Google Data Analytics: Case Study #1

S.Watson

1/23/2022

Contents

Introduction

This R Markdown document will be comprised of the analysis performed for Case Study #1 for the Google Data Analytics Professional Certificate. This document will provide an overview of the business case, a step-by-step walk through of data preparation, processing and analysis, a summary of findings, recommendations, and potential future steps.

Business Case

Cyclistic, a fictional bike-share company, based in Chicago, IL, has requested your services. The marketing director, Lily Moreno, believes the company's future success depends upon maximizing the number of annual memberships. The company would like to better understand the differences between how *casual* riders and annual *members* utilize Cyclistic's bikes in hopes of designing a new marketing strategy to convert *casual* riders into annual *member*. Note that customers who purchase "single-ride" or "full-day" passes are considered *casual* riders and customers who purchase annual memberships are considered *members*.

The goal of this analysis is to identify difference between casual riders and annual members.

Data Preparation

The section details the data preparation utilized for this analysis. The raw data files can be located here. The last twelve (12) months of data were utilized for this analysis (January 2021 through December 2021).

First, the necessary packages were installed and loaded for allow for data loading, cleaning, analysis and visualization.

```
# Install packages utilized for analysis if install is required - remove and
# run code

# install.packages('lubricate') install.packages('tidyverse')
# install.packages('skimr')

# Load packages required for data analysis

# Load tidyverse package for data import, cleaning, analysis, and visualization
library(tidyverse)
# Load lubricate package to handle date/time analysis
library(lubridate)
# Load hms library to handle time conversion (part of tidyverse package)
library(hms)
# Load skimr package for data analysis
library(skimr)
```

Please confirm working directory location to allow for data import.

```
getwd()
```

[1] "C:/Users/Stephanie/Documents/GradSchool/Coursera/Google_Data_Analytics/8-CapstoneProject/Week2/Case_1"

If the data files are not located in the working directory, then they can be moved to the working directory, mapped to the working directory when they are loaded, or the working directory can be set to their file path location at the beginning of the code chunk. The following code can be copied into the respective code chunk to change working directory to location where data files are stored.

```
# Change working directory (if required), uncomment below code and add
# applicable file path

# setwd()
```

Next, the individual data files were loaded into RStudio as dataframes.

Note that the following steps (loading data files, create new dataframes) will consume a fair amount of memory. Please be mindful of memory usage.

```
# Set directory to location of raw data files
setwd(paste("C:/Users/Stephanie/Documents/GradSchool", "/Coursera/Google_Data_Analytics",
    "/8-CapstoneProject/Week2/Case_1/Raw-Data/CSV_files/2021", sep = ""))

# Load in raw CSV files from 2021
Jan_2021_raw <- read_csv("202101-divvy-tripdata.csv")
Feb_2021_raw <- read_csv("202102-divvy-tripdata.csv")
Mar_2021_raw <- read_csv("202103-divvy-tripdata.csv")
```

```r
Apr_2021_raw <- read_csv("202104-divvy-tripdata.csv")
May_2021_raw <- read_csv("202105-divvy-tripdata.csv")
June_2021_raw <- read_csv("202106-divvy-tripdata.csv")
July_2021_raw <- read_csv("202107-divvy-tripdata.csv")
Aug_2021_raw <- read_csv("202108-divvy-tripdata.csv")
Sep_2021_raw <- read_csv("202109-divvy-tripdata.csv")
Oct_2021_raw <- read_csv("202110-divvy-tripdata.csv")
Nov_2021_raw <- read_csv("202111-divvy-tripdata.csv")
Dec_2021_raw <- read_csv("202112-divvy-tripdata.csv")
```

Lastly, the individual monthly dataframes were compiled into one dataframe, which represents all 2021 data. This one dataframe was saved in the event further analysis is required on the raw data. The individual monthly dataframes were removed to free up memory space.

```r
# Combine all monthly files into one yearly dataframe
Total_2021_trips_raw <- bind_rows(Jan_2021_raw, Feb_2021_raw, Mar_2021_raw, Apr_2021_raw,
    May_2021_raw, June_2021_raw, July_2021_raw, Aug_2021_raw, Sep_2021_raw, Oct_2021_raw,
    Nov_2021_raw, Dec_2021_raw)

# Write combined data to csv files (mapped from Week2/Case_1 directory)
write_csv(Total_2021_trips_raw, "Raw-Data/CSV_files/2021/Total_2021_Trips_raw.csv",
    col_names = TRUE)

# remove monthly data to free up memory space.
rm(Jan_2021_raw, Feb_2021_raw, Mar_2021_raw, Apr_2021_raw, May_2021_raw, June_2021_raw,
    July_2021_raw, Aug_2021_raw, Sep_2021_raw, Oct_2021_raw, Nov_2021_raw, Dec_2021_raw)
gc()
```

A view and summary of statistics on the raw data was performed.

```r
# Review combined data to ensure all data was merged correctly
kable(head(Total_2021_trips_raw), caption = "The First 6 row of raw dataframe")
```

Table 1: The First 6 row of raw dataframe

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
E19E6F1B8D4C42ED	electric_bike	2021-01-23 16:14:19	2021-01-23 16:24:44	California Ave & Cortez St	17660	NA	NA	41.90034	-87.69674	41.89	-87.72	member
DC88F20C2C55F27F	electric_bike	2021-01-27 18:43:08	2021-01-27 18:47:12	California Ave & Cortez St	17660	NA	NA	41.90033	-87.69671	41.90	-87.69	member
EC45C94683FE3F27	electric_bike	2021-01-21 22:35:54	2021-01-21 22:37:14	California Ave & Cortez St	17660	NA	NA	41.90031	-87.69664	41.90	-87.70	member
4FA453A75AE377DB	electric_bike	2021-01-07 13:31:13	2021-01-07 13:42:55	California Ave & Cortez St	17660	NA	NA	41.90040	-87.69666	41.92	-87.69	member
BE5E8EB4E7263A0B	electric_bike	2021-01-23 02:24:02	2021-01-23 02:24:45	California Ave & Cortez St	17660	NA	NA	41.90033	-87.69670	41.90	-87.70	casual

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
5D8969F88C773979	electric_bike	2021-01-09 14:24:07	2021-01-09 15:17:54	California Ave & Cortez St	17660	NA	NA	41.90041	-87.69676	41.94	-87.71	casual

```r
# Obtain combined data column names to be utilized in data processing
colnames(Total_2021_trips_raw)
```

```
##  [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"
```

```r
# Obtain summary statistics on combined raw data
str(Total_2021_trips_raw)
```

```
## spec_tbl_df [5,595,063 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5595063] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA4
##  $ rideable_type     : chr [1:5595063] "electric_bike" "electric_bike" "electric_bike" "electric_bike
##  $ started_at        : POSIXct[1:5595063], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:5595063], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:5595063] "California Ave & Cortez St" "California Ave & Cortez St" "Cal
##  $ start_station_id  : chr [1:5595063] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:5595063] NA NA NA NA ...
##  $ end_station_id    : chr [1:5595063] NA NA NA NA ...
##  $ start_lat         : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:5595063] "member" "member" "member" "member" ...
##  - 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>
```

In total, there are 5,595,063 rows and 13 columns in the combined 2021 raw dataframe. The column names and data type are summarized below.

