Cyclistic R Project

Manuel Chavez

2023-06-24

Loading Packages ——————————————————

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.0     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.1     ✓ tibble    3.2.0
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force
all conflicts to become errors

library(scales)

##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

Importing Datasets ——————————————————

```
jan22 <- read_csv("csv_files/202201-divvy-tripdata.csv")

## Rows: 103770 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────

## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```r
feb22 <- read_csv("csv_files/202202-divvy-tripdata.csv")
```

```
## Rows: 115609 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```r
mar22 <- read_csv("csv_files/202203-divvy-tripdata.csv")
```

```
## Rows: 284042 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```r
apr22 <- read_csv("csv_files/202204-divvy-tripdata.csv")
```

```
## Rows: 371249 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```r
may22 <- read_csv("csv_files/202205-divvy-tripdata.csv")
```

```
## Rows: 634858 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```r
jun22 <- read_csv("csv_files/202206-divvy-tripdata.csv")
```

```
## Rows: 769204 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```r
jul22 <- read_csv("csv_files/202207-divvy-tripdata.csv")
```

```
## Rows: 823488 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name,
s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
aug22 <- read_csv("csv_files/202208-divvy-tripdata.csv")
```

```
## Rows: 785932 Columns: 13
## ── Column specification
```

```
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
sep22 <- read_csv("csv_files/202209-divvy-tripdata.csv")
```

```
## Rows: 701339 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
oct22 <- read_csv("csv_files/202210-divvy-tripdata.csv")
```

```
## Rows: 558685 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
nov22 <- read_csv("csv_files/202211-divvy-tripdata.csv")
```

```
## Rows: 337735 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
dec22 <- read_csv("csv_files/202212-divvy-tripdata.csv")
```

```
## Rows: 181806 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id,
end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

Inspecting column names and datatypes ——————————————-

All month data frames were inspected using the glimpse() function. The July df is shown here. The July df was removed because its variable data types did not match the other month's data types.

```
glimpse(jul22)
```

```
## Rows: 823,488
## Columns: 13
## $ ride_id            <chr> "954144C2F67B1932", "292E027607D218B6",
"5776585258…
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike",
"cl…
## $ started_at         <chr> "7/5/2022 8:12", "7/26/2022 12:53", "7/3/2022
13:58…
## $ ended_at           <chr> "7/5/2022 8:24", "7/26/2022 12:55", "7/3/2022
14:06…
## $ start_station_name <chr> "Ashland Ave & Blackhawk St", "Buckingham
Fountain …
## $ start_station_id   <chr> "13224", "15541", "15541", "15541",
"TA1307000117",…
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th
St"…
## $ end_station_id     <chr> "KA1503000043", "623", "623", "TA1307000164",
"TA13…
## $ start_lat          <dbl> 41.90707, 41.86962, 41.86962, 41.86962,
41.89147, 4…
## $ start_lng          <dbl> -87.66725, -87.62398, -87.62398, -87.62398, -
```

```
87.626…
## $ end_lat            <dbl> 41.88918, 41.87277, 41.87277, 41.79526,
41.93625, 4…
## $ end_lng            <dbl> -87.63851, -87.62398, -87.62398, -87.59647, -
87.652…
## $ member_casual      <chr> "member", "casual", "casual", "casual",
"member", "…

rm(jul22)

# July df imported again with appropriate variable data types
jul22 <- read_csv("csv_files/202207-divvy-tripdata.csv",
                  col_types = cols(started_at = col_datetime(format =
"%m/%d/%Y %H:%M"),
                                   ended_at = col_datetime(format = "%m/%d/%Y
%H:%M")))
# seconds were not included as all observations result in NA values
```

Data Cleaning

```
# Combine all data frames
all_2022 <- bind_rows(jan22, feb22, mar22, apr22, may22, jun22,
                      jul22, aug22, sep22, oct22, nov22, dec22)

# Remove columns with id in name
all_2022_v2 <- all_2022 %>%
  select(-ends_with('id'))

# Remove duplicates
all_2022_v3 <- all_2022_v2 %>%
  distinct(started_at, ended_at, .keep_all = TRUE)

# Add a ride_length column
all_2022_v4 <- all_2022_v3 %>%
  mutate(
    ride_length = ended_at - started_at,
    .before = start_station_name)

# Change ride_length to numeric data type
all_2022_v4$ride_length = as.numeric(as.difftime(all_2022_v4$ride_length))

# Remove negative ride lengths
all_2022_v4 %>%
  arrange(ride_length)

## # A tibble: 5,397,143 × 11
##    rideable_type started_at          ended_at            ride_…¹ start…²
end_s…³
##    <chr>         <dttm>              <dttm>                <dbl> <chr>
```

