Case Study 1 Cyclistic

John

2024-1-26

Contents

1 Ask **2**
- 1.1 What is the problem to be solved? . . . 2
- 1.2 How can the insights from the analysis drive business decisions? . . . 2
- 1.3 Business Task . . . 2
- 1.4 Key Stakeholders . . . 3

2 Prepare **3**
- 2.1 Where is the data located? . . . 3
- 2.2 How is the data organized? . . . 3
- 2.3 Are there issues with bias or credibility in this data? Does the data ROCCC? . . . 4
- 2.4 How are licensing, privacy, security, and accessibility addressed? . . . 5
- 2.5 How was the data's integrity verified? . . . 5
- 2.6 How does it help answer the question? . . . 5
- 2.7 Are there any problems with the data? . . . 5
- 2.8 Description of all Data Sources used . . . 6

3 Process **7**
- 3.1 What tools were chosen for the analysis and why? . . . 7
- 3.2 Is the data's integrity ensured? . . . 7
- 3.3 What steps were taken to ensure that the data is clean? . . . 7
- 3.4 How can the data be verified as clean and ready to analyze? . . . 8
- 3.5 Is the cleaning process documented so the results can be reviewed and shared? . . . 8
- 3.6 Documentation of any cleaning or manipulation of data . . . 8

4 Analyze **13**
- 4.1 How should the data be organized to perform analysis on it? . . . 13
- 4.2 Has the data been properly formatted? . . . 13
- 4.3 What surprises were discovered in the data? . . . 13
- 4.4 What trends or relationships were found in the data? . . . 13
- 4.5 How will these insights help answer the business questions? . . . 15
- 4.6 Analysis Summary . . . 16

5 Share **16**
- 5.1 Was the question of how annual members and casual riders use Cyclistic bikes differently answered? . . . 16
- 5.2 What story does the data tell? . . . 16
- 5.3 How do the findings relate to the original question? . . . 16
- 5.4 Who is the audience? What is the best way to communicate with them? . . . 16
- 5.5 Can data visualization help share the findings? . . . 17
- 5.6 Is the presentation accessible to the audience? . . . 17
- 5.7 Supporting visualizations and key findings. . . . 17

6 Act **23**
6.1 What is the final conclusion based on the analysis? . . . 23
6.2 How could the team and business apply the insights? . . . 23
6.3 What next steps would the data analyst or the stakeholders take based on the findings? . . . 23
6.4 Is there additional data that could be used to expand on the findings? . . . 23
6.5 The top three recommendations based on the analysis. . . . 23

List of Figures

1 Boxplot of ride_length for casual riders and members . . . 11
2 Boxplot of ride_length with negative values removed for casual riders and members . . . 12
3 Zoomed in Boxplot of ride_length . . . 14
4 Column chart of number_of_rides by weekday for casual riders and members . . . 17
5 Column chart of average_duration by weekday for casual riders and members . . . 18
6 Column chart of median_duration by weekday for casual riders and members . . . 19
7 Map of the starting points of bike rides: blue dots are for casual riders and red dots are for members . . . 21
8 Map of the ending points of bike rides: blue dots are for casual riders and red dots are for members . . . 22

List of Tables

1 Column Names . . . 4
2 ‘all_trips‘ tibble structure without ‘ride_id‘, ‘rideable_type‘, ‘start_station_name‘, ‘start_station_id‘, ‘end_station_name‘, and ‘end_station_id‘ . . . 4
3 Some of ‘start_station_name‘, including NA . . . 6
4 Number of NAs for each column in ‘all_trips‘ . . . 6
5 Number of NaNs for each column in ‘all_trips‘ . . . 7
6 Number of NAs in all columns of ‘all_trips_v2‘ . . . 8
7 Number of NAs in all columns of ‘all_trips_v3‘ . . . 9
8 Unique values of ‘member_casual‘ variable . . . 9
9 Unique values of ‘rideable_type‘ variable . . . 9
10 Summary Statistics of ‘ride_length‘ . . . 13
11 Summary Statistics of ‘ride_length‘ by ‘day_of_week‘ . . . 15
12 Three Statistics of ‘ride_length‘ to be plotted . . . 16

1 Ask

1.1 What is the problem to be solved?

The problem to be solved is to answer the following question: how do annual members and casual riders use Cyclistic bikes differently?

1.2 How can the insights from the analysis drive business decisions?

It is believed that the company’s future success depends on maximizing the number of annual memberships. The insights from understanding how casual riders and annual members use Cyclistic bikes differently will help the marketing analyst team design a new marketing strategy to convert casual riders into annual members.

1.3 Business Task

Design marketing strategies using digital media that will convert casual riders to become annual members.

1.4 Key Stakeholders

- Lily Moreno: The director of marketing and the manager. Responsible for the development of campaigns and initiatives to promote the bike-share program: including email, social media, and other channels.
- Cyclistic executive team: The detail-oriented executive team will decide whether to approve the recommended marketing program.

2 Prepare

2.1 Where is the data located?

The data is located in an Amazon AWS S3 data bucket called "divvy-tripdata" [3], which was by Divvy, "Chicagoland's bike share system across Chicago and Evanston that provides residents and visitors with a convenient, fun and affordable transportation option for getting around and exploring Chicago." According to Divvy Bikes' Data License Agreement: "The City of Chicago owns all right, title, and interest in the Data." [2]

2.2 How is the data organized?

On the data bucket page "divvy-tripdata" [3], the data appears to be split into separate CSV files starting from 2013 up to now, each CSV contained within one zip file. It started out with 2013 in one zip file, then 2014 to 2017 were each split into two files per year, 2018 to 2019 were each split into quarters, the first quarter of 2020 (January to March) was recorded into one file, then all subsequent zip files are for each month following March 2020.

As a separate note, the description document of this case study [6] mentioned "In 2016, Cyclistic launched a successful bike-share offering. . . . ", implying that the company's operation has not started until the year 2016, although the data set started from 2013.

The scope of this case study is for the past 12 months [6]. The following R code chunk reads the CSV files for the time frame December 2022 to November 2023.

```
# Upload Divvy datasets (csv files) here
december_2022 <- read_csv("202212-divvy-tripdata.csv")
january_2023 <- read_csv("202301-divvy-tripdata.csv")
february_2023 <- read_csv("202302-divvy-tripdata.csv")
march_2023 <- read_csv("202303-divvy-tripdata.csv")
april_2023 <- read_csv("202304-divvy-tripdata.csv")
may_2023 <- read_csv("202305-divvy-tripdata.csv")
june_2023 <- read_csv("202306-divvy-tripdata.csv")
july_2023 <- read_csv("202307-divvy-tripdata.csv")
august_2023 <- read_csv("202308-divvy-tripdata.csv")
september_2023 <- read_csv("202309-divvy-tripdata.csv")
october_2023 <- read_csv("202310-divvy-tripdata.csv")
november_2023 <- read_csv("202311-divvy-tripdata.csv")
```

The tibbles created from reading all of the above 12 CSV files are put into a list.

```
list_of_dfs <- list(december_2022, january_2023, february_2023, march_2023,
                    april_2023, may_2023, june_2023, july_2023,
                    august_2023, september_2023, october_2023, november_2023)
```

The fields in all tibbles of `list_of_dfs` are the same as indicated in the following results, and the column names are also shown below in Table 1. For the hidden code which generates the following results, please refer to [4].