Column Name	Data Type	Column Description
ride_id	character	Unique ride ID
rideable_type	character	Type of bike utilized for ride
started_at	datetime	Date/time ride was started (S3: POSIXct)
ended_at	datetime	Date/time ride was ended (S3: POSIXct)
start_station_name	character	Name of ride start station
start_station_id	character	Unique ID for ride start station
end_station_name	character	Name of ride end station
end_station_id	character	Unique ID for ride end station
start_lat	numeric	Latitude of start station
start_lng	numeric	Longitude of start station
end_lat	numeric	Latitude of end station
end_lng	numeric	Longitude of end station
member_casual	character	Type of rider

Data Processing

This section details the data processing for this analysis. The raw data mentioned above will be cleaned in preparation for data analysis. Data not required for this analysis were removed, new columns were created for ride length, weekday, month, and year, and the data was sorted based on the ride start date/time. The raw 2021 dataframe was removed to free up memory space.

```
# Utilize select statement to remove unnecessary rows (start/end station,
# ride_id and start/end lat/lng), add new columns for ride_length, day_of_week,
# year, and month and sort data by start date/time in ascending order

Total_2021_trips_clean <- Total_2021_trips_raw %>%
    mutate(ride_length = int_length(interval(ymd_hms(started_at), ymd_hms(ended_at))),
        day_of_week = wday(ymd_hms(started_at), label = TRUE, abbr = FALSE), month_ = month(ymd_hms(sta
            label = TRUE, abbr = FALSE), year_ = year(ymd_hms(started_at))) %>%
    arrange(started_at) %>%
    select(rideable_type, started_at, ended_at, member_casual, ride_length, day_of_week,
        month_, year_)

# The raw 2021 dataframe was removed to free up memory space.
rm(Total_2021_trips_raw)
gc()
```

A view and summary of statistics on the clean data was performed.

```
kable(head(Total_2021_trips_clean), caption = "The First 6 row of new dataframe")
```

Table 3: The First 6 row of new dataframe

rideable_type	started_at	ended_at	member_casual	ride_length	day_of_week	month_	year_
electric_bike	2021-01-01 00:02:05	2021-01-01 00:12:39	member	634	Friday	January	2021
classic_bike	2021-01-01 00:02:24	2021-01-01 00:08:39	member	375	Friday	January	2021
classic_bike	2021-01-01 00:06:55	2021-01-01 00:26:36	member	1181	Friday	January	2021

rideable_type	started_at	ended_at	member_casual	ride_length	day_of_week	month_	year_
electric_bike	2021-01-01 00:12:13	2021-01-01 00:20:06	member	473	Friday	January	2021
classic_bike	2021-01-01 00:12:21	2021-01-01 00:12:33	member	12	Friday	January	2021
classic_bike	2021-01-01 00:12:27	2021-01-01 00:12:30	casual	3	Friday	January	2021

```
str(Total_2021_trips_clean)
```

```
## tibble [5,595,063 x 8] (S3: tbl_df/tbl/data.frame)
##  $ rideable_type: chr [1:5595063] "electric_bike" "classic_bike" "classic_bike" "electric_bike" ...
##  $ started_at   : POSIXct[1:5595063], format: "2021-01-01 00:02:05" "2021-01-01 00:02:24" ...
##  $ ended_at     : POSIXct[1:5595063], format: "2021-01-01 00:12:39" "2021-01-01 00:08:39" ...
##  $ member_casual: chr [1:5595063] "member" "member" "member" "member" ...
##  $ ride_length  : num [1:5595063] 634 375 1181 473 12 ...
##  $ day_of_week  : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ month_       : Ord.factor w/ 12 levels "January"<"February"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year_        : num [1:5595063] 2021 2021 2021 2021 2021 ...
```

```
skim_without_charts(Total_2021_trips_clean)
```

Table 4: Data summary

Name	Total_2021_trips_clean
Number of rows	5595063
Number of columns	8
Column type frequency:	
character	2
factor	2
numeric	2
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
rideable_type	0	1	11	13	0	3	0
member_casual	0	1	6	6	0	2	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
day_of_week	0	1	TRUE	7	Sat: 991047, Sun: 857285, Fri: 810508, Wed: 756142
month_	0	1	TRUE	12	Jul: 822410, Aug: 804352, Sep: 756147, Jun: 729595

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ride_length	0	1	1316.12	10700.09	-3482	405	720	1307	3356649
year_	0	1	2021.00	0.00	2021	2021	2021	2021	2021

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2021-01-01 00:02:05	2021-12-31 23:59:48	2021-08-01 01:52:11	4677998
ended_at	0	1	2021-01-01 00:08:39	2022-01-03 17:32:18	2021-08-01 02:21:55	4671372

Based on the statistics summary, there are ride times with values less than 0 seconds (negative times) and greater than 1 day (86400 seconds).

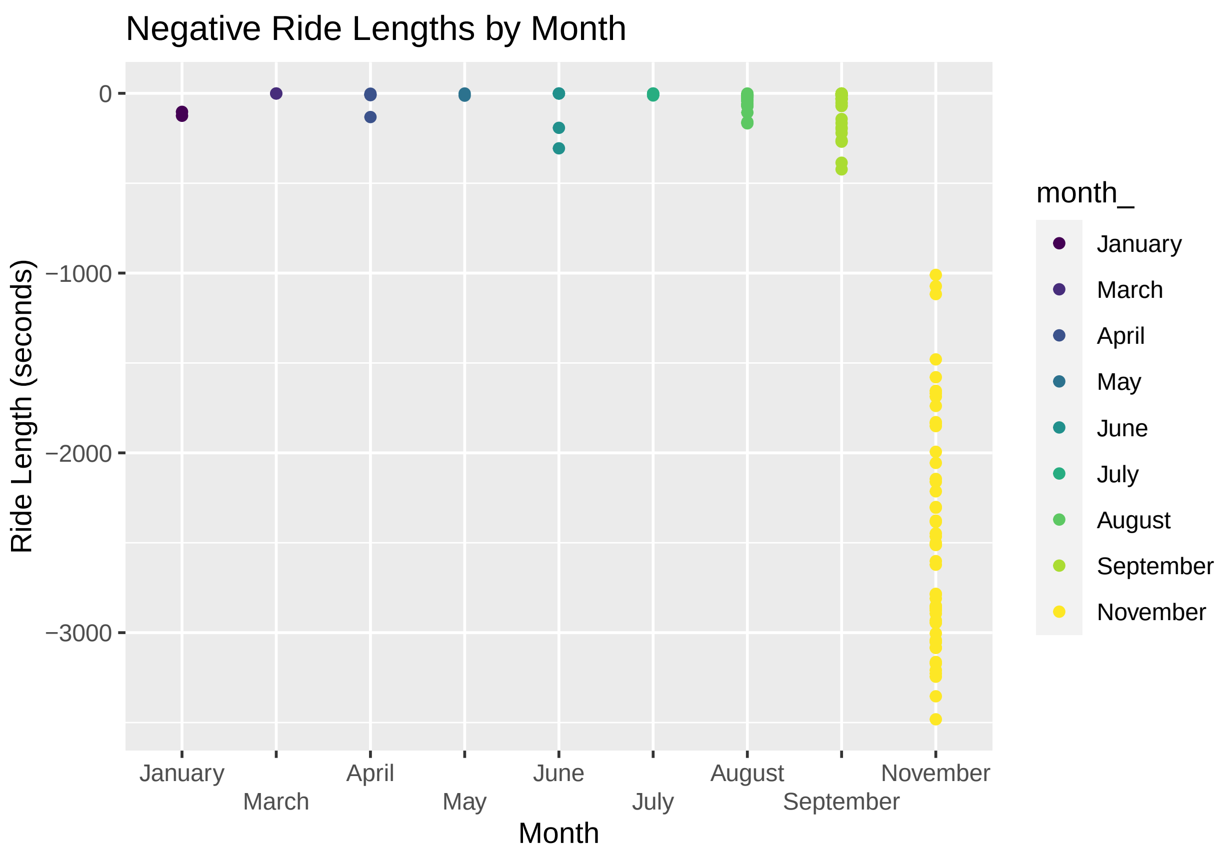
The dataframe was queried for the negative values.

```
count(Total_2021_trips_clean[which(Total_2021_trips_clean$ride_length < 0), ])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   147
```

There are 147 rows with negative ride times. The below chart displays the number of negative ride lengths per month.

```
ggplot(Total_2021_trips_clean[which(Total_2021_trips_clean$ride_length < 0), ]) +
    geom_point(aes(x = month_, y = ride_length, color = month_)) + labs(x = "Month",
    y = "Ride Length (seconds)", title = "Negative Ride Lengths by Month") + guides(x = guide_axis(n.do
```




```r
# Count the number of negative ride lengths in the month of November
count(Total_2021_trips_clean[which(Total_2021_trips_clean$ride_length < 0 & Total_2021_trips_clean$month
    "November"), ])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    53
```

It appears that nine (9) months have at least one ride length that is negative, with most being a few seconds in length. The month of November has a significant number of negative ride lengths (53) with values significantly larger than previous months. The cause for these anomalies should be investigated.

