
title: 'Relatorio de apoio a campanhas futuras para o Marketing'

author: "André Christian F G Almeida"

date: "2024-08-20"

output:

html_document: default

Nesta reunião, foi apresentado a Cyclistic, uma empresa de mobilidade com foco na sustentabilidade e compartilhamento de bicicletas de diversos tipos como: reclináveis, triciclos manuais e de carga. Hoje, ela conta com mais de 5.800 unidades e cerca de 692 estações de compartilhamento em Chicago.

Identificação da tarefa de negócios, desafios e principais stakeholders.

Hoje a Cyclistic tem um desafio: Aumentar a taxa de conversão de clientes casuais (aqueles que assinam os planos diários e mensais) em membros Cyclistic (aqueles que assinam planos anuais). Através dos dados podemos realizar análises exploratórias e descritivas para descobrir o padrão de utilização dos membros, como: horas utilizadas, trajeto efetuado, quantidade de uso em determinado período etc.

Principais tarefas

- Baixar os dados de utilização de todos os clientes dos últimos 12 meses
- Realizar a estruturação dos dados
- Realizar a limpeza dos dados
- Realizar a análise exploratória dos dados
- Realizar a análise descritiva dos dados
- Criar as visualizações das **três principais** recomendações com base na análise
- Apresentar para os stakeholders


Stakeholders e seu objetivo

A Lily Moreno, diretora de Marketing e responsável pelas campanhas de e-mail, mídias e outros canais juntamente com a sua equipe de marketing desejam descobrir tendências através dos dados históricos.

Conclusão

Ao analisar os dados fornecidos e gerar insights, podemos auxiliar o setor de Marketing a criar campanhas estratégicas, a fim de converter os clientes casuais em membros Cyclistic respondendo as seguintes perguntas:

1. Como os membros Cyclistic e ciclistas casuais usam as bicicletas da Cyclistic de forma diferente?
2. Por que os ciclistas casuais iriam querer adquirir planos planos anuais da Cyclistic?
3. Como a Cyclistic pode usar a mídia digital para influenciar os passageiros casuais a se tornarem membros cyclistic?

 Com esta entrega, finalizamos a primeira etapa de um processo de análise de dados: Perguntar

Preparação e Armazenamento dos Dados

Os dados foram fornecidos através de um link (<https://divvy-tripdata.s3.amazonaws.com/index.html>) sob uma licença (<https://www.divvybikes.com/data-license-agreement>) pela Divvy, empresa operada pela Motivate International Inc., uma entidade reconhecida e respeitada no setor de compartilhamento de bicicletas.

Coletados de forma automática pelos sistema de compartilhamento, os dados sugerem uma coleta consistente e precisa, no entanto, foram identificados valores nulos, que serão tratados e validados.

Embora os dados abrangem um período extenso (desde 2015), iremos focar na atualidade e analisar somente os últimos 12 meses, garantindo uma análise baseada em informações recentes e relevantes. Por se tratar de uma análise anônima, não há preocupações com vies de relato/relatório, no entanto, vale ressaltar que é possível termos vieses na coleta, como o impacto de eventos sazonais ou alterações no serviço.

Levando em consideração os pontos cruciais citados, como: Precisão, Atualidade, Objetividade, Vies, Integridade e Licenciamento, podemos garantir que os dados utilizados serão confiáveis e apropriados para análise. Fortalecendo a confiabilidade as conclusões tiradas.

Etapas de processamento - Download dos arquivos - Extração de cada um das respectivas pastas zips - Junção dos arquivos csv's extraídos na pasta "ProjetoAnaliseCyclistic" - Criar o dataframe com todos os arquivos

```
# Definição do diretório dos arquivos
diretorio <- "C:/Users/André/Documents/ProjetoBike/ProjetoAnaliseCyclistic"
```

```
# Leitura de cada arquivo csv
arquivos <- list.files(diretorio, pattern = "*.csv", full.names = TRUE)
```

```
# Listagem dos arquivos existentes na pasta
leitura_arq <- function(file){
  read.csv(file)
}
```

```
# Criação do dataframe com os arquivos existentes
df_historico <- map_df(arquivos, leitura_arq)
```

```
# Visualização das variáveis e seus tipos
str(df_historico)
```

```
## 'data.frame':    7058826 obs. of  13 variables:
## $ ride_id          : chr  "0D9FA920C3062031" "92485E5FB5888ACD" "FB144B3FC8300187" "DDEB
93BC2CE9AA77" ...
## $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "classic_bike"
...
## $ started_at        : chr  "2023-05-07 19:53:48" "2023-05-06 18:54:08" "2023-05-21 00:40:
21" "2023-05-10 16:47:01" ...
## $ ended_at          : chr  "2023-05-07 19:58:32" "2023-05-06 19:03:35" "2023-05-21 00:44:
36" "2023-05-10 16:59:52" ...
## $ start_station_name: chr  "Southport Ave & Belmont Ave" "Southport Ave & Belmont Ave" "H
alsted St & 21st St" "Carpenter St & Huron St" ...
## $ start_station_id  : chr  "13229" "13229" "13162" "13196" ...
## $ end_station_name  : chr  "" "" "" "Damen Ave & Cortland St" ...
## $ end_station_id    : chr  "" "" "" "13133" ...
## $ start_lat         : num  41.9 41.9 41.9 41.9 42 ...
## $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
## $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
## $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
## $ member_casual     : chr  "member" "member" "member" "member" ...
```

