Cyclistic Analysis

Set up

Load of libraries:

```
library("tidyverse")
library("lubridate")
library(dplyr)
library(knitr)
library(geosphere)
library(mapview)
```

Here we load all the different CSV into a single data frame.

```
df <- list.files(path = "./dataset", pattern = "*.csv", full.names = TRUE) %>%
  lapply(read_csv) %>%
  bind_rows
```

First we want to see some general stats of the data.

```
head(df)
```

```
## # A tibble: 6 x 13
##   ride_id        ridea~1 started_at          ended_at            start~2 start~3
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 0A1B623926EF4~ docked~ 2021-07-02 14:44:36 2021-07-02 15:19:58 Michig~ 13001
## 2 B2D5583A5A5E7~ classi~ 2021-07-07 16:57:42 2021-07-07 17:16:09 Califo~ 17660
## 3 6F264597DDBF4~ classi~ 2021-07-25 11:30:55 2021-07-25 11:48:45 Wabash~ SL-012
## 4 379B58EAB20E8~ classi~ 2021-07-08 22:08:30 2021-07-08 22:23:32 Califo~ 17660
## 5 6615C1E4EB08E~ electr~ 2021-07-28 16:08:06 2021-07-28 16:27:09 Califo~ 17660
## 6 62DC2B32872F9~ electr~ 2021-07-29 17:09:08 2021-07-29 17:15:00 Califo~ 17660
## # ... with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names 1: rideable_type,
## #   2: start_station_name, 3: start_station_id
## # i Use `colnames()` to see all variable names
```

```
summary(df)
```

```
##    ride_id          rideable_type        started_at
##  Length:5900385     Length:5900385     Min.   :2021-07-01 00:00:22
##  Class :character   Class :character   1st Qu.:2021-08-26 07:57:58
##  Mode  :character   Mode  :character   Median :2021-10-27 17:35:55
##                                        Mean   :2021-12-12 00:11:36
##                                        3rd Qu.:2022-04-25 13:41:23
##                                        Max.   :2022-06-30 23:59:58
##
##     ended_at                   start_station_name start_station_id
##  Min.   :2021-07-01 00:04:51   Length:5900385     Length:5900385
##  1st Qu.:2021-08-26 08:11:00   Class :character   Class :character
##  Median :2021-10-27 17:49:46   Mode  :character   Mode  :character
##  Mean   :2021-12-12 00:31:53
##  3rd Qu.:2022-04-25 13:57:17
##  Max.   :2022-07-13 04:21:06
##
##  end_station_name   end_station_id       start_lat       start_lng
```

```
##  Length:5900385     Length:5900385     Min.   :41.64   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :45.64   Max.   :-73.80
##
##     end_lat         end_lng       member_casual
##  Min.   :41.39   Min.   :-88.97   Length:5900385
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.17   Max.   :-87.49
##  NA's   :5374    NA's   :5374
```

Data aggregation

To better understand the data and arrive at a conclusion, we add some new columns.

```
df$ride_length <- df$ended_at - df$started_at
df$day_of_week <- wday(df$started_at)
df$year <- year(df$started_at)
df$month <- month(df$started_at)
df$day <- day(df$started_at)
df$hour <- hour(df$started_at)

df$ride_distance <- distGeo(matrix(c(df$start_lng, df$start_lat), ncol = 2), matrix(c(df$end_lng, df$en
df$ride_distance <- df$ride_distance / 1000
```

Consistency of features

For features like rideable_type and member_casual, we can check their unique values to verify if there are some strange labels. But as we can see, the values seem correct.

```
unique(df$rideable_type)
```

```
## [1] "docked_bike"   "classic_bike"  "electric_bike"
```

```
unique(df$member_casual)
```

```
## [1] "casual" "member"
```

As ride_id must be unique, we can check for duplicated values, but none was found.

```
n_occur <- data.frame(table(df$ride_id))
n_occur[n_occur$Freq > 1,]
```

```
## [1] Var1 Freq
## <0 rows> (or 0-length row.names)
```

Check the length for ride_id, it is always 16.

```
unique(nchar(as.character(df$ride_id)))
```

```
## [1] 16
```

Wrong data as rides shorter than a minute have been dropped from the dataset.

```
df <- subset(df, ride_length > 60)
```

