Projeto final de Data Analytics do Google

Alan

2023-07-25

library(knitr)  
caminho\_imagem <- "C:/Users/Alanf/Downloads/Certificados Google/certificado-profissional-de-analise-de-dados-do-google.png"  
include\_graphics(caminho\_imagem)

 ## Introdução e Objetivo do Estudo Este projeto representa a parte principal do programa de certificação do Google Analytical.

Objetivo do Estudo O objetivo deste estudo de caso é analisar o comportamento dos membros anuais e dos passageiros não membros, com a finalidade de desenvolver uma estratégia de marketing que incentive os não membros a aderirem ao programa de membros.

## Perguntar

Para alcançar o objetivo proposto, o estudo busca responder às seguintes questões:

1. Qual é o padrão de comportamento dos membros anuais em comparação com os passageiros não membros?
2. Quais são os principais motivos pelos quais os passageiros não membros não aderem ao programa de membros?
3. Quais ações de marketing podem ser implementadas para atrair os não membros e incentivá-los a se tornarem membros anuais?

As respostas a essas perguntas serão fundamentais para orientar a estratégia futura de marketing e aumentar o número de adesões ao programa de membros.

1. Como os membros anuais e os ciclistas casuais usam as bicicletas ciclísticas de maneira diferente?
2. Por que os ciclistas casuais comprariam assinaturas anuais da Cyclistic?
3. Como a Cyclistic pode usar a mídia digital para influenciar ciclistas casuais a se tornarem membros?

## Preparar

O conjunto de dados referente ao trimestre nº 2 de 2019 ao trimestre nº 1 de 2020 foi baixado a partir do link fornecido e salvo na minha área de trabalho local.

## Processar

Os passos realizados na plataforma R para preparar o arquivo final para análise foram os seguintes:

1. Uniformização dos nomes das colunas: Todas as colunas no conjunto de dados foram padronizadas para terem o mesmo nome em todo o arquivo. Isso garante consistência nos dados e facilita a manipulação e análise.
2. Remoção de colunas extras: Antes de mesclar os arquivos, foram identificadas quaisquer colunas que não eram comuns a todos os arquivos e, em seguida, essas colunas foram removidas. Isso garante que apenas as informações relevantes para a análise sejam mantidas.
3. Mesclagem das colunas: As colunas foram mescladas em um único arquivo usando a junção externa completa. Isso significa que todas as linhas dos arquivos foram mantidas, mesmo que não houvesse correspondência em alguma coluna específica.
4. Renomeação das linhas: A coluna que indicava se um registro era de um assinante/membro foi renomeada para “Membro”, enquanto a coluna que indicava se era de um cliente/casual foi renomeada para “Não Membro”. Essa renomeação torna os dados mais claros e compreensíveis.
5. Identificação e registro de outliers: Foram identificados quaisquer valores atípicos nos dados e eles foram registrados em uma coluna separada chamada “Outliers”. Essa informação é importante para avaliar a qualidade e a confiabilidade dos dados.

Após a conclusão desses passos, o arquivo foi extraído como um arquivo CSV e está pronto para ser usado na análise. O conjunto de dados preparado é uma valiosa fonte de informações para a análise final, que pode revelar insights importantes sobre o comportamento dos membros e não membros, bem como possíveis oportunidades de melhoria para o negócio.

## Pré-análise dos dados: criamos as seguintes colunas

1. Ride Length in Minutess – Isso foi feito subtraindo o horário de início do percurso até o horário final.
2. Weekday – calculado a partir da coluna da hora de início
3. Day name – calculado a partir da coluna da hora de início
4. Day number - calculado a partir da coluna da hora de início
5. Yea-Month-Quarter - calculado a partir da coluna da hora de início
6. Da mesma forma, o número do mês, o nome do mês e os trimestres foram criados

# Carregando Bibliotecas Necessárias  
  
library(tidyverse) # coleção de pacotes R projetados para tornar a manipulação e visualização de dados mais fácil e eficiente.

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate) # fornece funções para trabalhar com datas e horários de forma mais conveniente.  
  
  
library(dplyr) # Carregar o pacote "dplyr", que é uma parte do "tidyverse" e fornece uma gramática para a manipulação de dados.  
  
library(here) # fornece uma maneira mais conveniente de referenciar arquivos e diretórios dentro do projeto.

## here() starts at C:/Users/Alanf/Downloads

library(skimr)# fornece funções para criar resumos descritivos de conjuntos de dados.  
  
library(janitor) # fornece funções para a limpeza e transformação de dados.

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

library(hablar) # fornece funções para análise de texto.

##   
## Attaching package: 'hablar'  
##   
## The following object is masked from 'package:skimr':  
##   
## n\_unique  
##   
## The following object is masked from 'package:forcats':  
##   
## fct  
##   
## The following object is masked from 'package:dplyr':  
##   
## na\_if  
##   
## The following object is masked from 'package:tibble':  
##   
## num

library(gapminder) # informações sobre indicadores sociais e econômicos de países.  
  
library(scales) # fornece funções para formatar números e datas em gráficos.

##   
## Attaching package: 'scales'  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

library(ggrepel) # fornece funções para reposicionar automaticamente rótulos em gráficos ggplot2 para evitar sobreposições.  
  
library(conflicted) # é usado para gerenciar conflitos entre funções de pacotes diferentes.  
  
library(stats) # fornece funções estatísticas básicas e recursos essenciais. No entanto, precisamos carregar o pacote dplyr após carregar o pacote stats para evitar conflitos de função entre dplyr::filter e stats::filter.  
  
options(warn=-1) # os avisos não serão exibidos. Isso é útil quando você deseja suprimir avisos durante a execução do código.

## Lendo os Dataframes

# Load the data files  
divvy\_tips\_2019\_Q2 <- read.csv("C:/Users/Alanf/Downloads/Projeto\_Google\_Analytics/Divvy\_Trips\_2019\_Q2.csv")  
divvy\_tips\_2019\_Q3 <- read.csv("C:/Users/Alanf/Downloads/Projeto\_Google\_Analytics/Divvy\_Trips\_2019\_Q3.csv")  
divvy\_tips\_2019\_Q4 <- read.csv("C:/Users/Alanf/Downloads/Projeto\_Google\_Analytics/Divvy\_Trips\_2019\_Q4.csv")  
divvy\_tips\_2020\_Q1 <- read.csv("C:/Users/Alanf/Downloads/Projeto\_Google\_Analytics/Divvy\_Trips\_2020\_Q1.csv")

# Entendendo sobre os dados  
  
cat(" divvy\_tips\_2019\_Q2 Dataset: -","\n", "Number of Rows are", nrow(divvy\_tips\_2019\_Q2), "\n", "Number of Columns are", ncol(divvy\_tips\_2019\_Q2))

## divvy\_tips\_2019\_Q2 Dataset: -   
## Number of Rows are 1108163   
## Number of Columns are 12

cat("\n","---------------------------","\n")

##   
## ---------------------------

cat(" divvy\_tips\_2019\_Q3 Dataset: -","\n", "Number of Rows are", nrow(divvy\_tips\_2019\_Q3), "\n", "Number of Columns are", ncol(divvy\_tips\_2019\_Q3))

## divvy\_tips\_2019\_Q3 Dataset: -   
## Number of Rows are 1640718   
## Number of Columns are 12

cat("\n","---------------------------","\n")

##   
## ---------------------------

cat(" divvy\_tips\_2019\_Q4 Dataset: -","\n", "Number of Rows are", nrow(divvy\_tips\_2019\_Q4), "\n", "Number of Columns are", ncol(divvy\_tips\_2019\_Q4))

## divvy\_tips\_2019\_Q4 Dataset: -   
## Number of Rows are 704054   
## Number of Columns are 12

cat("\n","---------------------------","\n")

##   
## ---------------------------

cat(" divvy\_tips\_2020\_Q1 Dataset: -","\n", "Number of Rows are", nrow(divvy\_tips\_2020\_Q1), "\n", "Number of Columns are", ncol(divvy\_tips\_2020\_Q1))

## divvy\_tips\_2020\_Q1 Dataset: -   
## Number of Rows are 426887   
## Number of Columns are 13

# Revisando o nome das colunas em cada conjunto de dados fornecido   
  
cat("Columns Name in 2019\_Q2 Data Set:-", "\n","\n")

## Columns Name in 2019\_Q2 Data Set:-   
##

print(colnames(divvy\_tips\_2019\_Q2))

