usam as bicicletas com menor intensidade durante os meses mais frios. Um ponto a ser observado é que durante os meses mais quentes os ciclistas casuais o usam com mais frequência do que os membros associados, e de que os membros usam o serviço mais continuamente independente do período.
- O maior número de viagens para ciclistas casuais e membros ocorre entre junho e setembro de 2019.

Gráfico 5. Analisando a duração média das viagens por tipo de usuário e mês

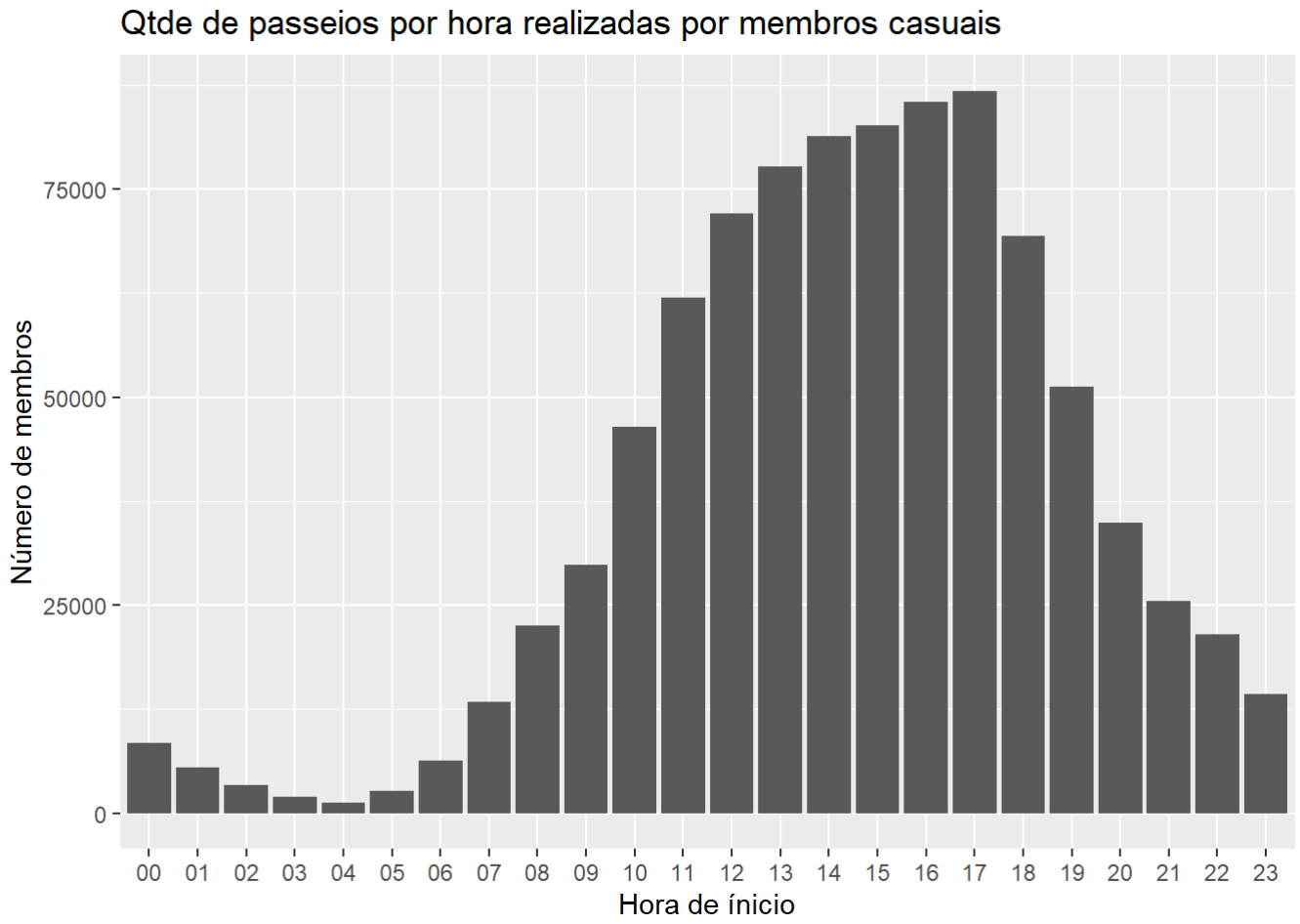
```
ggplot(all_trips_month)+
  geom_col(aes(x=month, y=average_duration/60, fill=member_casual))+
  labs(title= "Duração média das viagens por mês", x = "Mês", y = "Duração média das viagens
 (minutos)", fill="Membros")
```



