Google Data Analytics Professional Certificate Capstone Project: How does a bike-share navigate speedy success?

By Fletcher Henson

May 30, 2022

Case Study: Help Cyclistic, a bike-share company, to convert casual members into annual members

In this article I will show my approach to solve the Google Data Analytics Professional Certificate Capstone Project.

As I learned from the Google Data Analytics program, I will use the six phases of the data analysis process: **ask**, **prepare**, **process**, **analyze**, **share, and act** to help complete the case study.

About the Company

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. The bikes can be unlocked from one station and returned to any other station in the system anytime.

The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, team wants to understand how casual riders and annual members use Cyclistic bikes differently.

Business Task

Help analyze data to answer this question: **How do annual members and casual riders use Cyclistic bikes differently?**

Step 1: Ask

There are three questions that Cyclistic's marketing team is looking to answer:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

Step 2: Prepare

For this step, I collected data that has been organized in monthly .csv files from the source provided for this project (Data Source).

Sub-questions:

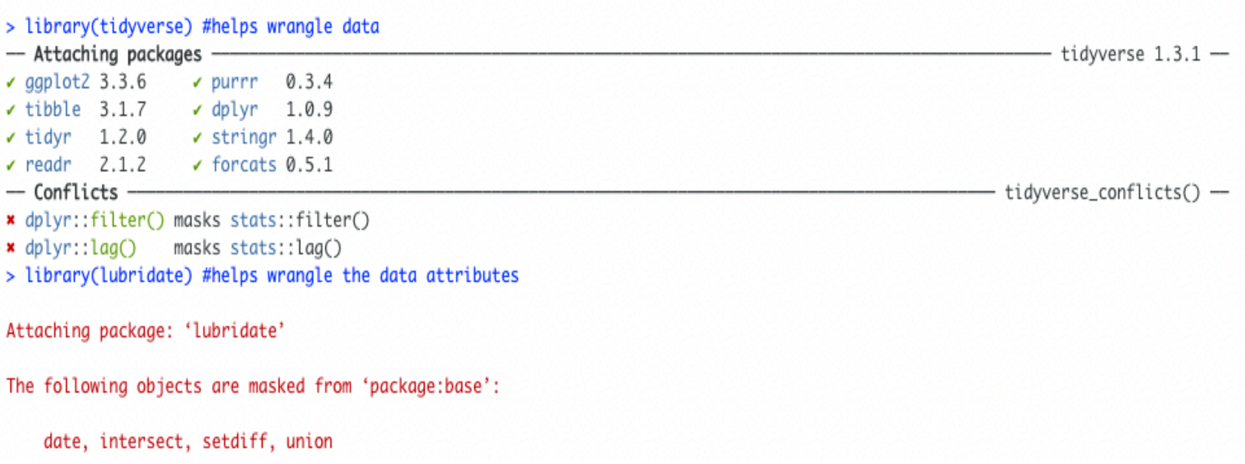
1. **Are there issues with bias or credibility in this data?** This data is from 2021 and has been collected by Cyclistic directly. It includes all rides from 2021, and therefore it is not a sample from the whole dataset. Since the data has been collected from the company directly, it is possible to assume that there are no issues around bias or credibility with this data.

2. **How are you addressing Licensing, Privacy, Security, and Accessibility?** The data is protected under this license (License). The data does not contain any personal information of the users, therefore there are no privacy concerns.