## [1] "X01...Rental.Details.Rental.ID"   
## [2] "X01...Rental.Details.Local.Start.Time"   
## [3] "X01...Rental.Details.Local.End.Time"   
## [4] "X01...Rental.Details.Bike.ID"   
## [5] "X01...Rental.Details.Duration.In.Seconds.Uncapped"  
## [6] "X03...Rental.Start.Station.ID"   
## [7] "X03...Rental.Start.Station.Name"   
## [8] "X02...Rental.End.Station.ID"   
## [9] "X02...Rental.End.Station.Name"   
## [10] "User.Type"   
## [11] "Member.Gender"   
## [12] "X05...Member.Details.Member.Birthday.Year"

cat("--------------------------------------------------","\n","\n")

## --------------------------------------------------   
##

cat("Columns Name in 2019\_Q3 Data Set:-", "\n","\n")

## Columns Name in 2019\_Q3 Data Set:-   
##

print(colnames(divvy\_tips\_2019\_Q3))

## [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"

cat("--------------------------------------------------","\n","\n")

## --------------------------------------------------   
##

cat("Columns Name in 2019\_Q4 Data Set:-", "\n","\n")

## Columns Name in 2019\_Q4 Data Set:-   
##

print(colnames(divvy\_tips\_2019\_Q4))

## [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"

cat("--------------------------------------------------","\n","\n")

## --------------------------------------------------   
##

cat("Columns Name in 2020\_Q1 Data Set:-", "\n","\n")

## Columns Name in 2020\_Q1 Data Set:-   
##

print(colnames(divvy\_tips\_2020\_Q1))

## [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"

Tornando cada coluna e posição uniforme.

Antes de unir os conjuntos de dados dos trimestres de 2019-T2 a 2020-T1 no projeto de análise, é fundamental garantir que os nomes das colunas e a ordem das colunas sejam uniformes em todos os arquivos. Isso é essencial para evitar erros na combinação dos dados e obter resultados precisos e consistentes.

A uniformidade dos nomes das colunas significa que cada coluna deve ter o mesmo nome em todos os conjuntos de dados, independentemente do trimestre. Além disso, a ordem das colunas deve ser a mesma em todos os arquivos, ou seja, a posição das colunas deve ser consistente em toda a base de dados.

Ao combinar os conjuntos de dados, o R utilizará os nomes das colunas para identificar quais colunas devem ser unidas. Se os nomes das colunas não estiverem uniformes ou se as colunas estiverem em posições diferentes em diferentes arquivos, isso pode levar a erros de correspondência e análise incorreta dos dados.

divvy\_tips\_2019\_Q2<-divvy\_tips\_2019\_Q2 %>%  
rename(Rental\_ID=X01...Rental.Details.Rental.ID,  
 From\_Start\_Date\_Time=X01...Rental.Details.Local.Start.Time,  
 To\_End\_Date\_Time=X01...Rental.Details.Local.End.Time,  
 Rental\_Bike\_ID=X01...Rental.Details.Bike.ID,  
 Rental\_Duration\_Second=X01...Rental.Details.Duration.In.Seconds.Uncapped,  
 Start\_Station\_ID=X03...Rental.Start.Station.ID,  
 Start\_Station\_Name=X03...Rental.Start.Station.Name,  
 End\_Station\_ID=X02...Rental.End.Station.ID,  
 End\_Station\_Name=X02...Rental.End.Station.Name,  
 User\_Type=User.Type,  
 Gender=Member.Gender,  
 Birth\_Year=X05...Member.Details.Member.Birthday.Year)  
  
divvy\_tips\_2019\_Q3<-divvy\_tips\_2019\_Q3 %>%   
 rename(Rental\_ID=trip\_id,  
 From\_Start\_Date\_Time=start\_time,  
 To\_End\_Date\_Time=end\_time,  
 Rental\_Bike\_ID=bikeid,  
 Rental\_Duration\_Second=tripduration,  
 Start\_Station\_ID=from\_station\_id,  
 Start\_Station\_Name=from\_station\_name,  
 End\_Station\_ID=to\_station\_id,  
 End\_Station\_Name=to\_station\_name,  
 User\_Type=usertype,  
 Gender=gender,  
 Birth\_Year=birthyear)  
  
divvy\_tips\_2019\_Q4<-divvy\_tips\_2019\_Q4 %>%   
 rename(Rental\_ID=trip\_id,  
 From\_Start\_Date\_Time=start\_time,  
 To\_End\_Date\_Time=end\_time,  
 Rental\_Bike\_ID=bikeid,  
 Rental\_Duration\_Second=tripduration,  
 Start\_Station\_ID=from\_station\_id,  
 Start\_Station\_Name=from\_station\_name,  
 End\_Station\_ID=to\_station\_id,  
 End\_Station\_Name=to\_station\_name,  
 User\_Type=usertype,  
 Gender=gender,  
 Birth\_Year=birthyear)  
  
divvy\_tips\_2020\_Q1<-divvy\_tips\_2020\_Q1 %>%   
 rename(Rental\_ID=ride\_id,  
 From\_Start\_Date\_Time=started\_at,  
 To\_End\_Date\_Time=ended\_at,  
 Start\_Station\_Name=start\_station\_name,  
 Start\_Station\_ID=start\_station\_id,  
 End\_Station\_Name=end\_station\_name,  
 End\_Station\_ID=end\_station\_id,  
 User\_Type=member\_casual)

# Removendo as colunas que não são necessárias  
  
divvy\_tips\_2019\_Q2<-divvy\_tips\_2019\_Q2 %>%  
 select(-c(Rental\_ID,Rental\_Bike\_ID,Rental\_Duration\_Second,  
 Gender, Birth\_Year))  
  
divvy\_tips\_2019\_Q3<-divvy\_tips\_2019\_Q3 %>%   
 select(-c(Rental\_ID,Rental\_Bike\_ID,Rental\_Duration\_Second,  
 Gender,Birth\_Year))  
  
divvy\_tips\_2019\_Q4<-divvy\_tips\_2019\_Q4 %>%   
 select(-c(Rental\_ID,Rental\_Bike\_ID,Rental\_Duration\_Second,  
 Gender,Birth\_Year))  
  
divvy\_tips\_2020\_Q1<-divvy\_tips\_2020\_Q1 %>%   
 select(c(From\_Start\_Date\_Time,To\_End\_Date\_Time,  
 Start\_Station\_ID,Start\_Station\_Name,End\_Station\_ID,  
 End\_Station\_Name,User\_Type))

# Revisando a estrutura de dados do conjunto de dados fornecido  
  
cat("------------------------------","\n")

## ------------------------------

cat("2019\_Q2 Dataset", "\n")

## 2019\_Q2 Dataset

cat("------------------------------","\n")

## ------------------------------

str(divvy\_tips\_2019\_Q2)

## 'data.frame': 1108163 obs. of 7 variables:  
## $ From\_Start\_Date\_Time: chr "2019-04-01 00:02:22" "2019-04-01 00:03:02" "2019-04-01 00:11:07" "2019-04-01 00:13:01" ...  
## $ To\_End\_Date\_Time : chr "2019-04-01 00:09:48" "2019-04-01 00:20:30" "2019-04-01 00:15:19" "2019-04-01 00:18:58" ...  
## $ Start\_Station\_ID : int 81 317 283 26 202 420 503 260 211 211 ...  
## $ Start\_Station\_Name : chr "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...  
## $ End\_Station\_ID : int 56 59 174 133 129 426 500 499 211 211 ...  
## $ End\_Station\_Name : chr "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...  
## $ User\_Type : chr "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...

cat("------------------------------","\n")

## ------------------------------

cat("2019\_Q3 Dataset", "\n")

## 2019\_Q3 Dataset

cat("------------------------------","\n")

## ------------------------------

str(divvy\_tips\_2019\_Q3)

## 'data.frame': 1640718 obs. of 7 variables:  
## $ From\_Start\_Date\_Time: chr "2019-07-01 00:00:27" "2019-07-01 00:01:16" "2019-07-01 00:01:48" "2019-07-01 00:02:07" ...  
## $ To\_End\_Date\_Time : chr "2019-07-01 00:20:41" "2019-07-01 00:18:44" "2019-07-01 00:27:42" "2019-07-01 00:27:10" ...  
## $ Start\_Station\_ID : int 117 381 313 313 168 300 168 313 43 43 ...  
## $ Start\_Station\_Name : chr "Wilton Ave & Belmont Ave" "Western Ave & Monroe St" "Lakeview Ave & Fullerton Pkwy" "Lakeview Ave & Fullerton Pkwy" ...  
## $ End\_Station\_ID : int 497 203 144 144 62 232 62 144 195 195 ...  
## $ End\_Station\_Name : chr "Kimball Ave & Belmont Ave" "Western Ave & 21st St" "Larrabee St & Webster Ave" "Larrabee St & Webster Ave" ...  
## $ User\_Type : chr "Subscriber" "Customer" "Customer" "Customer" ...

cat("------------------------------","\n")

## ------------------------------

cat("2019\_Q4 Dataset", "\n")