✅ Com esta entrega, finalizamos a segunda etapa de um processo de análise de dados: Preparar para exploração

Processamento dos dados

Para esta etapa temos que: * Verificar se há valores nulos * Verificar se há valores duplicados * Conversão das colunas `started_at` e `ended_at` de texto para data * Criação das colunas `duration_trip`, que mostra a duração de cada viagem * Criação da coluna `day_of_the_week`, que mostra que dia da semana cada viagem começou

```
# Sumário estatístico do data frame
df_summary <- dfSummary(df_historico)
print(df_summary)
```

```
## Data Frame Summary
## df_historico
## Dimensions: 7058826 x 13
## Duplicates: 0
##
## -----
## No      Variable      Stats / Values      Freqs (% of Valid)      Graph
Valid      Missing
## -----
## -----
## 1      ride_id      1. 011C8EF97AB0F30D      2 ( 0.0%)
7058826      0
##      [character]      2. 01406457A85B0AFF      2 ( 0.0%)
(100.0%)      (0.0%)
##      3. 02606FBC7F8537EE      2 ( 0.0%)
##      4. 0354FD0756337B59      2 ( 0.0%)
##      5. 048C715F1DE0D8C0      2 ( 0.0%)
##      6. 05D27072A33A290C      2 ( 0.0%)
##      7. 0625A51D397A68F9      2 ( 0.0%)
##      8. 076F662AFB9CAEC1      2 ( 0.0%)
##      9. 07DBFDA3C91006AE      2 ( 0.0%)
##      10. 0AE60485F97A473D      2 ( 0.0%)
##      [ 7058605 others ]      7058806 (100.0%)      IIIIIIIII
IIIIIIIIII
##
## 2      rideable_type      1. classic_bike      3437335 (48.7%)      IIIIIIIII
I      7058826      0
##      [character]      2. docked_bike      62447 ( 0.9%)
(100.0%)      (0.0%)
##      3. electric_bike      3559044 (50.4%)      IIIIIIIII
II
##
## 3      started_at      1. 2023-07-13 17:38:08      8 ( 0.0%)
7058826      0
##      [character]      2. 2023-05-10 17:20:41      7 ( 0.0%)
(100.0%)      (0.0%)
##      3. 2023-05-16 17:20:12      7 ( 0.0%)
##      4. 2023-05-19 16:18:05      7 ( 0.0%)
##      5. 2023-05-21 13:32:30      7 ( 0.0%)
##      6. 2023-06-02 16:36:57      7 ( 0.0%)
##      7. 2023-06-07 18:19:47      7 ( 0.0%)
##      8. 2023-06-17 11:36:31      7 ( 0.0%)
##      9. 2023-06-23 17:51:51      7 ( 0.0%)
##      10. 2023-06-30 15:49:32      7 ( 0.0%)
##      [ 6055515 others ]      7058755 (100.0%)      IIIIIIIII
IIIIIIIIII
##
## 4      ended_at      1. 2023-12-08 09:54:28      18 ( 0.0%)
7058826      0
##      [character]      2. 2023-08-03 07:16:39      15 ( 0.0%)
(100.0%)      (0.0%)
##      3. 2023-09-14 17:09:23      14 ( 0.0%)
##      4. 2024-06-08 12:51:04.436      14 ( 0.0%)
##      5. 2023-08-12 07:48:46      13 ( 0.0%)
```