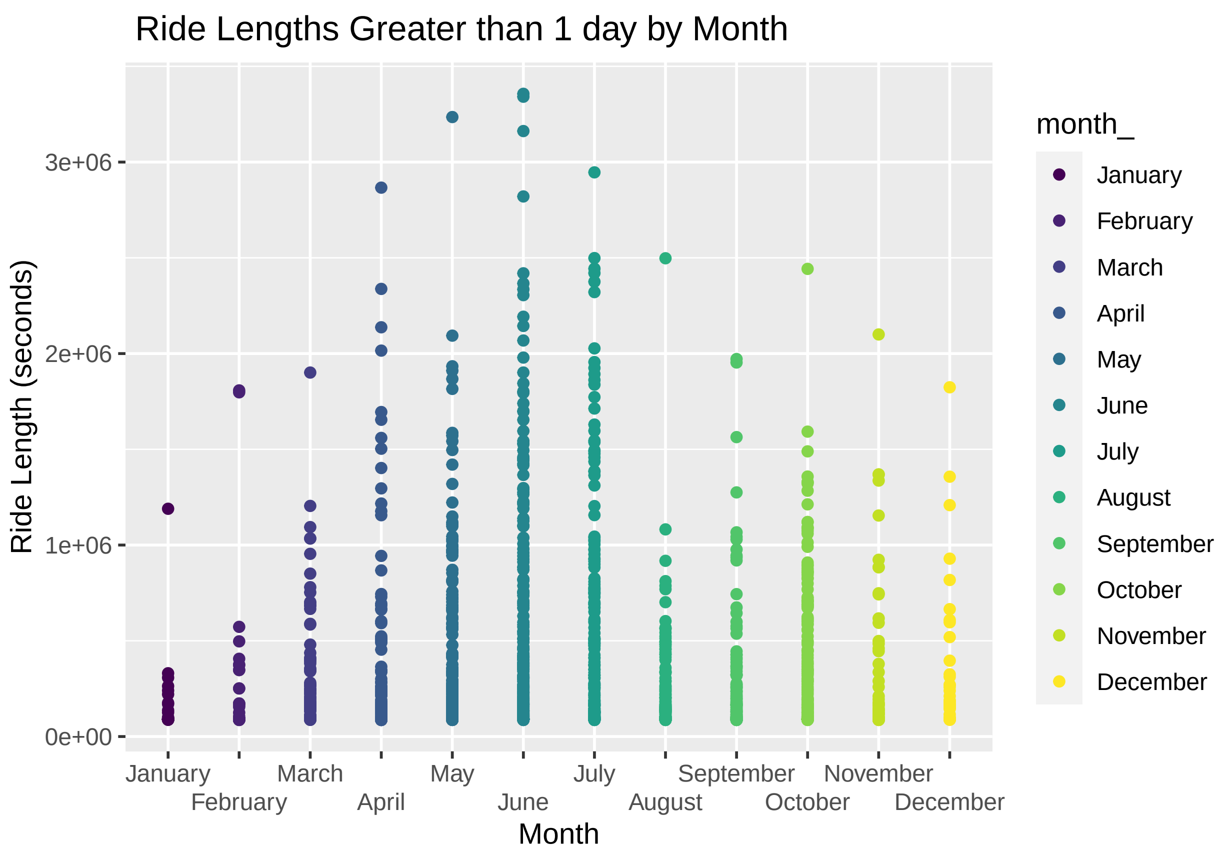
Next, the dataframe was queried for the ride lengths which are greater than 1 day (84600 seconds).

```r
# Count number of ride length that are greater than 84600 seconds (1 day)
count(Total_2021_trips_clean[which(Total_2021_trips_clean$ride_length > 86400), ])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  4016
```

There are 4,016 rows with ride times greater than 1 day. The below chart displays the number of ride lengths that exceed 1 day per month.

```r
# scatter plot of ride lengths greater than 1 day per month
ggplot(Total_2021_trips_clean[which(Total_2021_trips_clean$ride_length > 86400),
    ]) + geom_point(aes(x = month_, y = ride_length, color = month_)) + labs(x = "Month",
    y = "Ride Length (seconds)", title = " Ride Lengths Greater than 1 day by Month") +
    guides(x = guide_axis(n.dodge = 2))
```



Based on this graph, it appears that each month has ride lengths greater than 1 day. The higher numbers in spring/summer is consistent with the increased number of rides during this time period.

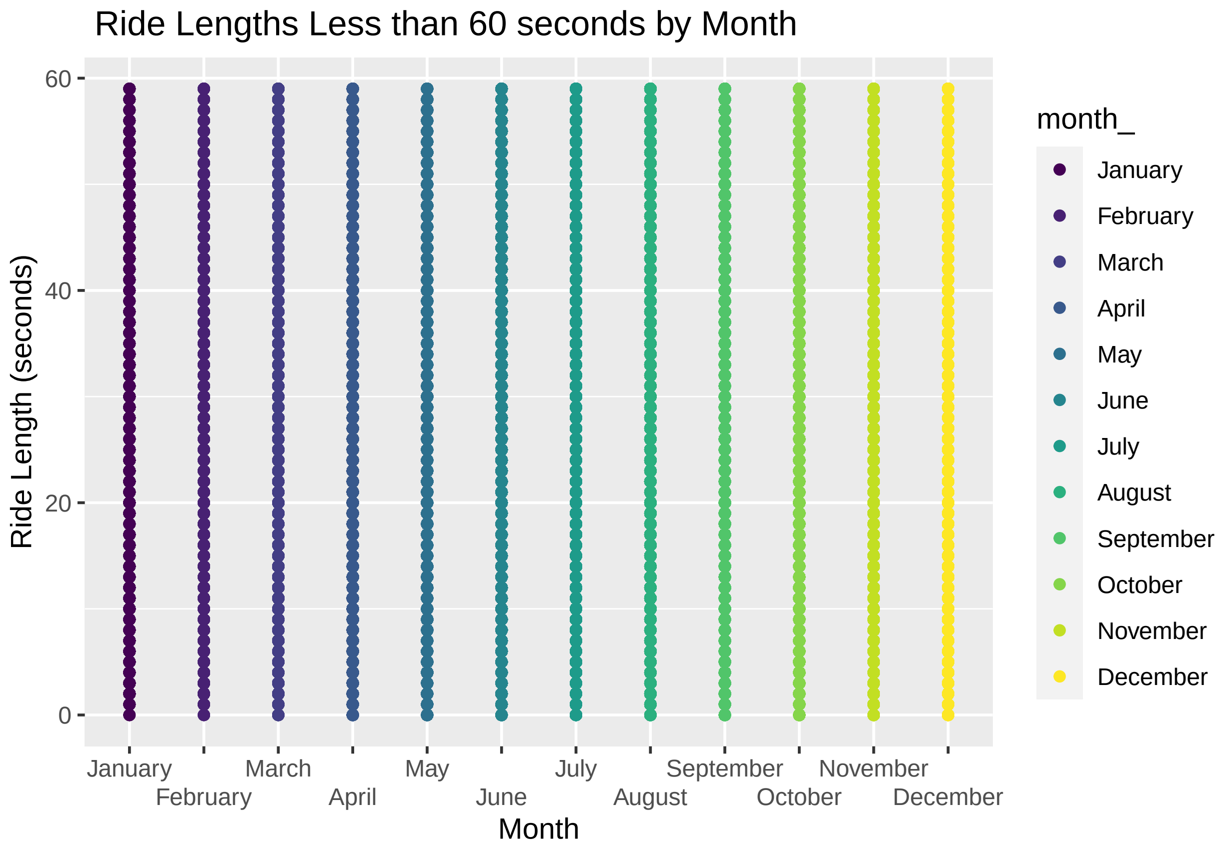
Also, based on the information provided on Divvy website (located here), ride lengths less than 60 seconds were removed as these trips could be *'potentially false starts or users trying to redock a bike'*.

```r
# count number of ride lengths that a positive and less than 60 seconds
count(Total_2021_trips_clean[which((Total_2021_trips_clean$ride_length < 60) & (Total_2021_trips_clean$r
    0)), ])
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 85086
```