## 2019\_Q4 Dataset

cat("------------------------------","\n")

## ------------------------------

str(divvy\_tips\_2019\_Q4)

## 'data.frame': 704054 obs. of 7 variables:  
## $ From\_Start\_Date\_Time: chr "2019-10-01 00:01:39" "2019-10-01 00:02:16" "2019-10-01 00:04:32" "2019-10-01 00:04:32" ...  
## $ To\_End\_Date\_Time : chr "2019-10-01 00:17:20" "2019-10-01 00:06:34" "2019-10-01 00:18:43" "2019-10-01 00:43:43" ...  
## $ Start\_Station\_ID : int 20 19 84 313 210 156 84 156 156 336 ...  
## $ Start\_Station\_Name : chr "Sheffield Ave & Kingsbury St" "Throop (Loomis) St & Taylor St" "Milwaukee Ave & Grand Ave" "Lakeview Ave & Fullerton Pkwy" ...  
## $ End\_Station\_ID : int 309 241 199 290 382 226 142 463 463 336 ...  
## $ End\_Station\_Name : chr "Leavitt St & Armitage Ave" "Morgan St & Polk St" "Wabash Ave & Grand Ave" "Kedzie Ave & Palmer Ct" ...  
## $ User\_Type : chr "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...

cat("------------------------------","\n")

## ------------------------------

cat("2020\_Q1 Dataset", "\n")

## 2020\_Q1 Dataset

cat("------------------------------","\n")

## ------------------------------

str(divvy\_tips\_2020\_Q1)

## 'data.frame': 426887 obs. of 7 variables:  
## $ From\_Start\_Date\_Time: chr "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:26" "2020-01-06 16:17:07" ...  
## $ To\_End\_Date\_Time : chr "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09 19:32:17" "2020-01-06 16:25:56" ...  
## $ Start\_Station\_ID : int 239 234 296 51 66 212 96 96 212 38 ...  
## $ Start\_Station\_Name : chr "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...  
## $ End\_Station\_ID : int 326 318 117 24 212 96 212 212 96 100 ...  
## $ End\_Station\_Name : chr "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ...  
## $ User\_Type : chr "member" "member" "member" "member" ...

cat("------------------------------","\n")

## ------------------------------

# Verificando o tipo de usuário distinto em cada conjunto de dados  
  
cat("------------------------------","\n")

## ------------------------------

cat("Distinct User Type in 2019\_Q2 Data set are:","\n")

## Distinct User Type in 2019\_Q2 Data set are:

print(unique(divvy\_tips\_2019\_Q2$User\_Type))

## [1] "Subscriber" "Customer"

cat("------------------------------","\n", "\n")

## ------------------------------   
##

cat("------------------------------","\n")

## ------------------------------

cat("Distinct User Type in 2019\_Q3 Data set are:","\n")

## Distinct User Type in 2019\_Q3 Data set are:

print(unique(divvy\_tips\_2019\_Q3$User\_Type))

## [1] "Subscriber" "Customer"

cat("------------------------------","\n", "\n")

## ------------------------------   
##

cat("------------------------------","\n")

## ------------------------------

cat("Distinct User Type in 2019\_Q4 Data set are:","\n")

## Distinct User Type in 2019\_Q4 Data set are:

print(unique(divvy\_tips\_2019\_Q4$User\_Type))

## [1] "Subscriber" "Customer"

cat("------------------------------","\n", "\n")

## ------------------------------   
##

cat("------------------------------","\n")

## ------------------------------

cat("Distinct User Type in 2020\_Q1 Data set are:","\n")

## Distinct User Type in 2020\_Q1 Data set are:

print(unique(divvy\_tips\_2020\_Q1$User\_Type))

## [1] "member" "casual"

cat("------------------------------","\n", "\n")

## ------------------------------   
##

# Nome de linhas uniformes no dataset para tipo de usuário  
  
divvy\_tips\_2019\_Q2$User\_Type<-recode(divvy\_tips\_2019\_Q2$User\_Type,  
 "Subscriber" ="Member",  
 "Customer" ="Not Member")  
  
divvy\_tips\_2019\_Q3$User\_Type<-recode(divvy\_tips\_2019\_Q3$User\_Type,  
 "Subscriber" ="Member",  
 "Customer" ="Not Member")  
  
divvy\_tips\_2019\_Q4$User\_Type<-recode(divvy\_tips\_2019\_Q4$User\_Type,  
 "Subscriber" ="Member",  
 "Customer" ="Not Member")  
  
divvy\_tips\_2020\_Q1$User\_Type<-recode(divvy\_tips\_2020\_Q1$User\_Type,  
 "member" ="Member",  
 "casual" ="Not Member")

# Combine todos os quatro conjuntos do dataset  
  
divvy\_tips\_2019\_2020<-bind\_rows(divvy\_tips\_2019\_Q2,divvy\_tips\_2019\_Q3,  
 divvy\_tips\_2019\_Q4,divvy\_tips\_2020\_Q1)  
  
# Revise a coluna Tipo de usuário após a modificação   
  
cat("O membro agora deve ser apenas de dois tipos: 'Member' e 'Not Member'", "\n")

## O membro agora deve ser apenas de dois tipos: 'Member' e 'Not Member'

print(unique(divvy\_tips\_2019\_2020$User\_Type))

## [1] "Member" "Not Member"

# Revise as junções no datasets  
  
print(str(divvy\_tips\_2019\_2020))

## 'data.frame': 3879822 obs. of 7 variables:  
## $ From\_Start\_Date\_Time: chr "2019-04-01 00:02:22" "2019-04-01 00:03:02" "2019-04-01 00:11:07" "2019-04-01 00:13:01" ...  
## $ To\_End\_Date\_Time : chr "2019-04-01 00:09:48" "2019-04-01 00:20:30" "2019-04-01 00:15:19" "2019-04-01 00:18:58" ...  
## $ Start\_Station\_ID : int 81 317 283 26 202 420 503 260 211 211 ...  
## $ Start\_Station\_Name : chr "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...  
## $ End\_Station\_ID : int 56 59 174 133 129 426 500 499 211 211 ...  
## $ End\_Station\_Name : chr "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...  
## $ User\_Type : chr "Member" "Member" "Member" "Member" ...  
## NULL

# Formatando as colunas de data de caractere para data e hora  
  
divvy\_tips\_2019\_2020$From\_Start\_Date\_Time<-ymd\_hms(divvy\_tips\_2019\_2020$From\_Start\_Date\_Time)  
divvy\_tips\_2019\_2020$To\_End\_Date\_Time<-ymd\_hms(divvy\_tips\_2019\_2020$To\_End\_Date\_Time)

Para a realização da análise dos dados, foi criada uma nova coluna denominada “duração da viagem em minutos”. Essa coluna foi obtida a partir da subtração do horário de término da viagem pelo horário de início da mesma. Essa medida permitirá uma compreensão mais detalhada da duração das viagens realizadas.

divvy\_tips\_2019\_2020$Ride\_Length\_min<-  
as.numeric(as.character(difftime(divvy\_tips\_2019\_2020$To\_End\_Date\_Time,divvy\_tips\_2019\_2020$From\_Start\_Date\_Time)/60))

Os dados com duração de viagem em minutos menor que zero foram excluídos da análise utilizando o comando de filtro. Optamos por considerar apenas os registros cujo tempo de viagem em minutos é maior que zero, garantindo que tenhamos apenas valores válidos e coerentes para prosseguir com a análise dos dados.

library(conflicted)  
conflicts\_prefer(dplyr::filter) # informando ao R que prefere usar a função dplyr::filter quando houver um conflito com a função filter do pacote stats.