Table 1: Column Names

x
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

Table 2: 'all_trips' tibble structure without 'ride_id', 'rideable_type', 'start_station_name', 'start_station_id', 'end_station_name', and 'end_station_id'

started_at	ended_at	start_lat	start_lng	end_lat	end_lng	member_casual
2022-12-05 10:47:18	2022-12-05 10:56:34	41.91824	-87.65711	41.92217	-87.63889	member
2022-12-18 06:42:33	2022-12-18 07:08:44	41.94011	-87.64545	41.92217	-87.63889	casual
2022-12-13 08:47:45	2022-12-13 08:59:51	41.88592	-87.65113	41.89435	-87.62280	member
2022-12-13 18:50:47	2022-12-13 19:19:48	41.83846	-87.63541	41.88137	-87.67493	member
2022-12-14 16:13:39	2022-12-14 16:27:50	41.89595	-87.66773	41.92008	-87.67785	casual
2022-12-02 15:24:47	2022-12-02 15:34:14	41.87068	-87.62571	41.88314	-87.63724	member

```
## [1] "The column names are all the same."
```

The data types for the fields in all of the 12 CSV files are also the same as indicated below. For the hidden code which generates the following results, please refer to [4].

```
## [1] "The column data types are all the same."
```

The following code binds the data from all 12 CSV files into one tibble.

```
all_trips <- bind_rows(list_of_dfs)
```

The tibble is structured as shown in Table 2, where some of the columns are shown and only the first six rows are listed.

2.3 Are there issues with bias or credibility in this data? Does the data ROCCC?

The data doesn't appear to possess a bias that would systematically skew the results in a certain direction. The data consists of Cyclistic bike trips that took place between the beginning of December 2022 to the end of November 2023 (12 months). Sampling bias may not apply here, for example, this dataset does not miss a single season as it covers the entire contiguous 12 months. The data were automatically recorded by the geotracking of the bicycles, it would be difficult for observer bias to occur.

For the other two biases, interpretation and confirmation, they are a posterior to data collection and must be kept in mind during the analysis. More specifically, interpretation bias is the tendency to always interpret

ambiguous situations in a positive or negative way, and confirmation bias is the tendency to search for, or interpret information in a way that confirms preexisting beliefs [6].

Identifying the data source as a good data source with ROCCC:

- Reliable: This dataset is identified for use in Google Data Analytics certificate program [6] module 8. It is trusted that the information in the dataset has been vetted and proven fit for use in this case study.
- Original: The data has been made available by Motivate International Inc. under a Data License Agreement (with Lyft Bikes and Scooters, LLC) [6].
- Comprehensive: For the purposes of this case study, the datasets are appropriate and will enable the business questions to be answered.
- Current: The data covers the time from the beginning of December 2022 to the end of November 2023 (12 months).
- Citing: The data is cited since the Google Data Analytics Certificate program [6] cites the data provided by Divvy (Cyclistic) through geotracking of their bikes, and "Divvy is a program of the Chicago Department of Transportation (CDOT), which owns the city's bikes, stations and vehicles." [2]

2.4 How are licensing, privacy, security, and accessibility addressed?

Licensing is addressed in the Data License Agreement provided by Cyclistic (Divvy Bikes), which "grants [the user] a non-exclusive, royalty-free, limited, perpetual license to access, reproduce, analyze, copy, modify, distribute in [their] product or service and use the Data for any lawful purpose." [2]

The dataset provided does not contain personally identifiable information (PII).

Security at the source is addressed by the Lyft Privacy Policy, which states that they "take reasonable and appropriate measures designed to protect [the user's] personal information." [5] The usage of the dataset in this case study is limited to the purpose of completing the module 8 of Google Data Analytics certificate program [6].

The dataset is referred to by the Google Data Analytics Certificate Program [6] and is accessible at [3].

2.5 How was the data's integrity verified?

The accuracy of the data [3] is good in regards to the dates and times of the bike rides. The data is almost complete with a portion of the total records missing values for station fields like `start_station_name` and `start_station_id` to be further clarified later in the document. Additionally, there are missing values for end station latitude and longitude, and there are some inconsistency in `started_at` and `ended_at` for some trips because they yield negative differences. All of this will be elaborated later in this document. The data can also be considered trustworthy since the data is owned by the City of Chicago [2].

2.6 How does it help answer the question?

The dataset contains all relevant information of individual trips for the past 12 months, which would help to understand how the cansual riders and annual members use the the bike-share service in the past 12 months.

2.7 Are there any problems with the data?

There exist some NA in the `start_station_name` field as shown in Table 3. For the hidden code which generates Table 3, please refer to [4].

As stated above, there are NAs for `start_station_name`. Table 4 shows NA counts for that column and any other columns.

Table 3: Some of ‘start_station_name‘, including NA

start_station_name
Aberdeen St & Jackson Blvd
900 W Harrison St
63rd St Beach
410
2112 W Peterson Ave
NA

Table 4: Number of NAs for each column in ‘all_trips‘

	x
ride_id	0
rideable_type	0
started_at	0
ended_at	0
start_station_name	869289
start_station_id	869421
end_station_name	922436
end_station_id	922577
start_lat	0
start_lng	0
end_lat	6879
end_lng	6879
member_casual	0

```
all_trips_na_counts_1 <- colSums(is.na(all_trips))
kable(all_trips_na_counts_1, caption = "Number of NAs for each column in `all\\_trips`")
```

There are no NaN values in the fields as shown in Table 5.

```
all_trips_nan_counts <- all_trips %>%
  summarize_all(~ sum(is.nan(.)))
all_trips_nan_counts_vector <- as.numeric(all_trips_nan_counts)
names(all_trips_nan_counts_vector) <- c("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")
kable(all_trips_nan_counts_vector, caption = "Number of NaNs for each column in `all\\_trips`")
```

2.8 Description of all Data Sources used

The data source used in this case study is a data bucket called divvy-tripdata [3], which was produced by Divvy, “Chicagoland’s bike share system across Chicago and Evanston that provides residents and visitors with a convenient, fun and affordable transportation option for getting around and exploring Chicago.” [1] According to Divvy Bikes’ Data License Agreement: “The City of Chicago owns all right, title, and interest in the Data.” [2] This is most likely first-party data. For this case study, the company will be referred to as Cyclistic.

Table 5: Number of NaNs for each column in ‘all_trips‘

	x
ride_id	0
rideable_type	0
started_at	0
ended_at	0
start_station_name	0
start_station_id	0
end_station_name	0
end_station_id	0
start_lat	0
start_lng	0
end_lat	0
end_lng	0
member_casual	0

3 Process

3.1 What tools were chosen for the analysis and why?

R and RStudio, R is a programming language frequently used for statistical analysis, visualization, and other data analysis. This project started with spreadsheets, but it soon became apparent that spreadsheets could not handle the large dataset as it takes time to process the data. Then the tool for this project was switched to R/RStudio. So far, R/RStudio has shown that it can handle processing the large dataset for this project.

3.2 Is the data's integrity ensured?

The accuracy of the data is good in regards to the dates and times of the bike rides. The data is almost complete with a portion of the total records missing values for station fields like `start_station_name` and `start_station_id`. Please refer to the end of the “Prepare” section for more details. The data can also be considered trustworthy since the data is owned by the City of Chicago. [2]

3.3 What steps were taken to ensure that the data is clean?

1. Check for consistency in columns and data types from the CSV files in the section “Prepare”, subsection “How is the data organized?”.
2. Merge the multiple data sets into a single data set in the section “Prepare”, subsection “How is the data organized?”.
3. Inspect the new data set for any NAs or NaNs in the section “Prepare”, subsection “Are there any problems with the data?”.
4. Remove columns that have high NA counts or are not necessary for the analysis in the section “Process”, subsection “Documentation of any cleaning or manipulation of data”.
5. Remove rows that contain remaining NAs, preferably when NA counts are small in the section “Process”, subsection “Documentation of any cleaning or manipulation of data”.
6. Inspect the resulting data set for any more NAs in the section “Process”, subsection “Documentation of any cleaning or manipulation of data”.
7. Check the unique values for all string variables in the section “Process”, subsection “Documentation of any cleaning or manipulation of data”.
8. Create and format new fields necessary for the analysis in the section “Process”, subsection “Documentation of any cleaning or manipulation of data”.
9. Check the data type of ride_length and convert to numeric type if necessary in the section “Process”, subsection “Documentation of any cleaning or manipulation of data”.

