Ride Patterns of City of Chicago Bikers

Babatunde Ademusire

2024-01-01

Introduction

I've been a data analyst since 2019, mostly analyzing research data and publishing those projects I co-author in international journals. In October 2023, I decided to upskill and branch out to other fields where a data analyst is needed. The first step I took was to enroll in the Google Data Analytics Professional Certificate. This case study is a capstone project I completed as one of the requirements to obtain my certification.

In this case study, I analyzed data owned by the City of Chicago Divvy bicycle sharing service, operated by Lyft Bikes and Scooters, LLC, and made publicly available by Motivate International Inc.

Problem Statement

The business task was to determine how annual members (riders who pay for an annual pass) differ from casual riders (riders who pay for a single-ride pass or a day pass) based on their bike usage. This would help provide insights into how to convert these casual riders into annual members.

Basis

Finance analysts working on this project had concluded that annual members are more profitable than casual riders, therefore stakeholders want to have more riders become annual members. Rather than targeting the entire population of Chicago, stakeholders want the marketing team to target casual riders in their campaign to have more annual members. They believe that with the right strategy, casual riders will be more likely to become annual members. But first, they want to know how this subset of their users differ from annual members. To mimic the Professor from the hit Netflix series, Money Heist, "this is where I come in".

Data

Data Description

The description of the data sources has been provided above. Specifically, the data analyzed in this project was from the first quarter of 2022. There were 503421 observations and 13 variables, including ride id, the type of bicycle used, when each trip started and ended, the station names, ids, longitudes, and latitudes where each trip started and ended, and the type of rider it was (casual rider or an annual member). Riders' personally identifiable information was not included in the dataset, so it was impossible to compare casual riders with annual members using sociodemographic characteristics.

Data Cleaning

The data was in three different CSV files, one for each month in the quarter, so my first intuition was to merge this data into one using Google Sheet, make a copy of the combined dataset, clean and transform it, before transferring to a SQL database management system or RStudio, where I would do further analysis.

However, due to the sheer size of the dataset, running basic tasks such as extracting data from a column and using same to autofill new columns using functions and formulas took several minutes. Thus, I decided to use R for the data analysis from start to finish.

As a first step in my RStudio Cloud, I created a new folder in my working directory and imported the data files into it. I used this command to get the path to my working directory.

```
getwd()
```

```
## [1] "C:/Users/HP/Documents/web-apps/coursera/google-data-analytics/city-of-chicago-bikers"
```

Then, I installed the tidyverse package, and loaded the tidyverse and knitr packages which were needed to complete the analysis.

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

In order to merge different files into one in R, the column names should match. Therefore, I read each csv file and named the resulting data frame.

```
jan_2022 <- read.csv('2022-divvy-tripdata/202201-divvy-tripdata.csv')
feb_2022 <- read.csv('2022-divvy-tripdata/202202-divvy-tripdata.csv')
mar_2022 <- read.csv('2022-divvy-tripdata/202203-divvy-tripdata.csv')
```

Subsequently, I verified that the column names are the same in each data frame

```
colnames(jan_2022)
```

```
##  [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"
```

```
colnames(feb_2022)
```

```
##  [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"
```

```
colnames(mar_2022)
```

```
##  [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"
```

Since they are, I removed these data frames in order to reduce the RAM used for this analysis and save resources.

```
rm(jan_2022, feb_2022, mar_2022)
```

I then merged the datasets into one.

```
cyclistic <- list.files(path='2022-divvy-tripdata', full.names = TRUE) %>%
  lapply(read_csv) %>%
  bind_rows
```

I looked up the first 6 rows in the data frame - just to get a sense of what the data was like - using the `head` function and inspected the structure of the data using the `str` function.

```r
head(cyclistic)
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 C2F7DD78E82EC875 electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44
## 2 A6CF8980A652D272 electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17
## 3 BD0F91DFF741C66D classic_bike  2022-01-25 04:53:40 2022-01-25 04:58:01
## 4 CBB80ED419105406 classic_bike  2022-01-04 00:18:04 2022-01-04 00:33:00
## 5 DDC963BFDDA51EEA classic_bike  2022-01-20 01:31:10 2022-01-20 01:37:12
## 6 A39C6F6CC0586C0B classic_bike  2022-01-11 18:48:09 2022-01-11 18:51:31
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