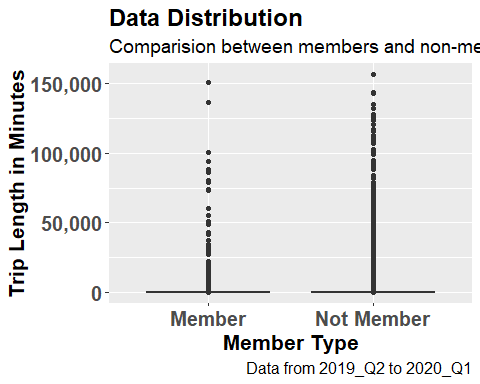
## [conflicted] Will prefer dplyr::filter over any other package.

divvy\_tips\_2019\_2020<-filter(divvy\_tips\_2019\_2020, Ride\_Length\_min>0)  
head(divvy\_tips\_2019\_2020)

## From\_Start\_Date\_Time To\_End\_Date\_Time Start\_Station\_ID  
## 1 2019-04-01 00:02:22 2019-04-01 00:09:48 81  
## 2 2019-04-01 00:03:02 2019-04-01 00:20:30 317  
## 3 2019-04-01 00:11:07 2019-04-01 00:15:19 283  
## 4 2019-04-01 00:13:01 2019-04-01 00:18:58 26  
## 5 2019-04-01 00:19:26 2019-04-01 00:36:13 202  
## 6 2019-04-01 00:19:39 2019-04-01 00:23:56 420  
## Start\_Station\_Name End\_Station\_ID End\_Station\_Name User\_Type  
## 1 Daley Center Plaza 56 Desplaines St & Kinzie St Member  
## 2 Wood St & Taylor St 59 Wabash Ave & Roosevelt Rd Member  
## 3 LaSalle St & Jackson Blvd 174 Canal St & Madison St Member  
## 4 McClurg Ct & Illinois St 133 Kingsbury St & Kinzie St Member  
## 5 Halsted St & 18th St 129 Blue Island Ave & 18th St Member  
## 6 Ellis Ave & 55th St 426 Ellis Ave & 60th St Member  
## Ride\_Length\_min  
## 1 7.433333  
## 2 17.466667  
## 3 4.200000  
## 4 5.950000  
## 5 16.783333  
## 6 4.283333

Realizar a visualização dos dados utilizando o gráfico de caixa (boxplot) para compreender a distribuição dos dados e identificar possíveis valores discrepantes (outliers).

# Rotulagem de gráficos  
graphM<-labs(title = "Data Distribution",  
 subtitle= "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x = "Member Type",  
 y ="Trip Length in Minutes")  
  
# Temas dos Gráficos  
graphTh<-theme(plot.title = element\_text(size=18,face="bold"),  
 plot.subtitle = element\_text(size=14),  
 plot.caption = element\_text(size=12),  
 axis.title.x = element\_text(size=16, face="bold"),  
 axis.title.y= element\_text(size=16, face="bold"),  
 axis.text.x=element\_text(size=15,face="bold"),  
 axis.text.y=element\_text(size=15,face="bold"))  
  
# Box-Plot  
options(repr.plot.width=12, repr.plot.height=8)  
ggplot(data=divvy\_tips\_2019\_2020)+   
 geom\_boxplot(mapping = aes(x=User\_Type,y=Ride\_Length\_min))+  
 scale\_y\_continuous(label = scales::comma)+   
 graphM +   
 graphTh

 Com o objetivo de facilitar a análise, adicionamos novas colunas contendo informações como o dia da semana, nome do dia, número do mês, nome do mês, hora e ano, a partir dos dados da coluna “From\_Start\_Date\_Time”. Isso nos permitirá realizar análises mais detalhadas e identificar padrões relacionados ao tempo nos dados.

divvy\_tips\_2019\_2020<- divvy\_tips\_2019\_2020 %>%  
 mutate(Day\_of\_Week = wday(From\_Start\_Date\_Time),   
 Day\_Name = wday(From\_Start\_Date\_Time,label = TRUE, abbr = FALSE),  
 Month\_Number=month(From\_Start\_Date\_Time),  
 Month\_Name =months(From\_Start\_Date\_Time),  
 quarter\_Number = quarters(From\_Start\_Date\_Time),  
 Year = year(From\_Start\_Date\_Time),  
 Hour= hour(From\_Start\_Date\_Time))

Nossa análise inclui a avaliação da porcentagem de outliers em cada categoria de tipo de usuário presente no dataset. Ao identificar esses outliers, poderemos entender melhor como eles afetam nossas análises e tomar decisões mais precisas ao considerar cada grupo separadamente. Isso nos permitirá explorar se existem diferenças significativas nos dados entre os tipos de usuários e como os outliers podem influenciar nossas conclusões e insights. Com essa informação em mãos, estaremos mais preparados para realizar análises mais robustas e embasadas nos dados, considerando a presença de valores extremos em cada categoria de usuário.

# Primeiro quartil  
  
Q1<-divvy\_tips\_2019\_2020 %>%   
 group\_by(User\_Type) %>%   
 summarize(Q1= quantile(Ride\_Length\_min,probs=0.25))  
  
Q1

## # A tibble: 2 × 2  
## User\_Type Q1  
## <chr> <dbl>  
## 1 Member 6.03  
## 2 Not Member 15.0

# Terceiro quartil  
  
Q3<-divvy\_tips\_2019\_2020 %>%   
 group\_by(User\_Type) %>%   
 summarize(Q3= quantile(Ride\_Length\_min,probs=0.75))  
Q3

## # A tibble: 2 × 2  
## User\_Type Q3  
## <chr> <dbl>  
## 1 Member 16.2  
## 2 Not Member 45.1

# Amplitude Interquartil  
  
IQR<-divvy\_tips\_2019\_2020 %>%   
 group\_by(User\_Type) %>%   
 summarize(IQR= quantile(Ride\_Length\_min,probs=0.75)-quantile(Ride\_Length\_min,probs=0.25))  
IQR

## # A tibble: 2 × 2  
## User\_Type IQR  
## <chr> <dbl>  
## 1 Member 10.1  
## 2 Not Member 30.0

Vamos aplicar esses cálculos ao conjunto de dados que temos disponível.

# Limite Inferior   
  
Lower\_Bound<-divvy\_tips\_2019\_2020 %>%   
 group\_by(User\_Type) %>%   
 summarize(Lower\_Bound= quantile(Ride\_Length\_min,probs=0.25)-  
 1.5\*(quantile(Ride\_Length\_min,probs=0.75)-quantile(Ride\_Length\_min,probs=0.25)))  
Lower\_Bound

## # A tibble: 2 × 2  
## User\_Type Lower\_Bound  
## <chr> <dbl>  
## 1 Member -9.19  
## 2 Not Member -30.0

# Limite superior  
  
Upper\_Bound<-divvy\_tips\_2019\_2020 %>%   
 group\_by(User\_Type) %>%   
 summarize(Upper\_Bound= quantile(Ride\_Length\_min,probs=0.75)+  
 1.5\*(quantile(Ride\_Length\_min,probs=0.75)-quantile(Ride\_Length\_min,probs=0.25)))  
Upper\_Bound

## # A tibble: 2 × 2  
## User\_Type Upper\_Bound  
## <chr> <dbl>  
## 1 Member 31.4  
## 2 Not Member 90.2

# Combinando limites inferior e superior  
  
Outlier\_prm<-cbind(Lower\_Bound,Upper\_Bound)  
Outlier\_prm<-Outlier\_prm %>% select(-3)  
Outlier\_prm

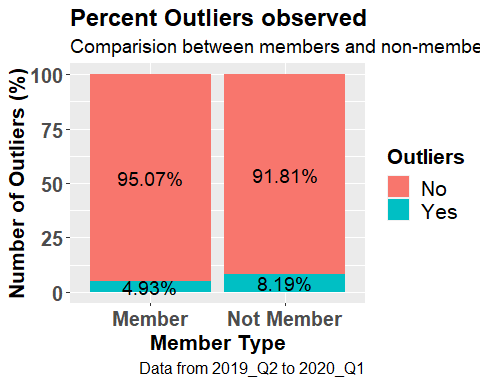
## User\_Type Lower\_Bound Upper\_Bound  
## 1 Member -9.191667 31.40833  
## 2 Not Member -30.025000 90.17500

# Dados mesclados com Dataset Orginal  
  
divvy\_tips\_2019\_2020<- merge(divvy\_tips\_2019\_2020,Outlier\_prm, by ="User\_Type", all=TRUE)  
# Criando a coluna Outliers como True e False  
  
divvy\_tips\_2019\_2020$Outliers <-   
 divvy\_tips\_2019\_2020$Ride\_Length\_min<= divvy\_tips\_2019\_2020$Lower\_Bound | divvy\_tips\_2019\_2020$Ride\_Length\_min>= divvy\_tips\_2019\_2020$Upper\_Bound  
  
# Converta TRUE e FALSE em Yes e No  
  
divvy\_tips\_2019\_2020$Outliers[divvy\_tips\_2019\_2020$Outliers== TRUE]<- "Yes"  
divvy\_tips\_2019\_2020$Outliers[divvy\_tips\_2019\_2020$Outliers== FALSE]<- "No"  
  
# Verificando a coluna Outliers e deve haver apenas dois valores "Yes" e "No".  
  
print(unique(divvy\_tips\_2019\_2020$Outliers))

## [1] "No" "Yes"

Para identificar a porcentagem de outliers em cada categoria de tipo de usuário, podemos realizar uma visualização de dados usando um gráfico de boxplot para cada grupo separadamente. O boxplot nos ajudará a identificar os valores atípicos e entender melhor a dispersão dos dados em relação aos quartis.

