Caso A

Rafael Rodriguez

2024-08-29

Cyclistic historical bike trip data

Context

You are a junior data analyst working on the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations. In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. Moreno believes there is a solid opportunity to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.

Stakeholders - Who is your audience?

Cyclistic Lily Moreno - The director of marketing Cyclistic marketing analytics team Cyclistic executive team

Problem

How do annual members and casual riders use Cyclistic bikes differently?

What type of data should I collect?

This data can be downloaded Population: Years: 2004 - 2024 Months: January – December Sample: Years: 2019 - 2020 Months: Oct-19 / mar-20

Hosted

https://divvy-tripdata.s3.amazonaws.com/index.html

Licensed

https://divvybikes.com/data-license-agreement dataset present restrictions agree to license-agreement

Dataset Upload

```
A <- read_csv("D:/KIKE/Cursos/SENATEC/Curso 8. Capstone/Modulo 2/Caso A/csv Datos Caso A/Divvy_Trips_20
```

```
## 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.
```

```
B <- read_csv("D:/KIKE/Cursos/SENATEC/Curso 8. Capstone/Modulo 2/Caso A/csv Datos Caso A/Divvy_Trips_20
```

```
## 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.
```

Verify the column names

```
colnames(A)
```

```
##  [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(B)
```

```
##  [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"
```

Rename columns file A

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

Join the datasets

```
D<- C %>%
  select(ride_id, started_at,ended_at,start_station_id,start_station_name,end_station_id,end_station_nar

E <- B %>%
  select(ride_id, started_at,ended_at,start_station_id,start_station_name,end_station_id,end_station_nar

Cyclist<- rbind(D,E)
```

```
Cyclist_TV <- Cyclist %>%
  mutate(Tiempo_viaje = as.numeric(difftime(ended_at, started_at, units = "secs"))) %>%
  mutate(Tiempo_viaje = as.integer(Tiempo_viaje)) %>%
  mutate(Tiempo_viaje = sprintf("%02d:%02d:%02d",
                                Tiempo_viaje %/% 3600,
                                (Tiempo_viaje %% 3600) %/% 60,
                                Tiempo_viaje %% 60)) %>%
  select(ride_id, ended_at, started_at, Tiempo_viaje) %>%
  arrange(Tiempo_viaje)
head(Cyclist_TV)
```

```
## # A tibble: 6 x 4
##   ride_id  ended_at            started_at          Tiempo_viaje
##   <chr>    <dttm>              <dttm>              <chr>
## 1 25625850 2019-11-03 01:01:26 2019-11-03 01:57:48 -1:03:38
## 2 25625849 2019-11-03 01:01:52 2019-11-03 01:55:33 -1:06:19
## 3 25625851 2019-11-03 01:08:27 2019-11-03 01:58:17 -1:10:10
## 4 25625843 2019-11-03 01:03:02 2019-11-03 01:51:59 -1:11:03
## 5 25625846 2019-11-03 01:04:53 2019-11-03 01:53:10 -1:11:43
## 6 25625845 2019-11-03 01:04:53 2019-11-03 01:53:08 -1:11:45
```