```
<chr>
##  1 electric_bike 2022-09-28 11:04:32 2022-09-21 06:31:11 -621201 Cornel…
<NA>
##  2 classic_bike  2022-10-13 14:42:10 2022-10-13 11:53:28  -10122 Wilton…
Wilton…
##  3 electric_bike 2022-06-07 19:23:03 2022-06-07 17:05:38   -8245 <NA>
<NA>
##  4 electric_bike 2022-06-07 19:15:39 2022-06-07 17:05:37   -7802 <NA>
Kostne…
##  5 electric_bike 2022-06-07 19:14:47 2022-06-07 17:05:42   -7745 Base -… W
Armi…
##  6 electric_bike 2022-06-07 19:14:46 2022-06-07 17:07:45   -7621 W Armi… W
Armi…
##  7 electric_bike 2022-06-07 19:11:33 2022-06-07 17:05:24   -7569 <NA>
<NA>
##  8 electric_bike 2022-06-07 19:13:27 2022-06-07 17:07:57   -7530 <NA>
<NA>
##  9 electric_bike 2022-06-07 19:06:49 2022-06-07 17:09:43   -7026 <NA>
<NA>
## 10 electric_bike 2022-06-07 18:47:01 2022-06-07 17:05:41   -6080 <NA>
<NA>
## # … with 5,397,133 more rows, 5 more variables: start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>,
and
## #   abbreviated variable names ¹ride_length, ²start_station_name,
## #   ³end_station_name

all_2022_v5 <- all_2022_v4 %>%
  filter(ride_length >= 0)

# Find day of week from starting_at date
all_2022_v6 <- all_2022_v5 %>%
  mutate(day_of_week = (wday(all_2022_v5$started_at,label = TRUE)))

# Arrange, rename, and relocate columns for clarity
all_2022_v7 <- all_2022_v6 %>%
  arrange(started_at) %>%
  relocate(member_casual, rideable_type, day_of_week) %>%
  rename(member_type = member_casual, bike_type = rideable_type)

# Adding and removing additional columns
all_2022_v8 <-  all_2022_v7 %>%
  mutate(
    weekday = (wday(all_2022_v7$started_at,label = TRUE)),
    month = (month(all_2022_v7$started_at, label = TRUE)),
    hour_of_day = (hour(all_2022_v7$started_at))
  ) %>%
  select(-day_of_week, -start_station_name, -end_station_name,
         -start_lat, -start_lng, -end_lat, -end_lng)
```

```r
# Remove excess data frames
rm(jan22, feb22, mar22, apr22, may22, jun22,
   jul22, aug22, sep22, oct22, nov22, dec22,
   all_2022, all_2022_v2, all_2022_v3, all_2022_v4,
   all_2022_v5, all_2022_v6)
```

Data Analysis ————————————————————————

```r
# How many rides for the entire year by member type?
all_2022_v8 %>%
  count(member_type) %>%
  group_by(member_type) %>%
  arrange(n)

## # A tibble: 2 × 2
## # Groups:   member_type [2]
##   member_type       n
##   <chr>         <int>
## 1 casual      2190629
## 2 member      3206415

# How many rides for the entire year by bike type?
all_2022_v7 %>%
  count(member_type, bike_type) %>%
  group_by(bike_type, member_type) %>%
  arrange(n)

## # A tibble: 5 × 3
## # Groups:   bike_type, member_type [5]
##   member_type bike_type           n
##   <chr>       <chr>           <int>
## 1 casual      docked_bike    169964
## 2 casual      classic_bike   843194
## 3 casual      electric_bike 1177471
## 4 member      electric_bike 1564272
## 5 member      classic_bike  1642143

# What is the avg ride length (in seconds) per member type?
all_2022_v7 %>%
  group_by(member_type) %>%
  summarise(avg_ride_length = mean(ride_length))

## # A tibble: 2 × 2
##   member_type avg_ride_length
##   <chr>                 <dbl>
## 1 casual                1800.
## 2 member                 769.