##		6. 2023-10-04 09:59:13	12 (0.0%)	
##		7. 2024-01-18 08:13:56	12 (0.0%)	
##		8. 2024-01-18 09:09:41	12 (0.0%)	
##		9. 2023-08-08 08:27:26	11 (0.0%)	
##		10. 2023-09-06 15:12:53	11 (0.0%)	
##		[6068680 others]	7058694 (100.0%)	IIIIIIII
IIIIIIIIII				
##				
## 5	start_station_name	1. (Empty string)	1138502 (16.1%)	III
7058826	0			
##	[character]	2. Streeter Dr & Grand Ave	81568 (1.2%)	
(100.0%)	(0.0%)			
##		3. DuSable Lake Shore Dr & M	54038 (0.8%)	
##		4. Michigan Ave & Oak St	48812 (0.7%)	
##		5. DuSable Lake Shore Dr & N	48232 (0.7%)	
##		6. Clark St & Elm St	43251 (0.6%)	
##		7. Kingsbury St & Kinzie St	43092 (0.6%)	
##		8. Clinton St & Washington B	40569 (0.6%)	
##		9. Wells St & Concord Ln	40210 (0.6%)	
##		10. Theater on the Lake	38811 (0.5%)	
##		[1696 others]	5481741 (77.7%)	IIIIIIII
IIIIIIII				
##				
## 6	start_station_id	1. (Empty string)	1138502 (16.1%)	III
7058826	0			
##	[character]	2. 13022	81568 (1.2%)	
(100.0%)	(0.0%)			
##		3. 13300	54038 (0.8%)	
##		4. 13042	48812 (0.7%)	
##		5. LF-005	48232 (0.7%)	
##		6. TA1307000039	43251 (0.6%)	
##		7. KA1503000043	43092 (0.6%)	
##		8. WL-012	40569 (0.6%)	
##		9. TA1308000050	40210 (0.6%)	
##		10. TA1308000001	38811 (0.5%)	
##		[1654 others]	5481741 (77.7%)	IIIIIIII
IIIIIIII				
##				
## 7	end_station_name	1. (Empty string)	1199873 (17.0%)	III
7058826	0			
##	[character]	2. Streeter Dr & Grand Ave	82466 (1.2%)	
(100.0%)	(0.0%)			
##		3. DuSable Lake Shore Dr & M	51891 (0.7%)	
##		4. DuSable Lake Shore Dr & N	51319 (0.7%)	
##		5. Michigan Ave & Oak St	49138 (0.7%)	
##		6. Clark St & Elm St	42521 (0.6%)	
##		7. Kingsbury St & Kinzie St	42459 (0.6%)	
##		8. Clinton St & Washington B	41189 (0.6%)	
##		9. Wells St & Concord Ln	40602 (0.6%)	
##		10. Theater on the Lake	39344 (0.6%)	
##		[1713 others]	5418024 (76.8%)	IIIIIIII
IIIIIIII				
##				
## 8	end_station_id	1. (Empty string)	1199873 (17.0%)	III
7058826	0			
##	[character]	2. 13022	82466 (1.2%)	

```

(100.0%)  (0.0%)
##          3. 13300          51891 ( 0.7%)
##          4. LF-005          51319 ( 0.7%)
##          5. 13042          49138 ( 0.7%)
##          6. TA1307000039      42521 ( 0.6%)
##          7. KA1503000043      42459 ( 0.6%)
##          8. WL-012          41189 ( 0.6%)
##          9. TA1308000050      40602 ( 0.6%)
##         10. TA1308000001       39344 ( 0.6%)
##          [ 1666 others ]      5418024 (76.8%)      IIIIIIII
IIIIIII
##
## 9   start_lat      Mean (sd) : 41.9 (0)      865186 distinct values
: :      7058826      0
##      [numeric]      min < med < max:
: :      (100.0%)      (0.0%)
##          41.6 < 41.9 < 42.1
: : .
##          IQR (CV) : 0 (0)
: : :
##
. : : : .
##
## 10  start_lng      Mean (sd) : -87.6 (0)      820046 distinct values
:      7058826      0
##      [numeric]      min < med < max:
:      (100.0%)      (0.0%)
##          -87.9 < -87.6 < -87.5
. :
##          IQR (CV) : 0 (0)
: :
##
. : : .
##
## 11  end_lat      Mean (sd) : 41.9 (0.1)      14124 distinct values
:      7049308      9518
##      [numeric]      min < med < max:
:      (99.9%)      (0.1%)
##          0 < 41.9 < 42.2
:
##          IQR (CV) : 0.1 (0)
:
##
:
##
## 12  end_lng      Mean (sd) : -87.6 (0.1)      14243 distinct values      :
7049308      9518
##      [numeric]      min < med < max:      :
(99.9%)      (0.1%)
##          -88.2 < -87.6 < 0      :
##          IQR (CV) : 0 (0)      :
##
##
## 13  member_casual  1. casual      2584797 (36.6%)      IIIIIIII
7058826      0
##      [character]      2. member      4474029 (63.4%)      IIIIIIII

```

```
IIII      (100.0%)  (0.0%)
```

```
## -----  
-----
```

💡 Identificamos que existem valores `ride_id` duplicados e colunas com valores em branco. Não iremos preencher as colunas com valores estatísticos pois a quantidade de registro é baixa, dessa forma, iremos excluir as linhas com valores nulos/vázios.

```
# Remove as linhas duplicadas com base na coluna ride_id  
df_historico_limpo <- df_historico %>%  
  distinct(ride_id, .keep_all = TRUE)
```

```
# Verificação se ainda há linhas duplicadas  
existe_duplicadas <- df_historico_limpo %>%  
  filter(duplicated(ride_id) | duplicated(ride_id, fromLast = TRUE)) %>%  
  nrow() > 0  
print(existe_duplicadas)
```