- Os ciclistas casuais usam as bicicletas por mais tempo do que os membros associados, contudo os membros são mais consistentes no uso das bicicletas.
- A duração média da viagem para membros é de 15 minutos, cerca de 3 a 4 vezes menor que a de usuários casuais.

Gráfico 6. Analisando o número de passeios de bicicleta por hora de início para ciclistas casuais

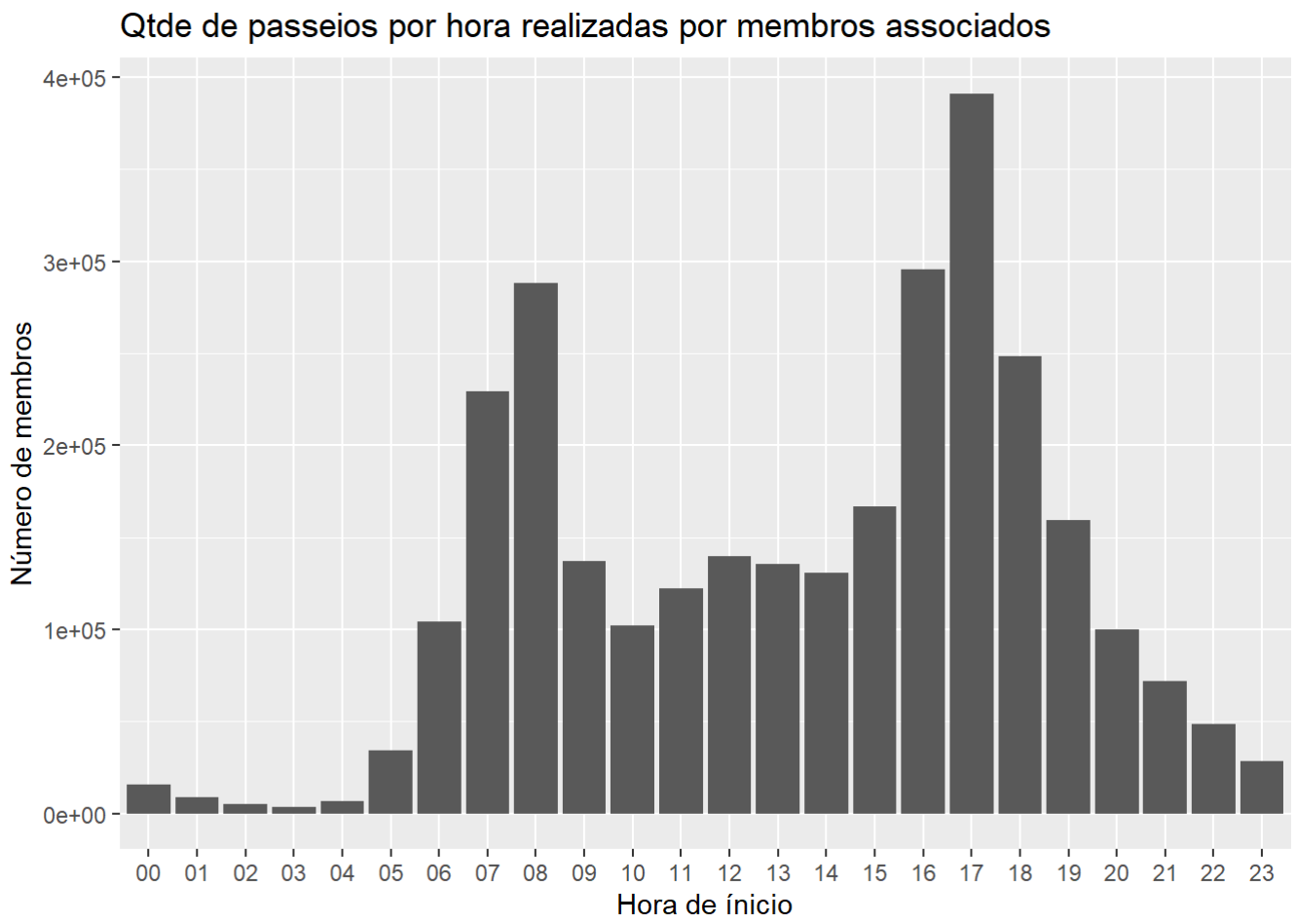
```
ggplot(qtd_ride_casual_by_hour, ) +
  geom_col(aes(x=started_hour,y=number_of_casual_riders)) +
  labs(title = "Qtde de passeios por hora realizadas por membros casuais",x="Hora de ínicio",
y="Número de membros")
```