Table 6: Number of NAs in all columns of 'all_trips_v2'

	x
ride_id	0
rideable_type	0
started_at	0
ended_at	0
start_lat	0
start_lng	0
end_lat	6879
end_lng	6879
member_casual	0

10. Check the values of ride_length and filter out the rows with negative values in the section "Process", subsection "Documentation of any cleaning or manipulation of data".

3.4 How can the data be verified as clean and ready to analyze?

The data must have all of the required information, consistent data types, no NAs, and no outliers that would have any effect on the analysis. The dataset `all_trips_v3`, which is created in section "Process", subsection "Documentation of any cleaning or manipulation of data", is clear and will be used for the rest of the analysis steps.

3.5 Is the cleaning process documented so the results can be reviewed and shared?

The steps for the cleaning process and the data manipulation is documented clearly in this R markdown document.

3.6 Documentation of any cleaning or manipulation of data

Drop the fields `start_station_name`, `start_station_id`, `end_station_name`, and `end_station_id` due to NA counts. Refer to section "Prepare", subsection "Are there any problems with the data?" This analysis will not use these four fields.

```
all_trips_v2 <- all_trips %>%
  select(-c(start_station_name, start_station_id, end_station_name, end_station_id))
```

Check the NA counts after removal of four fields, as shown in Table 6.

```
all_trips_na_counts_2 <- colSums(is.na(all_trips_v2))
kable(all_trips_na_counts_2, caption = "Number of NAs in all columns of `all\\_trips\\_v2`")
```

Remove the rows that still have NAs.

```
all_trips_v3 <- na.omit(all_trips_v2)
```

Check the NA counts after removal of rows with NAs, as shown in Table 7.

```
all_trips_na_counts_3 <- colSums(is.na(all_trips_v3))
kable(all_trips_na_counts_3, caption = "Number of NAs in all columns of `all\\_trips\\_v3`")
```

Table 7: Number of NAs in all columns of 'all_trips_v3'

	x
ride_id	0
rideable_type	0
started_at	0
ended_at	0
start_lat	0
start_lng	0
end_lat	0
end_lng	0
member_casual	0

Table 8: Unique values of 'member_casual' variable

Var1	Freq
casual	2046594
member	3624137

Use glimpse to examine the resulting data set.

```
glimpse(all_trips_v3)
```

```
## Rows: 5,670,731
## Columns: 9
## $ ride_id       <chr> "65DBD2F447EC51C2", "0C201AA7EA0EA1AD", "E0B148CCB358A49~
## $ rideable_type <chr> "electric_bike", "classic_bike", "electric_bike", "class~
## $ started_at    <dttm> 2022-12-05 10:47:18, 2022-12-18 06:42:33, 2022-12-13 08~
## $ ended_at      <dttm> 2022-12-05 10:56:34, 2022-12-18 07:08:44, 2022-12-13 08~
## $ start_lat     <dbl> 41.91824, 41.94011, 41.88592, 41.83846, 41.89595, 41.870~
## $ start_lng     <dbl> -87.65711, -87.64545, -87.65113, -87.63541, -87.66773, -~
## $ end_lat       <dbl> 41.92217, 41.92217, 41.89435, 41.88137, 41.92008, 41.883~
## $ end_lng       <dbl> -87.63889, -87.63889, -87.62280, -87.67493, -87.67785, -~
## $ member_casual <chr> "member", "casual", "member", "member", "casual", "membe~
```

Check the unique values and associated counts of `member_casual` in Table 8.

```
table(all_trips_v3$member_casual) %>%
  kable(caption = "Unique values of `member\\_casual` variable")
```

Check the unique values and associated counts of `rideable_type` in Table 9.

```
table(all_trips_v3$rideable_type) %>%
  kable(caption = "Unique values of `rideable\\_type` variable")
```

Table 9: Unique values of 'rideable_type' variable

Var1	Freq
classic_bike	2660114
docked_bike	77996
electric_bike	2932621

Create date-related columns using the `started_at` column.

```
#The default format is yyyy-mm-dd
all_trips_v3$date <- as.Date(all_trips_v3$started_at)
all_trips_v3$month <- format(as.Date(all_trips_v3$date), "%m")
all_trips_v3$day <- format(as.Date(all_trips_v3$date), "%d")
all_trips_v3$year <- format(as.Date(all_trips_v3$date), "%Y")
all_trips_v3$day_of_week <- format(as.Date(all_trips_v3$date), "%A")
```

Create the `ride_length` column using `difftime()` to get duration of a bike ride.

```
all_trips_v3$ride_length <- difftime(all_trips_v3$ended_at,all_trips_v3$started_at)
```

Format `ride_length` to numeric data type for use in analysis.

```
is.numeric(all_trips_v3$ride_length)
```

```
## [1] FALSE
```

```
all_trips_v3$ride_length <- as.numeric(as.character(all_trips_v3$ride_length))
is.numeric(all_trips_v3$ride_length)
```

```
## [1] TRUE
```

Use a boxplot to create a visual for the range of `ride_length` values for casual riders and members as shown in Figure 1.

```
all_trips_v3 %>%
  ggplot(aes(member_casual, ride_length)) +
  geom_boxplot()
```

Check the summary of the clean dataset, especially, for example, the minimum value of `ride_length` for negative calculations.

```
summary(all_trips_v3)
```

```
##    ride_id          rideable_type        started_at
##  Length:5670731     Length:5670731     Min.   :2022-12-01 00:01:22.00
##  Class :character   Class :character   1st Qu.:2023-05-10 19:35:06.50
##  Mode  :character   Mode  :character   Median :2023-07-13 05:22:15.00
##                                        Mean   :2023-07-03 13:58:04.04
##                                        3rd Qu.:2023-09-07 09:30:55.00
##                                        Max.   :2023-11-30 23:59:14.00
##     ended_at                       start_lat       start_lng
##  Min.   :2022-12-01 00:03:41.00   Min.   :41.63   Min.   :-87.94
##  1st Qu.:2023-05-10 19:51:07.50   1st Qu.:41.88   1st Qu.:-87.66
##  Median :2023-07-13 05:35:35.00   Median :41.90   Median :-87.64
##  Mean   :2023-07-03 14:13:12.75   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:2023-09-07 09:45:11.50   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :2023-12-01 17:00:47.00   Max.   :42.07   Max.   :-87.46
##     end_lat         end_lng       member_casual          date
##  Min.   : 0.00   Min.   :-88.16   Length:5670731     Min.   :2022-12-01
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   1st Qu.:2023-05-10
##  Median :41.90   Median :-87.64   Mode  :character   Median :2023-07-13
##  Mean   :41.90   Mean   :-87.65                      Mean   :2023-07-02
##  3rd Qu.:41.93   3rd Qu.:-87.63                      3rd Qu.:2023-09-07
##  Max.   :42.18   Max.   :  0.00                      Max.   :2023-11-30
##     month               day                year           day_of_week
##  Length:5670731     Length:5670731     Length:5670731     Length:5670731
```