# How many rides per month by member type?
all_2022_v8 %>%
  count(month, member_type) %>%
  group_by(month, member_type) %>%
  #filter(member_type== "member") %>%
  arrange(desc(n))
```

```
## # A tibble: 24 × 3
## # Groups:   month, member_type [24]
##    month member_type      n
##    <ord> <chr>        <int>
##  1 Aug   member      426926
##  2 Sep   member      404566
##  3 Jun   member      400097
##  4 Jun   casual      369005
##  5 Aug   casual      358867
##  6 May   member      354398
##  7 Oct   member      349656
##  8 Sep   casual      296654
##  9 May   casual      280383
## 10 Jul   member      278563
## # … with 14 more rows
```

```
# How many rides per weekday by member type?
all_2022_v8 %>%
  count(weekday, member_type) %>%
  group_by(weekday, member_type) %>%
  #filter(member_type== "member") %>%
  arrange(desc(n))
```

```
## # A tibble: 14 × 3
## # Groups:   weekday, member_type [14]
##    weekday member_type      n
##    <ord>   <chr>        <int>
##  1 Thu     member      510454
##  2 Wed     member      502978
##  3 Tue     member      498902
##  4 Mon     member      457424
##  5 Fri     member      447650
##  6 Sat     casual      441942
##  7 Sat     member      420021
##  8 Sun     member      368986
##  9 Sun     casual      365280
## 10 Fri     casual      316613
## 11 Thu     casual      292517
## 12 Mon     casual      264646
## 13 Wed     casual      259637
## 14 Tue     casual      249994
```

```
# How many rides per hour of day by member type?
all_2022_v8 %>%
  count(hour_of_day, member_type) %>%
  group_by(hour_of_day, member_type) %>%
  #filter(member_type== "member") %>%
  arrange(desc(n))
```

```
## # A tibble: 48 × 3
## # Groups:   hour_of_day, member_type [48]
##    hour_of_day member_type      n
##          <int> <chr>        <int>
##  1          17 member      330229
##  2          16 member      277835
##  3          18 member      269149
##  4          15 member      212531
##  5          17 casual      204436
##  6           8 member      197586
##  7          19 member      195906
##  8          16 casual      185021
##  9          18 casual      183457
## 10          12 member      179948
## # … with 38 more rows
```

```
# What station was the most popular by member type?
# Creating first df
start_station_counts_by_member <- all_2022_v7 %>%
  group_by(member_type) %>%
  count(start_station_name) %>%
  rename(station_name = start_station_name) %>%
  drop_na() %>%
  arrange(desc(n)) %>%
  filter(n> 19000)

# Creating second df
end_station_counts_by_member <- all_2022_v7 %>%
  group_by(member_type) %>%
  count(end_station_name) %>%
  rename(station_name = end_station_name) %>%
  drop_na() %>%
  arrange(desc(n)) %>%
  filter(n> 19000)

# Joining both data frames
joined_station_counts <- merge(start_station_counts_by_member,
                               end_station_counts_by_member,
                               by = "station_name") %>%
  select(station_name, member_type.x, n.x, n.y) %>%
  mutate(total_count = n.x + n.y) %>%
  rename(member_type = member_type.x) %>%
  arrange(desc(total_count))

# What is the avg ride length per month?
all_2022_v7 %>%
  mutate(month = (month(all_2022_v7$started_at, label = TRUE, abbr = FALSE)))
```

```
%>%
  group_by(month) %>%
  summarise(avg_bike_length = mean(ride_length)) %>%
  arrange(desc(avg_bike_length))
```

```
## # A tibble: 12 × 2
##    month     avg_bike_length
##    <ord>               <dbl>
##  1 July                1544.
##  2 June                1361.
##  3 May                 1266.
##  4 August              1240.
##  5 September           1159.
##  6 March               1110.
##  7 April               1058.
##  8 October             1041.
##  9 January              916.
## 10 February             854.
## 11 November             850.
## 12 December             810.
```