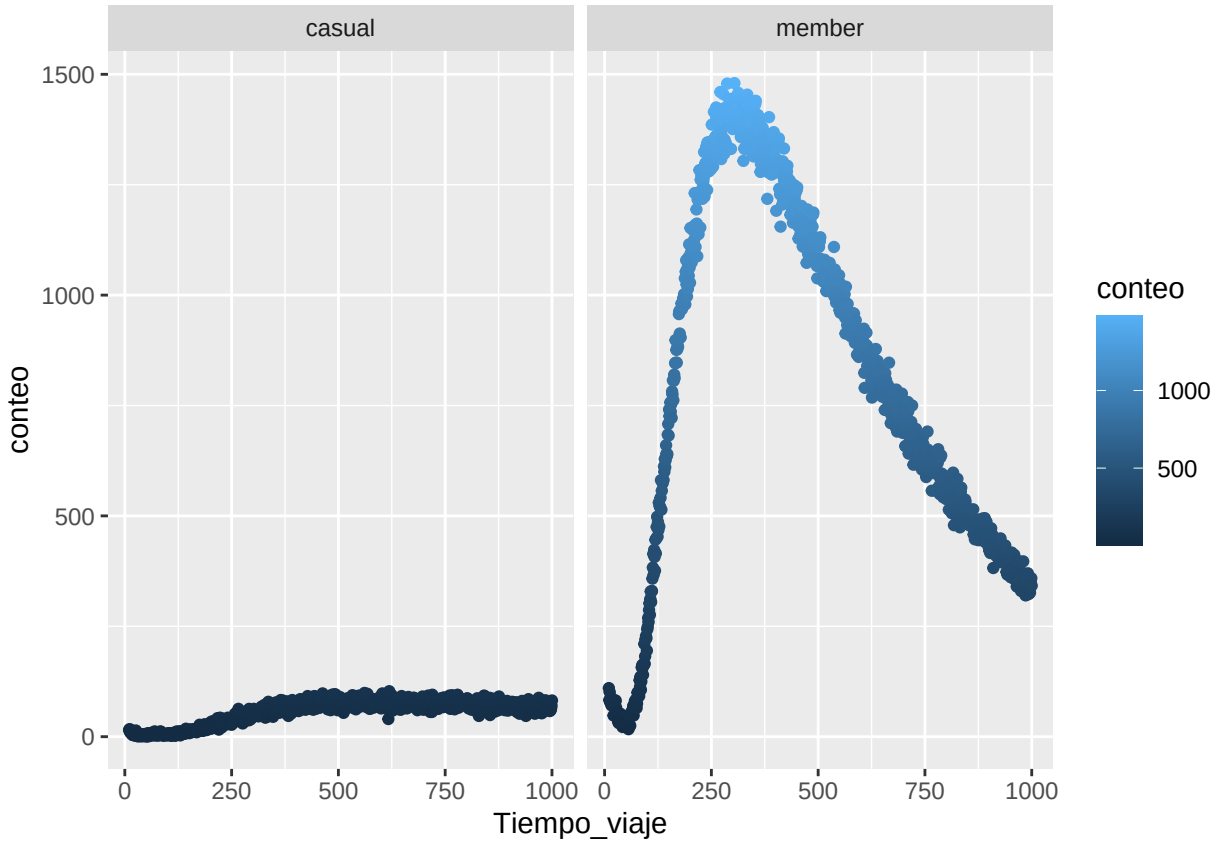
```
Cyclist_1 <- Cyclist %>%
  mutate(member_casual = str_replace_all(member_casual, "Subscriber", "member")) %>%
  mutate(member_casual = str_replace_all(member_casual, "Customer", "casual")) %>%
  filter(complete.cases(.))

Cyclist_PR <- Cyclist_1 %>%
  mutate(Tiempo_viaje = as.integer(difftime(ended_at, started_at, units = "secs"))) %>%
  mutate(Tiempo_viaje = round(Tiempo_viaje))  %>%
  mutate(Dia_semana = wday(started_at)) %>%
  mutate(started_at = as.POSIXct(started_at)) %>%
```

```
  mutate(year = year(started_at)) %>%
  select(ride_id, ended_at, started_at, Tiempo_viaje, Dia_semana, year, member_casual) %>%
  filter(Tiempo_viaje >= 10 | is.na(Tiempo_viaje) ) %>%
  arrange(Tiempo_viaje)
head(Cyclist_PR)
```

```
## # A tibble: 6 x 7
##   ride_id  ended_at            started_at          Tiempo_viaje Dia_semana  year
##   <chr>    <dttm>              <dttm>                     <dbl>      <dbl> <dbl>
## 1 49D0806~ 2020-01-18 07:18:31 2020-01-18 07:18:21           10          7  2020
## 2 D4FA53A~ 2020-01-22 12:22:48 2020-01-22 12:22:38           10          4  2020
## 3 FFD9F7F~ 2020-01-31 08:44:45 2020-01-31 08:44:35           10          6  2020
## 4 39FA76A~ 2020-01-04 14:01:11 2020-01-04 14:01:01           10          7  2020
## 5 97F1E28~ 2020-01-09 07:56:20 2020-01-09 07:56:10           10          5  2020
## 6 5DB6B15~ 2020-01-12 14:31:15 2020-01-12 14:31:05           10          1  2020
## # i 1 more variable: member_casual <chr>
```