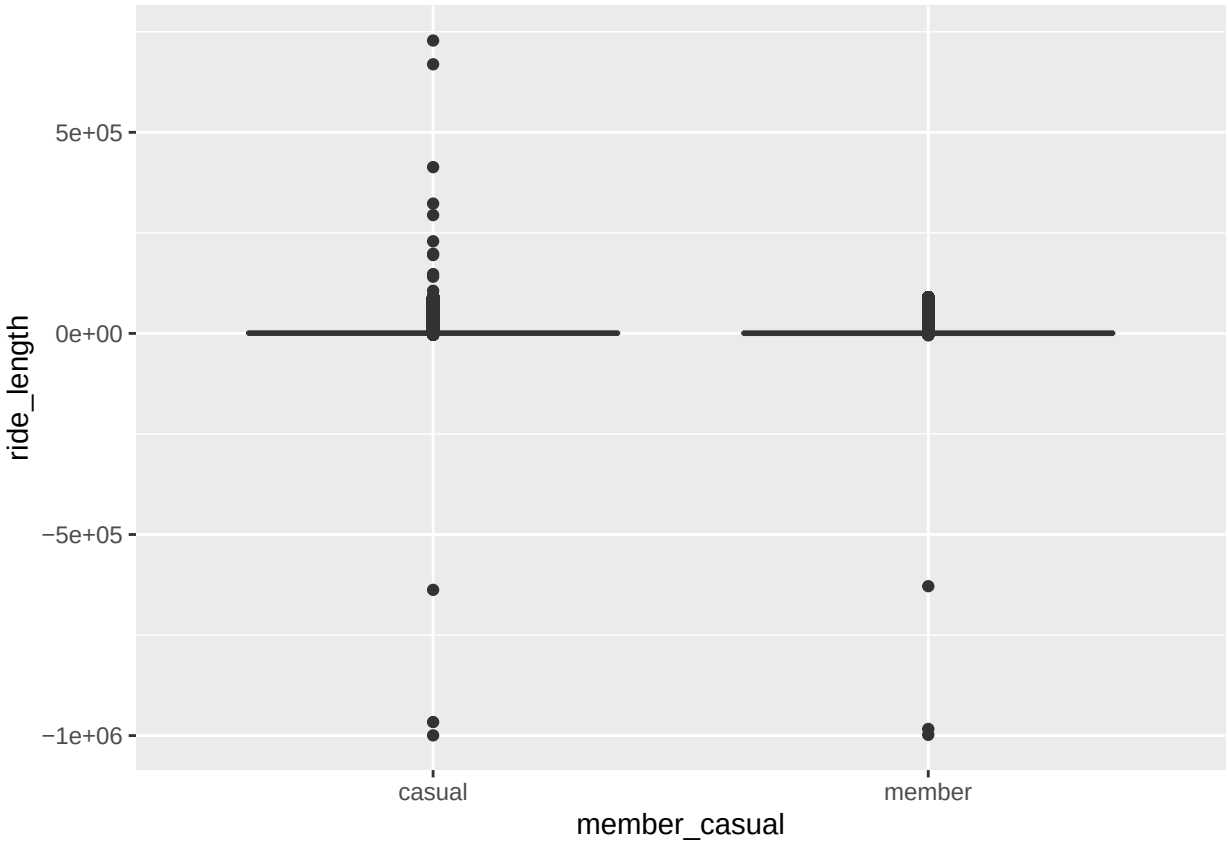

Figure 1: Boxplot of ride_length for casual riders and members

```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   ride_length
##  Min.   :-999391.0
##  1st Qu.:    325.0
##  Median :    571.0
##  Mean   :    908.7
##  3rd Qu.:   1013.0
##  Max.   : 728178.0
```

Remove rows where `ride_length` is less than 0.

```
all_trips_v3 <- all_trips_v3[!(all_trips_v3$ride_length < 0),]
```

Use another boxplot to show that the negative `ride_length` values have been removed as shown in Figure 2.

```
all_trips_v3 %>%
  ggplot(aes(member_casual, ride_length)) +
  geom_boxplot()
```

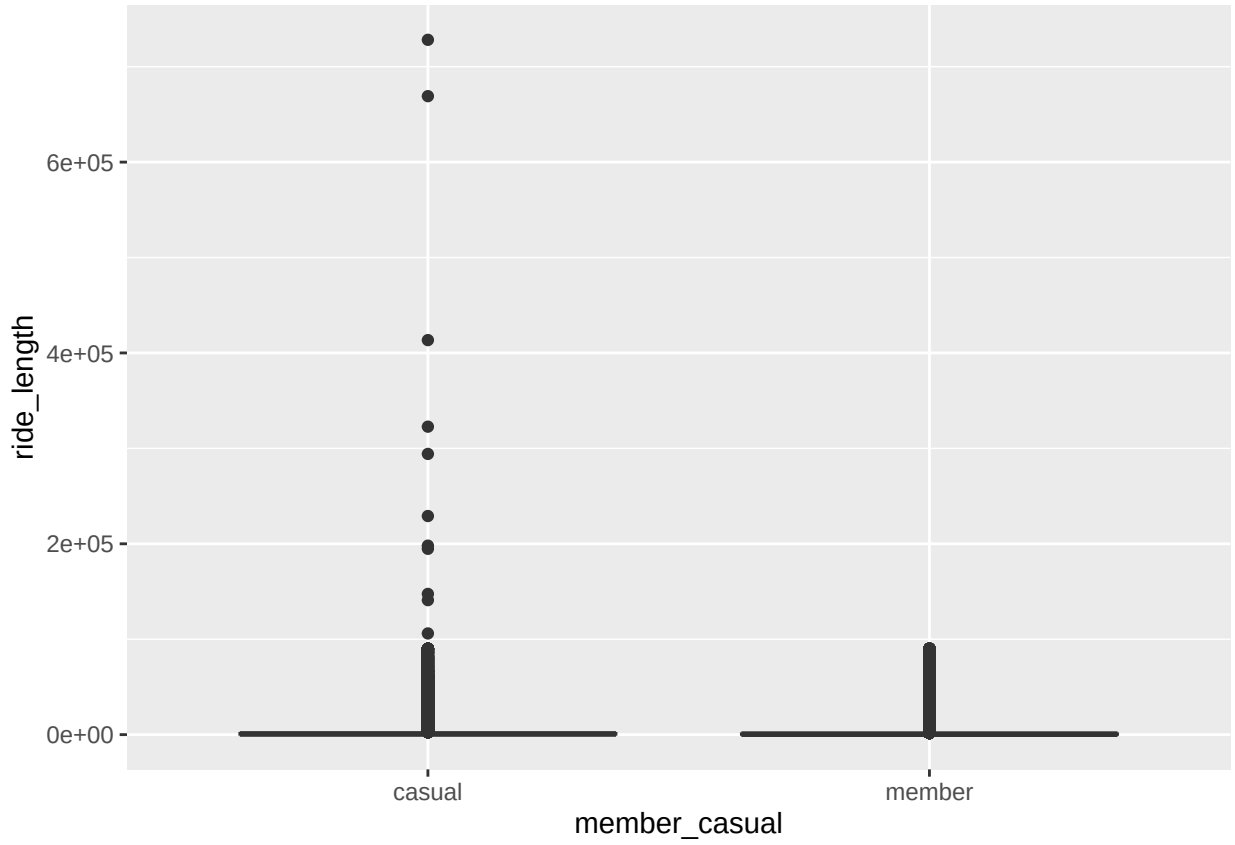

Figure 2: Boxplot of ride_length with negative values removed for casual riders and members

Table 10: Summary Statistics of ‘ride_length‘

member_casual	mean	minimum	q25	median	q75	maximum
casual	1238.426	0	397	709	1321	728178
member	724.076	0	296	511	876	89996

4 Analyze

4.1 How should the data be organized to perform analysis on it?

The monthly data frames were stacked together into one data frame with the `bind_rows()` function. Columns containing high NA counts were removed while the rows they were in are maintained. Rows with remaining NAs were removed. New columns were created, calculated, and formatted for analysis. Rows that contained negative `ride_length` values were removed.

4.2 Has the data been properly formatted?

Yes. Additionally, in the section “Analyze”, subsection “What trends or relationships were found in the data?” the order for day of the week will be added.

4.3 What surprises were discovered in the data?

Some of the values for `ride_length` were negative as discussed in section “Process”, subsection “Documentation of any cleaning or manipulation of data”.