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

```
# Verificação se há pelo menos um valor nulo ou vazio em cada linha  
verifica_vazio_ou_nulo <- function(x) {  
  if (is.character(x)) {  
    return(is.na(x) | x == "")  
  } else {  
    return(is.na(x))  
  }  
}
```

```
# Aplicar a função para cada linha do dataframe  
linha_com_problemas <- apply(df_historico_limpo, 1, function(row) any(verifica_vazio_ou_nulo  
(row)))
```

```
# Filtrar o dataframe para remover as linhas com erros e manter o dataframe original  
df_historico_limpo <- df_historico_limpo[!linha_com_problemas, ]
```

```
# Exibição da quantidade de linhas antes e depois da limpeza  
numero_de_linhas_antes <- nrow(df_historico)  
numero_de_linhas_depois <- nrow(df_historico_limpo)  
  
print(paste("Número de linhas antes da limpeza: ", numero_de_linhas_antes))
```

```
## [1] "Número de linhas antes da limpeza: 7058826"
```

```
print(paste("Número de linhas depois da limpeza: ", numero_de_linhas_depois))
```

```
## [1] "Número de linhas depois da limpeza: 5272212"
```

```
# Adição da coluna "duration_trip" ao dataframe
# conversão da colunas started_at e ended_at para "difftime"
df_historico_limpo$started_at <- ymd_hms(df_historico_limpo$started_at)
df_historico_limpo$ended_at <- ymd_hms(df_historico_limpo$ended_at)
# Cálculo da duração e conversão para minutos arredondados para cima
df_historico_limpo <- df_historico_limpo %>%
  mutate(duration_trip = ceiling(difftime(ended_at, started_at, units = "mins")) #em minutos
```

```
# Criação da coluna day_of_the_week
df_historico_limpo <- df_historico_limpo %>%
  mutate(day_of_the_week = wday(started_at, label = TRUE))
```

✅ Com esta entrega, finalizamos a terceira etapa de um processo de análise de dados: Processar

Análise Exploratória e Descritiva do Dataframe

```
# Criação de um novo sumário com os dados limpos
df_summary <- dfSummary(df_historico_limpo)
print(df_summary)
```



```
## Data Frame Summary
## df_historico_limpo
## Dimensions: 5272212 x 15
## Duplicates: 0
##
## -----
##
## No      Variable      Stats / Values      Freqs (% of Valid)      Grap
## h      Valid      Missing
## -----
##
## 1      ride_id      1. 00000065B3150FF2      1 ( 0.0%)
5272212      0
##      [character]      2. 0000089D36728778      1 ( 0.0%)
(100.0%)      (0.0%)
##
##      3. 0000140539E54BF2      1 ( 0.0%)
##      4. 000018486CBEF63A      1 ( 0.0%)
##      5. 000019B6351D28DF      1 ( 0.0%)
##      6. 00001F1414872B6A      1 ( 0.0%)
##      7. 000023AB39488896      1 ( 0.0%)
##      8. 00002447D8785F1C      1 ( 0.0%)
##      9. 000028BC6D9349BE      1 ( 0.0%)
##      10. 00002E1E858F6AB7      1 ( 0.0%)
##      [ 5272202 others ]      5272202 (100.0%)      IIII
IIIIIIIIIIIIIIII
##
## 2      rideable_type      1. classic_bike      3429041 (65.0%)      IIII
IIIIIIIIII      5272212      0
##      [character]      2. docked_bike      60687 ( 1.2%)
(100.0%)      (0.0%)
##      3. electric_bike      1782484 (33.8%)      IIII
II
##
## 3      started_at      min : 2023-05-01 00:00:33      4673083 distinct values      .
:      .      5272212      0
##      [POSIXct, POSIXt]      med : 2023-09-29 17:04:20.5      : :
: :      :      (100.0%)      (0.0%)
##      max : 2024-06-30 23:54:52.215      : :
: : .      : :
##      range : 1y 1m 29d 23H 54M 19.2S      : :
: : : . . : : :
##
: : : : : : :
##
## 4      ended_at      min : 2023-05-01 00:04:28      4684973 distinct values      .
:      .      5272212      0
##      [POSIXct, POSIXt]      med : 2023-09-29 17:20:45.5      : :
: :      :      (100.0%)      (0.0%)
##      max : 2024-06-30 23:59:57.93      : :
: : .      : :
##      range : 1y 1m 29d 23H 55M 29.9S      : :
: : : . : : :
##
: : : : : : :
##
```

