Capstone

Justin P

2025-06-05

Capstone Project

Project Description:This R Markdown file is the main source of data cleaning and data analytics for the Google Data Analytics Capstone Project. It attempts to answer the question: How do annual members and casual riders use Cyclistic bikes differently? This is part of Track A from the Capstone Project portion of the course using the bike-share dataset, Cyclistic, available by Motivate International Inc.

Install packages (if needed)

Load environment and relevant packages

```
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.1     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.2.1
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(readr)
library(tidyr)
library(dplyr)
```

Load Datasets

```
Q1_2019_df <- read_csv("C:/Users/Justin/Documents/Coursera/Google Data Analytics Professional Ce
rtificate/Course 8/Case_Study1/Data/Divvy_Trips_2019_Q1 - Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 365069 Columns: 12
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (6): start_time, end_time, from_station_name, to_station_name, usertype,...
## dbl (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num (1): tripduration
##
## 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.
```

```
Q1_2020_df <- read_csv("C:/Users/Justin/Documents/Coursera/Google Data Analytics Professional Ce
rtificate/Course 8/Case_Study1/Data/Divvy_Trips_2020_Q1 - Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 426887 Columns: 13
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, en...
##
## 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.
```

Look at Q1-2019

```
head(Q1_2019_df)
```

```
## # A tibble: 6 × 12
##    trip_id start_time          end_time       bikeid tripduration from_station_id
##      <dbl> <chr>               <chr>           <dbl>        <dbl>           <dbl>
## 1 21742443 2019-01-01 0:04:37 2019-01-01 0:…   2167          390             199
## 2 21742444 2019-01-01 0:08:13 2019-01-01 0:…   4386          441              44
## 3 21742445 2019-01-01 0:13:23 2019-01-01 0:…   1524          829              15
## 4 21742446 2019-01-01 0:13:45 2019-01-01 0:…    252         1783             123
## 5 21742447 2019-01-01 0:14:52 2019-01-01 0:…   1170          364             173
## 6 21742448 2019-01-01 0:15:33 2019-01-01 0:…   2437          216              98
## # ℹ 6 more variables: from_station_name <chr>, to_station_id <dbl>,
## #   to_station_name <chr>, usertype <chr>, gender <chr>, birthyear <dbl>
```

Column names:

```
colnames(Q1_2019_df)
```

```
##  [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"
```

Look at Q1-2020

```
head(Q1_2020_df)
```

```
## # A tibble: 6 × 13
##   ride_id  rideable_type started_at ended_at start_station_name start_station_id
##   <chr>    <chr>         <chr>      <chr>    <chr>                         <dbl>
## 1 EACB191… docked_bike   2020-01-2… 2020-01… Western Ave & Lel…              239
## 2 8FED874… docked_bike   2020-01-3… 2020-01… Clark St & Montro…              234
## 3 789F3C2… docked_bike   2020-01-0… 2020-01… Broadway & Belmon…              296
## 4 C9A388D… docked_bike   2020-01-0… 2020-01… Clark St & Randol…               51
## 5 943BC3C… docked_bike   2020-01-3… 2020-01… Clinton St & Lake…               66
## 6 6D9C8A6… docked_bike   2020-01-1… 2020-01… Wells St & Hubbar…              212
## # ℹ 7 more variables: end_station_name <chr>, end_station_id <dbl>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>
```

Column names:

```
colnames(Q1_2020_df)
```

```
##  [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"
```

Data Cleaning

Make 2019 & 2020 column names consistent

Column mapping {2019:2020}: {trip_id: ride_id, start_time: started_at, end_time: ended_at, from_station_id: start_station_id, from_station_name: start_station_name, to_station_id: end_station_id, usertype: member_casual, ride_length: ride_length, day_of_week: day_of_week}