Step 3: Process

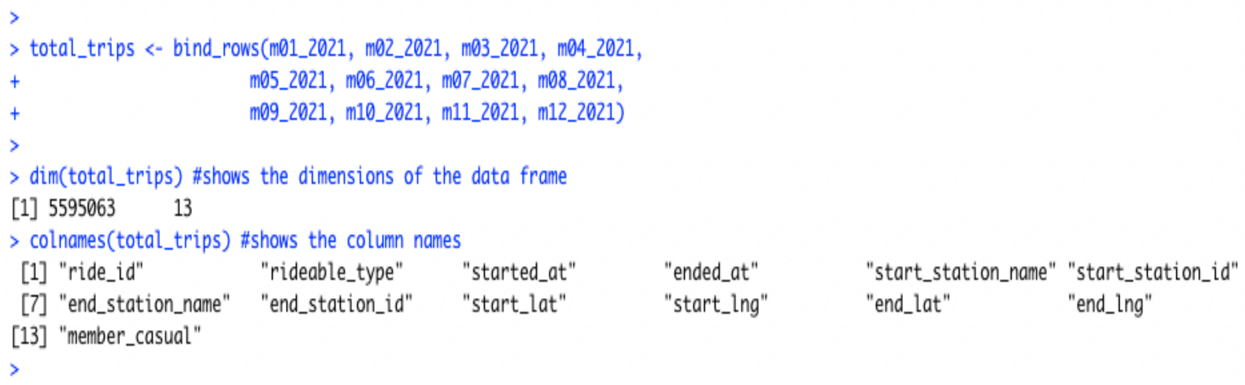
In this step, I processed the data to get it ready for the next step which is where I will look for insights in data that will help me answer the stakeholder questions. For this step, I have chosen to use R since we are working with very large datasets.

Below, I started by importing the relevant packages that are needed to begin:



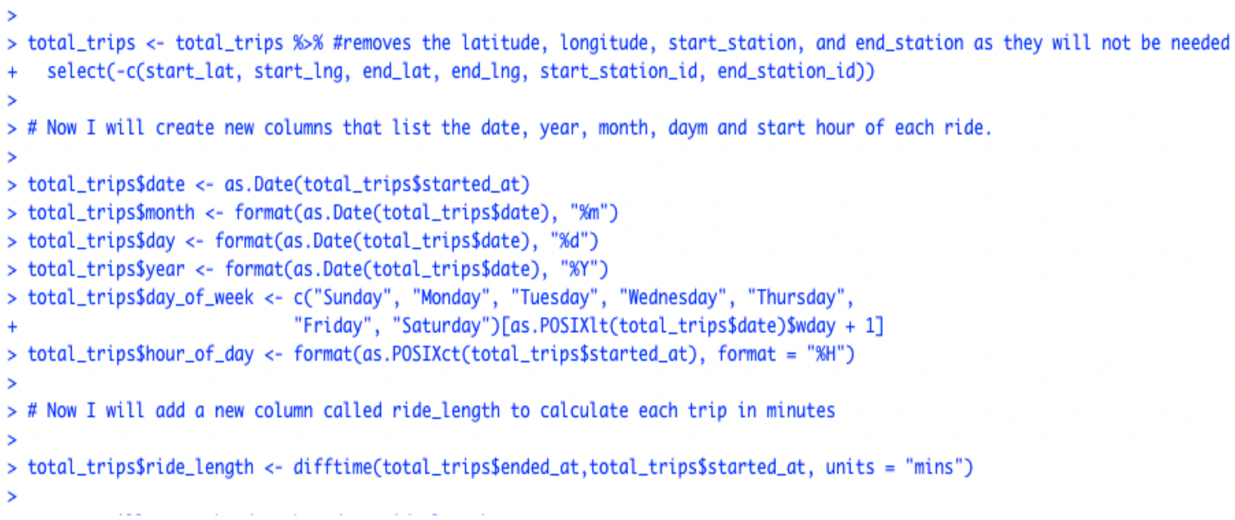
```
> library(tidyverse) #helps wrangle data
── Attaching packages ─────────────────────────────────────────────── tidyverse 1.3.1 ──
✔ ggplot2 3.3.6     ✔ purrr   0.3.4
✔ tibble  3.1.7     ✔ dplyr   1.0.9
✔ tidyr   1.2.0     ✔ stringr 1.4.0
✔ readr   2.1.2     ✔ forcats 0.5.1
── Conflicts ────────────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
> library(lubridate) #helps wrangle the data attributes

Attaching package: ‘lubridate’

The following objects are masked from ‘package:base’:

    date, intersect, setdiff, union
```

Next, I will combine all the files from the past 12 months of data given for this project into one, new file.