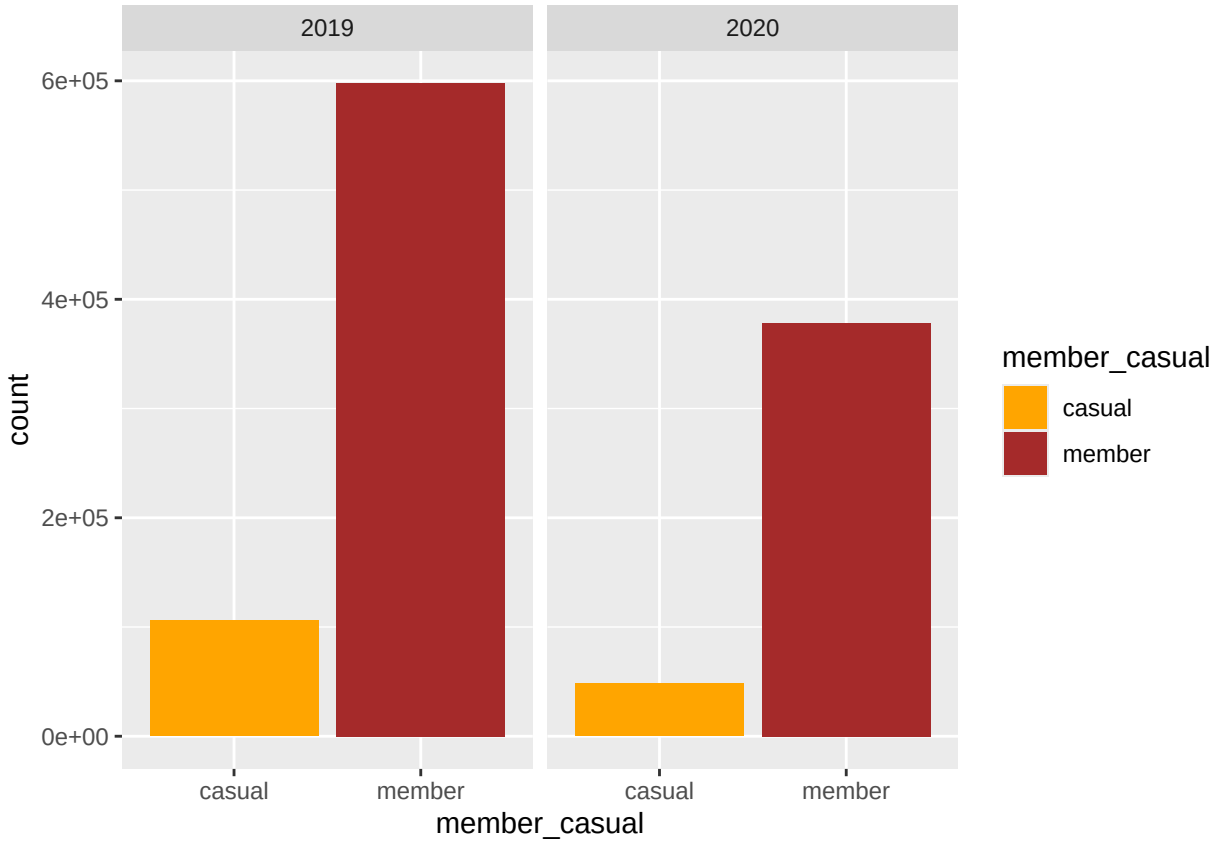
Evaluate the trip time vs type of member

```
Cyclist_PR_grouped <- Cyclist_PR %>%
  group_by(Tiempo_viaje, member_casual) %>%
  summarise(conteo = n(), .groups = 'keep')

df_Cyclist_PR_grouped <- as.data.frame(Cyclist_PR_grouped)

df_Cyclist_PR_grouped_1<- df_Cyclist_PR_grouped %>%
  filter(Tiempo_viaje <=1000)

ggplot(data = df_Cyclist_PR_grouped_1) +
  geom_point(mapping = aes(x=Tiempo_viaje, y=conteo, color= conteo))+
    facet_grid(~member_casual)
```



Evaluate the type of members quantity by year

```
Cyclist_2 <- Cyclist_1 %>%
  mutate(year = year(started_at)) %>%
  mutate(month= month(started_at))

ggplot(data = Cyclist_2) +
  geom_bar(mapping = aes(x=member_casual, fill = member_casual))+
  facet_grid(~year)+
  scale_fill_manual(values = c("casual" = "orange", "member" = "brown"))
```



Evaluate trip measurement by month

```
Cyclist_2_grouped <- Cyclist_2 %>%
  group_by(year, month, member_casual) %>%
  summarise(conteo = n()) %>%
  ungroup() %>%
  group_by(year) %>%
  mutate(Percent_year = round(conteo / sum(conteo) * 100, 1)) %>%
  ungroup() %>%
  group_by(year, month) %>%
  mutate(Percent_month_year = round(conteo / sum(conteo) * 100, 1))
```

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

```
print(Cyclist_2_grouped)
```

```
## # A tibble: 12 x 6
## # Groups:   year, month [6]
##     year month member_casual conteo Percent_year Percent_month_year
##    <dbl> <dbl> <chr>          <int>        <dbl>              <dbl>
##  1  2019    10 casual         71035         10.1               19.1
##  2  2019    10 member        300751         42.7               80.9
```

```
##  3  2019    11 casual         18729          2.7               10.6
##  4  2019    11 member        158447         22.5               89.4
##  5  2019    12 casual         16430          2.3               10.6
##  6  2019    12 member        138662         19.7               89.4
##  7  2020     1 casual          7785          1.8                5.4
##  8  2020     1 member        136099         31.9               94.6
##  9  2020     2 casual         12870          3                  9.2
## 10  2020     2 member        126715         29.7               90.8
## 11  2020     3 casual         27824          6.5               19.4
## 12  2020     3 member        115593         27.1               80.6
```

```r
Cyclist_3_grouped <- Cyclist_2 %>%
  group_by(month) %>%
  summarise(conteo = n()) %>%
    arrange(desc(conteo))
print(Cyclist_3_grouped)
```

```
## # A tibble: 6 x 2
##   month conteo
##   <dbl>  <int>
## 1    10 371786
## 2    11 177176
## 3    12 155092
## 4     1 143884
## 5     3 143417
## 6     2 139585
```

Recommendations

- Do marketing campaigns by station and date.
- Create a marketing strategy for the winter station.