Columns not present in both: Q1_2019: {bikeid, tripduration, gender, birthyear} | Q1_2020: {rideable_type, start_lat, start_lng, end_lat, end_lng}

```
library(dplyr)

Q1_2020_df_renamed <- Q1_2020_df %>%
  rename(
    trip_id           = ride_id,
    start_time        = started_at,
    end_time          = ended_at,
    from_station_id   = start_station_id,
    from_station_name = start_station_name,
    to_station_id     = end_station_id,
    to_station_name     = end_station_name,
    usertype          = member_casual,
  )

colnames(Q1_2020_df_renamed)
```

```
##  [1] "trip_id"           "rideable_type"     "start_time"
##  [4] "end_time"          "from_station_name" "from_station_id"
##  [7] "to_station_name"   "to_station_id"     "start_lat"
## [10] "start_lng"         "end_lat"           "end_lng"
## [13] "usertype"
```

```
head(Q1_2020_df_renamed)
```

```
## # A tibble: 6 × 13
##   trip_id    rideable_type start_time end_time from_station_name from_station_id
##   <chr>      <chr>         <chr>      <chr>    <chr>                       <dbl>
## 1 EACB19130… docked_bike   2020-01-2… 2020-01… Western Ave & Le…             239
## 2 8FED874C8… docked_bike   2020-01-3… 2020-01… Clark St & Montr…             234
## 3 789F3C21E… docked_bike   2020-01-0… 2020-01… Broadway & Belmo…             296
## 4 C9A388DAC… docked_bike   2020-01-0… 2020-01… Clark St & Rando…              51
## 5 943BC3CBE… docked_bike   2020-01-3… 2020-01… Clinton St & Lak…              66
## 6 6D9C8A693… docked_bike   2020-01-1… 2020-01… Wells St & Hubba…             212
## # ℹ 7 more variables: to_station_name <chr>, to_station_id <dbl>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   usertype <chr>
```

Check the structure of the dataframes

```
str(Q1_2019_df)
```

```
## spc_tbl_ [365,069 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:365069] 21742443 21742444 21742445 21742446 21742447 ...
##  $ start_time       : chr [1:365069] "2019-01-01 0:04:37" "2019-01-01 0:08:13" "2019-01-01 0:
13:23" "2019-01-01 0:13:45" ...
##  $ end_time         : chr [1:365069] "2019-01-01 0:11:07" "2019-01-01 0:15:34" "2019-01-01 0:
27:12" "2019-01-01 0:43:28" ...
##  $ bikeid           : num [1:365069] 2167 4386 1524 252 1170 ...
##  $ tripduration     : num [1:365069] 390 441 829 1783 364 ...
##  $ from_station_id  : num [1:365069] 199 44 15 123 173 98 98 211 150 268 ...
##  $ from_station_name: chr [1:365069] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racin
e Ave & 18th St" "California Ave & Milwaukee Ave" ...
##  $ to_station_id    : num [1:365069] 84 624 644 176 35 49 49 142 148 141 ...
##  $ to_station_name  : chr [1:365069] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St
(*)" "Western Ave & Fillmore St (*)" "Clark St & Elm St" ...
##  $ usertype         : chr [1:365069] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender           : chr [1:365069] "Male" "Female" "Female" "Male" ...
##  $ birthyear        : num [1:365069] 1989 1990 1994 1993 1994 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_character(),
##   ..   end_time = col_character(),
##   ..   bikeid = col_double(),
##   ..   tripduration = col_number(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   usertype = col_character(),
##   ..   gender = col_character(),
##   ..   birthyear = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(Q1_2020_df_renamed)
```

```
## spc_tbl_ [426,887 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trip_id          : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96"
"C9A388DAC6ABF313" ...
##  $ rideable_type    : chr [1:426887] "docked_bike" "docked_bike" "docked_bike" "docked_bike"
...
##  $ start_time       : chr [1:426887] "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09
19:29:26" "2020-01-06 16:17:07" ...
##  $ end_time         : chr [1:426887] "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09
19:32:17" "2020-01-06 16:25:56" ...
##  $ from_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Br
oadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ from_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 ...
##  $ to_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd"
"Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ...
##  $ to_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat        : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ start_lng        : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat          : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ end_lng          : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ usertype         : chr [1:426887] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Join the two data files

In R

First, the two dataframe columns must be aligned

```
all_cols <- c("trip_id", "start_time", "end_time", "from_station_id", "from_station_name", "to_s
tation_id", "to_station_name", "usertype")
all_cols
```

```
## [1] "trip_id"           "start_time"        "end_time"
## [4] "from_station_id"   "from_station_name" "to_station_id"
## [7] "to_station_name"   "usertype"
```

Get all relevant columns for 2019 and 2020

```
df_2019 <- Q1_2019_df[all_cols]
df_2020 <- Q1_2020_df_renamed[all_cols]
```

Add a year column to each dataframe to keep track of users per year

```
df_2019["Year"] = 2019
df_2020["Year"] = 2020
```

Last adjustments, making sure columns are of the same type

```
df_2019$trip_id <- as.character(df_2019$trip_id) # double to character
```

Finally combine the 2019 and 2020 data frames