```
# What is the avg ride_length per weekday?
all_2022_v7 %>%
  group_by(day_of_week) %>%
  summarise(avg_bike_length = mean(ride_length)) %>%
  arrange(desc(avg_bike_length))
```

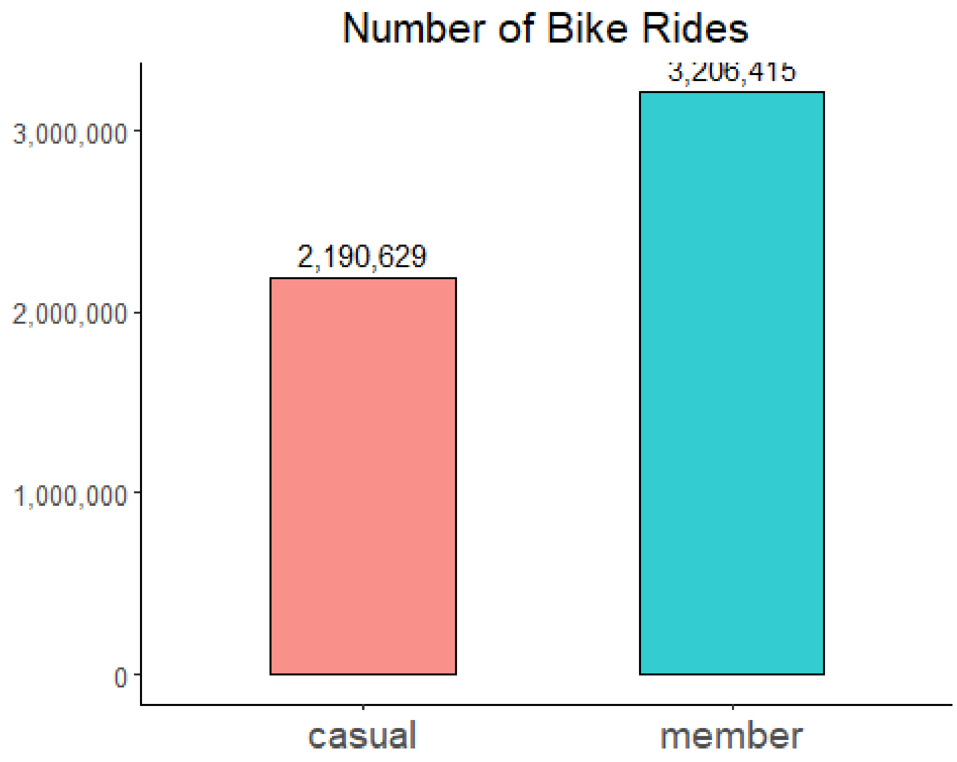
```
## # A tibble: 7 × 2
##   day_of_week avg_bike_length
##   <ord>                 <dbl>
## 1 Sun                   1476.
## 2 Sat                   1454.
## 3 Fri                   1163.
## 4 Mon                   1127.
## 5 Thu                   1048.
## 6 Tue                   1021.
## 7 Wed                   1003.
```