```r
str(cyclistic)
```

```
## spc_tbl_ [503,421 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:503421] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB8
##  $ rideable_type     : chr [1:503421] "electric_bike" "electric_bike" "classic_bike" "classic_bike"
##  $ started_at        : POSIXct[1:503421], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...
##  $ ended_at          : POSIXct[1:503421], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...
##  $ start_station_name: chr [1:503421] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffiel
##  $ start_station_id  : chr [1:503421] "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr [1:503421] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave &
##  $ end_station_id    : chr [1:503421] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num [1:503421] 42 42 41.9 42 41.9 ...
##  $ start_lng         : num [1:503421] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:503421] 42 42 41.9 42 41.9 ...
##  $ end_lng           : num [1:503421] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:503421] "casual" "casual" "member" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The next thing I did was to check if there were columns with missing values. Six columns contained missing values, which I kept in mind, should I have to use data in these columns in subsequent steps of the analysis.

```r
cbind(
  lapply(
    lapply(cyclistic, is.na), sum
```

```
  )
)
```

```
##                     [,1]
## ride_id                0
## rideable_type          0
## started_at             0
## ended_at               0
## start_station_name 82086
## start_station_id   82086
## end_station_name   89439
## end_station_id     89439
## start_lat              0
## start_lng              0
## end_lat              429
## end_lng              429
## member_casual          0
```

These are the columns I am most interested in:

- `rideable_type`: the type of bicycle used for the trip
- `started_at`: the date and time the trip started
- `ended_at`: the date and time the trip ended
- `member_casual`: the type of rider it was, casual or an annual member.

At this point, it was imperative for me to confirm that there were no bad data in these columns.

How many types of bikes were there? Three distinct types were revealed by calling the `table` function. If there were trailing commas in the bike names, there would be one or more bike types seemingly repeated in the output of the function.

```
table(cyclistic$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##        248920         10680        243821
```

Intuitively, I knew the `started_at` and `ended_at` columns contained only the datetime data type values. If it were not so, the data type would have been character. More so, the time a trip ended should always be later than when it started. But this was not so, which implied bad data.

```
filter(cyclistic, ended_at < started_at)
```

```
## # A tibble: 2 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 2D97E3C98E165D80 classic_bike  2022-03-05 11:00:57 2022-03-05 10:55:01
## 2 7407049C5D89A13D electric_bike 2022-03-05 11:38:04 2022-03-05 11:37:57
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

Also, I found out that there were twenty-six cases where the trip started and ended at the same times. This is something I would like to clarify with the stakeholders the possible reasons why these might have occurred.

```
filter(cyclistic, ended_at == started_at)
```

```
## # A tibble: 26 x 13
##    ride_id          rideable_type started_at          ended_at
```

```
##    <chr>            <chr>         <dttm>              <dttm>
##  1 C2E047DDF019C255 electric_bike 2022-01-18 19:25:42 2022-01-18 19:25:42
##  2 8D3E8E511FEB11CC electric_bike 2022-01-21 01:05:35 2022-01-21 01:05:35
##  3 A753A729011B4289 electric_bike 2022-01-09 10:39:48 2022-01-09 10:39:48
##  4 0C63D14D261205FA electric_bike 2022-01-28 15:28:11 2022-01-28 15:28:11
##  5 4B0FC5ACEE52EBF9 electric_bike 2022-01-18 19:38:26 2022-01-18 19:38:26
##  6 6174209419E9E4F4 classic_bike  2022-02-17 14:20:58 2022-02-17 14:20:58
##  7 FA080339DCF1E51A classic_bike  2022-02-12 15:07:50 2022-02-12 15:07:50
##  8 2CF1A9AEEB7A2BC1 electric_bike 2022-02-17 09:39:01 2022-02-17 09:39:01
##  9 21582FCE30A61620 classic_bike  2022-02-16 14:32:32 2022-02-16 14:32:32
## 10 CF76E9E68F30EE55 classic_bike  2022-02-24 07:41:45 2022-02-24 07:41:45
## # i 16 more rows
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

For now, I would only remove those instances of bad data where the time the trip ended was earlier than when it started.

```
cyclistic_v02 <- filter(cyclistic, ended_at >= started_at)
```

After this, I confirmed that there were only two categories of riders.

```
table(cyclistic_v02$member_casual)
```

```
##
## casual member
## 129816 373603
```

And with this, dearest readers - mimicking Lady Whistledown in Bridgerton, another hit Netflix series - I have come to the end of the data cleaning process. Up next is data transformation.