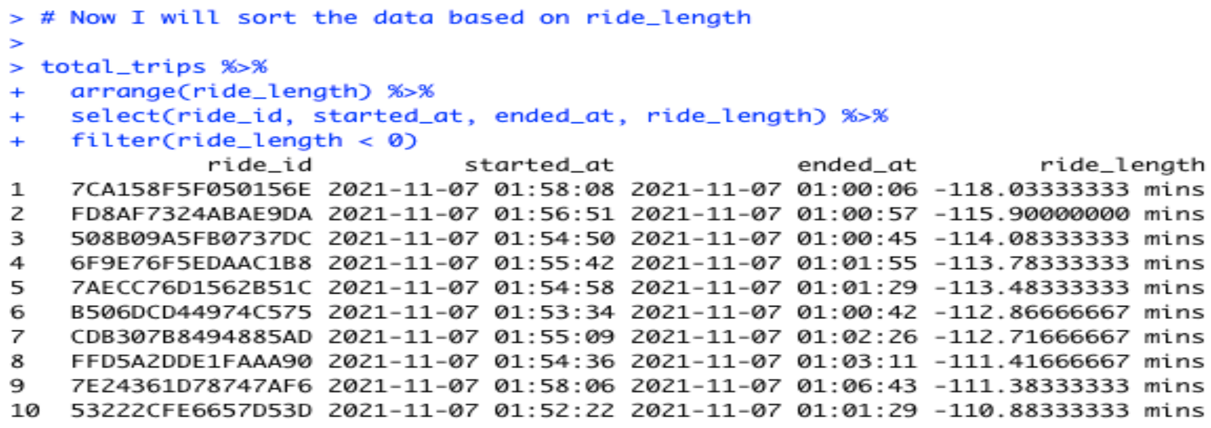
```
>
> total_trips <- bind_rows(m01_2021, m02_2021, m03_2021, m04_2021,
+                          m05_2021, m06_2021, m07_2021, m08_2021,
+                          m09_2021, m10_2021, m11_2021, m12_2021)
>
> dim(total_trips) #shows the dimensions of the data frame
[1] 5595063      13
> colnames(total_trips) #shows the column names
 [1] "ride_id"            "rideable_type"      "started_at"         "ended_at"           "start_station_name" "start_station_id"  
 [7] "end_station_name"   "end_station_id"     "start_lat"          "start_lng"          "end_lat"            "end_lng"           
[13] "member_casual"     
>
```

Now I created some new columns that are going to list the year, month, day, date, and start hour of each ride, and added a column that calculated each trip in minutes.



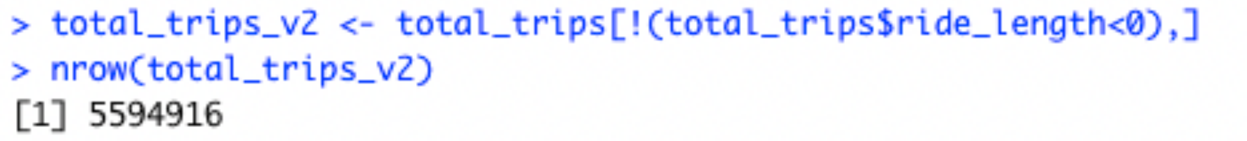
```
>
> total_trips <- total_trips %>% #removes the latitude, longitude, start_station, and end_station as they will not be needed
+   select(-c(start_lat, start_lng, end_lat, end_lng, start_station_id, end_station_id))
>
> # Now I will create new columns that list the date, year, month, daym and start hour of each ride.
>
> total_trips$date <- as.Date(total_trips$started_at)
> total_trips$month <- format(as.Date(total_trips$date), "%m")
> total_trips$day <- format(as.Date(total_trips$date), "%d")
> total_trips$year <- format(as.Date(total_trips$date), "%Y")
> total_trips$day_of_week <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
+                              "Friday", "Saturday")[as.POSIXlt(total_trips$date)$wday + 1]
> total_trips$hour_of_day <- format(as.POSIXct(total_trips$started_at), format = "%H")
>
> # Now I will add a new column called ride_length to calculate each trip in minutes
>
> total_trips$ride_length <- difftime(total_trips$ended_at,total_trips$started_at, units = "mins")
>
```

After this I sorted the data on ride length and found a few errors (i.e., negative values).