## 5	start_station_name	1. Streeter Dr & Grand Ave	76893 (1.5%)	
5272212	0			
##	[character]	2. DuSable Lake Shore Dr & M	51031 (1.0%)	
(100.0%)	(0.0%)			
##		3. Michigan Ave & Oak St	45174 (0.9%)	
##		4. DuSable Lake Shore Dr & N	44538 (0.8%)	
##		5. Kingsbury St & Kinzie St	39667 (0.8%)	
##		6. Clark St & Elm St	39643 (0.8%)	
##		7. Clinton St & Washington B	37158 (0.7%)	
##		8. Theater on the Lake	36036 (0.7%)	
##		9. Millennium Park	35808 (0.7%)	
##		10. Wells St & Concord Ln	35658 (0.7%)	
##		[1652 others]	4830606 (91.6%)	IIII
IIIIIIIIIIIIIIII				
##				
## 6	start_station_id	1. 13022	76893 (1.5%)	
5272212	0			
##	[character]	2. 13300	51031 (1.0%)	
(100.0%)	(0.0%)			
##		3. 13042	45174 (0.9%)	
##		4. LF-005	44538 (0.8%)	
##		5. KA1503000043	39667 (0.8%)	
##		6. TA1307000039	39643 (0.8%)	
##		7. WL-012	37158 (0.7%)	
##		8. TA1308000001	36036 (0.7%)	
##		9. 13008	35808 (0.7%)	
##		10. TA1308000050	35658 (0.7%)	
##		[1616 others]	4830606 (91.6%)	IIII
IIIIIIIIIIIIIIII				
##				
## 7	end_station_name	1. Streeter Dr & Grand Ave	79068 (1.5%)	
5272212	0			
##	[character]	2. DuSable Lake Shore Dr & M	49199 (0.9%)	
(100.0%)	(0.0%)			
##		3. DuSable Lake Shore Dr & N	48634 (0.9%)	
##		4. Michigan Ave & Oak St	45997 (0.9%)	
##		5. Kingsbury St & Kinzie St	39151 (0.7%)	
##		6. Clark St & Elm St	39055 (0.7%)	
##		7. Clinton St & Washington B	37836 (0.7%)	
##		8. Millennium Park	37154 (0.7%)	
##		9. Theater on the Lake	37070 (0.7%)	
##		10. Wells St & Concord Ln	36553 (0.7%)	
##		[1676 others]	4822495 (91.5%)	IIII
IIIIIIIIIIIIIIII				
##				
## 8	end_station_id	1. 13022	79068 (1.5%)	
5272212	0			
##	[character]	2. 13300	49199 (0.9%)	
(100.0%)	(0.0%)			
##		3. LF-005	48634 (0.9%)	
##		4. 13042	45997 (0.9%)	
##		5. KA1503000043	39151 (0.7%)	
##		6. TA1307000039	39055 (0.7%)	
##		7. WL-012	37836 (0.7%)	
##		8. 13008	37154 (0.7%)	
##		9. TA1308000001	37070 (0.7%)	

```

##          10. TA1308000050          36553 ( 0.7%)
##          [ 1632 others ]          4822495 (91.5%)          IIII
IIIIIIIIIIIIIIIIII
##
## 9      start_lat      Mean (sd) : 41.9 (0)          751217 distinct values
:          5272212      0
##      [numeric]      min < med < max:
: .          (100.0%)    (0.0%)
##          41.6 < 41.9 < 42.1
: :
##          IQR (CV) : 0 (0)
: : .
##
. . : : :
##
## 10     start_lng      Mean (sd) : -87.6 (0)          712443 distinct values
:          5272212      0
##      [numeric]      min < med < max:
:          (100.0%)    (0.0%)
##          -87.8 < -87.6 < -87.5
:
##          IQR (CV) : 0 (0)
: :
##
. : : :
##
## 11     end_lat        Mean (sd) : 41.9 (0.1)          10515 distinct values
:      5272212      0
##      [numeric]      min < med < max:
:      (100.0%)    (0.0%)
##          0 < 41.9 < 42.1
:
##          IQR (CV) : 0 (0)
:
##
:
##
## 12     end_lng        Mean (sd) : -87.6 (0.1)          10601 distinct values      :
5272212      0
##      [numeric]      min < med < max:      :
(100.0%)    (0.0%)
##          -87.8 < -87.6 < 0      :
##          IQR (CV) : 0 (0)      :
##          :
##
## 13     member_casual  1. casual          1900459 (36.0%)          IIII
III          5272212      0
##      [character]    2. member          3371753 (64.0%)          IIII
IIIIIIII    (100.0%)    (0.0%)
##
## 14     duration_trip  min : -54          1516 distinct values      :
5272212      0
##      [difftime]      med : 11      :
(100.0%)    (0.0%)
##          max : 12137      :
##          units : mins      :

```

```
##
##
## 15   day_of_the_week      1. dom                708832 (13.4%)      II
5272212    0
##      [ordered, factor]    2. seg                696114 (13.2%)      II
(100.0%)   (0.0%)
##                      3. ter                747136 (14.2%)      II
##                      4. qua                772903 (14.7%)      II
##                      5. qui                775433 (14.7%)      II
##                      6. sex                751385 (14.3%)      II
##                      7. sáb                820409 (15.6%)      III
## -----
-----
```

Análise exploratória

💡 O tipo de bicicleta com menor frequência de uso são as docked_bike (bicicletas cargueiras), com 1.2% dos registros e as classic_bike (bicicletas comuns) dominam, com 65.0% dos registros

💡 De 5272212 registros, 36% são de membros casuais e 64% de membros Cyclistic

💡 A duração média de uma viagem é de 11 minutos

```
# 💡 O dia em que os clientes mais utilizam o serviço é no sábado
freq_day_of_the_week <- table(df_historico_limpo$day_of_the_week)
moda <- names(freq_day_of_the_week)[which.max(freq_day_of_the_week)]
moda
```

```
## [1] "sáb"
```

```
# 💡 O tipo de bicicleta que mais utilizam são as bicicletas normais
freq_rideable_type <- table(df_historico_limpo$rideable_type)
moda <- names(freq_rideable_type)[which.max(freq_rideable_type)]
moda
```

```
## [1] "classic_bike"
```