Data Transformation

Three key metrics I was interested in were:

- the duration of each trip, in minutes
- the day of the week each trip started
- the month each trip happened

Thus, I created three new columns for each and used a formula and a few functions to deduce or extract these from the `started_at` and `ended_at` columns.

```
cyclistic_v02 <- cyclistic_v02 %>%
  mutate(
    ride_length = seconds_to_period(ended_at - started_at),
    day_of_week = wday(started_at, label = TRUE),
    month_of_year = month(started_at, label = TRUE, abbr = TRUE),
    .after = ended_at
  )
```

Also, I renamed the categories of riders in the `member_casual` column.

```
cyclistic_v02$member_casual[cyclistic_v02$member_casual == "casual"] <- "Casual Rider"
cyclistic_v02$member_casual[cyclistic_v02$member_casual == "member"] <- "Annual Member"
```

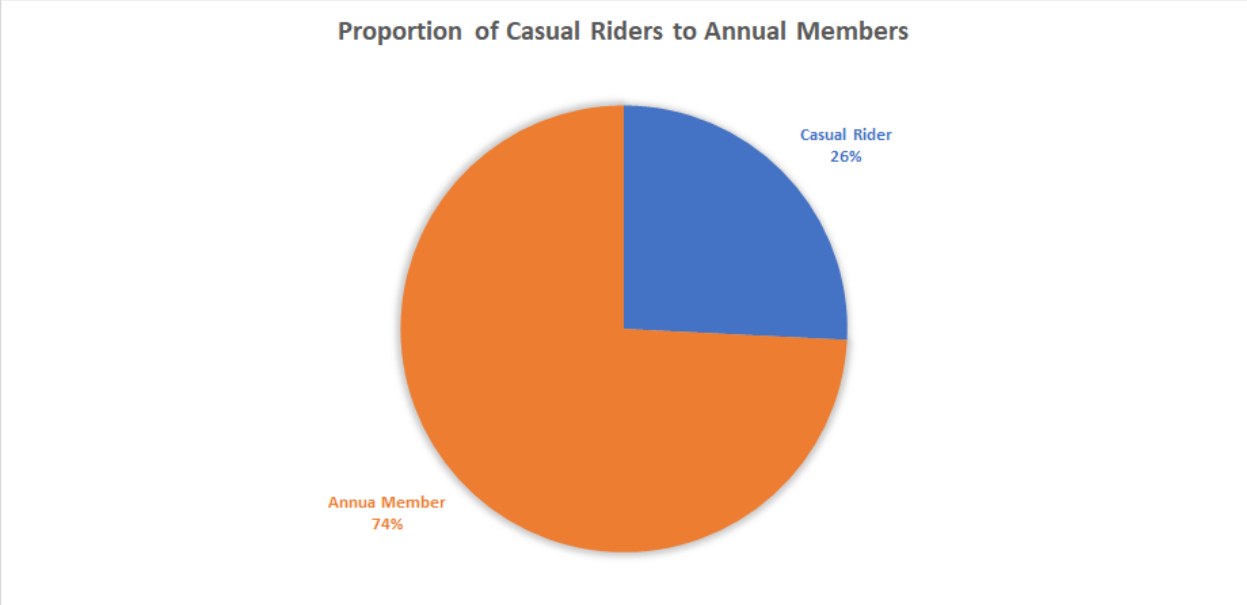
Results and Discussion

How many riders per group were there in the last quarter?

I generated a CSV file which contained data on the number of casual riders and annual members who made a trip using Divvy bikes in the last quarter.

```
write_csv(
  data.frame(table(cyclistic_v02$member_casual)),
  file = "riders.csv"
)
```

I opened the CSV file in Excel which I used to create the pie chart below. I could have used the `pie` function in R to create something similar *but I must sha use Excel in this project.* Concretely, out of the 503419 trips that were made, 74% (373603) of them were by annual members.