# Rótulo  
  
graphM1<-labs(title = "Percent Outliers observed",  
 subtitle= "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x = "Member Type",  
 y ="Number of Outliers (%)")  
  
# Temas  
  
graphTh1<-theme(legend.title=element\_text(size=16, face="bold"),  
 legend.text = element\_text(size=16),  
 plot.title = element\_text(size=18,face="bold"),  
 plot.subtitle = element\_text(size=14),  
 plot.caption = element\_text(size=12),  
 axis.title.x = element\_text(size=16, face="bold"),  
 axis.title.y= element\_text(size=16, face="bold"),  
 axis.text.x=element\_text(size=15, face= 'bold'),  
 axis.text.y=element\_text(size=15, face= 'bold'))  
  
# Gráfico de barras  
  
options(repr.plot.width=10, repr.plot.height=8)  
divvy\_tips\_2019\_2020 %>%   
 count(User\_Type,Outliers) %>%   
 group\_by(User\_Type) %>%   
 mutate(percnt=prop.table(n)\*100) %>%   
 ggplot(mapping=aes(User\_Type, percnt, fill=Outliers,  
 label = paste0(round(percnt,2), "%")))+  
 geom\_bar(stat="identity")+  
 geom\_text(position = position\_stack(vjust = .5),  
 size= 5)+  
 graphM1 +  
 graphTh1

 Identificamos que o total de outliers entre os membros corresponde a 4,93% do conjunto de dados, enquanto nos não membros esse valor é de 8,19%.

Agora, para remover os outliers tanto dos membros quanto dos não membros, utilizaremos o comando filter para manter o conjunto de dados apenas com os registros que não são outliers. Para fazer isso, adicionaremos uma nova coluna chamada “Outlier” e atribuiremos o valor “No” para os registros que não são outliers.

# Removendo outliers  
  
divvy\_tips\_2019\_2020<-filter(divvy\_tips\_2019\_2020, Outliers =="No")

Além disso, empregamos o mesmo conjunto de dados para realizar uma análise adicional utilizando o Excel Power Pivot. O arquivo Excel contendo o Power Pivot também está disponível no conjunto de dados, juntamente com um deck de apresentação em formato PPT.

# revendo o nome das colunas do dataset  
  
cat("Coulmns Name in the Datasets:-", "\n")

## Coulmns Name in the Datasets:-

cat("------------------------------","\n")

## ------------------------------

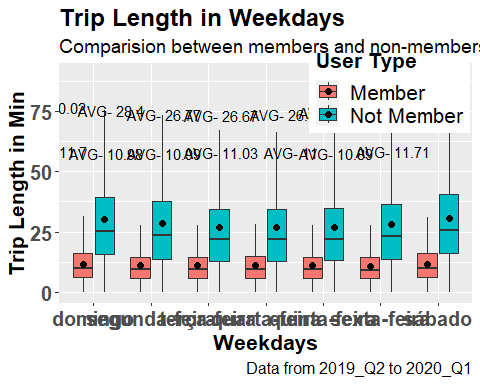
print(colnames(divvy\_tips\_2019\_2020))

## [1] "User\_Type" "From\_Start\_Date\_Time" "To\_End\_Date\_Time"   
## [4] "Start\_Station\_ID" "Start\_Station\_Name" "End\_Station\_ID"   
## [7] "End\_Station\_Name" "Ride\_Length\_min" "Day\_of\_Week"   
## [10] "Day\_Name" "Month\_Number" "Month\_Name"   
## [13] "quarter\_Number" "Year" "Hour"   
## [16] "Lower\_Bound" "Upper\_Bound" "Outliers"

## Visualização de Dados

### Duração da viagem em dias de semana

# Rótulo do gráfico  
  
graphM2<- labs(title = "Trip Length in Weekdays",  
 subtitle= "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x = "Weekdays",  
 y ="Trip Length in Min",  
 fill="User Type")  
  
# Temas do gráfico  
  
graphTh2<-theme(legend.position=c(0.80,0.90))  
  
  
# Box plot após a remoção dos Outliers   
  
options(repr.plot.width=15, repr.plot.height=8)  
ggplot(data=divvy\_tips\_2019\_2020,  
 mapping = aes(x=Day\_Name,y=Ride\_Length\_min, fill=User\_Type))+   
 geom\_boxplot(outlier.shape = NA)+   
 stat\_summary(fun=mean, geom = "point", size=2,  
 position=position\_dodge(width=0.75), color="black")+  
 stat\_summary(fun = mean, geom = "text", col = "black",   
 vjust = -8.9, hjust=1.1,size=4,  
 aes(label = paste("AVG-", round(..y.., digits = 2))))+  
 graphM2+  
 graphTh1+  
 graphTh2

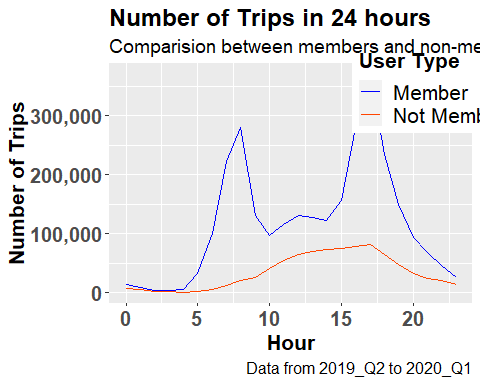
 1. Notavelmente, a duração média das viagens é menor para os membros em comparação com os não membros.

1. De forma interessante, durante os fins de semana, os não membros apresentaram uma média de duração de viagem 3 minutos maior do que nos dias úteis do mesmo grupo.
2. Observamos que não há diferença significativa no valor médio da duração das viagens entre os membros quando analisamos os dados dos diferentes dias da semana.

### Número de viagens em 24 horas

# Rótulo dos gráficos  
  
graphM3<- labs(title = "Number of Trips in 24 hours",  
 subtitle= "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x = "Hour",  
 y ="Number of Trips",  
 color="User Type")  
  
# Temas dos gráficos  
  
graphTh3<-theme(legend.position=c(0.89,0.90))  
   
# Gráfico de linhas - Número de viagens em 24 horas  
   
divvy\_tips\_2019\_2020 %>% group\_by(Hour,User\_Type) %>%  
 summarise(Total\_Rides = n()) %>%   
 ggplot(mapping = aes(x=Hour, y=Total\_Rides,   
 group=User\_Type, color=User\_Type))+  
 geom\_line()+ scale\_y\_continuous(label = scales::comma)+  
 scale\_color\_manual(values=c("blue", "orangered"))+  
 scale\_size\_manual(values=c(1.5, 1.5))+  
 graphM3 +  
 graphTh1 +  
 graphTh3

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

 ## Members 1. Observamos um aumento significativo no número de viagens das 4h às 7h30 e uma queda acentuada das 7h30 às 10h. 2. Além disso, notamos um leve aumento no número de viagens até o meio-dia, seguido por uma diminuição até as 14h. 3. Após as 15h, o número de viagens apresenta um aumento expressivo, atingindo o pico por volta das 18h.

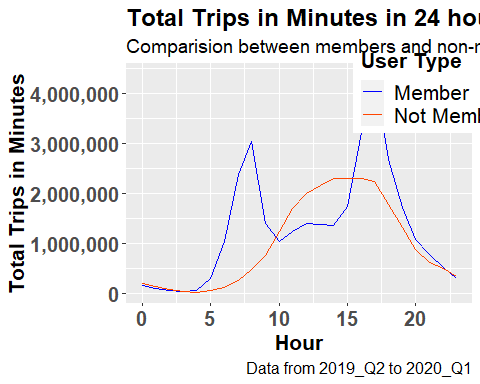
## Not Members

1. Notamos um aumento gradual no número de viagens das 6h às 17h, seguido de uma queda após as 17h.
2. Em relação ao grupo de Sócios, o número de viagens do grupo de Não Sócios é significativamente menor das 4h00 às 22h00.

### Total de viagens em minutos em 24 horas

# Rótulos do gráfico  
  
graphM4<- labs(title="Total Trips in Minutes in 24 hours",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Hour",  
 y ="Total Trips in Minutes",  
 color="User Type")  
  
# Gráfico de linhas - Total de viagens em minutos em 24 horas  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Hour, User\_Type) %>%   
 summarise(Total\_Trip\_Minutes =sum(Ride\_Length\_min)) %>%   
 ggplot(mapping = aes(x=Hour, y=Total\_Trip\_Minutes,  
 group=User\_Type, color=User\_Type))+  
 geom\_line()+   
 scale\_y\_continuous(label = scales::comma)+  
 scale\_color\_manual(values = c("blue","orangered"))+  
 scale\_size\_manual(values=c(1.5, 1.5))+  
 graphM4+  
 graphTh1+  
 graphTh3

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

 ## Members 1. Observamos que o número total de viagens e o padrão de minutos completos de viagens em 24 horas são idênticos.