There are 85,086 rows with positive ride times that are less than 60 seconds. The below chart displays the number of positive ride lengths that are less than 60 seconds day per month.

```
# scatter plot of ride lengths that a positive and less than 60 seconds
ggplot(Total_2021_trips_clean[which((Total_2021_trips_clean$ride_length < 60) & (Total_2021_trips_clean$
    0)), ]) + geom_point(aes(x = month_, y = ride_length, color = month_)) + labs(x = "Month",
    y = "Ride Length (seconds)", title = " Ride Lengths Less than 60 seconds by Month") +
    guides(x = guide_axis(n.dodge = 2))
```



Based on this graph, it appears that each month has positive ride lengths that are less than 60 seconds.

The ride lengths that are negative, less than 60 seconds, or greater than 1 day (89,249 samples out of over 5 million or ~1.6%), will be removed from the dataframe. In addition, rides with "docked_bike" were removed as this category only captures how long a bike stayed at a station.This cleaned data was stored in a new dataframe

```
Total_2021_trips_clean <- subset(Total_2021_trips_clean, (!((Total_2021_trips_clean$ride_length <
    60) | (Total_2021_trips_clean$ride_length > 86400)) & !(Total_2021_trips_clean$rideable_type ==
    "docked_bike")))

# Write cleaned data to csv file (mapped from Week2/Case_1 directory)
write_csv(Total_2021_trips_clean, "CleanData/Total_2021_Trips_clean.csv", col_names = TRUE)

skim_without_charts(Total_2021_trips_clean)
```

Table 9: Data summary

Name	Total_2021_trips_clean

Table 9: Data summary

Number of rows	5196779
Number of columns	8
Column type frequency:	
character	2
factor	2
numeric	2
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
rideable_type	0	1	12	13	0	2	0
member_casual	0	1	6	6	0	2	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
day_of_week	0	1	TRUE	7	Sat: 897107, Sun: 773205, Fri: 755911, Wed: 717060
month_	0	1	TRUE	12	Jul: 751989, Aug: 747712, Sep: 709750, Jun: 665909

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ride_length	0	1	1076.71	1794.3	60	404	700	1236	86397
year_	0	1	2021.00	0.0	2021	2021	2021	2021	2021

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2021-01-01 00:02:05	2021-12-31 23:59:48	2021-08-02 18:07:34	4403286
ended_at	0	1	2021-01-01 00:08:39	2022-01-01 03:59:48	2021-08-02 18:26:36	4398654

In total, there are 5,196,779 rows and 8 columns in the combined 2021 clean dataframe. The column names and data type are summarized below.

Column Name	Data Type	Column Description
rideable_type	character	Type of bike utilized for ride
started_at	datetime	Date/time ride was started (S3: POSIXct)
ended_at	datetime	Date/time ride was ended (S3: POSIXct)

Column Name	Data Type	Column Description
member_casual	character	Type of rider
ride_length	numeric	Ride length in seconds
day_of_week	ordinal	Weekday extracted from start date/time
month_	ordinal	Month extracted from start date/time
year_	numeric	Year extracted from start date/time

Data Analysis

This section details the data analysis of the clean data. The goal of this analysis is to identify differences in behavior between casual riders and annual members. The first subsection performs descriptive analytics (i.e. max, mean, median, etc) and the second subsection plots the cleaned data.

Descriptive Analysis

In this subsection, descriptive analytics is performed on the cleaned 2021 data in an attempt to understand the difference behaviors of casual riders and members.

First, the maximum, average, and median ride lengths were calculated based on member type for each bike type.

```
# Summarize data, calculate max, mean, and median ride length (in hh:mm:ss) by
# member type and bike type
kable(Total_2021_trips_clean %>%
    group_by(member_casual, rideable_type) %>%
    drop_na() %>%
    summarize(max_ride_length = hms(max(ride_length)), mean_ride_length = round_hms(hms(mean(ride_lengt
        2), median_ride_length = hms(median(ride_length))), caption = "2021 Max, Average, Median Ride L
```

Table 15: 2021 Max, Average, Median Ride Length by Member Type & Bike Type

member_casual	rideable_type	max_ride_length	mean_ride_length	median_ride_length
casual	classic_bike	23:59:55	00:26:40	00:16:14
casual	electric_bike	08:07:16	00:20:14	00:13:25
member	classic_bike	23:59:57	00:13:58	00:10:08
member	electric_bike	08:00:31	00:12:58	00:09:07

Based on this information, it appears that the *casual* rider spends more time on each bike type category.

Secondly, the maximum, average, and median ride lengths were calculated based on member type for day of the week.

```
# Summarize data, mean and max ride length (in hh:mm:ss) per day by member type
kable(Total_2021_trips_clean %>%
    group_by(member_casual, day_of_week) %>%
    drop_na() %>%
    summarize(max_ride_length = hms(max(ride_length)), mean_ride_length = round_hms(hms(mean(ride_lengt
        2), median_ride_length = hms(median(ride_length))), caption = "2021 Max, Average, Median Ride L
```

Table 16: 2021 Max, Average, Median Ride Length by Member Type & Day of Week

member_casual	day_of_week	max_ride_length	mean_ride_length	median_ride_length
casual	Sunday	23:53:49	00:27:38	00:17:26.0
casual	Monday	23:58:33	00:24:04	00:14:48.5
casual	Tuesday	23:55:45	00:21:34	00:13:28.0
casual	Wednesday	23:57:27	00:21:00	00:13:17.0
casual	Thursday	23:55:21	00:20:54	00:13:09.0
casual	Friday	23:53:23	00:22:22	00:14:09.0
casual	Saturday	23:59:55	00:26:06	00:16:43.0
member	Sunday	23:48:05	00:15:34	00:11:04.0
member	Monday	23:24:50	00:13:12	00:09:21.0
member	Tuesday	21:15:45	00:12:48	00:09:16.0
member	Wednesday	23:35:08	00:12:52	00:09:22.0
member	Thursday	23:25:55	00:12:46	00:09:16.0
member	Friday	23:46:34	00:13:18	00:09:36.0
member	Saturday	23:59:57	00:15:14	00:11:01.0

Based on this information, it appears that the *casual* rider spend on average more time utilizing the bikes than *member* customer. In addition, the top two (2) days for both the *casual* and *member* customers with respect to the highest average and median ride lengths are Sunday and Saturday respectively.