```
combined_df <- bind_rows(df_2019, df_2020)
head(combined_df)
```

```
## # A tibble: 6 × 9
##   trip_id  start_time   end_time from_station_id from_station_name to_station_id
##   <chr>    <chr>        <chr>              <dbl> <chr>                     <dbl>
## 1 21742443 2019-01-01 … 2019-01…             199 Wabash Ave & Gra…            84
## 2 21742444 2019-01-01 … 2019-01…              44 State St & Rando…           624
## 3 21742445 2019-01-01 … 2019-01…              15 Racine Ave & 18t…           644
## 4 21742446 2019-01-01 … 2019-01…             123 California Ave &…           176
## 5 21742447 2019-01-01 … 2019-01…             173 Mies van der Roh…           35
## 6 21742448 2019-01-01 … 2019-01…              98 LaSalle St & Was…           49
## # ℹ 3 more variables: to_station_name <chr>, usertype <chr>, Year <dbl>
```

In SQL

Export data frames and move to SQL to join dataset

```
write.csv(Q1_2019_df, "2019_partial_clean.csv", row.names = FALSE)
write.csv(Q1_2020_df_renamed, "2020_partial_clean.csv", row.names = FALSE)
```

Move to SQL and join datasets

This code was run in BigQuery. And will not work with the current markdown.

Using BigQuery as an easy SQL server. Steps: 1) Create a new project. 2) Create a new dataset, call it bicycle. 3) Upload “2019_partial_clean.csv” and “2020_partial_clean.csv” to the bicycle data set as tables (2019, 2020). 4) Run the code below in BigQuery.

```
#SELECT
#  CAST(trip_id AS STRING) AS trip_id, # recast trip_id from int64 to string
#  start_time,
#  end_time,
#  CAST(from_station_id AS STRING) AS from_station_id, # make sure from_station_id is a string
#  from_station_name,
#  CAST(to_station_id AS STRING) AS to_station_id, # make sure to_station_id is a string
#  to_station_name,
#  usertype,
#  2019 AS Year, # Add a "Year" column for all users for 2019.
#  TIMESTAMP_DIFF(end_time, start_time, MINUTE) AS trip_duration # find the trip duration for ea
ch user
#FROM `project-1-461316.bicycle.2019`

#UNION ALL # combine all rows and columns from both tables

#SELECT
#  CAST(ride_id AS STRING) AS trip_id, # recast trip_id from int64 to string
#  started_at AS start_time,
#  ended_at AS end_time,
#  CAST(start_station_id AS STRING) AS from_station_id,  # make sure start_station_id is a strin
g
#  start_station_name AS from_station_name,
#  CAST(end_station_id AS STRING) AS to_station_id,  #  # make sure end_station_id is a string
#  end_station_name AS to_station_name,
#  member_casual AS usertype,
#  2020 AS Year, # Add a "Year" column for all users for 2019.
#  TIMESTAMP_DIFF(ended_at, started_at, MINUTE) AS trip_duration # find the trip duration for ea
ch user