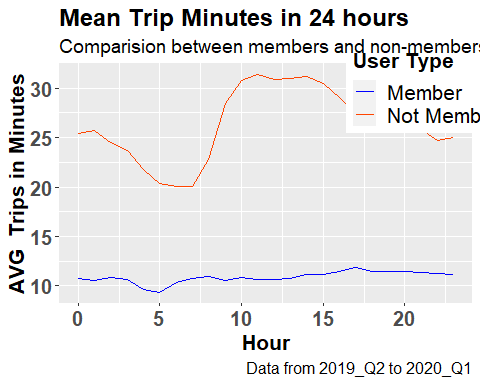
## Not Members

1. Da mesma forma, notamos que o número total de viagens e o total de minutos de viagens em 24 horas são idênticos.
2. Além disso, podemos observar que o total de minutos de viagens utilizadas por um não associado é maior das 10h às 15h do que o de um associado.
3. Com base no estudo, podemos inferir que a maioria dos membros provavelmente pertence a grupos de funcionários que trabalham em turnos.

### Mean Trip Minutes in 24 hours

# Rótulos do gráfico  
  
graphM5<- labs(title="Mean Trip Minutes in 24 hours",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Hour",  
 y ="AVG Trips in Minutes",  
 color="User Type")  
  
  
# Gráfico de linhas - Minutos médios de viagem em 24 horas  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Hour, User\_Type) %>%   
 summarise(AVG\_Trip\_Minutes =mean(Ride\_Length\_min)) %>%   
 ggplot(mapping = aes(x=Hour, y=AVG\_Trip\_Minutes,  
 group=User\_Type, color=User\_Type))+  
 geom\_line()+ scale\_color\_manual(values = c("blue","orangered"))+  
 scale\_size\_manual(values=c(1.5, 1.5))+  
 graphM5+  
 graphTh1+  
 graphTh3

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

 ### Members: - O tempo médio da viagem dos membros varia de aproximadamente 9 a 11 horas ao longo do dia.

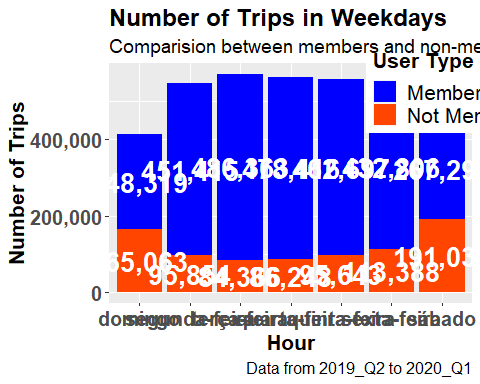
### Not Members:

1. O tempo médio de viagem dos não membros diminui de 12h à meia-noite até às 6h da manhã.
2. Observa-se um aumento acentuado no tempo médio da viagem após as 7h até às 11h. Permanece estável até as 14h e gradualmente diminui até as 22h, com um leve aumento após as 22h.
3. Em geral, o tempo médio de viagem é aproximadamente duas vezes maior para os não membros em comparação com os membros.
4. Realizaremos uma análise mais detalhada do número de viagens em várias estações entre membros e não membros.

### Número de viagens em dias de semana

# Rótulo do gráfico  
  
graphM6<- labs(title="Number of Trips in Weekdays",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Hour",  
 y ="Number of Trips",  
 fill="User Type")  
# Tema do gráfico  
  
graphTh3<-theme(legend.position=c(0.93,0.90))  
  
# gráfico de barras empilhadas - Número de viagens em dias de semana  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Day\_Name,User\_Type) %>%   
 summarise(Total\_rides\_wk=n()) %>%   
 ggplot(mapping = aes(x=Day\_Name, y=Total\_rides\_wk,  
 fill=User\_Type))+  
 geom\_bar(stat = "identity")+   
 scale\_y\_continuous(label = scales::comma)+  
 scale\_fill\_manual(values = c("blue","orangered"))+  
 geom\_text(aes(label = paste0(comma(round(Total\_rides\_wk,2)))),  
 position = position\_stack(vjust = .5),  
 size=7,color="white", fontface="bold")+  
 graphM6+  
 graphTh1+  
 graphTh3

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

 ## Members: 1. O número de viagens dos membros foi reduzido em mais da metade nos fins de semana em comparação com segunda-feira a sexta-feira. 2. O número de viagens dos membros aumentou a partir de segunda-feira, atingiu o máximo na quarta-feira e diminuiu ligeiramente a partir de quarta-feira.

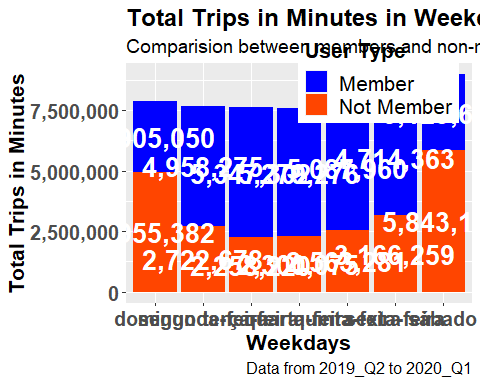
## Not Members:

1. O número de viagens dos não membros foi aproximadamente de 1,5 a 2,0 vezes maior nos fins de semana do que de segunda a sexta-feira.
2. O número de viagens dos não membros foi maior na sexta-feira, seguido por segunda-feira, excluindo os fins de semana.

### Total de Viagens em Minutos nos Dias da Semana

# Rótulo do gráfico  
  
graphM7<- labs(title="Total Trips in Minutes in Weekdays",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Weekdays",  
 y ="Total Trips in Minutes",  
 fill="User Type")  
  
# Tema do gráfico  
  
graphTh4<-theme(legend.position=c(0.73,0.94))  
  
# gráfico de barras - Total de minutos de viagens em dias da semana  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Day\_Name, User\_Type) %>%   
 summarise(Total\_trips\_min\_wkd =sum(Ride\_Length\_min)) %>%   
 ggplot(mapping=aes(x=Day\_Name,y=Total\_trips\_min\_wkd, fill=User\_Type))+  
 geom\_bar(stat="identity")+  
 scale\_y\_continuous(labels=scales::comma)+  
 scale\_fill\_manual(values=c("blue","orangered"))+  
 geom\_text(aes(label = paste0(comma(round(Total\_trips\_min\_wkd ,2)))),  
 position = position\_stack(vjust = .5),  
 size=7,color="white",fontface="bold")+  
 graphM7+  
 graphTh1+  
 graphTh4

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

 ## Members: 1. O número máximo de viagens em minutos foi maior na terça-feira e reduziu gradualmente após a terça-feira. 2. O valor mais baixo foi observado no domingo, seguido pelo sábado.

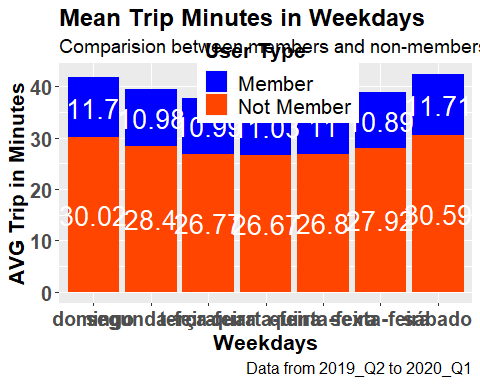
## Not Members:

1. O número máximo de viagens em minutos foi maior no sábado, seguido pelo domingo e sexta-feira.
2. A utilização das viagens foi observada como maior na sexta-feira, sábado e domingo.

## Média de Minutos por Viagem nos Dias de Semana

# Rótulo do gráfico  
  
graphM8<-labs(title="Mean Trip Minutes in Weekdays",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Weekdays",  
 y ="AVG Trip in Minutes",  
 fill="User Type")  
  
# Tema do gráfico  
  
graphTh4<-theme(legend.position=c(0.53,0.94))  
  
# gráfico de barras - Minutos médios de viagem em dias da semana  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Day\_Name,User\_Type) %>%   
 summarise(AVG\_Trip\_Min\_wkd=mean(Ride\_Length\_min)) %>%   
 ggplot(mapping = aes(x=Day\_Name, y=AVG\_Trip\_Min\_wkd,  
 fill=User\_Type))+  
 geom\_bar(stat = "identity")+  
 scale\_fill\_manual(values = c("blue","orangered"))+  
 geom\_text(aes(label= paste0(round(AVG\_Trip\_Min\_wkd,2))),  
 position = position\_stack(vjust = 0.5),  
 size=7,color="white")+  
 graphM8 +  
 graphTh1+  
 graphTh4

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

 ## Members: - A média de tempo de viagem em minutos é aproximadamente duas vezes menor do que a dos não membros. - Embora o número de viagens tenha diminuído nos fins de semana, o tempo médio de viagem é aproximadamente maior em 1 minuto.