4.4 What trends or relationships were found in the data?

As shown by the following boxplot (Figure 3) and summary statistics (Table 10): The `ride_length` mean, 25 percentile, median, and 75 percentile values for casual riders are all higher than those for members.

```
all_trips_v3 %>%
  ggplot(aes(member_casual, ride_length)) +
  geom_boxplot() +
  ylim(-1, 2200)
```

```
# Compute summary statistics
summary_stats <- all_trips_v3 %>%
  group_by(member_casual) %>%
  summarize(
    mean = mean(ride_length),
    minimum = min(ride_length),
    q25 = quantile(ride_length, 0.25),
    median = median(ride_length),
    q75 = quantile(ride_length, 0.75),
    maximum = max(ride_length)
  )
kable(summary_stats, caption = "Summary Statistics of `ride\\_length`")
```

The order of values in the `day_of_week` column was set to be applied to the next summary statistics for casual riders and members.

```
all_trips_v3$day_of_week <- ordered(all_trips_v3$day_of_week,
  levels=c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"))
```

The average `ride_length` of casual riders appears to be higher than the average `ride_length` of members

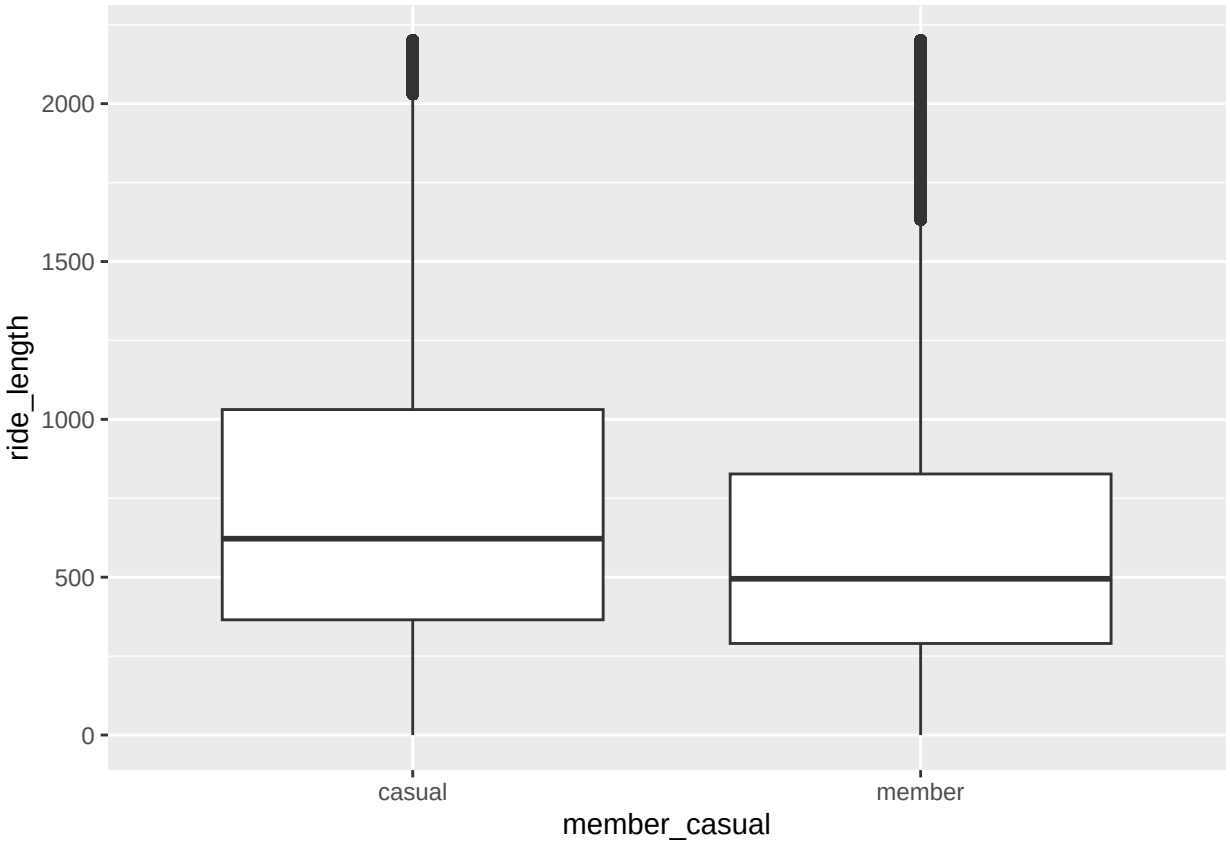

Figure 3: Zoomed in Boxplot of ride_length

Table 11: Summary Statistics of 'ride_length' by 'day_of_week'

member_casual	day_of_week	mean	minimum	q25	median	q75	maximum
casual	Sunday	1440.2958	0	451	833	1579	229104
casual	Monday	1218.3205	0	377	675	1289	89995
casual	Tuesday	1106.1613	0	365	635	1158	728178
casual	Wednesday	1059.4184	0	360	618	1100	322740
casual	Thursday	1082.0231	0	366	631	1124	413473
casual	Friday	1204.3415	0	393	696	1286	198050
casual	Saturday	1403.5057	0	456	831	1541	669136
member	Sunday	808.8823	0	308	550	979	89995
member	Monday	687.9945	0	285	486	831	89996
member	Tuesday	695.1957	0	293	499	845	89144
member	Wednesday	692.7765	0	293	501	842	89994
member	Thursday	693.0186	0	293	502	850	89995
member	Friday	720.8727	0	293	504	862	89996
member	Saturday	806.1553	0	318	559	977	89994

on each day of the week as shown in Table 11. The quantiles of casual riders is also higher than the quantiles of members by day of the week.

```
# Compute summary statistics with day of week
summary_stats_3 <- all_trips_v3 %>%
  group_by(member_casual, day_of_week) %>%
  summarize(
    mean = mean(ride_length),
    minimum = min(ride_length),
    q25 = quantile(ride_length, 0.25),
    median = median(ride_length),
    q75 = quantile(ride_length, 0.75),
    maximum = max(ride_length)
  )
kable(summary_stats_3, caption = "Summary Statistics of `ride\\_length` by `day\\_of\\_week`")
```

The number of rides for members by day of the week is higher than the number of rides for casual riders, especially during the weekdays, as shown in Table 12.

```
# analyze ridership data by type and weekday
all_trips_v3 %>%
  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
            average_duration = mean(ride_length), #the average duration
            median_duration = median(ride_length)) %>% #the median duration
  arrange(member_casual, weekday) %>% #sorts by usertype then weekday
  kable(caption = "Three Statistics of `ride\\_length` to be plotted")
```

4.5 How will these insights help answer the business questions?

The above insights will help find out what are the differences and similarities to how casual riders and annual members use Cyclistic bikes during the months of December 2022 to November 2023.

Table 12: Three Statistics of 'ride_length' to be plotted

member_casual	weekday	number_of_rides	average_duration	median_duration
casual	Sun	332373	1440.2958	833
casual	Mon	234090	1218.3205	675
casual	Tue	246195	1106.1613	635
casual	Wed	247254	1059.4184	618
casual	Thu	270439	1082.0231	631
casual	Fri	308708	1204.3415	696
casual	Sat	407395	1403.5057	831
member	Sun	402446	808.8823	550
member	Mon	491717	687.9945	486
member	Tue	575225	695.1957	499
member	Wed	578664	692.7765	501
member	Thu	587963	693.0186	502
member	Fri	521761	720.8727	504
member	Sat	466239	806.1553	559

4.6 Analysis Summary

Casual riders tend to ride longer in time and less often than members. This conclusion will be reinforced by the visuals in the following Share section.

One more observation which will be presented in the Share section is that casual riders are more spread out into the suburban neighborhoods of downtown Chicago.

5 Share

5.1 Was the question of how annual members and casual riders use Cyclistic bikes differently answered?

Yes, there are differences with how annual members and casual riders use Cyclistic bikes, referring to the above Analysis Summary.

5.2 What story does the data tell?

Annual members tend to have more bike rides than casual riders for every day of the week, especially on weekdays.

The `average_duration` and `median_duration` is higher for casual riders than annual members. The `average_duration` and `median_duration` of annual members is more consistent for each day of the week, see the later graphs in this document.

The bike rides of casual riders are more spread out in the Chicago area than the ones for annual members.

5.3 How do the findings relate to the original question?

The findings show the differences between annual members and casual riders in regards to the number of bike rides, average ride lengths, and so on.

5.4 Who is the audience? What is the best way to communicate with them?

The audience will consist of the director of marketing and the Cyclistic executive team. Visualizations along with good data storytelling will help in communicating the meaning of this dataset to the audience.