#FROM `project-1-461316.bicycle.2020`
```

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

Data Cleaning - advanced

Find all users and make sure each user has the same title of {“Subscriber”: “member”, “customer”: “causal”}

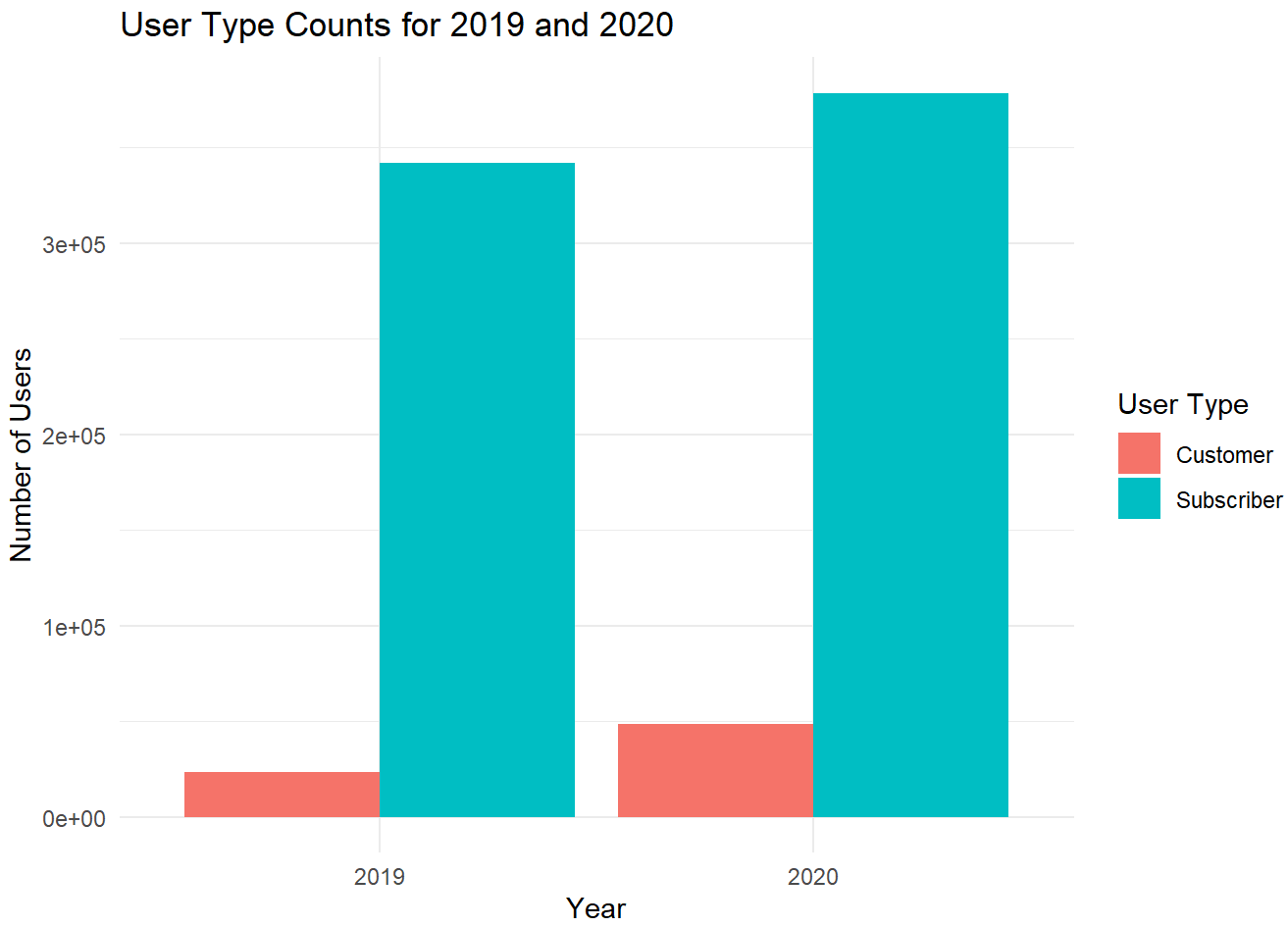
```
combined_df <- combined_df %>%
  mutate(usertype = case_when(
    usertype == "member" ~ "Subscriber",
    usertype == "casual" ~ "Customer",
    TRUE ~ usertype  # Keep other values unchanged
  ))
```

Data Analysis

Find the number of Subscribers and Customers for 2019 and 2020

```
# Summarize counts by usertype and year
summary_df <- combined_df %>%
  group_by(Year, usertype) %>%
  summarise(count = n(), .groups = "drop")

