Caso de estudio 1 Ciclystic

Ronal Castellanos

2023-07-12

About the company.

In 2016, Cyclistic launched a successful bike share offering. Since then, the program grew to a fleet of 5,824 geo-tagged and locked bikes at a network of 692 stations across Chicago. Bikes can be unlocked from one station and returned to any other station in the system at any time.

Until now, Cyclistic's marketing strategy was based on building overall brand recognition and appealing to broad consumer segments. One of the approaches that helped make this possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride passes or full-day passes are referred to as occasional riders. Customers who purchase annual memberships are called Cyclistic members.

Cyclistic's financial analysts concluded that annual memberships are much more profitable than occasional riders. While flexible pricing helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all new customers, Moreno believes there is plenty of opportunity to convert casual cyclists into members. She points out that casual cyclists already know about Cyclistic's program and have chosen Cyclistic for their mobility needs.

Moreno set a clear goal: Design marketing strategies aimed at converting casual cyclists into annual members. However, to do that, the marketing analyst team needs to better understand how annual members and casual cyclists differ, why casual cyclists would buy a membership, and how digital media might affect their marketing tactics. Moreno and his team are interested in analyzing Cyclistic's historical bike trip data to identify trends.

Translated with www.DeepL.com/Translator (free version)

Data Source

This data was obtained from from https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html) Data obtained for June 2022 to June 2023

Step 1: Prepare the files

1. First row removed from each .csv file, just one keep first header row
2. all files merged by cmd comand copy *.csv, ciclystic.csv

Step 2: Load packages

Packages intalled:

install.packages("tidyverse") install.packages("skirm") install.packages("skimr") install.packages("janitor")

Step 2.1: Loading packages

Step 3: Import and Clean data

Data loadd from .csv file merged.

```
rides<- read_csv("ciclystic.csv")
```

3.1 Filter null values

```
cleaned_rides <- na.omit(rides)
```

3.2 Convert datetime to same format

There is two types of dateformat in the file YYYY-mm-dd HH:MM:SS y MM/DD/YYY HH:MM They were matched to datetimestamp format

4.Analysis

Spliting date into day, month and year.

```
trips<-mutate(trips,travel_time=as.double(difftime(ended_at,started_at)/60))

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

4.1 Removing trips when performed quality checks

```
trips_v2 <- trips[!(trips$start_station_name == "HQ QR" | trips$travel_time<0),]
```

4.2 Descriptive analisys

Summarizing data, calculating mean, median, max and min values for the cleaned and filtered dataset.

```
aggregate(trips_v2$travel_time ~ trips_v2$member_casual, FUN = mean)
```

```
##   trips_v2$member_casual trips_v2$travel_time
## 1                 casual             22.71981
## 2                 member             12.19373
```

```
aggregate(trips_v2$travel_time ~ trips_v2$member_casual, FUN = median)
```

```
##   trips_v2$member_casual trips_v2$travel_time
## 1                 casual                   13
## 2                 member                    9
```

```
aggregate(trips_v2$travel_time ~ trips_v2$member_casual, FUN = max)
```

```
##   trips_v2$member_casual trips_v2$travel_time
## 1                 casual               32035
## 2                 member                1498
```

```
aggregate(trips_v2$travel_time ~ trips_v2$member_casual, FUN = min)
```

```
##   trips_v2$member_casual trips_v2$travel_time
## 1                 casual                   0
## 2                 member                   0
```

```
trips_v2$day_of_week <- ordered(trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "W
ednesday", "Thursday", "Friday", "Saturday"))

aggregate(trips_v2$travel_time ~ trips_v2$member_casual + trips_v2$day_of_week, FUN = mean)
```