What was the average duration per trip for the different rider types?

I used the `aggregate` function to generate the summary statistics, which I then displayed in a tabular form. The mean duration per trip for casual riders in Q1 2022 was 31M 20S and that of annual members was 11M 49S. This makes sense. Let me use this analogy to explain why: If it were possible to do a day subscription on Netflix, you would probably spend more time watching movies that day than you would if you were on a monthly (or annual) subscription. Something about getting maximum value for your money.

```
aggregate(ride_length ~ member_casual, data = cyclistic_v02, summary)
```

```
##   member_casual ride_length.Min. ride_length.1st Qu. ride_length.Median
## 1 Annual Member               0S              4M 39S             7M 46S
## 2  Casual Rider               0S              7M 14S            12M 54S
##        ride_length.Mean ride_length.3rd Qu. ride_length.Max.
## 1 11M 49.4547768620702S             13M 26S    1d 1H 59M 54S
## 2 31M 19.6221112959881S             24M 30S   23d 20H 34M 4S
```

```
summary_stat <- data.frame(
  Rider = c("Casual Rider", "Annual Member"),
  Minimum = c("0S", "0S"),
  Median = c("12M 54S", "7M 46S"),
```

```
  Mean = c("31M 20S", "11M 49S"),
  Maximum = c("23d 20H 34M 4S", "1d 1H 59M 54S")
)

kable(summary_stat, caption = "Descriptive statistics on trip duration group by rider type")
```

Table 1: Descriptive statistics on trip duration group by rider type

Rider	Minimum	Median	Mean	Maximum
Casual Rider	0S	12M 54S	31M 20S	23d 20H 34M 4S
Annual Member	0S	7M 46S	11M 49S	1d 1H 59M 54S

How about the average trip duration on different days?

The table below shows the daily average trip duration grouped by the category of rider.

```
avg_ride_duration_per_day <- aggregate(as.numeric(ride_length)/60 ~ member_casual + day_of_week, data =

avg_ride_duration_per_day <- rename(avg_ride_duration_per_day,
       Rider = member_casual,
       `Day of Week` = day_of_week,
       `Mean (in minutes)` = `as.numeric(ride_length)/60`)

kable(avg_ride_duration_per_day, caption = "Daily average trip duration")
```