```
# What is the avg ride_length per hour of day?
all_2022_v7 %>%
  mutate(hour_of_day = (hour(all_2022_v7$started_at))) %>%
  group_by(hour_of_day) %>%
  summarise(avg_bike_length = mean(ride_length)) %>%
  arrange(desc(avg_bike_length))
```

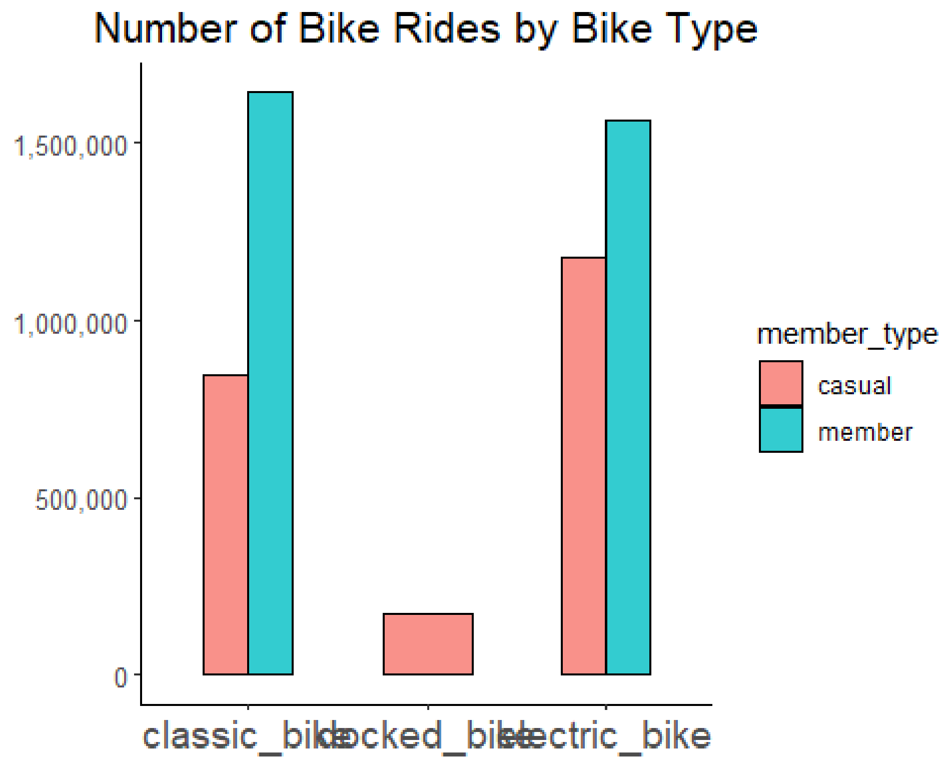
```
## # A tibble: 24 × 2
##    hour_of_day avg_bike_length
##          <int>           <dbl>
##  1           2           1796.
##  2           3           1794.
##  3           1           1680.
##  4           4           1490.
##  5          23           1408.
##  6           0           1386.
##  7          14           1307.
##  8          15           1293.
##  9          11           1280.
## 10          13           1274.
## # … with 14 more rows
```

Data Visualization ———————————————————————

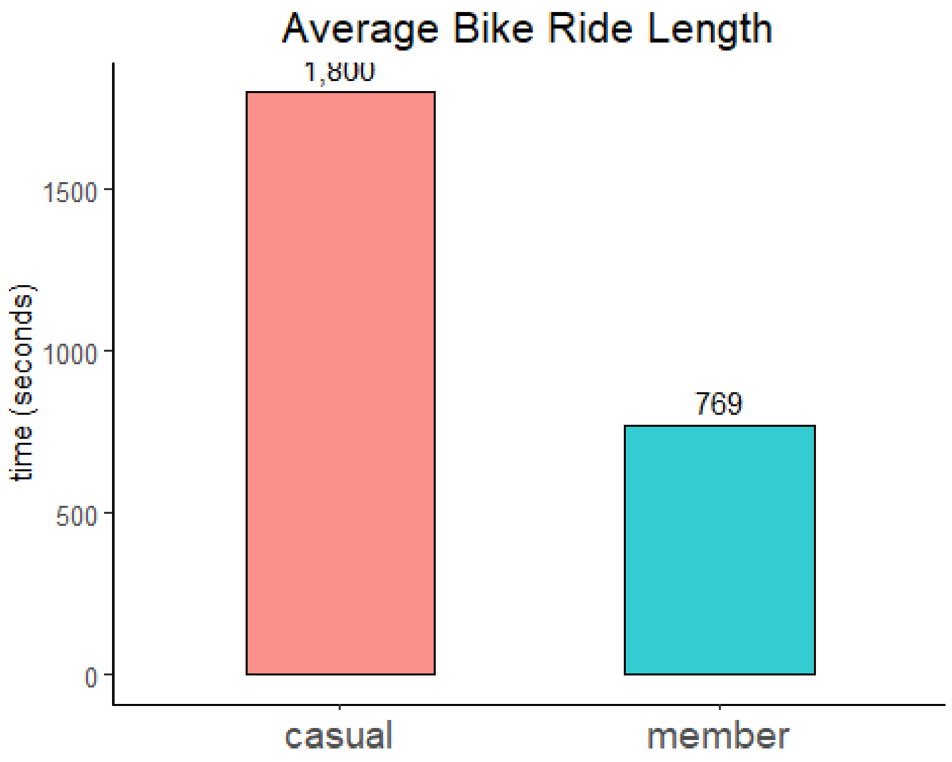
```r
# Plotting Number of Rides by member type
plot1 <- all_2022_v8 %>%
  count(member_type) %>%
  group_by(member_type) %>%
  arrange(n) %>%
  ggplot(aes(x = member_type, y = n, fill = member_type))+
    geom_col(color = "black", width = 0.5, alpha = 0.8)+
  labs(
    title = "Number of Bike Rides",
    x = NULL,
    y = NULL
  )+
  scale_y_continuous(labels = label_comma())+
  geom_text(
    aes(label = comma(n)),
    color = "black", size = 4, vjust = -.5
  )+
  theme_classic()+
  theme(