5.5 Can data visualization help share the findings?

Yes, especially for this case study where the following graphs will help communicate the conclusions.

5.6 Is the presentation accessible to the audience?

This data analysis report and the R code associated with it is available in [4].

5.7 Supporting visualizations and key findings.

Annual members tend to have more bike rides than casual riders for every day of the week, especially on the weekdays, as shown in Figure 4.

```
# Let's visualize the number of rides by rider type
all_trips_v3 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

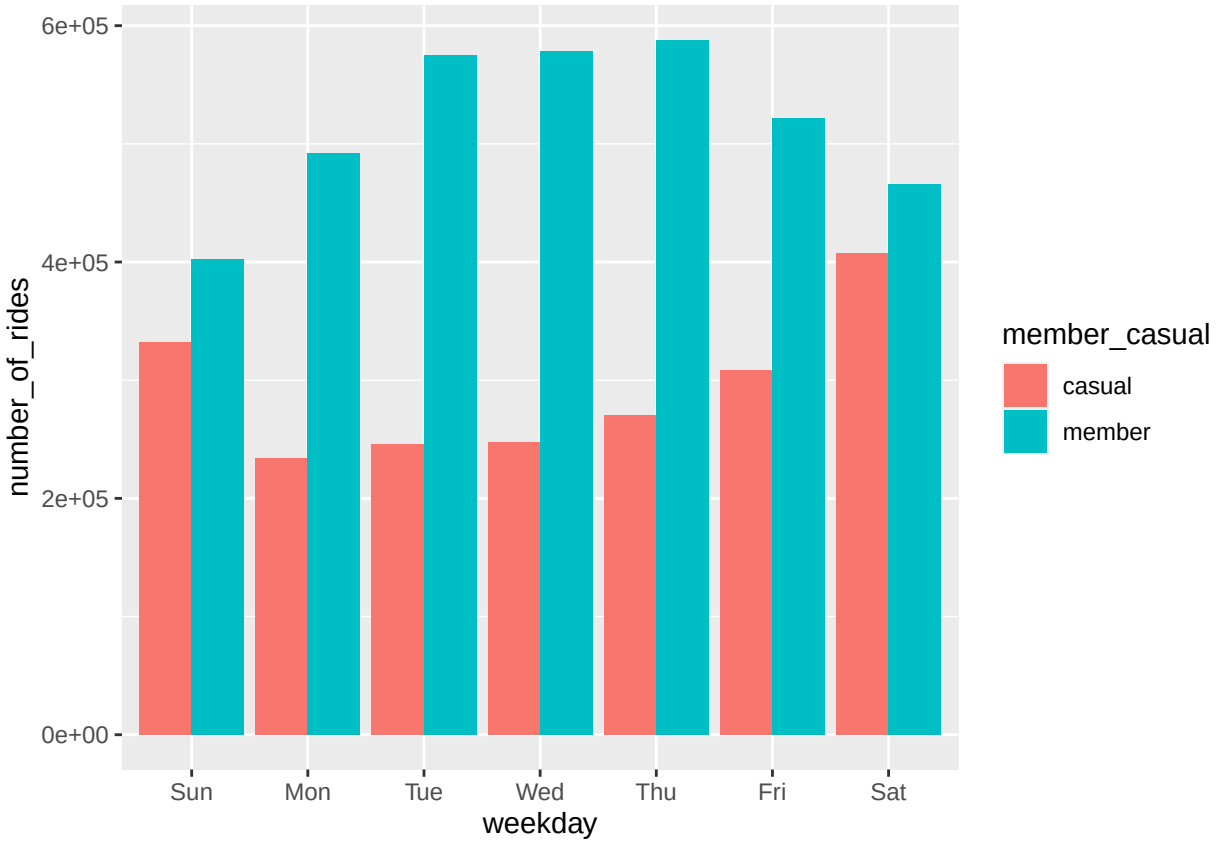

Figure 4: Column chart of number_of_rides by weekday for casual riders and members

Casual riders tend to have longer average bike rides than for annual members, especially on the weekends. Annual members tend to have more consistent average bike ride lengths than for casual riders, as shown in Figure 5.

```r
# Let's create a visualization for average duration
all_trips_v3 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

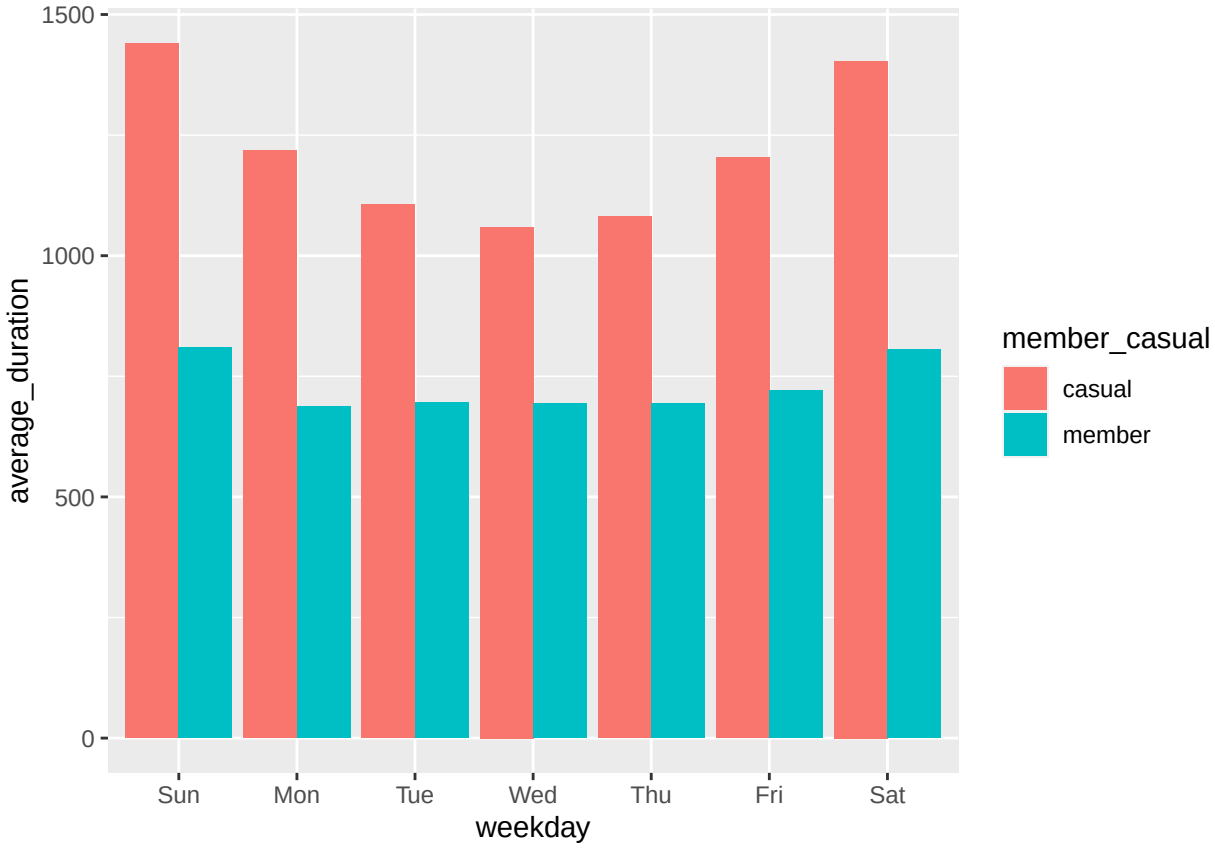

Figure 5: Column chart of average_duration by weekday for casual riders and members