Type transformation

Features like member_casual can be represented as a boolean type, so we can map this column and rename it.

```
df$is_member <- df$member_casual == 'member'
df$member_casual <- NULL
```

Dealing with null values

Null values:

```
colSums(is.na(df))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##             810999             810996             860669             860669
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               5352               5352
##        ride_length        day_of_week               year              month
##                  0                  0                  0                  0
##                day               hour      ride_distance          is_member
##                  0                  0               5352                  0
```

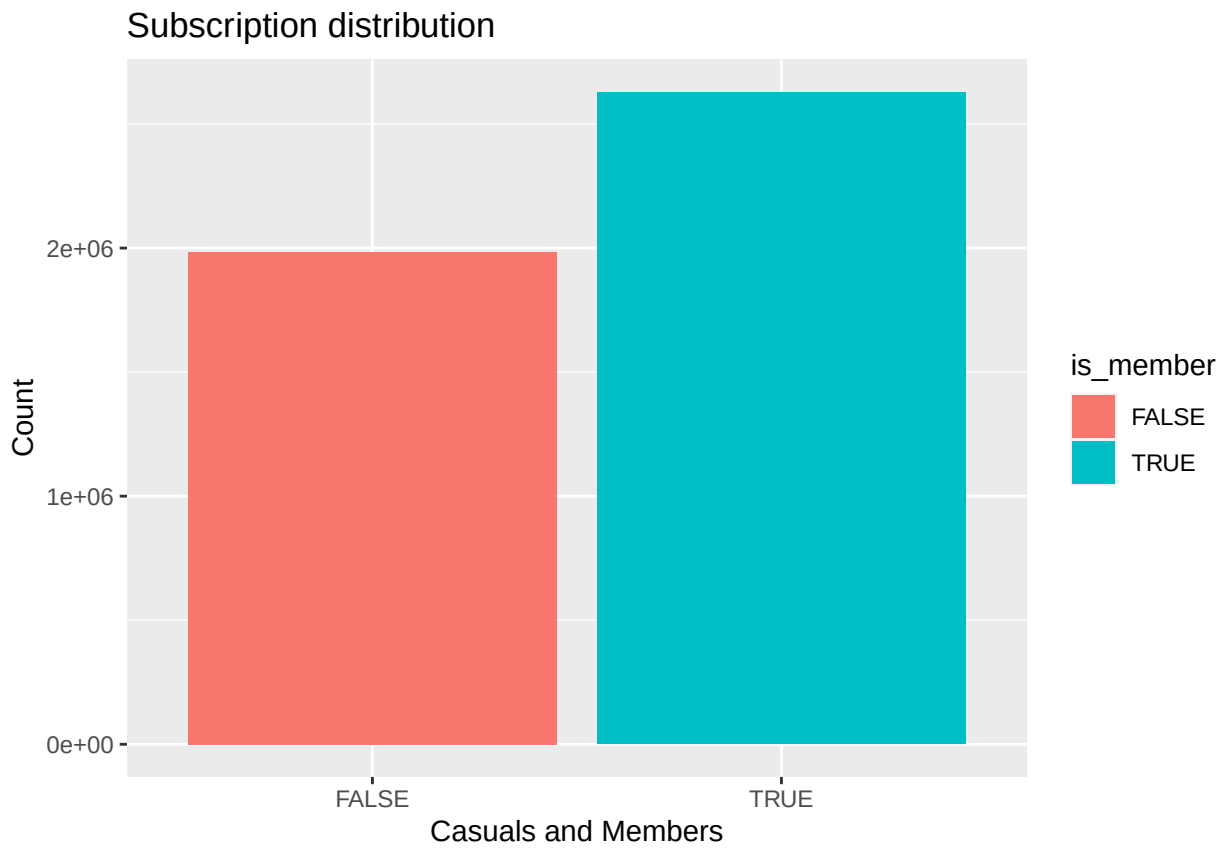
Dropping nulls found:

```
df <- subset(df, !is.na(df$start_station_id))
df <- subset(df, !is.na(df$end_station_id))
df <- subset(df, !is.na(df$start_station_name))
df <- subset(df, !is.na(df$end_station_name))
df <- subset(df, !is.na(df$end_lat))
df <- subset(df, !is.na(df$end_lng))
```