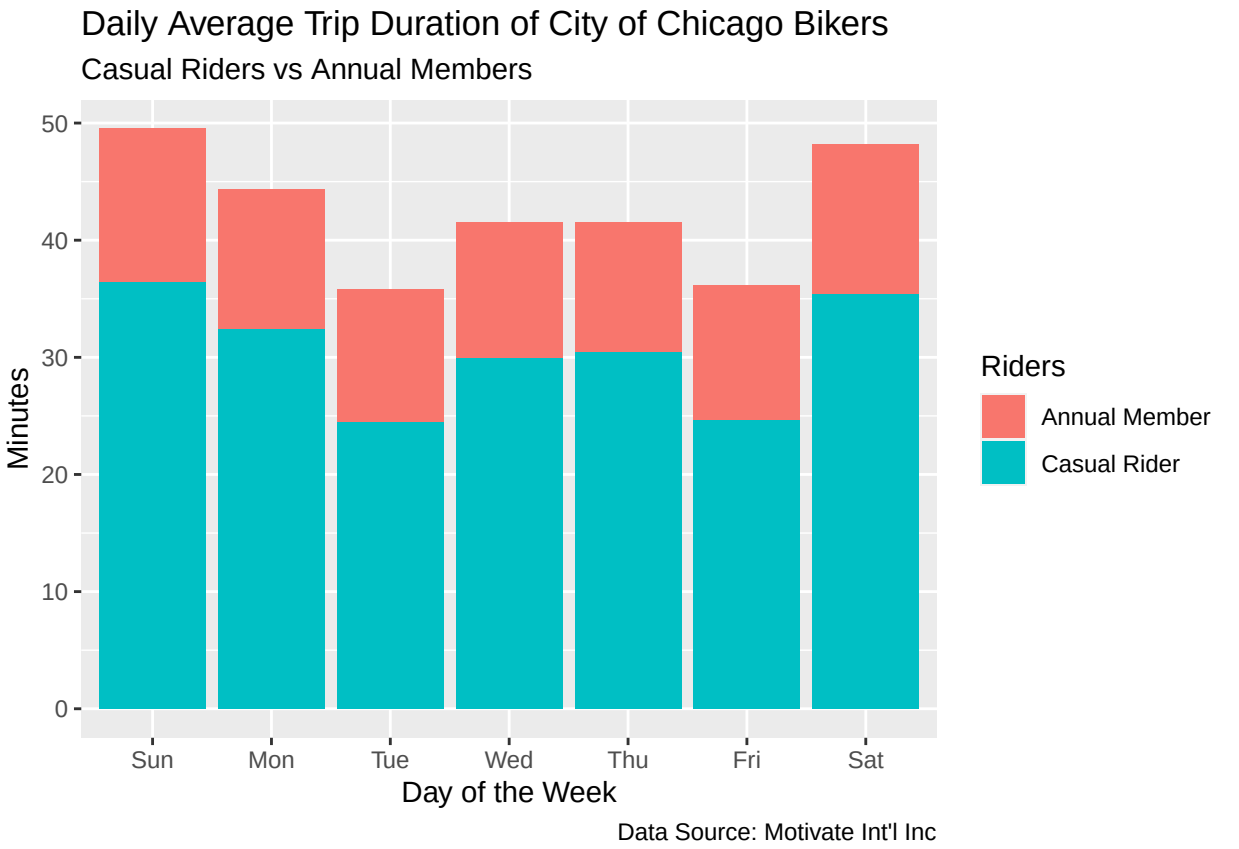
Table 2: Daily average trip duration

Rider	Day of Week	Mean (in minutes)
Annual Member	Sun	13.07604
Casual Rider	Sun	36.42812
Annual Member	Mon	11.99076
Casual Rider	Mon	32.39739
Annual Member	Tue	11.30266
Casual Rider	Tue	24.49017
Annual Member	Wed	11.63723
Casual Rider	Wed	29.89225
Annual Member	Thu	11.07391
Casual Rider	Thu	30.47099
Annual Member	Fri	11.46216
Casual Rider	Fri	24.65541
Annual Member	Sat	12.84091
Casual Rider	Sat	35.37999

This is better visualized using the chart below:

```
cyclistic_v02 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_duration = mean(as.numeric(ride_length))/60) %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "stack") +
  labs(title = "Daily Average Trip Duration of City of Chicago Bikers",
       subtitle = "Casual Riders vs Annual Members",
```