- Geralmente, a maioria das bicicletas é usada entre 10h às 19h. As hipótese possíveis seriam para fins de uso escolar, lazer (fim do dia) e atividades esportivas.

Gráfico 7. Analisando o número de passeios de bicicleta por hora de início para ciclistas menbros

```
ggplot(qtd_ride_member_by_hour) +
  geom_col(aes(x=started_hour,y=number_of_member_riders)) +
  labs(title = "Qtde de passeios por hora realizadas por membros associados",x="Hora de ínici
o",y="Número de membros")
```



- Geralmente, a maioria das bicicletas é usada entre 07h e 08h, e entre 16h e 18h. As hipótese possíveis seriam para fins de atividades esportivas e lazer (início e fim do dia).

FASE 6 - AGIR

Os membros usam consistentemente as bicicletas todos os dias por um curto período de tempo, por outro lado os membros casuais as usam de forma mais irregular, contudo por períodos maiores de tempo, onde o uso das bicicletas durante a semana é menor e apresentando um aumento gradativo durante os dias de final de semana e feriados. Essas observações sugerem que os membros utilizam o serviço mais como meio de transporte, enquanto os ciclistas casuais podem estar utilizando-as principalmente para lazer e recreação.

Sugestões:

a. Realizar campanhas publicitárias específicas entre os meses de junho a setembro para os ciclistas casuais oferecendo um pacote de serviços com descontos;
b. Realizar campanhas publicitárias específicas convidando os ciclistas a realizarem atividades esportivas em horários entre 6h e 8h e entre 16h e 19h, inclusise objetivando sustentabilidade e economia energética;

Um exemplo seria apresentar nossas bicicletas como uma alternativa ecológica aos carros e transporte público e criar um programa de conscientização ambiental, com recompensas para membros anuais que usam o serviço de forma consistente. Isso possibilitaria a oportunidade para membros casuais que estejam interessados em ecologia em realizar a mudança para membro associado.

c. Promoções e ofertas por tempo limitado objetivando os meses de baixo consumo e finais de semana específicos, poderiam atingir um número expressivo de membros casuais, que poderiam se tornar membros associados.
d. Campanhas de marketing poderiam ser planejadas para os meses mais quentes.

Nota: As campanhas podem incluir e-mail, mídias sociais e outros canais.

EXPORTAÇÂO DO ARQUIVO DE RESUMO

Criando um arquivo csv para visualização no Excel, Tableau ou em um software de apresentação.

```
counts <- aggregate(all_trips_v2$ride_duration ~ all_trips_v2$member_casual +
            all_trips_v2$day_of_week, FUN = mean)
```

```
write.csv(counts, file = 'D:/Coursera_CIEE/Cenario1/Datasets/avg_ride_length.csv')
```