The median bike duration of casual riders tend to be longer than for annual members, especially on the weekends. Annual members tend to have more consistent median bike ride lengths than for casual riders, as shown in Figure 6.

```r
# Let's create a visualization for median duration
all_trips_v3 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(median_duration = median(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = median_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

The cleaned dataset is split into two groups, one for casual riders and the other for annual members.

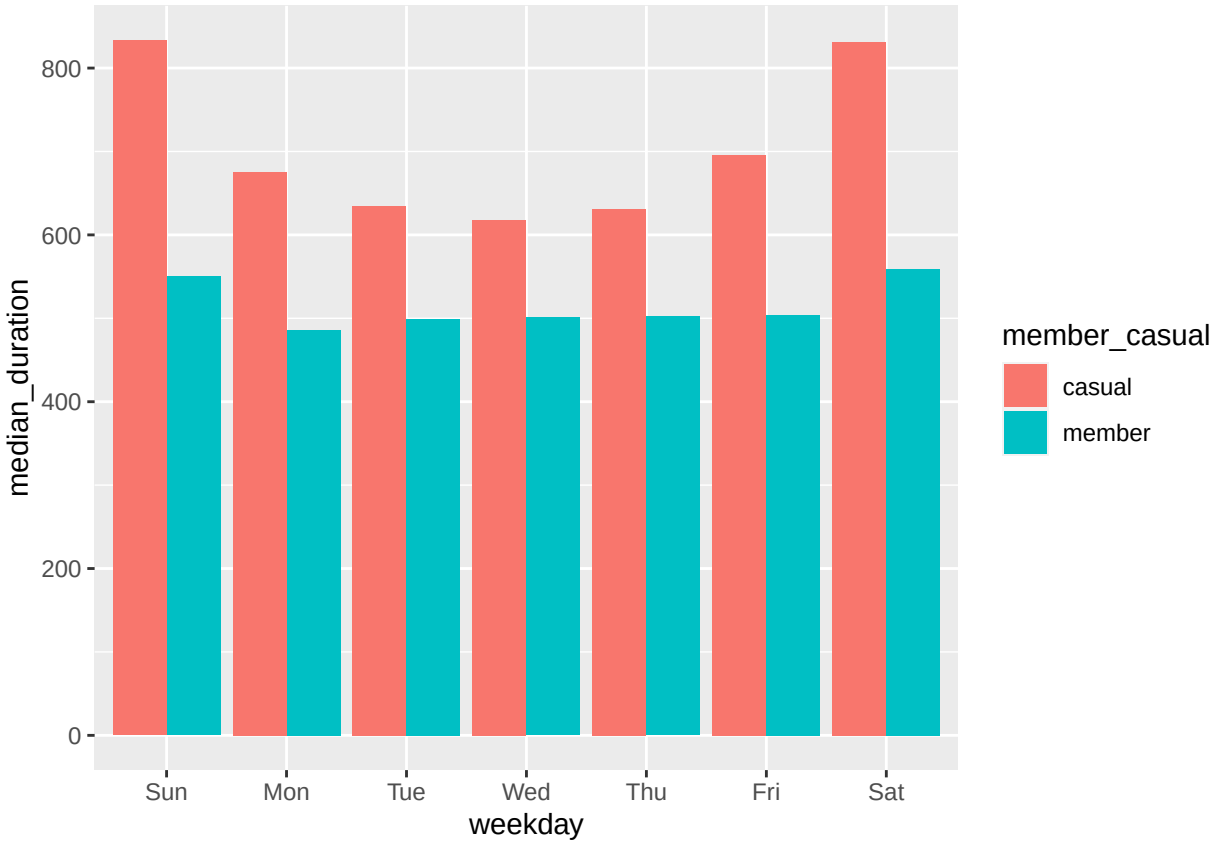

Figure 6: Column chart of median_duration by weekday for casual riders and members

```r
all_trips_v3_casual <- all_trips_v3 %>%
  filter(member_casual == "casual")

all_trips_v3_member <- all_trips_v3 %>%
  filter(member_casual == "member")
```

Create subsets of bike rides for casual riders and for annual members, then shuffle both subsets together randomly.

```r
# Set the seed for reproducibility
set.seed(123)

percentage <- 0.01
member_adjustment <- 1.771
all_trips_casual_subset <- all_trips_v3_casual %>%
  slice_sample(prop = percentage, replace = FALSE)
all_trips_member_subset <- all_trips_v3_member %>%
  slice_sample(prop = (percentage / member_adjustment), replace = FALSE)
all_trips_subset <- bind_rows(all_trips_casual_subset, all_trips_member_subset)
all_trips_shuffle_subset <- all_trips_subset[sample(nrow(all_trips_subset)), ]
```

The following maps uses two colors: blue is for casual riders and red is for annual members.

The starting points for casual riders are more spread out than for annual members as shown in Figure 7.

```r
# Create a leaflet map
my_start_map <- all_trips_shuffle_subset %>%
  leaflet() %>%
  addTiles()  # You can customize the map tiles using addProviderTiles() if needed

# Add circle markers with colors based on the categorical column
my_start_map <- my_start_map %>%
  addCircleMarkers(
    lng = ~start_lng,
    lat = ~start_lat,
    color = ~ifelse(member_casual == "casual", "blue", "red"),
    opacity = 0.7,
    fillOpacity = 0.7,
    radius = 4,
    weight = 1
  )

# Save the leaflet map as an HTML file
start_tmp_html <- tempfile(fileext = ".html")
saveWidget(my_start_map, start_tmp_html, selfcontained = FALSE)

# Use webshot to capture the leaflet map as an image
start_tmp_image <- tempfile(fileext = ".png")
webshot::webshot(start_tmp_html, file = start_tmp_image)
```

The ending points for casual riders are also more spread out than for annual members, as shown in Figure 8. There is at least one bike ride for each group that ends further away than the rest of their respective groups.

```r
# Create a leaflet map
my_end_map <- all_trips_shuffle_subset %>%
  leaflet() %>%
```

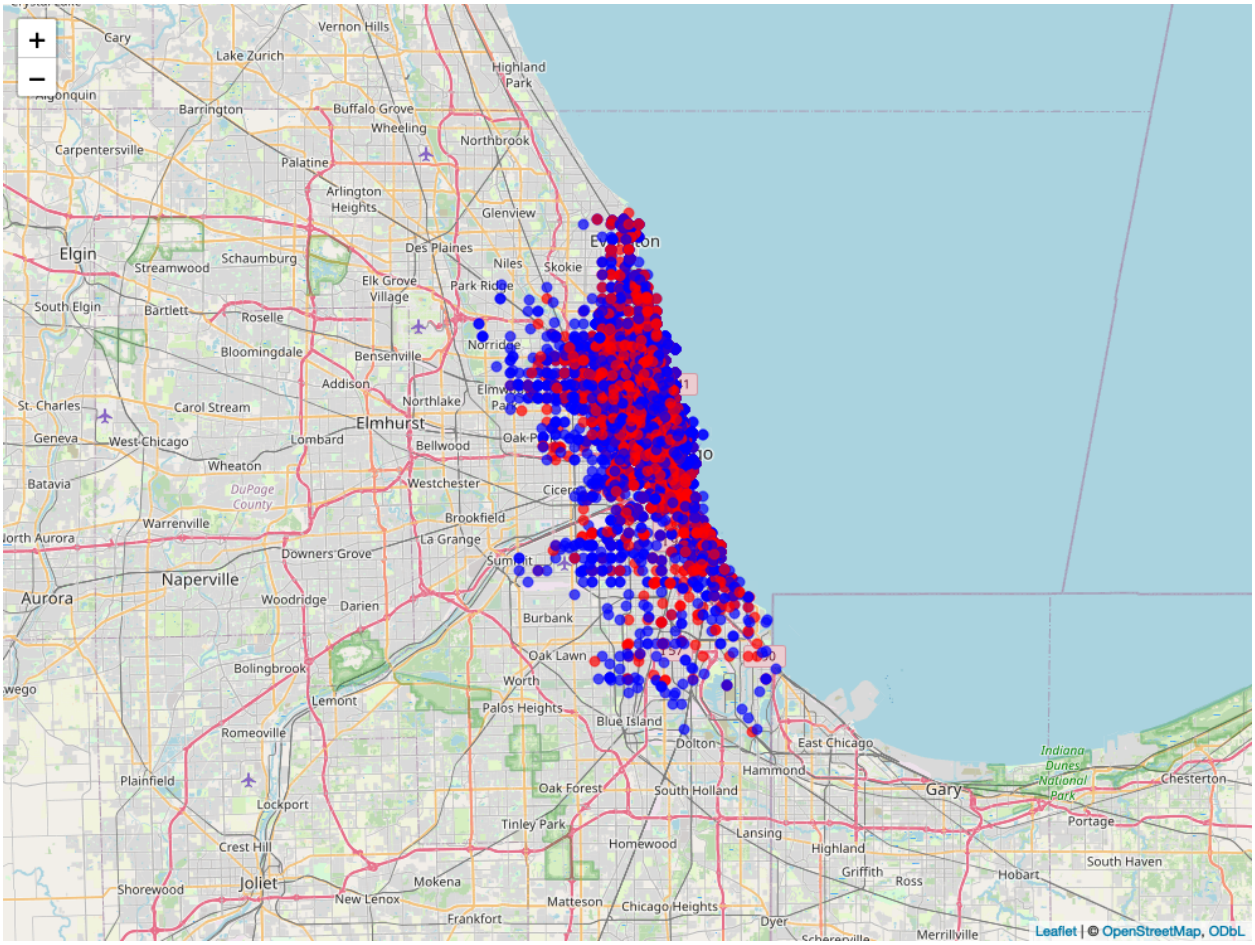

Figure 7: Map of the starting points of bike rides: blue dots are for casual riders and red dots are for members