## Not Members:

* A média de tempo de viagem em minutos é maior aos sábados e domingos, seguida por segunda-feira e sexta-feira.
* Embora o número de viagens seja menor em comparação com os membros, o tempo médio de viagem em minutos é duas vezes maior.

## Número de Viagens por Ano-Mês-Trimestre

Criamos uma coluna “Ano-Mês-Trimestre” multiplicando a coluna de ano por 100 e o número do mês. Combinamos isso com o número do trimestre usando o sinal “\_“.

# Criando Coluna Ano-Mês  
  
divvy\_tips\_2019\_2020$Year\_Month<-divvy\_tips\_2019\_2020$Year\*100+divvy\_tips\_2019\_2020$Month\_Number  
  
# Verificando o valor exclusivo da coluna Ano-Mês e classificando-os  
  
print(sort(unique(divvy\_tips\_2019\_2020$Year\_Month)))

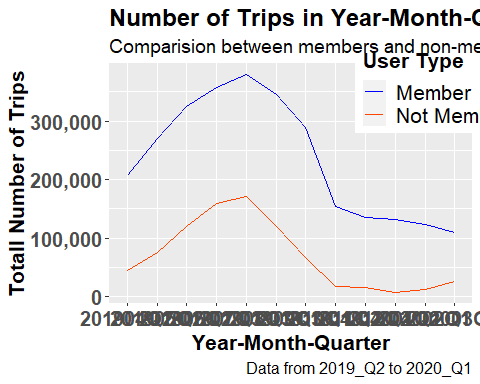
## [1] 201904 201905 201906 201907 201908 201909 201910 201911 201912 202001  
## [11] 202002 202003

# Criar coluna Ano-Mês\_Trimestre  
  
divvy\_tips\_2019\_2020$Year\_Month\_Quarter<-paste0(divvy\_tips\_2019\_2020$Year\_Month,divvy\_tips\_2019\_2020$quarter\_Number)  
  
# Verificando o valor unqiue da coluna Year-Month e classifique-os  
  
print(sort(unique(divvy\_tips\_2019\_2020$Year\_Month\_Quarter)))

## [1] "201904Q2" "201905Q2" "201906Q2" "201907Q3" "201908Q3" "201909Q3"  
## [7] "201910Q4" "201911Q4" "201912Q4" "202001Q1" "202002Q1" "202003Q1"

# Rótulo do gráfico  
  
graphM9<- labs(title="Number of Trips in Year-Month-Quarter",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Year-Month-Quarter",  
 y ="Totall Number of Trips",  
 color="User Type")  
  
# Tema do gráfico  
graphTh5<-theme(legend.position=c(0.90,0.90))  
  
# Gráfico de linhas - Número de viagens no ano-mês-trimestre  
  
divvy\_tips\_2019\_2020 %>%  
 group\_by(Year\_Month\_Quarter, User\_Type) %>%   
 summarise(Total\_Number\_Trips\_ymq=n()) %>%   
 ggplot(mapping = aes(x=Year\_Month\_Quarter,y=Total\_Number\_Trips\_ymq,  
 group=User\_Type,  
 color=User\_Type))+  
 geom\_line()+  
 scale\_y\_continuous(labels = scales::comma)+  
 scale\_color\_manual(values = c("blue","orangered"))+  
 scale\_size\_manual(values=c(1.5, 1.5))+  
 graphM9 +  
 graphTh1 +  
 graphTh5

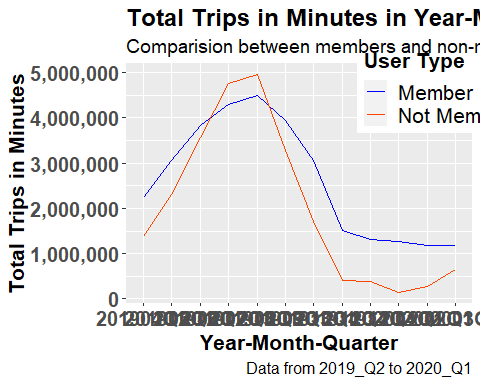
## `summarise()` has grouped output by 'Year\_Month\_Quarter'. You can override  
## using the `.groups` argument.

 ## Observações: 1. O número de viagens para membros e não membros aumentou a partir de abril de 2019 e atingiu o pico em julho de 2019. 2. Após agosto de 2019, houve uma tendência de queda até novembro. No caso dos membros, a tendência permaneceu estável até fevereiro de 2020, enquanto para os não membros, foi até março de 2020. 3. A tendência dos não membros diminuiu acentuadamente em comparação com os membros. O mesmo padrão observado nos membros após outubro de 2019 se manteve até novembro de 2019.

## Total de Viagens em Minutos por Ano-Mês-Trimestre.

# Rótulo do gráfico  
  
graphM10<- labs(title="Total Trips in Minutes in Year-Month-Quarter",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Year-Month-Quarter",  
 y ="Total Trips in Minutes",  
 color="User Type")  
  
# Gráfico de linhas - Total de viagens em minutos no ano-mês-trimestre  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Year\_Month\_Quarter,User\_Type) %>%   
 summarise(Total\_Ride\_Mins\_ymq= sum(Ride\_Length\_min)) %>%   
 ggplot(mapping=aes(x=Year\_Month\_Quarter, y=Total\_Ride\_Mins\_ymq,  
 group=User\_Type,  
 color=User\_Type))+  
 geom\_line()+  
 scale\_y\_continuous(labels = scales::comma)+  
 scale\_color\_manual(values = c("blue","orangered"))+  
 scale\_size\_manual(values=c(1.5,1.5))+  
 graphM10+  
 graphTh1 +  
 graphTh5

## `summarise()` has grouped output by 'Year\_Month\_Quarter'. You can override  
## using the `.groups` argument.

 ## Members: 1. Foi observado um padrão semelhante para membros no total de tempo de viagem em minutos, conforme seguimos a mesma abordagem utilizada para o número total de viagens. 2. O tempo de viagem em minutos reduziu gradualmente a partir de agosto de 2019 até o final de setembro de 2019. A partir de setembro de 2019, houve uma queda acentuada no tempo total de viagem até o final de outubro de 2019. Uma diminuição ligeira no tempo total de viagem foi observada a partir de novembro de 2019 e permaneceu estável de fevereiro a março de 2020.

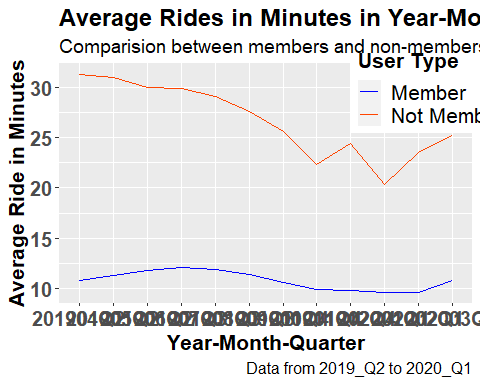
## Not Members:

1. O tempo total de viagem em minutos ultrapassou o dos membros a partir de meados de junho de 2019 e permaneceu em alta até o final de julho de 2019. O tempo total de viagem em minutos declinou acentuadamente de agosto de 2019 até outubro de 2019.
2. Houve um leve aumento no tempo total de viagem (em minutos) de novembro a dezembro de 2019. Esse tempo de viagem (em minutos) aumentou novamente a partir de fevereiro de 2019.

## Tempo Médio de Viagens em Minutos por Ano-Mês-Trimestre.

# Rótulo do gráfico  
  
graphM11<- labs(title="Average Rides in Minutes in Year-Month-Quarter",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Year-Month-Quarter",  
 y ="Average Ride in Minutes",  
 color="User Type")  
  
# Gráfico de Linhas - Média de Viagens em Minutos no Ano-Mês-Trimestre  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Year\_Month\_Quarter,User\_Type) %>%   
 summarise(AVG\_Ride\_Min\_ymq= mean(Ride\_Length\_min)) %>%   
 ggplot(mapping=aes(x=Year\_Month\_Quarter, y=AVG\_Ride\_Min\_ymq,  
 group=User\_Type,  
 color=User\_Type))+  
 geom\_line()+  
 scale\_y\_continuous(labels = scales::comma)+  
 scale\_color\_manual(values = c("blue","orangered"))+  
 scale\_size\_manual(values=c(1.5,1.5))+  
 graphM11 +  
 graphTh1 +  
 graphTh5

## `summarise()` has grouped output by 'Year\_Month\_Quarter'. You can override  
## using the `.groups` argument.

 ## Members: 1. O tempo médio de viagem (em minutos) aumenta de abril de 2019 até setembro de 2019. Permaneceu estável de outubro de 2019 até fevereiro de 2020. 2. O tempo médio de viagem (em minutos) aumentou ligeiramente a partir de meados de fevereiro de 2020.