    plot.title = element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 14),
    axis.text.y = element_text(size = 10),
    legend.position = "none"
  )
plot(plot1)
```



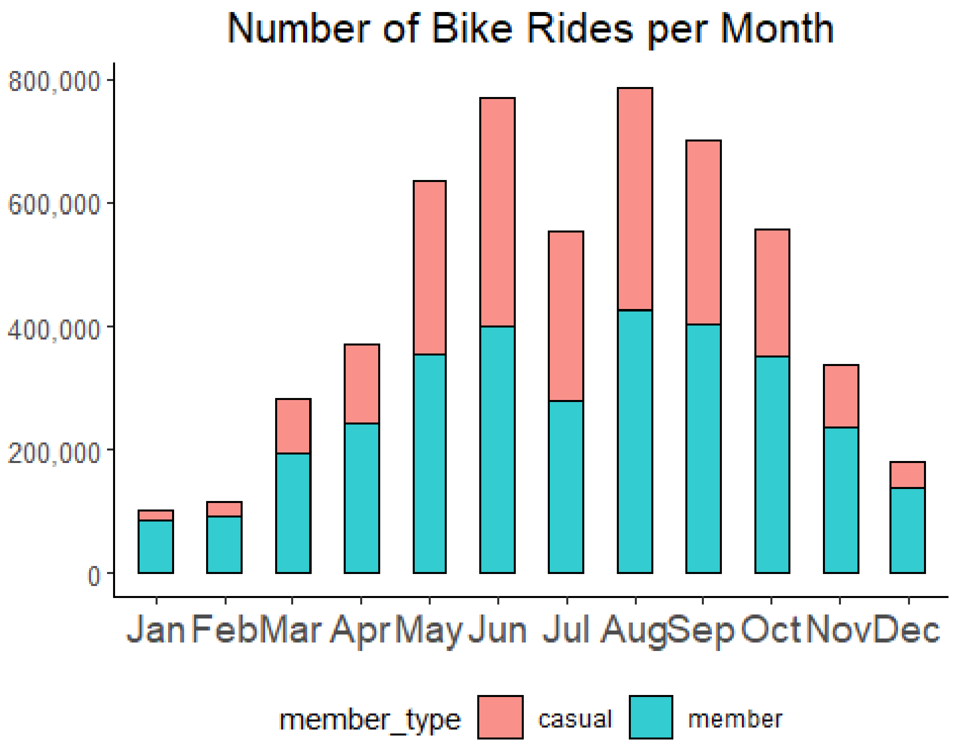
```
# Plotting Number of Bike Rides by bike type
plot2 <- all_2022_v8 %>%
  group_by(member_type) %>%
  count(bike_type) %>%
  ggplot(aes(x = bike_type, y = n, fill = member_type))+
  geom_col(color = "black", position = "dodge", width = 0.5, alpha = 0.8)+
  labs(
    title = "Number of Bike Rides by Bike Type",
    x = NULL,
    y = NULL
  )+
  scale_y_continuous(labels = label_comma())+
  theme_classic()+
  theme(
    plot.title = element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 14),
    axis.text.y = element_text(size = 10)
  )
plot(plot2)
```



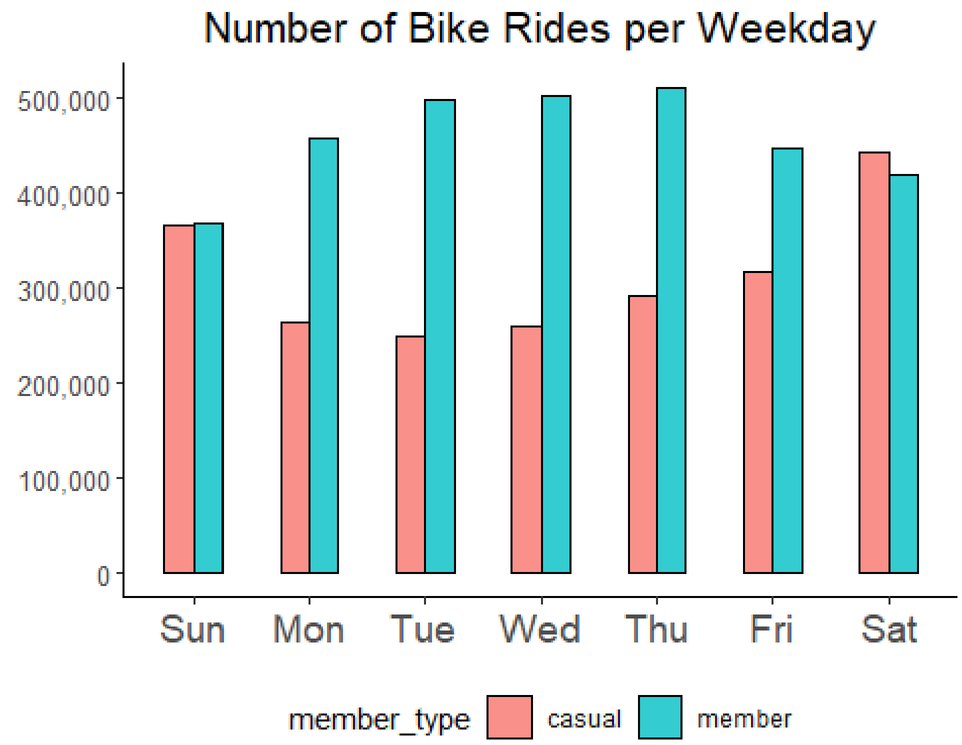
```r
# Plotting Avg Ride Length by member type
plot3 <- all_2022_v8 %>%
  group_by(member_type) %>%
  summarise(avg_ride_length = mean(ride_length)) %>%
  ggplot(aes(x = member_type, y = avg_ride_length, fill = member_type))+
    geom_col(color = "black", width = 0.5, alpha = 0.8)+
  labs(
    title = "Average Bike Ride Length",
    x = NULL,
    y = "time (seconds)"
  )+
  geom_text(
    aes(label = comma(avg_ride_length)),
    color = "black", size = 4, vjust = -.5
  )+
  theme_classic()+
  theme(