Next, the maximum, average, and median ride lengths were calculated based on member type for each month.

```
# Summarize data, mean,median, max ride length (in hh:mm:ss) per day by member
# type
kable(Total_2021_trips_clean %>%
    group_by(member_casual, month_) %>%
    drop_na() %>%
    summarize(max_ride_length = hms(max(ride_length)), mean_ride_length = round_hms(hms(mean(ride_lengt
        2), median_ride_length = hms(median(ride_length))), caption = "2021 Max, Average, Median Ride L
```

Table 17: 2021 Max, Average, Median Ride Length by Member Type & Month

member_casual	month_	max_ride_length	mean_ride_length	median_ride_length
casual	January	22:27:00	00:18:54	00:11:44
casual	February	23:47:38	00:27:30	00:15:07
casual	March	23:46:13	00:27:08	00:16:39
casual	April	23:58:33	00:26:40	00:16:02
casual	May	23:57:54	00:27:26	00:17:08
casual	June	23:55:45	00:25:52	00:16:15
casual	July	23:26:15	00:24:38	00:15:41
casual	August	23:52:05	00:23:58	00:15:20
casual	September	23:59:22	00:23:06	00:14:42
casual	October	23:53:49	00:20:46	00:13:07
casual	November	23:59:55	00:17:20	00:10:54
casual	December	23:15:40	00:16:42	00:10:35
member	January	23:24:50	00:12:52	00:08:50
member	February	22:41:33	00:16:08	00:10:20
member	March	23:38:48	00:14:04	00:10:09

member_casual	month_	max_ride_length	mean_ride_length	median_ride_length
member	April	23:25:55	00:14:44	00:10:34
member	May	22:44:53	00:14:42	00:10:40
member	June	23:53:44	00:14:40	00:10:48
member	July	23:22:34	00:14:18	00:10:36
member	August	22:27:41	00:14:06	00:10:19
member	September	22:31:43	00:13:44	00:09:57
member	October	23:35:08	00:12:26	00:08:48
member	November	23:59:57	00:11:18	00:07:48
member	December	20:30:52	00:11:00	00:07:44

Based on this information, *casual* rider average ride length peaks in early in the year (February through May).

Next, the total number of rides per month by member type was calculated.

```
# Summarize data, count number of rides per month by member type in descending
# order
ride_count_tbl <- Total_2021_trips_clean %>%
    group_by(member_casual, month_) %>%
    drop_na() %>%
    summarize(ride_count = n()) %>%
    arrange(desc(ride_count))

kable(ride_count_tbl, caption = "2021 Monthly Ride Count by Member Type Desc.")
```

Table 18: 2021 Monthly Ride Count by Member Type Desc.

member_casual	month_	ride_count
member	September	385902
member	August	385365
casual	July	378202
member	July	373787
member	October	367394
casual	August	362347
member	June	352612
casual	September	323848
casual	June	313297
member	May	269863
member	November	248663
casual	October	230864
casual	May	210095
member	April	197453
member	December	174856
member	March	142365
casual	April	110251
casual	November	97721
member	January	77562
casual	March	67513
casual	December	63781
member	February	38626
casual	January	15744

member_casual	month_	ride_count
casual	February	8668

```r
# Summarize data, average monthly rider by member type

kable(aggregate(ride_count ~ member_casual, ride_count_tbl, mean), caption = "2021 Average Ride Count by
```

Table 19: 2021 Average Ride Count by Member Type

member_casual	ride_count
casual	181860.9
member	251204.0

Based on this information, both *casual* customer and *member* customer demand peaks in 3Q (July, August, September). Both customer bases exceed their yearly average in May and drop below their averages in November.

Next, the ride count per month by member count and bike type was calculated.

```r
kable(count(Total_2021_trips_clean, member_casual, rideable_type, month_, member_casual,
    sort = TRUE), caption = "2021 Ride Count by Member Type, Bike Type, and Month Desc.")
```

Table 20: 2021 Ride Count by Member Type, Bike Type, and Month Desc.

member_casual	rideable_type	month_	n
member	classic_bike	August	269015
member	classic_bike	September	263028
member	classic_bike	July	261154
member	classic_bike	June	242933
casual	classic_bike	July	238365
casual	classic_bike	August	227270
member	classic_bike	October	207510
casual	classic_bike	September	193319
casual	classic_bike	June	185567
member	classic_bike	May	182290
member	electric_bike	October	159884
member	classic_bike	April	141773
casual	electric_bike	July	139837
casual	electric_bike	August	135077
casual	electric_bike	September	130529
member	electric_bike	November	128116
casual	electric_bike	June	127730
casual	electric_bike	October	126427
member	electric_bike	September	122874
casual	classic_bike	May	122388
member	classic_bike	November	120547
member	electric_bike	August	116350
member	electric_bike	July	112633
member	electric_bike	June	109679

member_casual	rideable_type	month_	n
member	classic_bike	March	105602
casual	classic_bike	October	104437
member	electric_bike	December	95105
casual	electric_bike	May	87707
member	electric_bike	May	87573
member	classic_bike	December	79751
casual	classic_bike	April	69938
casual	electric_bike	November	66223
member	electric_bike	April	55680
member	classic_bike	January	52835
casual	classic_bike	March	45088
casual	electric_bike	December	44203
casual	electric_bike	April	40313
member	electric_bike	March	36763
casual	classic_bike	November	31498
member	classic_bike	February	28746
member	electric_bike	January	24727
casual	electric_bike	March	22425
casual	classic_bike	December	19578
member	electric_bike	February	9880
casual	classic_bike	January	8188
casual	electric_bike	January	7556
casual	classic_bike	February	5588
casual	electric_bike	February	3080

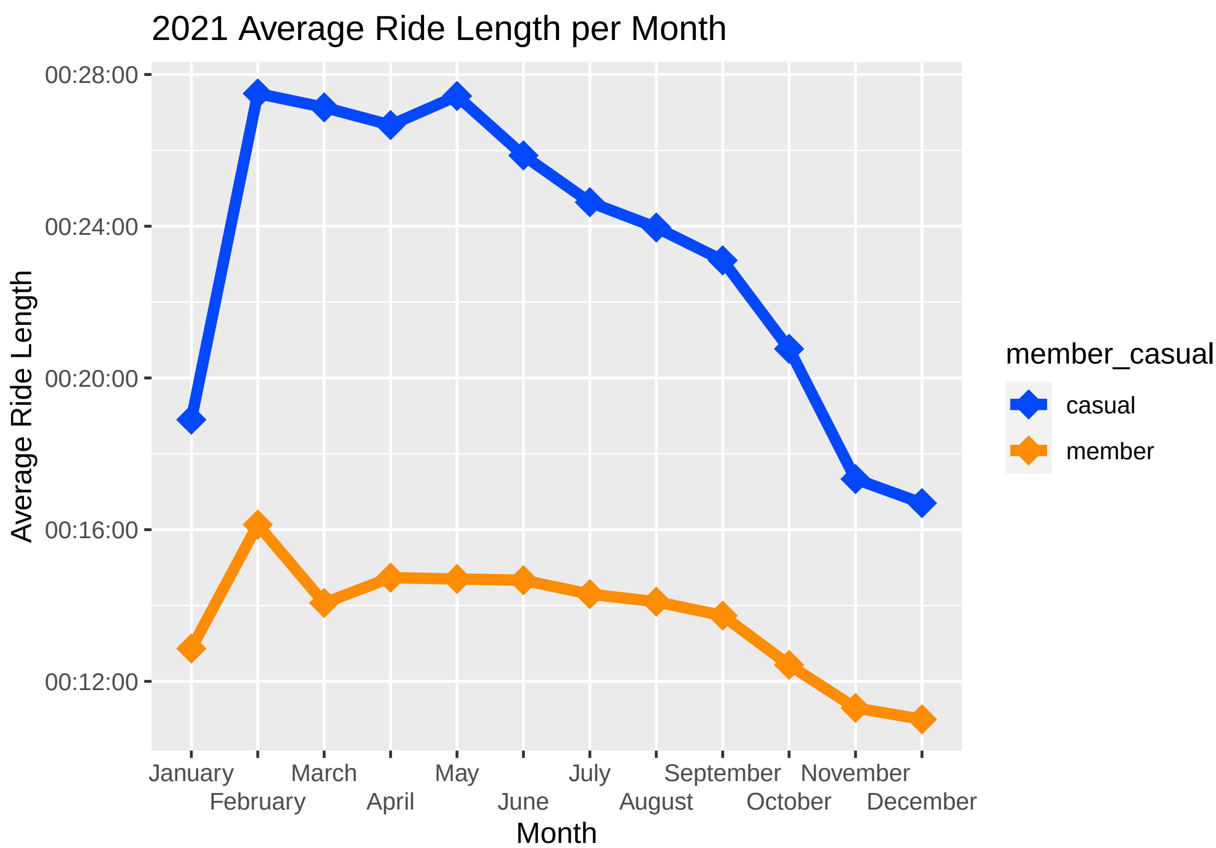
This information confirms that *casual* customer and *member* customer demand peaks in 3Q (July, August, September). This also indicates that the *classic* bike type appears to be the most popular among *casual* riders and *member* customers.