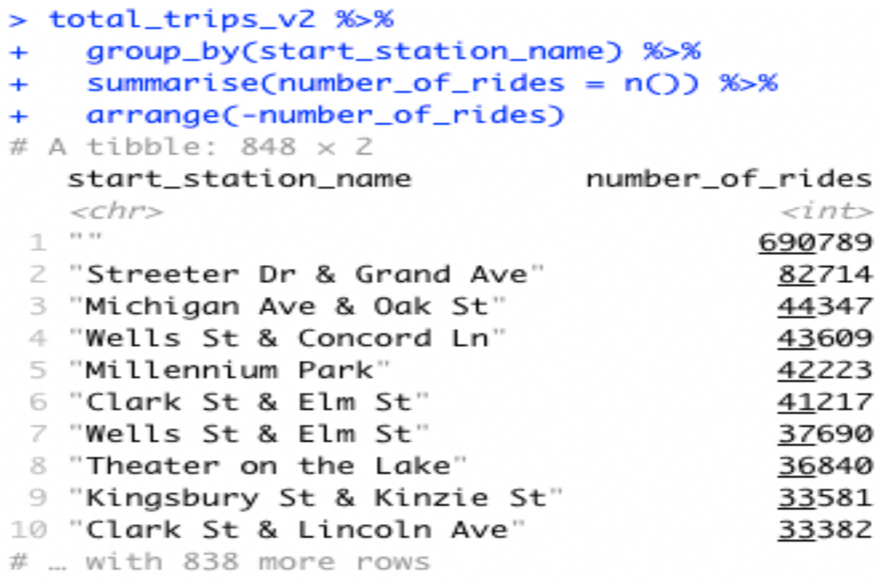
```
> # Now I will sort the data based on ride_length
>
> total_trips %>%
+   arrange(ride_length) %>%
+   select(ride_id, started_at, ended_at, ride_length) %>%
+   filter(ride_length < 0)
            ride_id          started_at            ended_at      ride_length
1  7CA158F5F050156E 2021-11-07 01:58:08 2021-11-07 01:00:06 -118.03333333 mins
2  FD8AF7324ABAE9DA 2021-11-07 01:56:51 2021-11-07 01:00:57 -115.90000000 mins
3  508B09A5FB0737DC 2021-11-07 01:54:50 2021-11-07 01:00:45 -114.08333333 mins
4  6F9E76F5EDAAC1B8 2021-11-07 01:55:42 2021-11-07 01:01:55 -113.78333333 mins
5  7AECC76D1562B51C 2021-11-07 01:54:58 2021-11-07 01:01:29 -113.48333333 mins
6  B506DCD44974C575 2021-11-07 01:53:34 2021-11-07 01:00:42 -112.86666667 mins
7  CDB307B8494885AD 2021-11-07 01:55:09 2021-11-07 01:02:26 -112.71666667 mins
8  FFD5A2DDE1FAAA90 2021-11-07 01:54:36 2021-11-07 01:03:11 -111.41666667 mins
9  7E24361D78747AF6 2021-11-07 01:58:06 2021-11-07 01:06:43 -111.38333333 mins
10 53222CFE6657D53D 2021-11-07 01:52:22 2021-11-07 01:01:29 -110.88333333 mins
```

I then removed the negative values from the dataset. In total, 147 negative values were removed.