    plot.title= element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 14),
    axis.text.y = element_text(size = 10),
    legend.position = "none"
  )
plot(plot3)
```



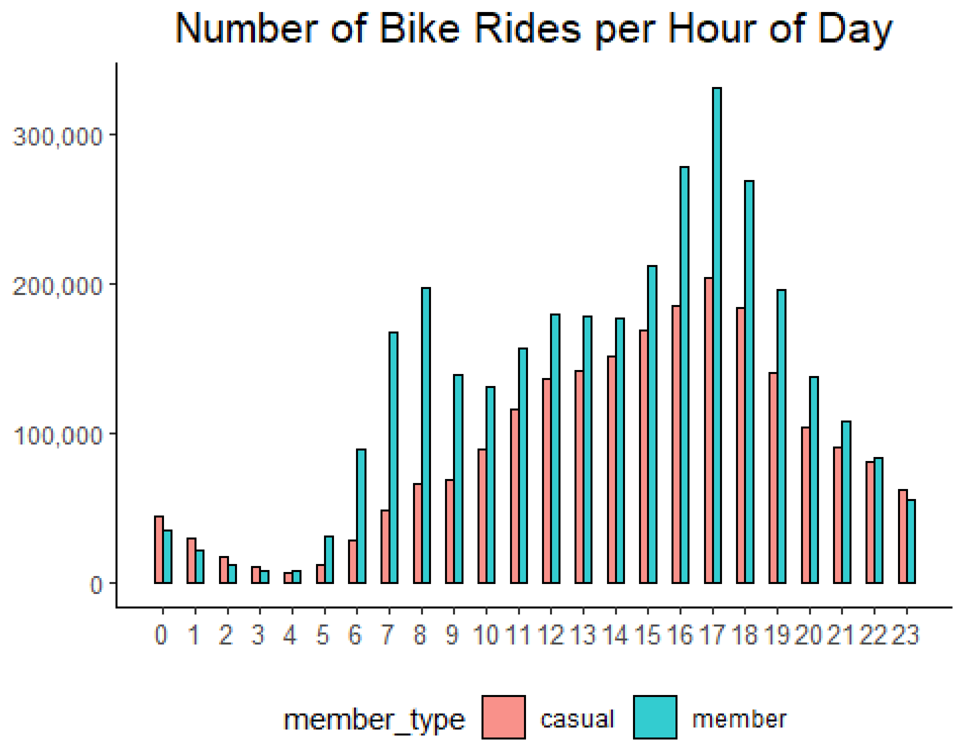
```
# Plotting Number of Bike Rides per Month
plot4 <- all_2022_v8 %>%
  group_by(member_type) %>%
  count(month) %>%
  ggplot(aes(x = month, y = n, fill = member_type))+
  geom_col(color= "black",  width=0.5, alpha= 0.8)+
  labs(
    title = "Number of Bike Rides per Month",
    x = NULL,
    y = NULL
  )+
  scale_y_continuous(labels = label_comma())+
  theme_classic()+
  theme(
    plot.title= element_text(size= 16, hjust = 0.5),
    axis.text.x = element_text(size=14),
    axis.text.y = element_text(size=10),
    legend.position= "bottom"
  )
plot(plot4)
```



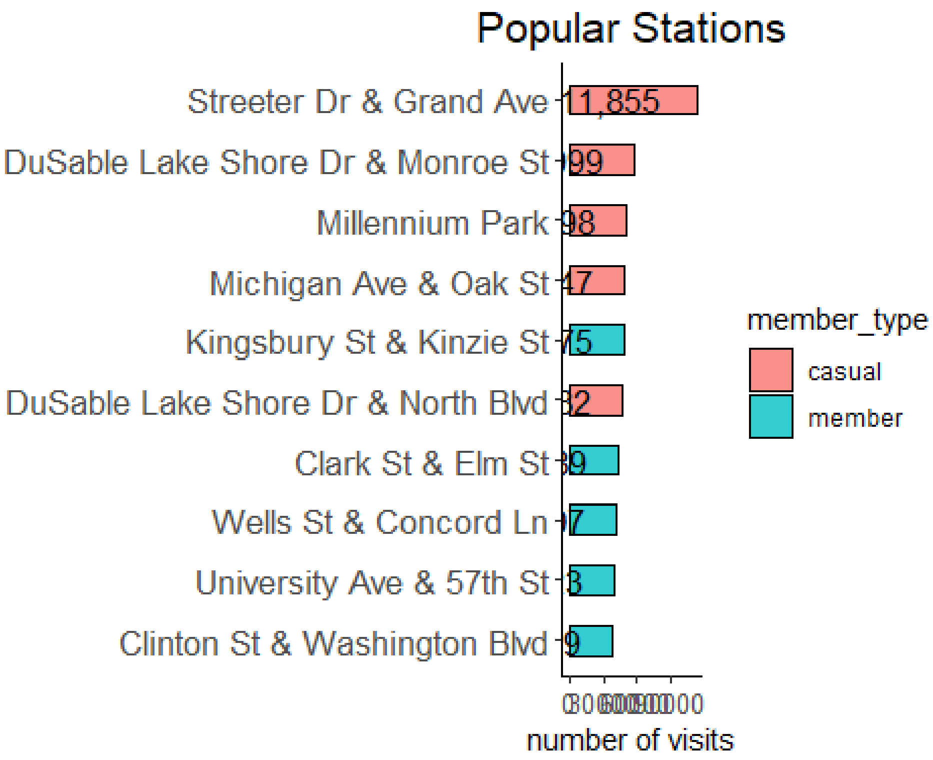
```
plot5 <- all_2022_v8 %>%
  group_by(member_type) %>%
  count(weekday) %>%
  ggplot(aes(x = weekday, y = n, fill = member_type))+
  geom_col(color= "black",  width = 0.5, alpha= 0.8, position = "dodge")+
  labs(
    title = "Number of Bike Rides per Weekday",
    x = NULL,
    y = NULL
  )+
  scale_y_continuous(labels = label_comma())+
  theme_classic()+
  theme(
    plot.title = element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 14),
    axis.text.y = element_text(size = 10),
    legend.position = "bottom"
  )
plot(plot5)
```