Data Visualization

In this subsection, data visualization is performed on the cleaned 2021 data in an attempt to understand the difference behaviors of casual riders and members.

First, the average ride length per month by member type was plotted.

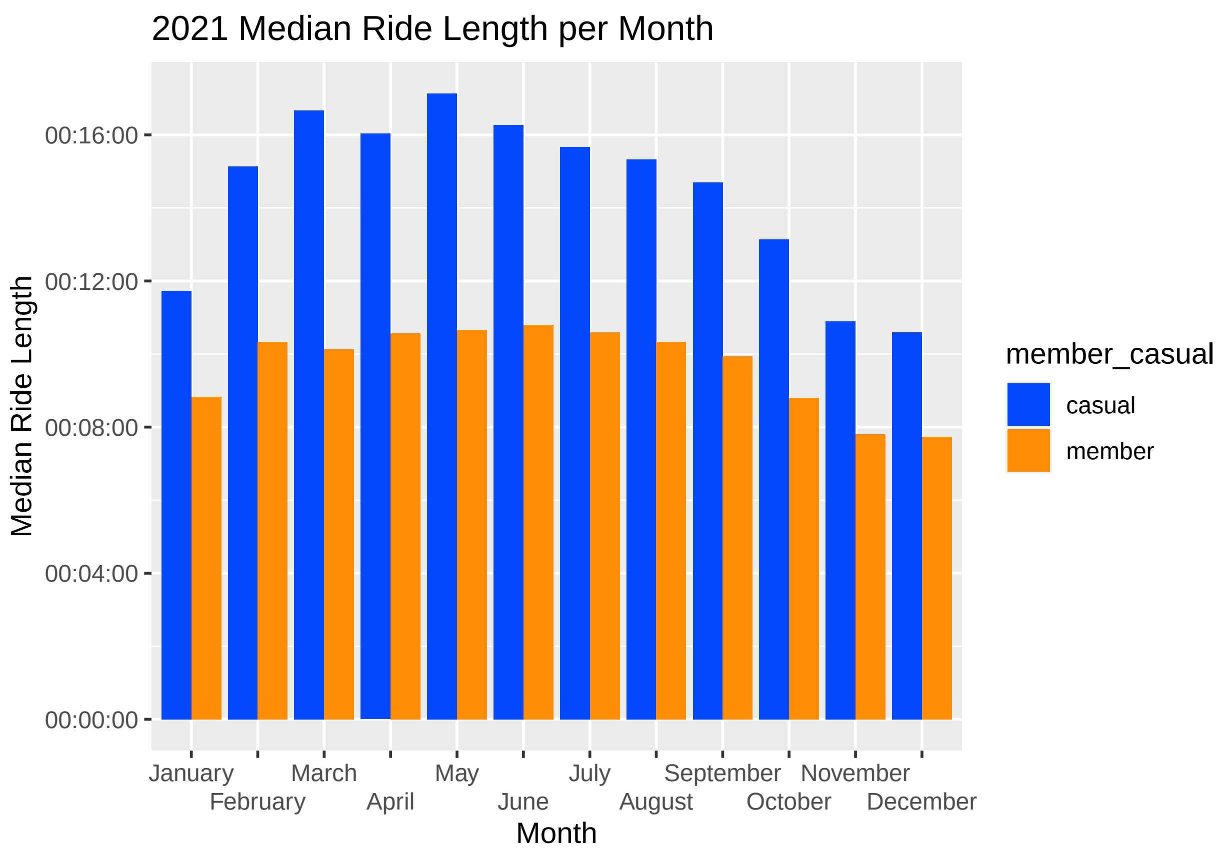
```
# Create line chart to display monthly average ride length per member type
Total_2021_trips_clean %>%
    group_by(member_casual, month_) %>%
    drop_na() %>%
    summarize(mean_ride_length = round_hms(hms(mean(ride_length)), 2)) %>%
    ggplot(aes(x = month_, y = mean_ride_length, group = member_casual, colour = member_casual)) +
    geom_line(size = 2) + geom_point(shape = "diamond", size = 5) + scale_color_manual(values = c("#0448
    "#FE8C04")) + labs(x = "Month", y = "Average Ride Length", title = "2021 Average Ride Length per Mon
    guides(x = guide_axis(n.dodge = 2))
```



As expected, the average ride length for *casual* customers is higher than that of *member* customers.

The median ride length was also plotted.

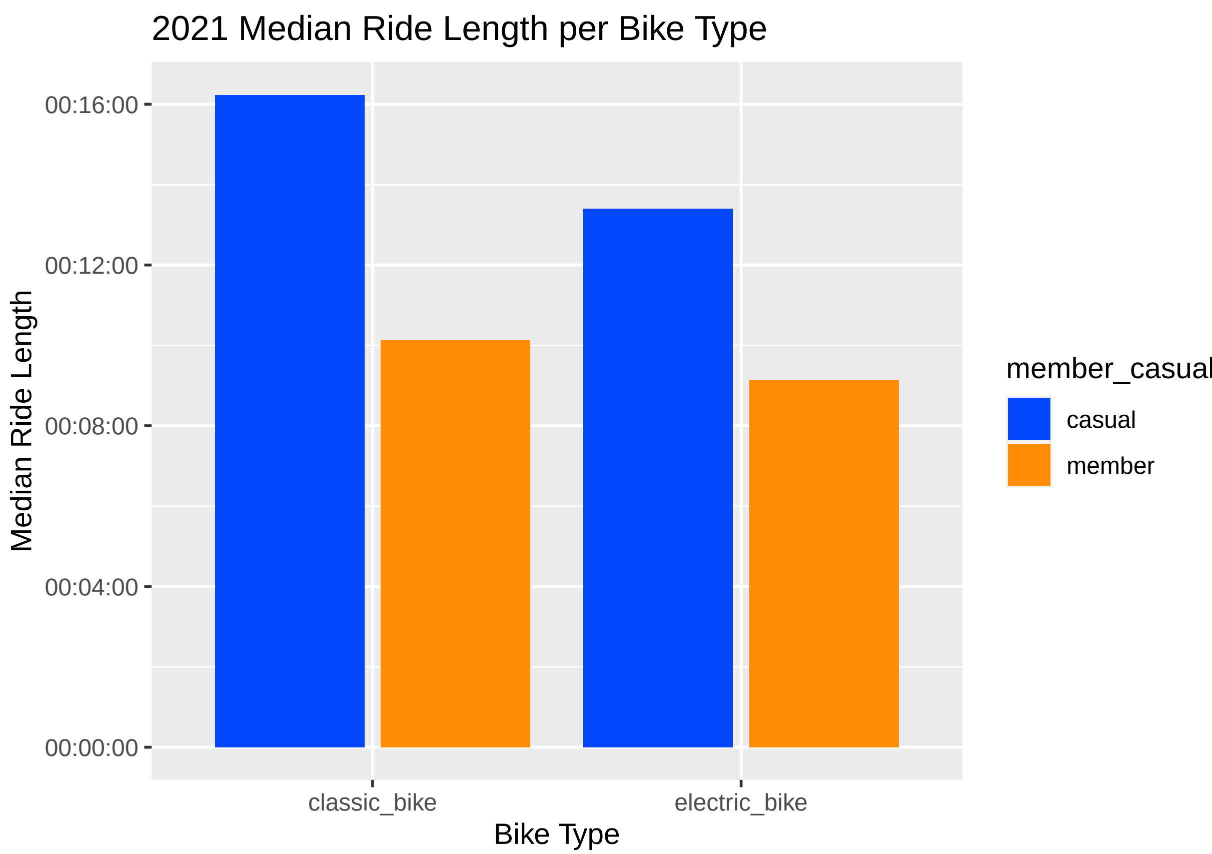
```
# Create column chart to display monthly average ride length per member type
Total_2021_trips_clean %>%
    group_by(member_casual, month_) %>%
    drop_na() %>%
    summarize(median_ride_length = round_hms(hms(median(ride_length)), 2)) %>%
    ggplot(aes(x = month_, y = median_ride_length, fill = member_casual)) + geom_col(position = "dodge")
    scale_fill_manual(values = c("#0448FE", "#FE8C04")) + labs(x = "Month", y = "Median Ride Length",
    title = "2021 Median Ride Length per Month") + guides(x = guide_axis(n.dodge = 2))
```