Analysis through plots

Comparative of members and casual riders. As we can see, there are slightly more subscribed clients than casuals.

```
ggplot(df, aes(is_member, fill=is_member)) +
    geom_bar() +
    labs(x="Casuals and Members", y="Count", title="Subscription distribution")
```




```
df %>%
    group_by(is_member) %>%
    summarize(count = n())
```

```
## # A tibble: 2 x 2
##   is_member   count
##   <lgl>       <int>
## 1 FALSE     1983915
## 2 TRUE      2627763
```

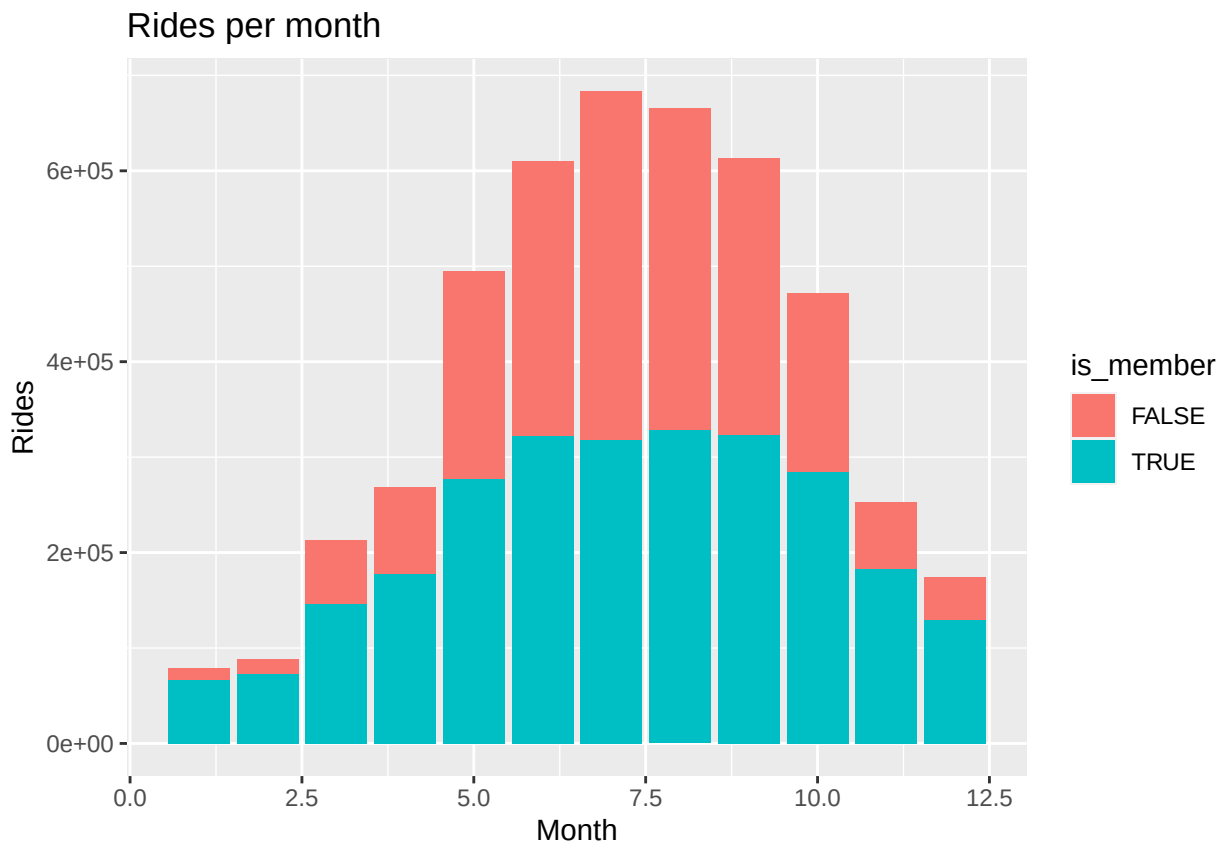
We can check which type of client does more kilometers:

```
df %>%
    group_by(is_member) %>%
    summarize(mean = mean(ride_distance))
```

```
## # A tibble: 2 x 2
##   is_member  mean
##   <lgl>     <dbl>
## 1 FALSE      2.23
## 2 TRUE       2.08
```