# Create the bar plot
ggplot(summary_df, aes(x = factor(Year), y = count, fill = usertype)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "User Type Counts for 2019 and 2020",
       x = "Year",
       y = "Number of Users",
       fill = "User Type") +
  theme_minimal()
```



Find the average trip duration for Customer and

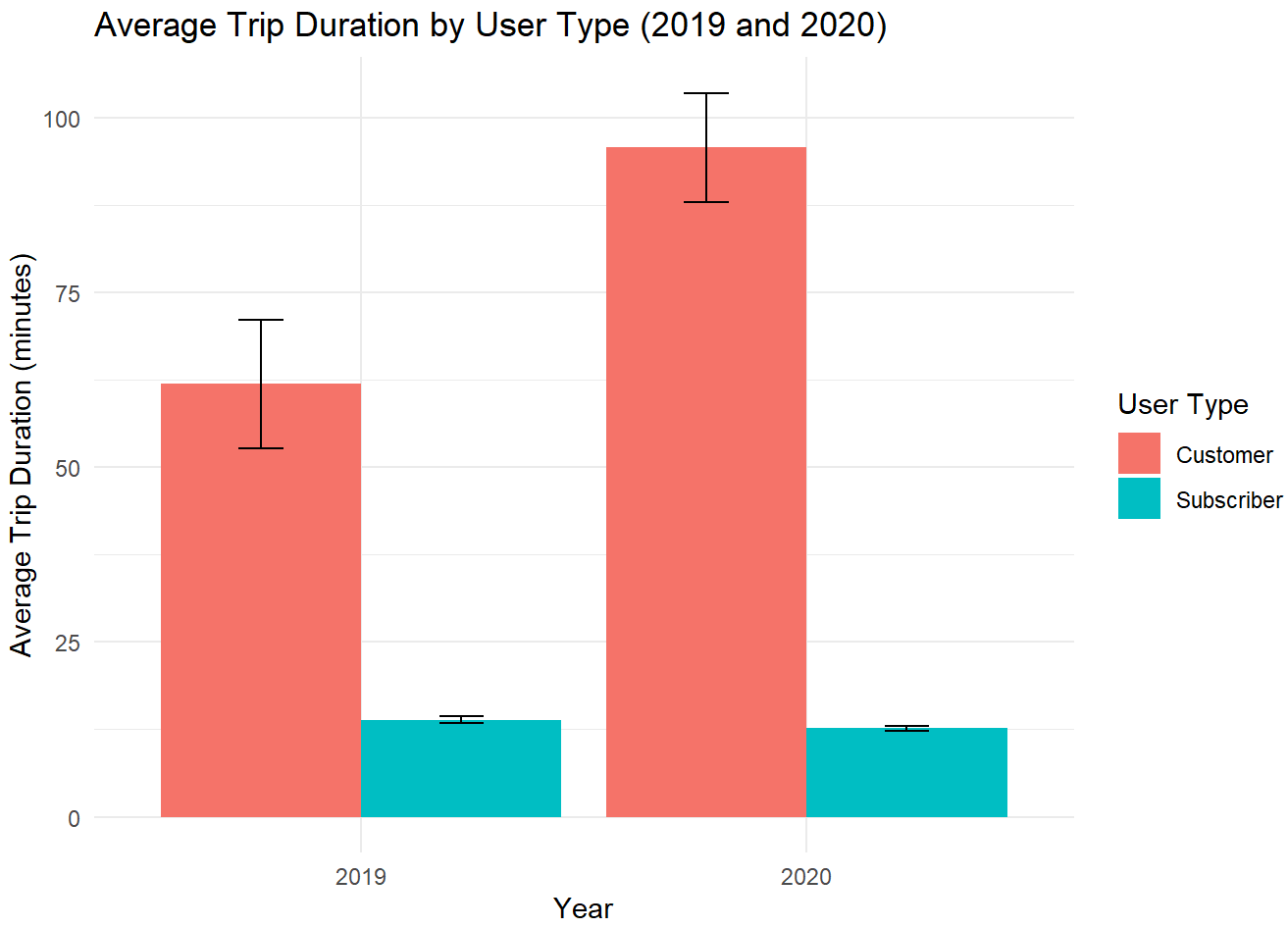
Subscriber

```
combined_df$start_time <- as.POSIXct(combined_df$start_time)
combined_df$end_time <- as.POSIXct(combined_df$end_time)