Análise exploratória

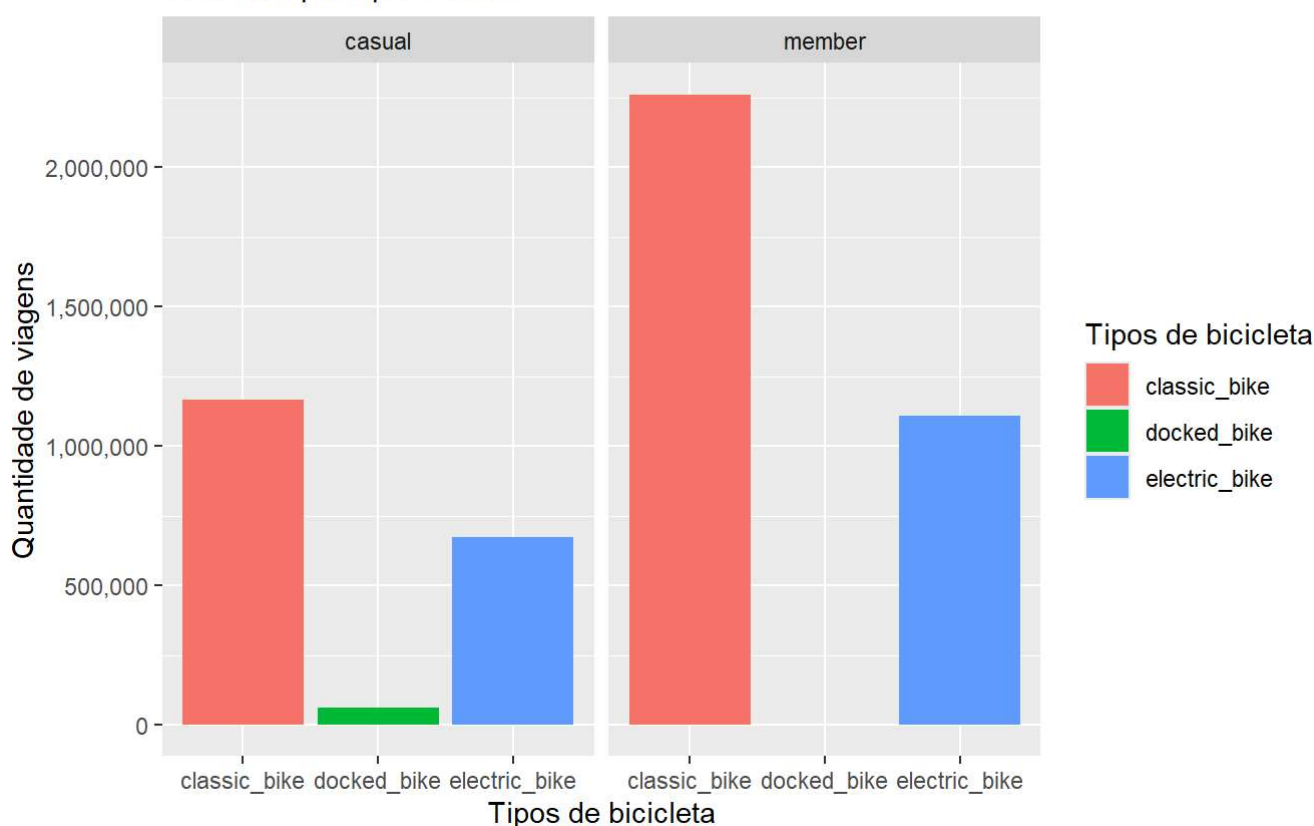
```

title_text <- str_wrap("Quantidade de viagens por tipo de bicicleta e membro")
subtitle_text <- str_wrap("Ambos os membros utilizam os modelos de bicicleta de forma parecida, sendo a clássica a principal escolha.")
ggplot(data = df_historico_limpo) +
  geom_bar(mapping = aes(x = rideable_type, fill = rideable_type)) +
  facet_wrap(~member_casual) +
  labs(title = title_text,
       subtitle = subtitle_text,
       y = "Quantidade de viagens",
       x = "Tipos de bicicleta",
       fill = "Tipos de bicicleta") +
  scale_y_continuous(labels = scales::comma_format()) # separação de milhares

```

Quantidade de viagens por tipo de bicicleta e membro

Ambos os membros utilizam os modelos de bicicleta de forma parecida, sendo a clássica a principal escolha.