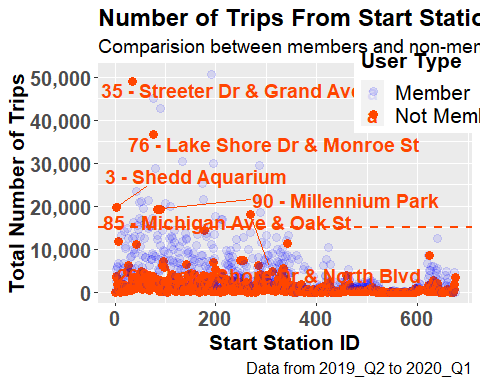
## Not Members:

1. O tempo médio de viagem (em minutos) diminuiu ligeiramente de abril de 2019 até meados de julho de 2019. A redução no tempo médio de viagem foi comparativamente maior de agosto de 2019 até outubro de 2019.
2. O tempo médio de viagem aumentou em novembro de 2019, diminuiu em dezembro de 2019 e aumentou a partir de janeiro de 2020.

## Número de Viagens a partir da Estação Inicial.

# Rótulo do gráfico  
  
graphM12<- labs(title="Number of Trips From Start Station",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Start Station ID",  
 y ="Total Number of Trips",  
 color="User Type",  
 alpha="User Type")  
  
# Gráfico de linhas - Número de viagens da estação inicial  
  
divvy\_tips\_2019\_2020 %>%   
 group\_by(Start\_Station\_ID, User\_Type, Start\_Station\_Name) %>%   
 summarise(Number\_of\_Trips\_sid=n()) %>%   
 ggplot(aes(x=Start\_Station\_ID, y=Number\_of\_Trips\_sid,  
 group=User\_Type,  
 color=User\_Type,  
 alpha=User\_Type)) +   
 geom\_point(size=3)+  
 geom\_text\_repel(data=divvy\_tips\_2019\_2020 %>%   
 group\_by(Start\_Station\_ID, User\_Type, Start\_Station\_Name) %>%  
 summarise(Number\_of\_Trips\_sid=n()) %>%  
 filter(User\_Type=="Not Member", Number\_of\_Trips\_sid>15000), # Filter data first  
 aes(label=paste0(Start\_Station\_ID, " - ", Start\_Station\_Name)),  
 size=5, vjust=1.2, hjust=1.2, fontface ="bold",  
 arrow = arrow(length = unit(0.01, "npc")), box.padding = 1)+  
 geom\_hline(yintercept=15000,  
 linetype="dashed", color = "orangered", size=1.0)+  
 scale\_y\_continuous(labels = scales::comma)+  
 scale\_color\_manual(values = c("blue","orangered"))+  
 graphM12+  
 graphTh1 +  
 graphTh5

## `summarise()` has grouped output by 'Start\_Station\_ID', 'User\_Type'. You can  
## override using the `.groups` argument.  
## `summarise()` has grouped output by 'Start\_Station\_ID', 'User\_Type'. You can  
## override using the `.groups` argument.

 ## Not Members: 1. O número de viagens dos não membros foi maior que 15.000 viagens nas seguintes estações de ID:

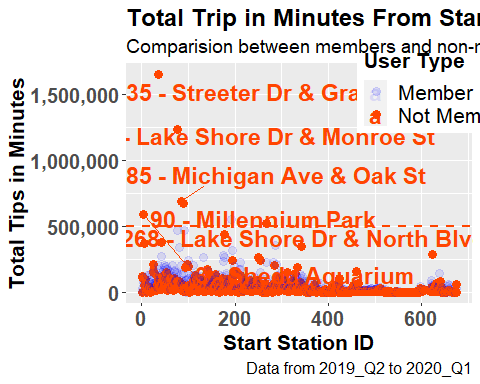
35 - [Streeter Dr & Grand Ave]; 76 - [Lake Shore Dr & Monroe St]; 90 - [Millennium Park]; 85 - [Michigan Ave & Oak St]; 3 - [Shedd Aquarium]; e 268 - [Lake Shore Dr & North Blvd]

O total de viagens “Não membros” também foi maior em comparação com os membros nessas estações.

## Tempo total de viagem a partir da Estação Inicial.

# Rótulo do gráfico  
  
graphM13<- labs(title="Total Trip in Minutes From Start Station",  
 subtitle = "Comparision between members and non-members",  
 caption = "Data from 2019\_Q2 to 2020\_Q1",  
 x= "Start Station ID",  
 y ="Total Tips in Minutes",  
 color="User Type",  
 alpha="User Type")  
  
  
# Gráfico de linhas - Número de viagens da estação inicial  
  
divvy\_tips\_2019\_2020 %>% group\_by(Start\_Station\_ID,User\_Type,Start\_Station\_Name) %>%   
 summarise(Total\_trip\_Min\_sid=sum(Ride\_Length\_min)) %>%   
 ggplot(aes(x=Start\_Station\_ID,y=Total\_trip\_Min\_sid,  
 group=User\_Type,  
 color=User\_Type,  
 alpha=User\_Type))+  
 geom\_point(size=3)+  
 geom\_hline(yintercept = 500000,linetype="dashed",   
 color = "orangered", size=1.0)+  
 geom\_text\_repel(data=divvy\_tips\_2019\_2020 %>%   
 group\_by(Start\_Station\_ID,User\_Type,Start\_Station\_Name) %>%   
 summarise(Total\_trip\_Min\_sid=sum(Ride\_Length\_min)) %>%   
 filter(User\_Type=="Not Member", Total\_trip\_Min\_sid>500000),  
 mapping = aes(x=Start\_Station\_ID,   
 y=Total\_trip\_Min\_sid,  
 label=paste0(Start\_Station\_ID," - ", Start\_Station\_Name)),   
 size=6, vjust=0.8, hjust=1.2, fontface ="bold",  
 arrow = arrow(length = unit(0.01, "npc")), box.padding = 1)+  
 scale\_y\_continuous(labels = scales::comma)+  
 scale\_color\_manual(values = c("blue","orangered"))+  
 graphM13+  
 graphTh1 +  
 graphTh5

## `summarise()` has grouped output by 'Start\_Station\_ID', 'User\_Type'. You can  
## override using the `.groups` argument.  
## `summarise()` has grouped output by 'Start\_Station\_ID', 'User\_Type'. You can  
## override using the `.groups` argument.

 ## Not Members: 1. O total de minutos de viagem dos não membros foi maior que 500.000 minutos nas seguintes estações de ID:

35 - [Streeter Dr & Grand Ave]; 76 - [Lake Shore Dr & Monroe St]; 90 - [Millennium Park]; 85 - [Michigan Ave & Oak St]; 3 - [Shedd Aquarium]; e 268 - [Lake Shore Dr & North Blvd]

O total de minutos “Não membros” também foi maior em comparação com os membros nessas estações.

## Conclusão e Recomendação:

Com o objetivo de impulsionar o número de viagens de curta e longa duração e incentivar a adesão ao programa de associação, recomendamos as seguintes ações estratégicas:

1. Oferecer incentivos, como descontos ou cupons, para encorajar viagens de curta duração entre os não membros através do programa de associação.
2. Estimular os membros a realizarem mais viagens nos finais de semana, criando pacotes familiares para viagens de longa duração e promovendo descontos e promoções para viagens de curta duração nesses dias específicos.
3. Durante os meses de junho e julho, que são períodos de pico de viagens tanto para membros quanto para não membros devido às férias de verão, desenvolver ofertas exclusivas e promoções para atrair mais usuários para o programa de associação.
4. Aproveitar a época de Natal, entre novembro e dezembro, quando há aumento nas viagens para não membros, para oferecer pacotes promocionais e descontos que incentivem ainda mais o uso do serviço.
5. Concentrar os esforços de marketing em estações específicas identificadas pelos ID [Name]:

* 35 - [Streeter Dr & Grand Ave]
* 76 - [Lake Shore Dr & Monroe St]
* 85 - [Michigan Ave & Oak St]
* 90 - [Millennium Park]
* 3 - [Shedd Aquarium]
* 268 - [Lake Shore Dr & North Blvd]

Essas estações têm um alto número de viagens de não membros, representando uma oportunidade para converter esses usuários em membros por meio de campanhas direcionadas e estratégias promocionais atraentes.

Com a implementação dessas estratégias, esperamos alcançar resultados positivos, aumentando a adesão ao programa de associação e estimulando o aumento geral das viagens tanto para membros quanto para não membros, o que pode contribuir para o crescimento e sucesso do serviço prestado pela empresa.