combined_df$trip_duration <- combined_df$end_time - combined_df$start_time
combined_df$trip_duration <- as.numeric(combined_df$trip_duration, units = "mins")
```

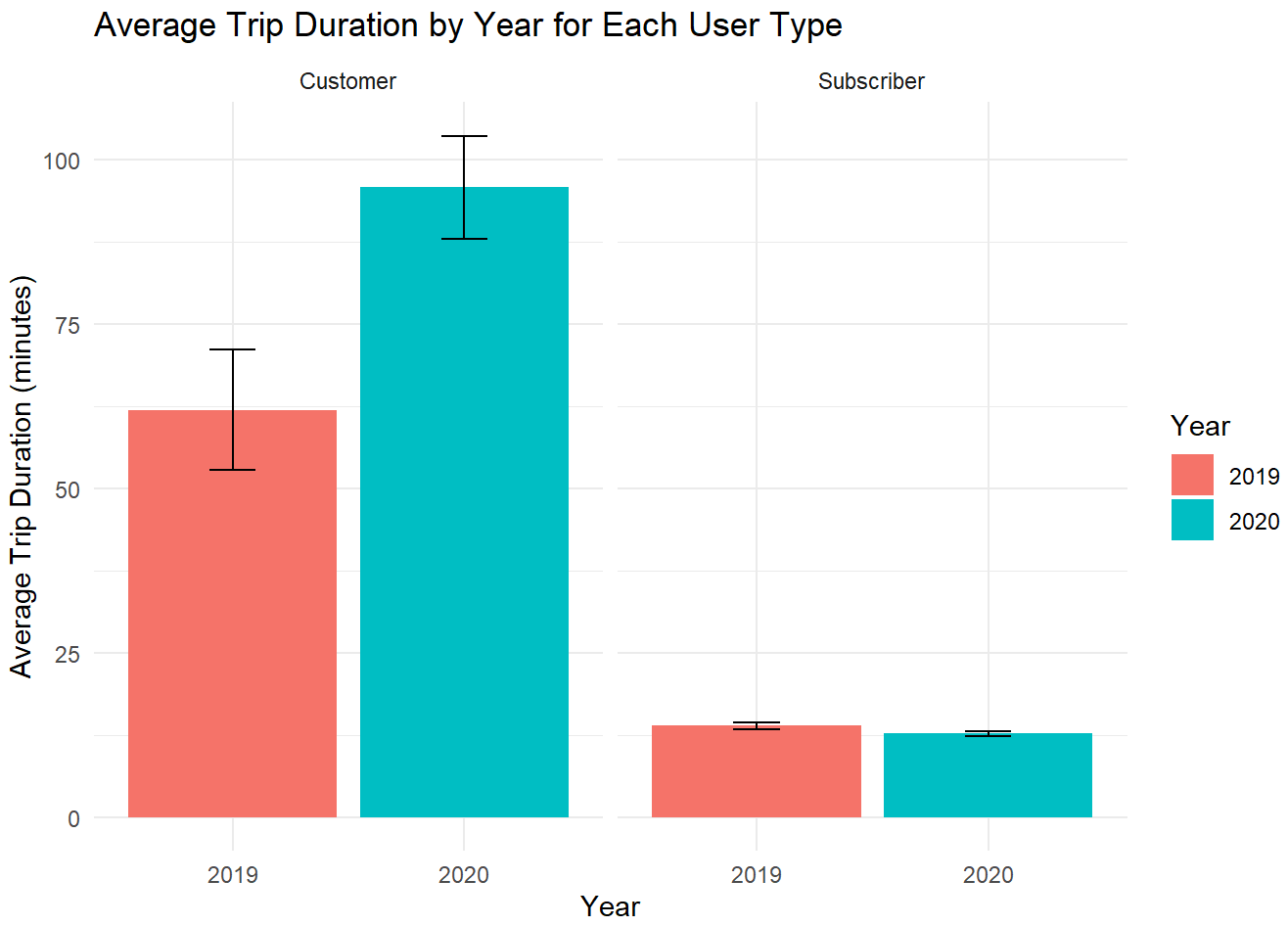
```
# Summarize the average trip duration by usertype and year
summary_df <- combined_df %>%
  group_by(Year, usertype) %>%
  summarise(
    average_duration = mean(trip_duration, na.rm = TRUE),
    sem = sd(trip_duration, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  )

ggplot(summary_df, aes(x = factor(Year), y = average_duration, fill = usertype)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  geom_errorbar(
    aes(ymin = average_duration - sem, ymax = average_duration + sem),
    position = position_dodge(width = 0.9),
    width = 0.2
  ) +
  labs(
    title = "Average Trip Duration by User Type (2019 and 2020)",
    x = "Year",
    y = "Average Trip Duration (minutes)",
    fill = "User Type"
  ) +
  theme_minimal()
```




```
# Calculate average and SEM
summary_df <- combined_df %>%
  group_by(Year, usertype) %>%
  summarise(
    average_duration = mean(trip_duration, na.rm = TRUE),
    sem = sd(trip_duration, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  )

# Plot with facet for usertype
ggplot(summary_df, aes(x = factor(Year), y = average_duration, fill = factor(Year))) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  geom_errorbar(
    aes(ymin = average_duration - sem, ymax = average_duration + sem),
    position = position_dodge(width = 0.9),
    width = 0.2
  ) +
  facet_wrap(~ usertype) +
  labs(
    title = "Average Trip Duration by Year for Each User Type",
    x = "Year",
    y = "Average Trip Duration (minutes)",
    fill = "Year"
  ) +
  theme_minimal()
```


Average Trip Duration by Year for Each User Type
Customer
Subscriber
Average Trip Duration (minutes)
100
75
50
25
0
2019
2020
2019
2020
Year
Year
2019
2020