```r
  addTiles()  # You can customize the map tiles using addProviderTiles() if needed

# Add circle markers with colors based on the categorical column
my_end_map <- my_end_map %>%
  addCircleMarkers(
    lng = ~end_lng,
    lat = ~end_lat,
    color = ~ifelse(member_casual == "casual", "blue", "red"),
    opacity = 0.7,
    fillOpacity = 0.7,
    radius = 4,
    weight = 1
  )

# Save the leaflet map as an HTML file
end_tmp_html <- tempfile(fileext = ".html")
saveWidget(my_end_map, end_tmp_html, selfcontained = FALSE)

# Use webshot to capture the leaflet map as an image
end_tmp_image <- tempfile(fileext = ".png")
webshot::webshot(end_tmp_html, file = end_tmp_image)
```

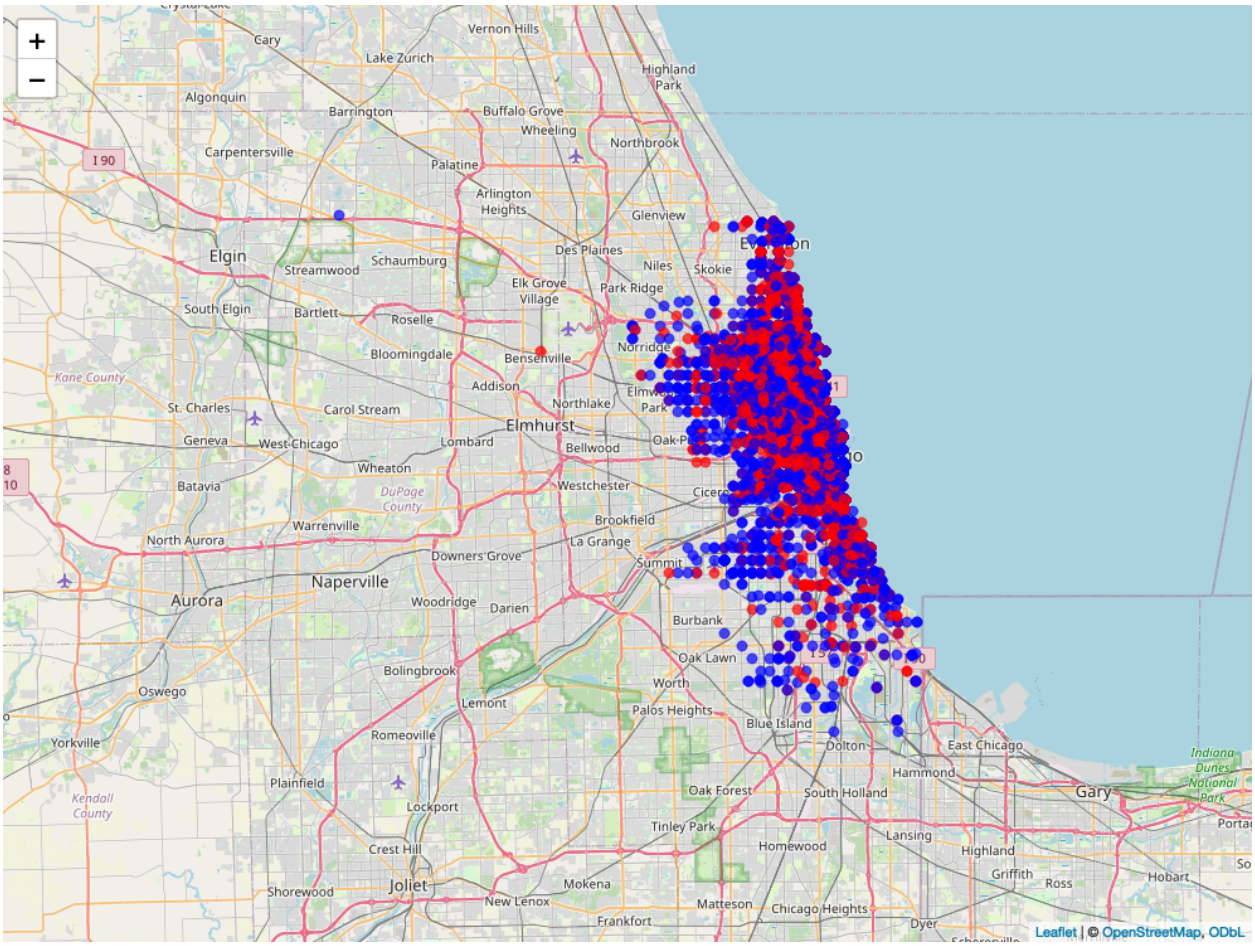

Figure 8: Map of the ending points of bike rides: blue dots are for casual riders and red dots are for members

6 Act

6.1 What is the final conclusion based on the analysis?

Annual members tend to have more bike rides than casual riders, but with shorter and more consistent ride durations throughout the week. Member bike rides are more focused on areas in downtown Chicago.

Casual riders tend to have longer bike rides than annual members, but the ride duration is not consistent for every day of the week. Casual riders ride bikes less frequently than annual members. Casual riders have their bike rides more spread out across the suburban areas around Chicago than annual members.

6.2 How could the team and business apply the insights?

From these insights, the team will find out why casual riders would buy memberships, find out how digital media can convert casual riders to annual members, then design a marketing strategy according to the new insights.

6.3 What next steps would the data analyst or the stakeholders take based on the findings?

Determine what led to the current members to buy their memberships. Then figure out possible reasons for casual riders to buy memberships.

6.4 Is there additional data that could be used to expand on the findings?

User ids could give a better idea on which bike rides were used by the same biker and the purpose of the frequent rides.

6.5 The top three recommendations based on the analysis.

Determine the main reasons for current members purchasing their memberships.

Determine how casual riders can benefit from becoming members.

Find out which locations are more frequently visited by Cyclistic bikers that can be applied to the marketing strategy.

References

[1] *About Divvy.* https://divvybikes.com/about.

[2] *Divvy Data License Agreement.* https://divvybikes.com/data-license-agreement.

[3] *divvy-tripdata.* https://divvy-tripdata.s3.amazonaws.com/index.html.

[4] *Google Data Analytics Certificate Program Module 8 Case Study 1 R Markdown file.* CS1_markdown. Rmd.

[5] *Lyft Privacy Policy.* https://www.lyft.com/privacy.

[6] *Google Data Analytics Certificate Program training materials.* 2023.