```
##    trips_v2$member_casual trips_v2$day_of_week trips_v2$travel_time
## 1                  casual              Sunday             26.16682
## 2                  member              Sunday             13.61741
## 3                  casual              Monday             22.81586
## 4                  member              Monday             11.56434
## 5                  casual             Tuesday             20.42292
## 6                  member             Tuesday             11.68593
## 7                  casual           Wednesday             19.62246
## 8                  member           Wednesday             11.68224
## 9                  casual            Thursday             20.17245
## 10                 member            Thursday             11.77837
## 11                 casual              Friday             21.80595
## 12                 member              Friday             12.01685
## 13                 casual            Saturday             25.44055
## 14                 member            Saturday             13.68233
```

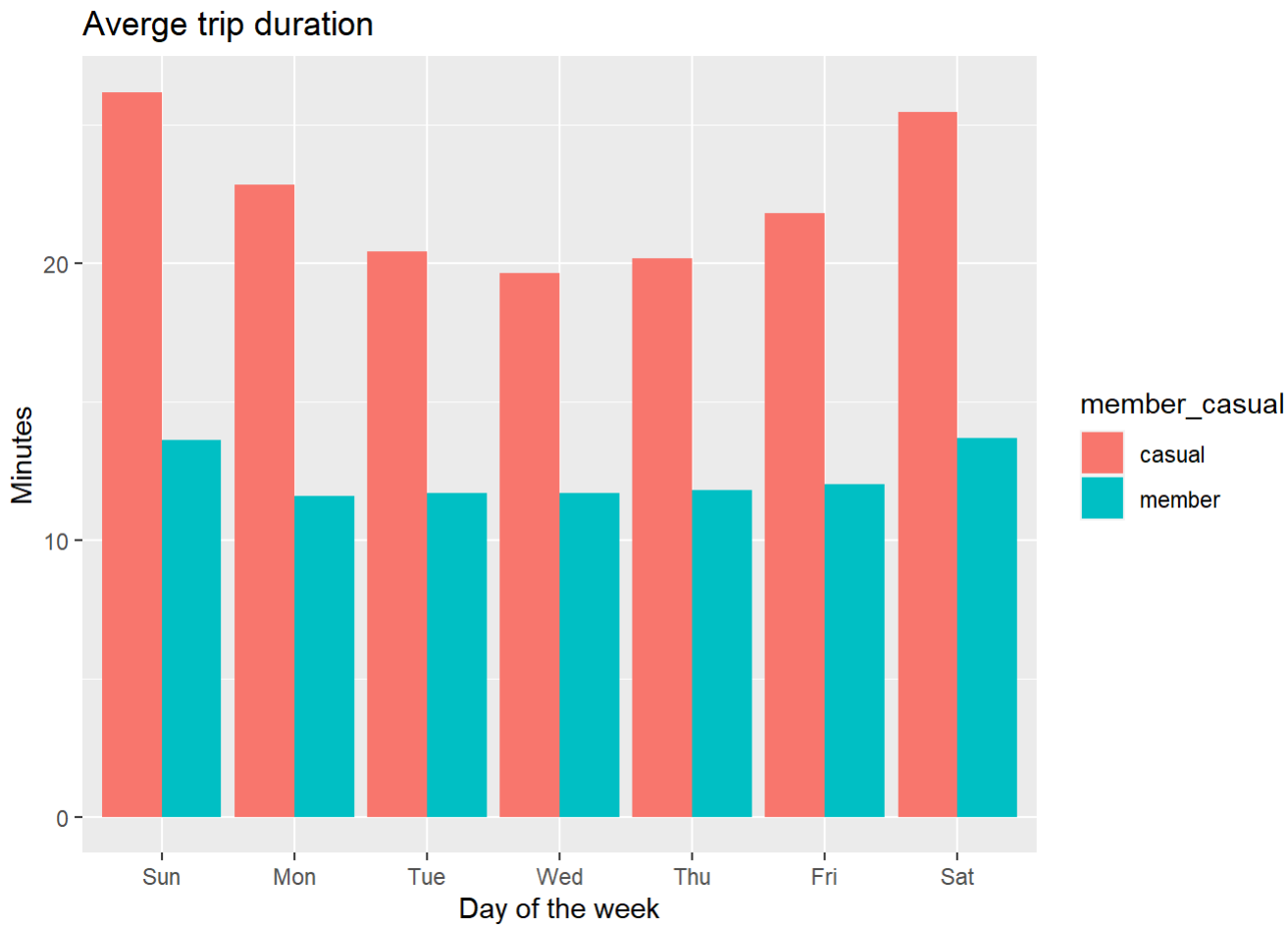
```
trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%                  #groups by usertype and weekday
  summarise(number_of_rides = n()                              #calculates the number of
rides and average duration
  ,average_duration = mean(travel_time)) %>%                # calculates the average duration
  arrange(member_casual, weekday)
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              289554             26.2
##  2 casual        Mon              194973             22.8
##  3 casual        Tue              202329             20.4
##  4 casual        Wed              218113             19.6
##  5 casual        Thu              233764             20.2
##  6 casual        Fri              258175             21.8
##  7 casual        Sat              350968             25.4
##  8 member        Sun              303828             13.6
##  9 member        Mon              374372             11.6
## 10 member        Tue              438804             11.7
## 11 member        Wed              456846             11.7
## 12 member        Thu              442115             11.8
## 13 member        Fri              387064             12.0
## 14 member        Sat              343719             13.7
```

5. Vizualizations

Graphs of average trip length for casual users and members per day. Graphs of numbers of trips per day for casual users and members.

```
# Let's create a visualization for average duration
trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(travel_time)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(title = "Averge trip duration",x="Day of the week",y="Minutes")
```




```
trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(travel_time)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(title = "Total rides per day",x="Day of the week",y="Number of trips")
```

6.Export data

Saving summaized data into .csv file

```
counts <- aggregate(trips_v2$travel_time ~ trips_v2$member_casual + trips_v2$day_of_week, FUN =
function(x) c(Sum = sum(x), Mean = mean(x)))
write.csv(counts, file = 'avg_ride_length.csv')
```