```r
plot6 <- all_2022_v8 %>%
  group_by(member_type) %>%
  count(hour_of_day) %>%
  ggplot(aes(x = hour_of_day, y = n, fill = member_type))+
  geom_col(color= "black",  width =0.5, alpha = 0.8, position = "dodge")+
  labs(
    title = "Number of Bike Rides per Hour of Day",
    x = NULL,
    y = NULL
  )+
  scale_y_continuous(labels = label_comma())+
  scale_x_continuous(breaks = pretty(all_2022_v8$hour_of_day, n = 20))+ #
creates ticks marks for all hours
  theme_classic()+
  theme(
    plot.title = element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 10),
    legend.position = "bottom"
  )
plot(plot6)
```



```r
plot7 <- joined_station_counts %>%
  select(station_name, member_type, total_count) %>%
  arrange(desc(total_count)) %>%
  ggplot(aes(x = fct_reorder(station_name, total_count),
             y = total_count, fill = member_type))+
  geom_col(color = "black",  width = 0.5, alpha = 0.8)+
  labs(
    title = "Popular Stations",
    x = NULL,
    y = "number of visits"
  )+
  geom_text(
    aes(label = comma(total_count)),
    color = "black", size = 4.5, vjust = 0.5, hjust = 1.3
  )+
  theme_classic()+
  theme(
    plot.title = element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 13),
    legend.position = "right"
  )+
  coord_flip()
plot(plot7)
```


Popular Stations
Streeter Dr & Grand Ave
DuSable Lake Shore Dr & Monroe St
Millennium Park
Michigan Ave & Oak St
Kingsbury St & Kinzie St
DuSable Lake Shore Dr & North Blvd
Clark St & Elm St
Wells St & Concord Ln
University Ave & 57th St
Clinton St & Washington Blvd
11,855
99
98
17
75
32
9
7
3
9
number of visits
member_type
🟥 casual
🟦 member