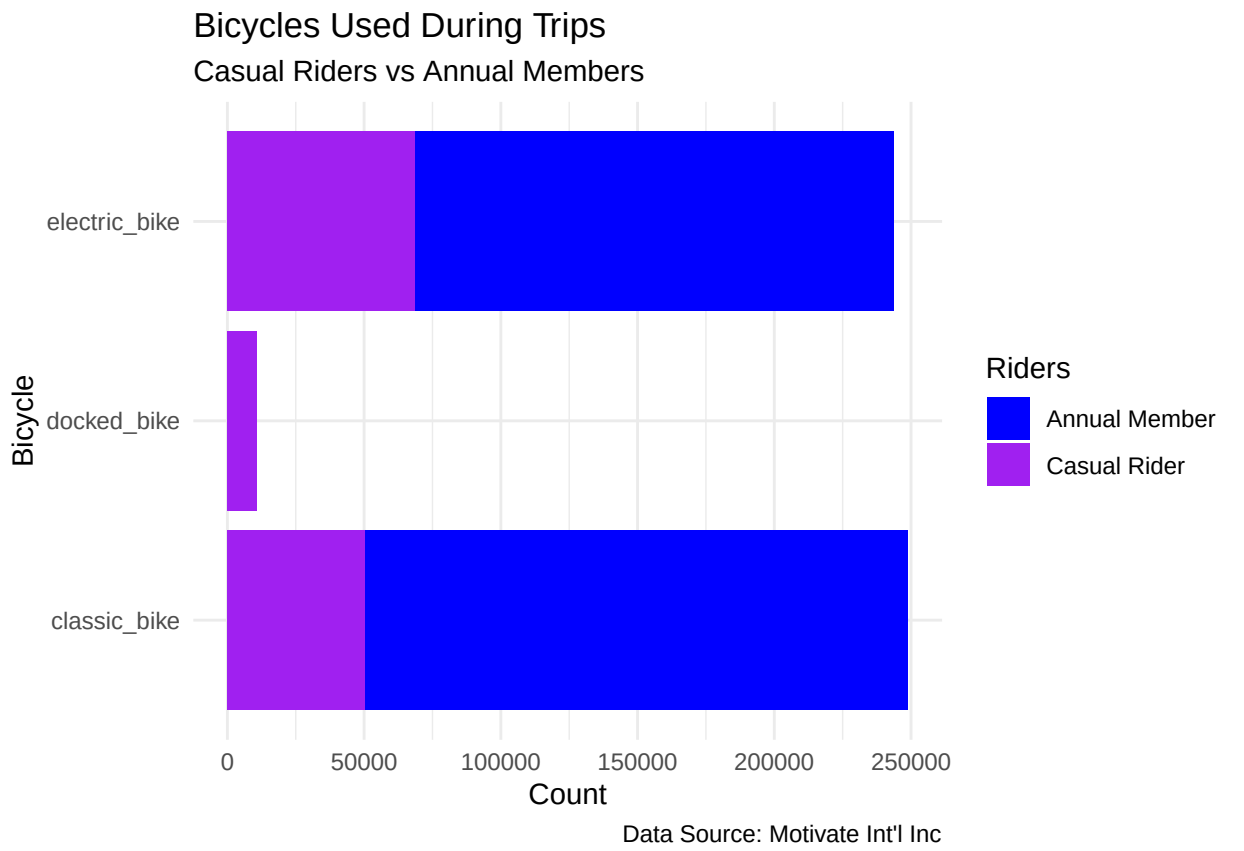
```
      caption = "Data Source: Motivate Int'l Inc",
      fill = "Riders") +
  xlab("Day of the Week") +
  ylab("Minutes")
```



What types of bicycles were used for the trips?

Casual riders used all types of bicycles, mostly electric bikes, closely followed by classic bikes, but also docked bikes. Annual members never used docked bikes. They embarked on a majority of their trips with the classic bicycles.

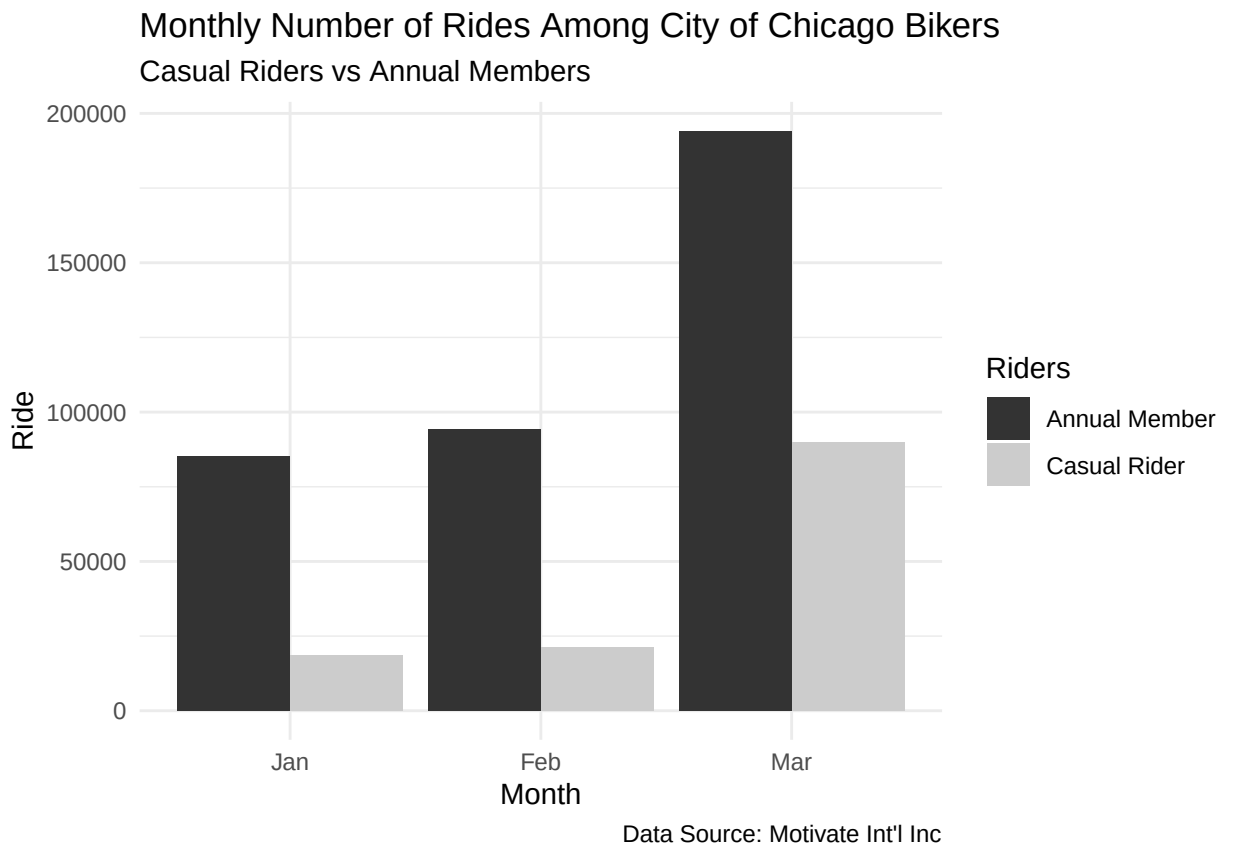
```
cyclistic_v02 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = rideable_type, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "stack") +
  coord_flip() +
  scale_fill_manual(values = c("blue", "purple")) +
  theme_minimal() +
  labs(title = "Bicycles Used During Trips",
       subtitle = "Casual Riders vs Annual Members",
       caption = "Data Source: Motivate Int'l Inc",
       fill = "Riders") +
  xlab("Bicycle") +
  ylab("Count")
```