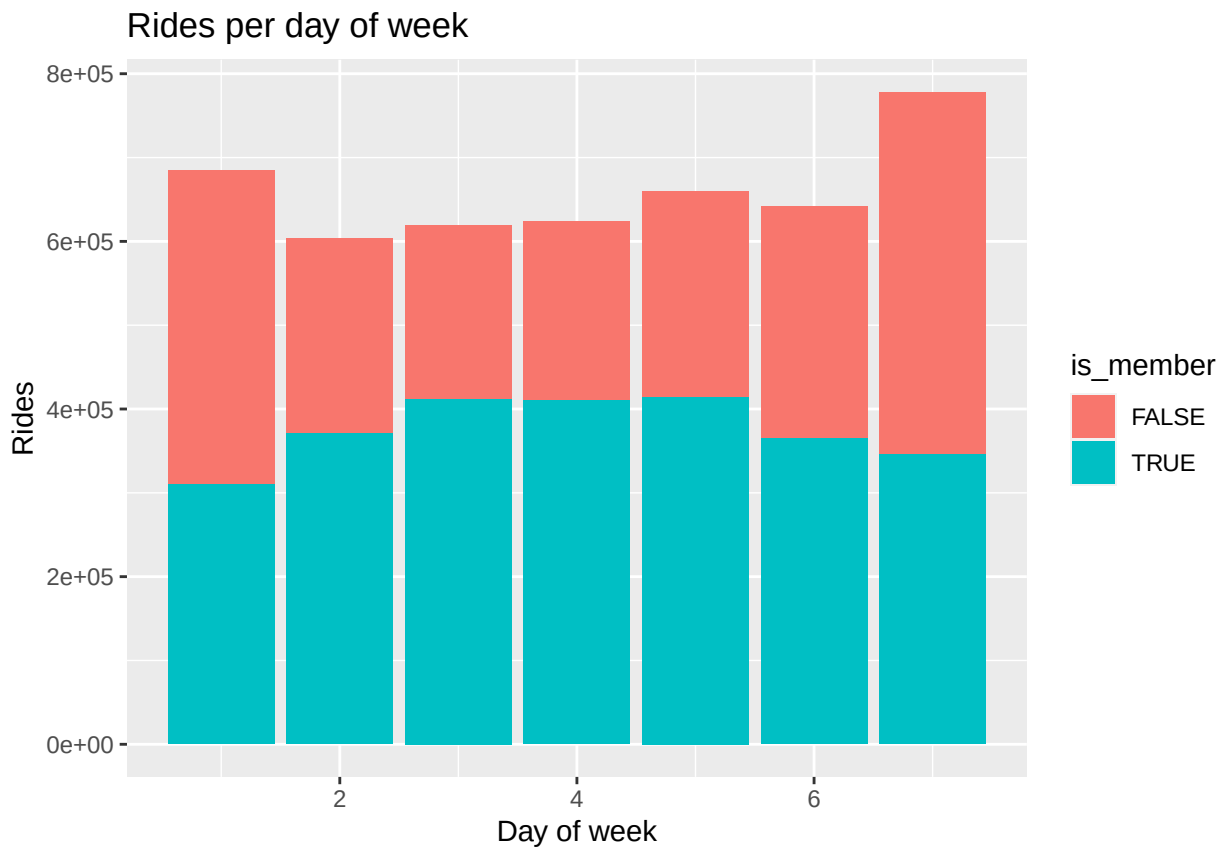
Distribution of rides through time.

```
df %>%
  ggplot(aes(month, fill=is_member)) +
    geom_bar() +
    labs(x="Month", y="Rides", title="Rides per month")
```



The months with more activity are the ones from May to October, this could be due to the good weather. It is also important to notice that casual riders are much more sensitive to this characteristic.

```
df %>%
  ggplot(aes(day_of_week, fill=is_member)) +
    geom_bar() +
    labs(x="Day of week", y="Rides", title="Rides per day of week")
```



As per the day of the week, Saturdays and Sundays are slightly more active than work days. We can see that on work days, there are many more members than casual riders. This can give us the hypothesis that clients that need to ride bikes between Monday to Friday (work, school...) found more attractive the membership status. And casual riders are more concentrated through the weekend.