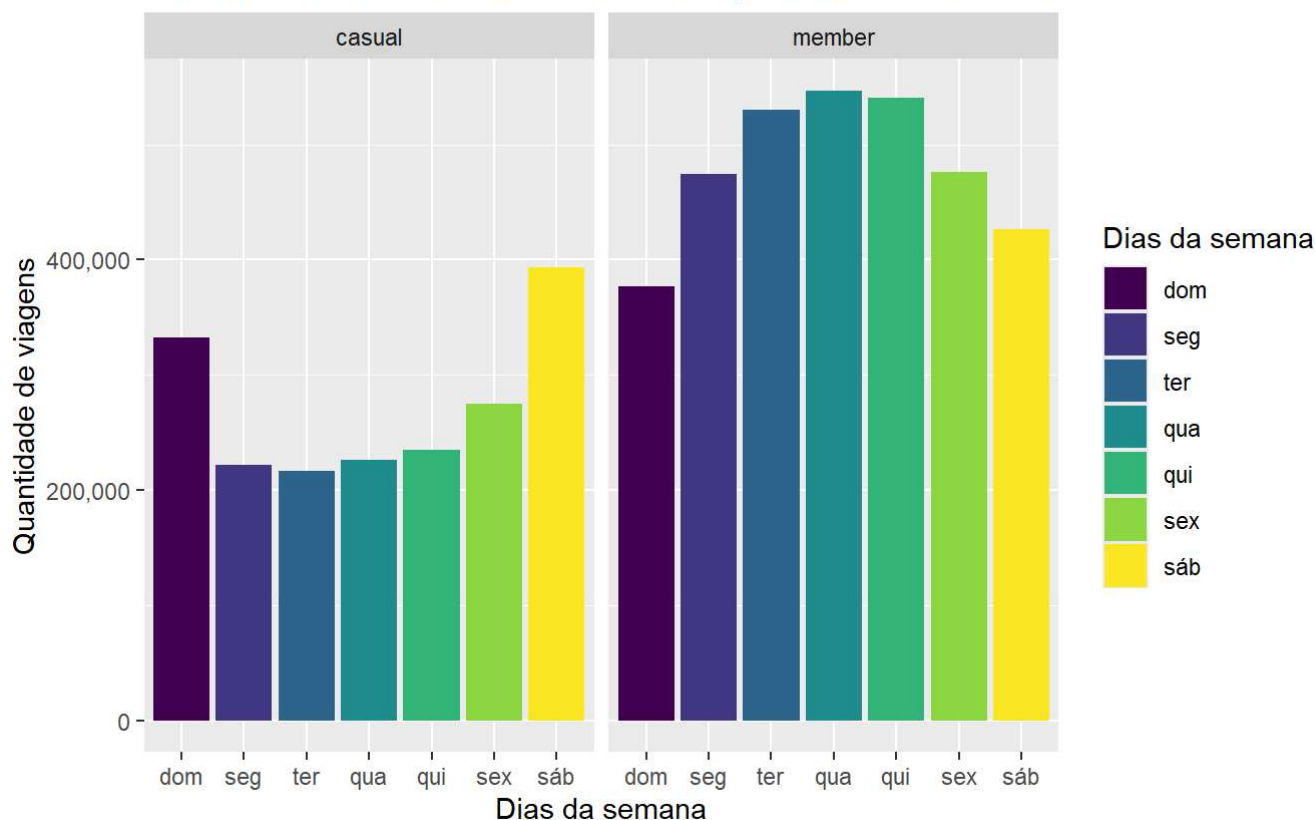
```

title_text <- str_wrap("Quantidade de viagens durante a semana por tipo de membro")
subtitle_text <- str_wrap("Os membros Cyclistic costumam utilizar as bicicletas durante a semana, enquanto os membros casuais tendem a utilizar o serviço aos finais de semana")
ggplot(data = df_historico_limpo) +
  geom_bar(mapping = aes(x = day_of_the_week, fill = day_of_the_week)) +
  facet_wrap(~member_casual) +
  labs(title = title_text,
       subtitle = subtitle_text,
       y = "Quantidade de viagens",
       x = "Dias da semana",
       fill = "Dias da semana") +
  scale_y_continuous(labels = scales::comma_format()) # separação de milhares