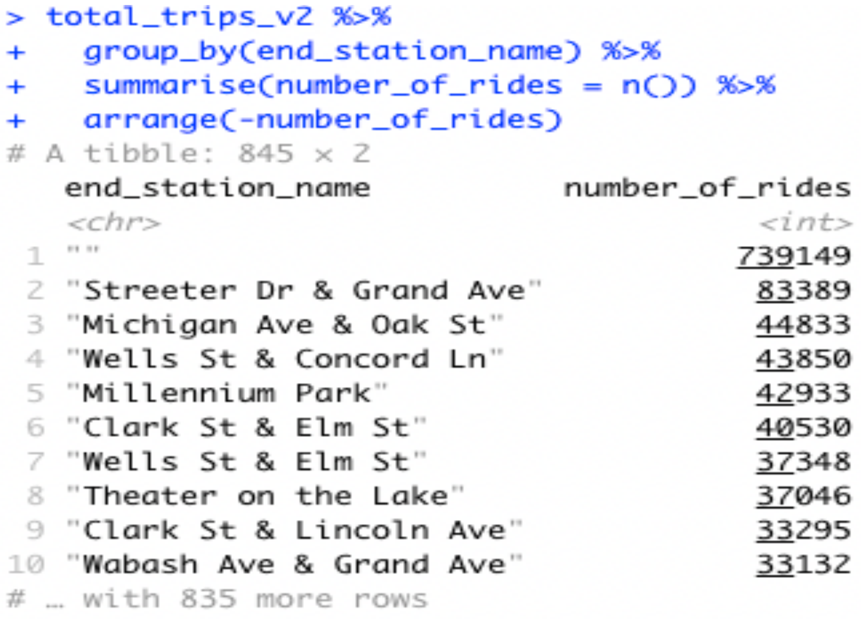


```
> total_trips_v2 <- total_trips[!(total_trips$ride_length<0),]
> nrow(total_trips_v2)
[1] 5594916
```

Next, I checked for NA values in the start station and end station columns.

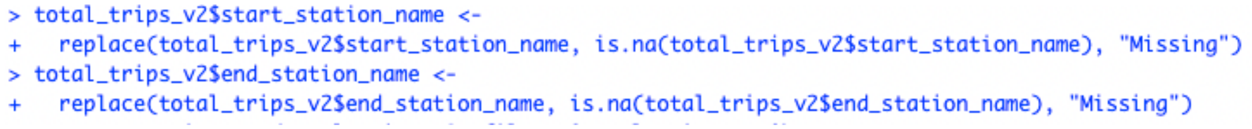


```
> total_trips_v2 %>%
+   group_by(start_station_name) %>%
+   summarise(number_of_rides = n()) %>%
+   arrange(-number_of_rides)
# A tibble: 848 × 2
   start_station_name           number_of_rides
   <chr>                                  <int>
 1 ""                                    690789
 2 "Streeter Dr & Grand Ave"              82714
 3 "Michigan Ave & Oak St"                44347
 4 "Wells St & Concord Ln"                43609
 5 "Millennium Park"                      42223
 6 "Clark St & Elm St"                    41217
 7 "Wells St & Elm St"                    37690
 8 "Theater on the Lake"                  36840
 9 "Kingsbury St & Kinzie St"             33581
10 "Clark St & Lincoln Ave"               33382
# … with 838 more rows
```



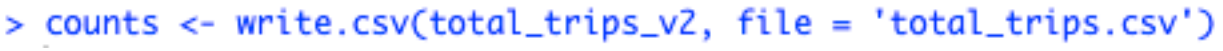
```
> total_trips_v2 %>%
+   group_by(end_station_name) %>%
+   summarise(number_of_rides = n()) %>%
+   arrange(-number_of_rides)
# A tibble: 845 × 2
   end_station_name           number_of_rides
   <chr>                                <int>
 1 ""                                  739149
 2 "Streeter Dr & Grand Ave"            83389
 3 "Michigan Ave & Oak St"              44833
 4 "Wells St & Concord Ln"              43850
 5 "Millennium Park"                    42933
 6 "Clark St & Elm St"                  40530
 7 "Wells St & Elm St"                  37348
 8 "Theater on the Lake"                37046
 9 "Clark St & Lincoln Ave"             33295
10 "Wabash Ave & Grand Ave"             33132
# … with 835 more rows
```

The NA values were kept in the dataset and assigned a different name so that they can be analyzed as well.



```
> total_trips_v2$start_station_name <-
+   replace(total_trips_v2$start_station_name, is.na(total_trips_v2$start_station_name), "Missing")
> total_trips_v2$end_station_name <-
+   replace(total_trips_v2$end_station_name, is.na(total_trips_v2$end_station_name), "Missing")
```

Lastly, I exported the data into a csv file so that we can upload it to Tableau for analysis.



```
> counts <- write.csv(total_trips_v2, file = 'total_trips.csv')
```

Tableau Dashboard:

https://public.tableau.com/app/profile/fletcher.henson/viz/GoogleDataAnalyticsCapstoneProject_16569528739880/Story1