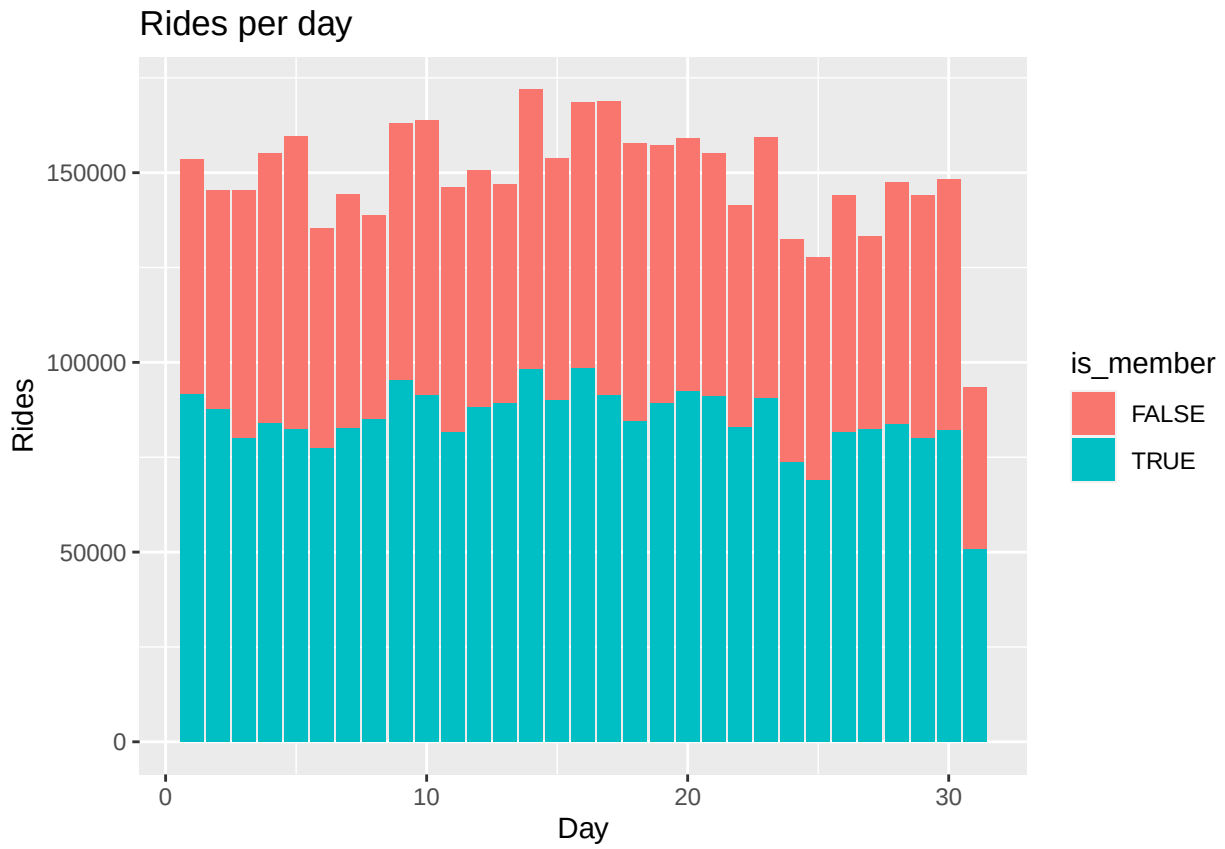
In addition, we can observe that on these days, the average time of the ride is bigger than the other days. People spend more time cycling at the weekend:

```
df %>%
    group_by(day_of_week) %>%
    summarize(mean = mean(ride_length))
```

```
## # A tibble: 7 x 2
##   day_of_week mean
##         <dbl> <drtn>
## 1           1 1463.6004 secs
## 2           2 1146.6151 secs
## 3           3  968.7724 secs
## 4           4  977.3685 secs
## 5           5 1032.2057 secs
## 6           6 1117.7515 secs
## 7           7 1417.2915 secs
```