This plot shows that the *casual* customers still have a 'typical' ride length that is greater than a *member* customer however the time length is not as drastic (monthly median within ~ 5 minutes).

Secondly, the average ride length by bike type was plotted by member type.

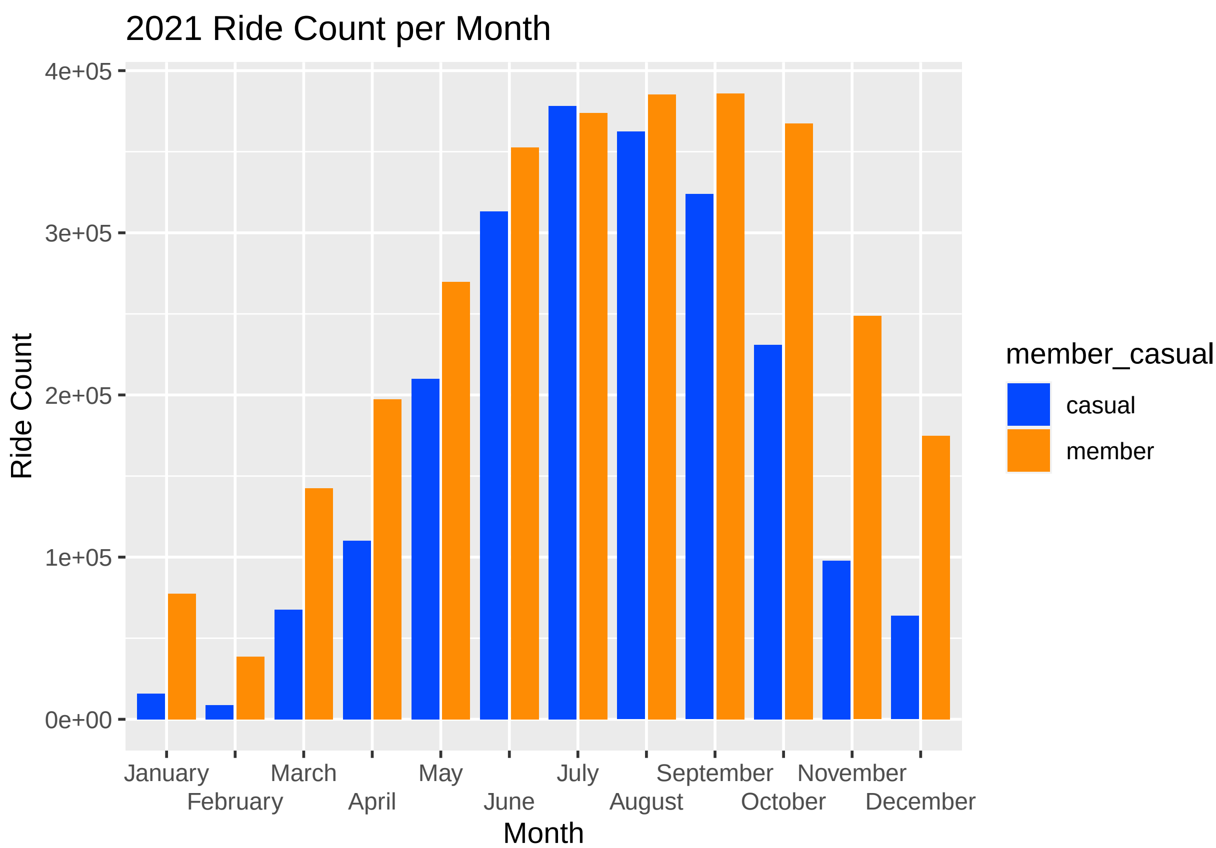
```
# Create column chart to display monthly ride count by member type
Total_2021_trips_clean %>%
    group_by(member_casual, rideable_type) %>%
    drop_na() %>%
    summarize(mean_ride_length = round_hms(hms(median(ride_length)), 2)) %>%
    ggplot() + geom_col(aes(x = rideable_type, y = mean_ride_length, fill = member_casual),
    position = "dodge2") + scale_fill_manual(values = c("#0448FE", "#FE8C04")) +
    labs(x = "Bike Type", y = "Median Ride Length", title = "2021 Median Ride Length per Bike Type")
```



As expected, *casual* customers spend more time on each bike type offered.

Next, the monthly ride count by member type was plotted.

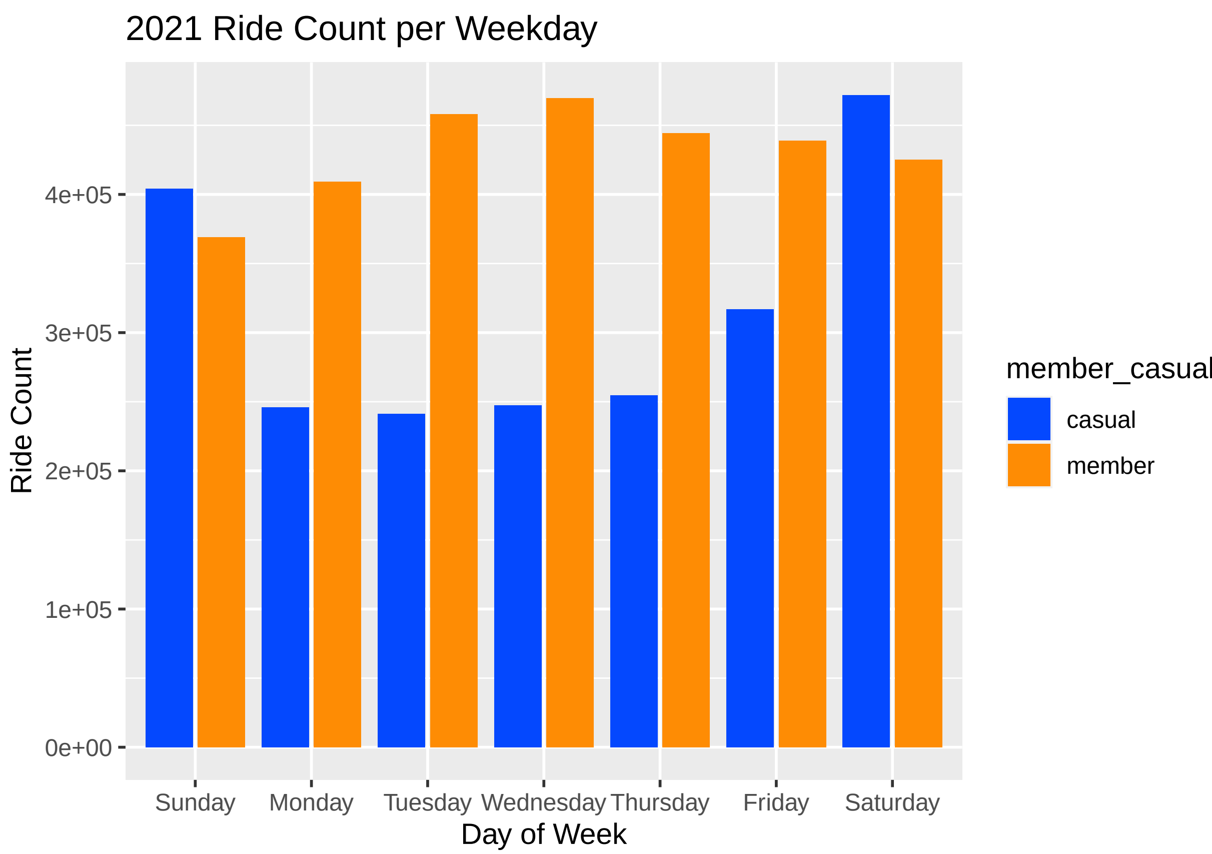
```
# Create bar chart to display monthly ride count by member type
Total_2021_trips_clean %>%
    group_by(member_casual, month_) %>%
    drop_na() %>%
    ggplot(aes(x = month_, fill = member_casual)) + geom_bar(position = "dodge2") +
    scale_fill_manual(values = c("#0448FE", "#FE8C04")) + labs(x = "Month", y = "Ride Count",
    title = "2021 Ride Count per Month") + guides(x = guide_axis(n.dodge = 2))
```