How many rides were there per month?

The number of monthly rides was lowest in January (18520/85250), increasing steadily, with a two to four-fold increase from February (21416/94193) to March (89880/194160), across both rider groups (casual riders/annual members).

```
cyclistic_v02 %>%
  group_by(member_casual, month_of_year) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = month_of_year, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  scale_fill_grey() +
  theme_minimal() +
  labs(title = "Monthly Number of Rides Among City of Chicago Bikers",
       subtitle = "Casual Riders vs Annual Members",
       caption = "Data Source: Motivate Int'l Inc",
       fill = "Riders") +
  xlab("Month") +
  ylab("Ride")
```



Conclusions

Here is a recap of my key insights from this analysis:

- Bikers rode the longest on weekends, particularly Sundays, and the duration of trips generally declined from Mondays to Fridays.
- Casual riders typically cycled for a longer duration than annual members.
- Casual riders mostly used electric bikes whereas annual members mostly used classic bikes.
- Casual riders sometimes used docked bikes while annual members never did.
- There was a two-fold and a four-fold increase in the number of rides undertaken by annual members and casual riders respectively from February to March 2022.

Recommendations

- Marketing efforts should be most intensive during weekends.
- Campaigns should be focused on those riders who use classic and electric bikes.
- Factors or company policies that might have caused the high surge in the number of rides from February to March should be investigated and capitalized on to drive further growth.

Limitations

The datasets did not include riders' personally identifiable information. As such, it cannot be determined if some casual riders kept coming back for more trips. Such riders are more likely to become annual members and could have been targeted in marketing campaigns.

More information is needed to understand why some riders (26) ended their trips at the same time they started them. Were they dissatisfied with Lyft Bikes and Scooters offering? Did they pay for the trip but had

to leave due to an emergency? Whatever their reasons may be, they are unlikely to influence the analysis done in this project significantly as their number was small and almost equally distributed between both rider groups (11 casual riders to 15 annual members).

Final Remarks

Thank you for reading this far. What do you like about this case study? What do you agree or disagree with? Please leave suggestions, questions, corrections or clarifications in the comments.

Oh! I'd definitely be doing more case studies. Please follow me so you don't miss out on the next one.

I am open to data analyst roles. Please refer me or reach out to me if you're hiring. Thank you.