```

Quantidade de viagens durante a semana por tipo de membro

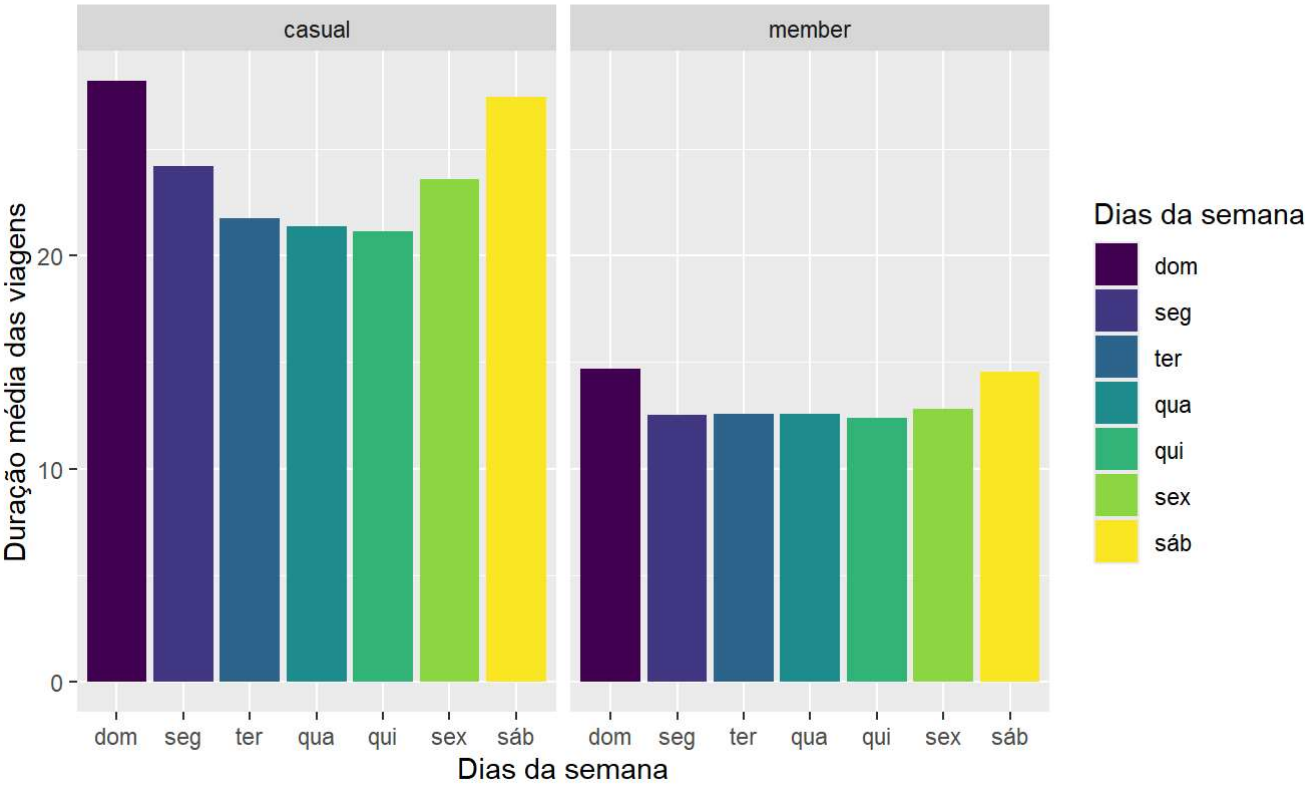
Os membros Cyclistic costumam utilizar as bicicletas durante a semana, enquanto os membros casuais tendem a utilizar o serviço aos finais de semana



```
title_text <- str_wrap("Duração média de viagens por dia da semana e tipo de membro")
subtitle_text <- str_wrap("O tempo médio de viagem dos membros casuais segue o padrão esperado, com viagens mais longas nos finais de semana. Enquanto os membros Cyclistic também apresentam um comportamento semelhante..")
df_historico_limpo %>%
  mutate(duration_trip_min = as.numeric(duration_trip, units = "mins")) %>%
  group_by(day_of_the_week, member_casual) %>%
    summarize(mean_duration_trip = mean(duration_trip_min), .groups = 'drop') %>%
  ggplot(aes(x = day_of_the_week, y = mean_duration_trip, fill = day_of_the_week)) +
    geom_bar(stat = "identity") +
    facet_wrap(~member_casual) +
  labs(title = title_text,
        subtitle = subtitle_text,
        y = "Duração média das viagens",
        x = "Dias da semana",
        fill = "Dias da semana")
```

Duração média de viagens por dia da semana e tipo de membro

O tempo médio de viagem dos membros casuais segue o padrão esperado, com viagens mais longas nos finais de semana. Enquanto os membros Cyclistic também apresentam um comportamento semelhante..



Resumo da análise

Analisando a utilização dos membros da Cyclistic do último ano (2023/05 - 2024-06) podemos concluir que:

- O dia mais comum de utilização do serviço é aos sábados e o tipo de bicicleta mais utilizado é a bicicleta clássica, enquanto as durações das viagens duram em média 11 minutos.
- **Membros casuais** costumam utilizar o serviço durante os finais de semana, dias normalmente relacionados com a prática de lazer.
- **Membros Cyclistic** tem registro de viagens maiores no meio da semana (terça a quinta), dias que podem ter uma maior demanda de idas aos escritórios diante do contexto global de modalidades de locais de trabalho. No entanto, a média de duração das viagens se destaca aos finais de semana, apontando que além de utilizá-las para trabalho, também utilizam para uma possível recreação.

✓ Com esta entrega, finalizamos a quinta etapa de um processo de análise de dados: Compartilhar

Recomendações com base na análise

Recomendações para orientar as futuras campanhas de marketing com base na análise:

1. Realização de pesquisa com os membros para entender quais são as principais empresas e instituição dos utilizadores do serviço para buscar parcerias corporativas ou descontos.
2. Captar mais dados a respeito dos membros para expandir o conhecimento, podendo ser possível traçar os diferentes tipos de persona.
3. Criação de campanhas para converter os membros casuais em membros Cyclistic através de benefícios e rede de acesso.

Recomendações de melhorias

- Configuração de uma API para conectar diretamente ao Docker disponibilizado para realizar o download automático sempre que estiver disponível.
- Estudar as possibilidades e necessidades de cruzar os dados de longitude e latitude de cada estação de início e fim para conseguir calcular a distância percorrida em cada viagem.

Metodologia

Decidimos utilizar a ferramenta R para realizar a análise para a equipe de marketing pela alta facilidade em realizar todo o processo de ETL. Precisamos instalar e configurar o ambiente R localmente para utilizar o terminal a partir do RStudio local, pois a limitação de hardware nas ferramentas online, como: Rstudio e GoogleColab impossibilitou a utilização.