The third quarter (July, August, September) produces the highest monthly riders for both *casual* and *member* customers.

Next, the ride count by weekday for each member type was plotted.

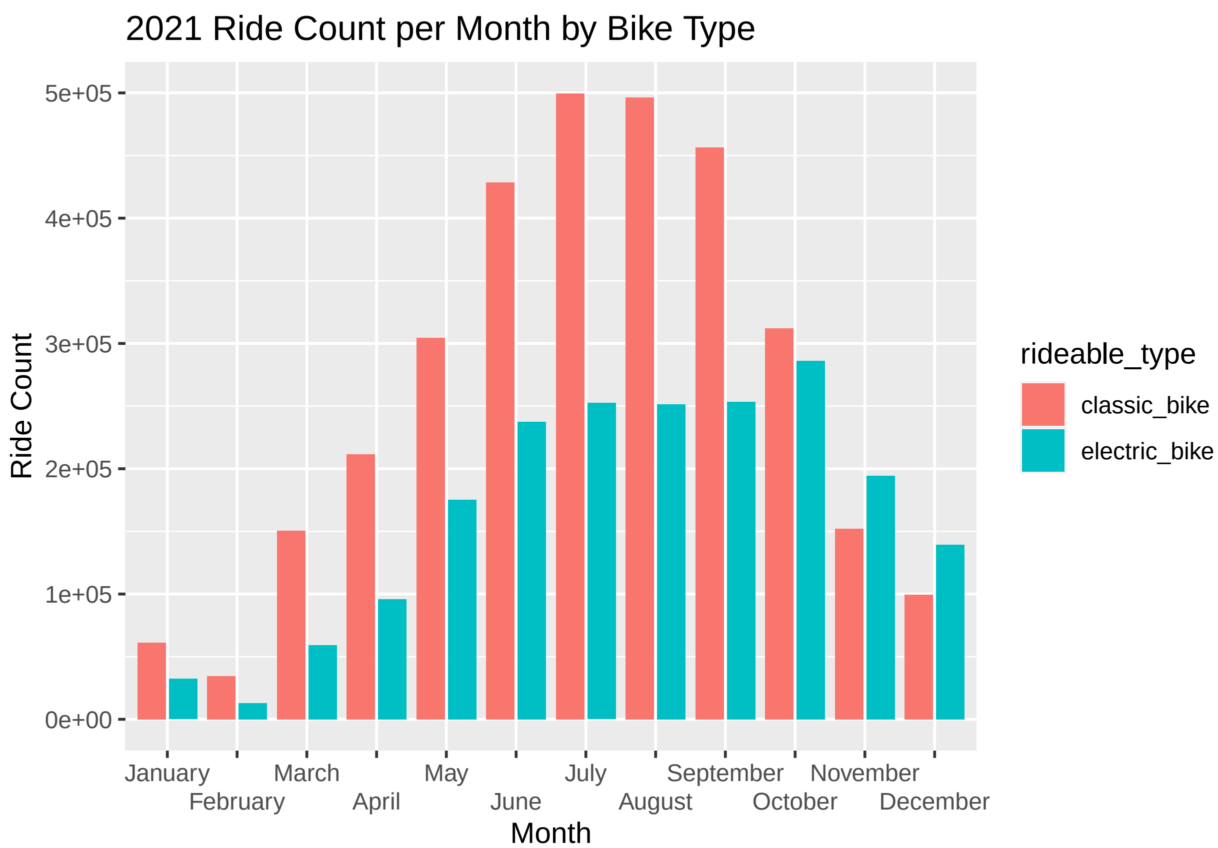
```
# Create bar chart to display ride count by member type for each weekday
Total_2021_trips_clean %>%
    group_by(member_casual, day_of_week) %>%
    drop_na() %>%
    ggplot(aes(x = day_of_week, fill = member_casual)) + geom_bar(position = "dodge2") +
    scale_fill_manual(values = c("#0448FE", "#FE8C04")) + labs(x = "Day of Week",
    y = "Ride Count", title = "2021 Ride Count per Weekday")
```



Casual customers prefer bike rides on the weekend while *member* customer demand is fairly throughout the week.

Lastly, the ride count by month for each bike type was plotted

```
# Create bar chart to display ride count by member type for each weekday
Total_2021_trips_clean %>%
    group_by(rideable_type, month_) %>%
    drop_na() %>%
    ggplot(aes(x = month_, fill = rideable_type)) + geom_bar(position = "dodge2") +
    labs(x = "Month", y = "Ride Count", title = "2021 Ride Count per Month by Bike Type") +
    guides(x = guide_axis(n.dodge = 2))
```



Both the *casual* and *member* customer have a preference for the classic bike.

Summary / Recommendations

This section summarizes the data analysis and provides recommendations based on the above analysis.

To reiterate, the goal of this analysis is to identify difference between casual riders and annual members.

In summary, the differences between casual riders and members are as follows.

1. *Casual* riders demand for bikes peaks on the weekends while *member* demand is fairly consistent throughout the week.

2. *Casual* riders spend more time on average on the rented bike versus *member* riders.

3. *Casual* riders peak demand for bikes is slightly early in the year (July vs. September) versus *member* riders

Based on the above analysis, it is recommended to perform the following to maximize annual membership by converting *casual* riders to *member*:

No	Recommendation
1	Run targeted marketing campaign on the **weekends** maximize *casual* riders customer pool
2	Run targeted marketing campaign in **spring/summer** to maximize *casual* riders customer pool
3	Possibly run targeted marketing campaign in on **classic** bikes to maximize *casual* riders customer pool

Note that an interactive Tableau dashboard of this analysis can be found here

The raw and cleaned dataset can be found here

Next Steps

- Investigate the cause of the negative ride length spike in November 2021
- Investigate ride lengths that last over 1 day
- Review previous years data to confirm if trends identified during the 2021 analysis are valid. The COVID-19 pandemic may have strewed the 2020 and 2021 data