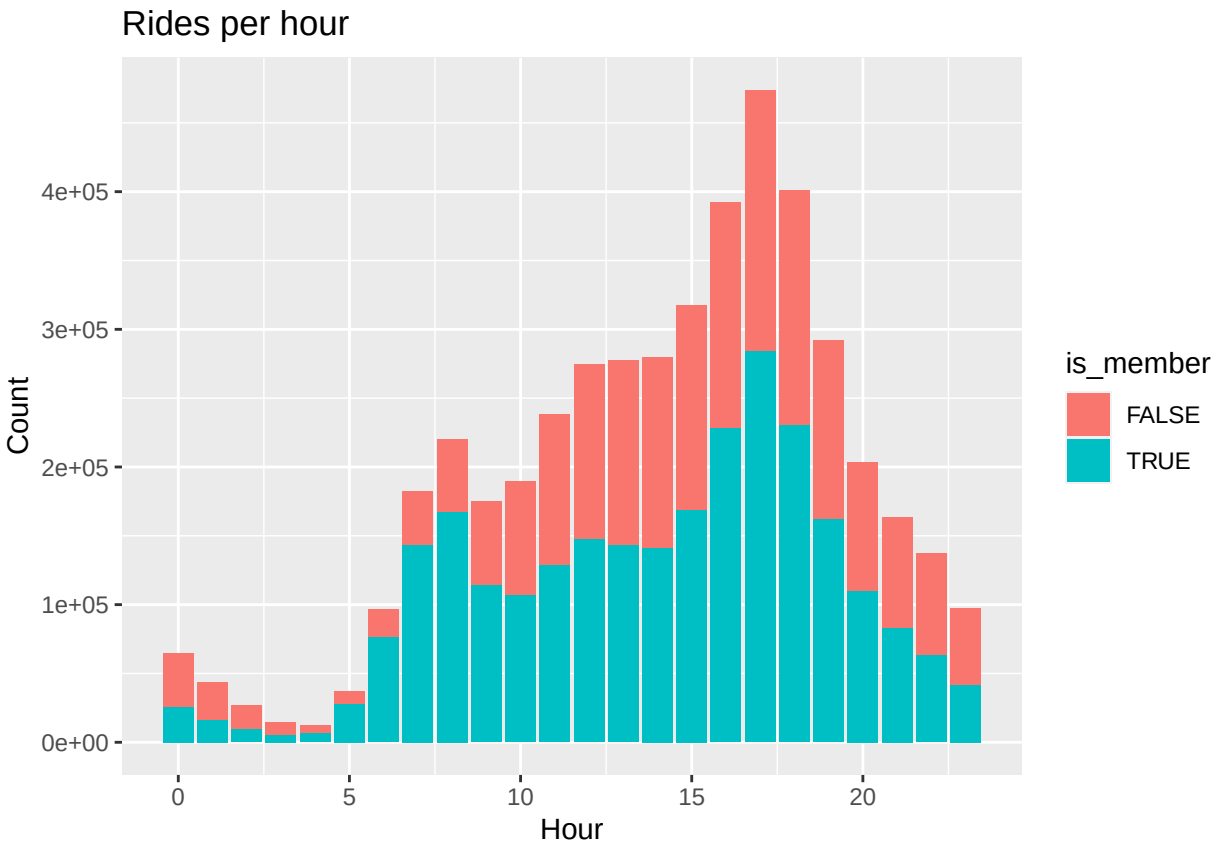
```
df %>%
    ggplot(aes(day, fill=is_member)) +
    labs(x="Day", y="Rides", title="Rides per day") +
```

```
    geom_bar()
```



If we inspect the days of the month with more rides, we can see that more or less the distribution is stable through all the days except the 31, but that is because half of the months don't have this day.

```
df %>%
    ggplot(aes(hour, fill=is_member)) +
    labs(x="Hour", y="Count", title="Rides per hour") +
    geom_bar()
```



The hours with more activity are from 8 AM to 7 PM, and there are important spikes with a high presence of members around those two hours, coinciding with the rush hours from work or school.

We can increase our scope in a hypothetical ad campaign if we focus the resources on those hours, and we can check as well the top 10 used ride stations, where we could arrive to more potential clients:

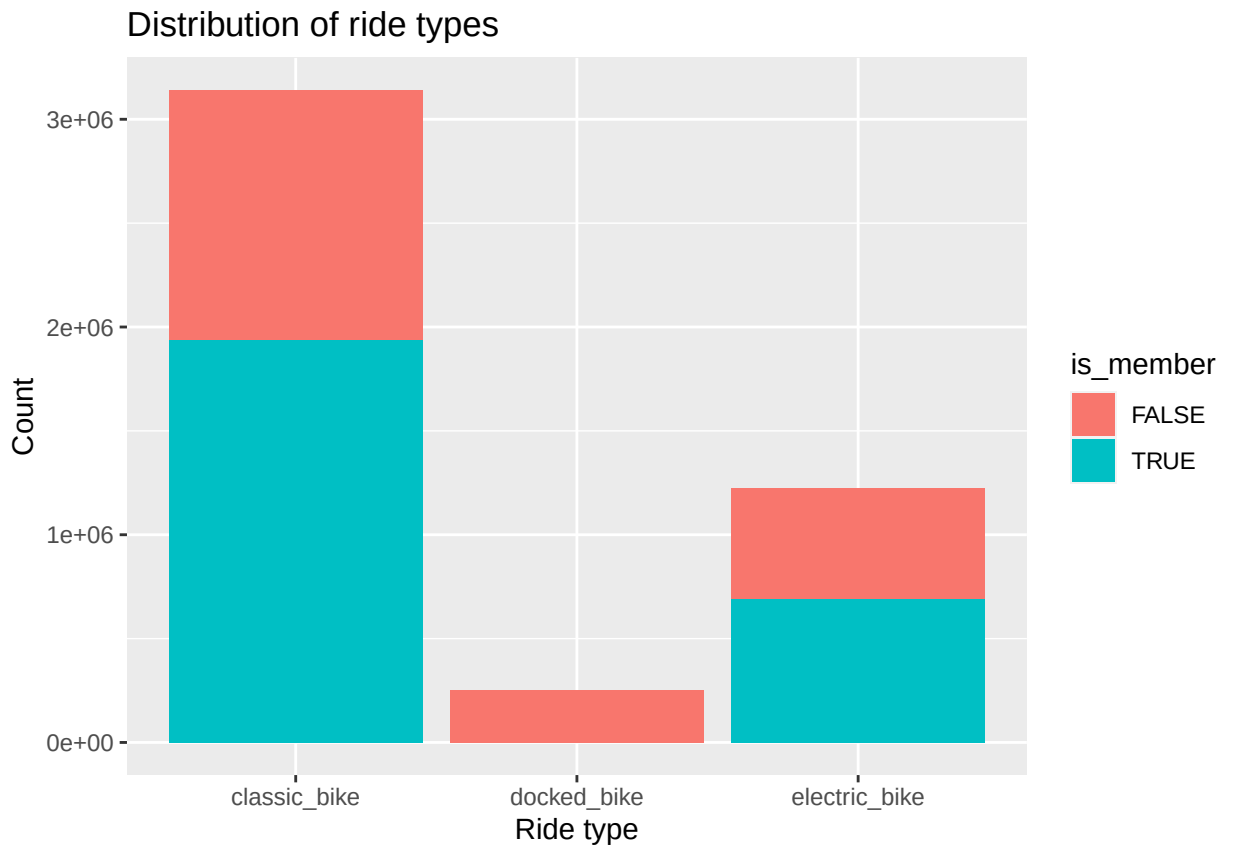
```
stations_rides <- df %>%
  group_by(start_station_name) %>%
  summarize(rides_per_station = n()) %>%
  arrange(desc(rides_per_station))

head(stations_rides, 10)
```

```
## # A tibble: 10 x 2
##    start_station_name                  rides_per_station
##    <chr>                                           <int>
##  1 Streeter Dr & Grand Ave                         78449
##  2 Michigan Ave & Oak St                           41211
##  3 Wells St & Concord Ln                           40732
##  4 DuSable Lake Shore Dr & North Blvd              38377
##  5 Millennium Park                                 37819
##  6 Clark St & Elm St                               37382
##  7 DuSable Lake Shore Dr & Monroe St               36184
##  8 Wells St & Elm St                               35023
##  9 Theater on the Lake                             34451
## 10 Kingsbury St & Kinzie St                        33983
```

As we have two types of bicycles (plus the docked ones), we can plot the distribution:

```
ggplot(df, aes(rideable_type, fill=is_member)) +
    geom_bar() +
    labs(x="Ride type", y="Count", title="Distribution of ride types")
```



Classic bikes are much more used than electric ones. And in this table, we can see the time spent on each type. Docked bikes are the most used, but this is because when they are docked the time still runs until they are parked at a proper station. Rides with classic bikes tend to be longer:

```
df %>%
    group_by(rideable_type) %>%
    summarize(mean = mean(ride_length))
```

```
## # A tibble: 3 x 2
##   rideable_type mean
##   <chr>         <drtn>
## 1 classic_bike  1072.4928 secs
## 2 docked_bike   3827.6853 secs
## 3 electric_bike  884.1244 secs
```

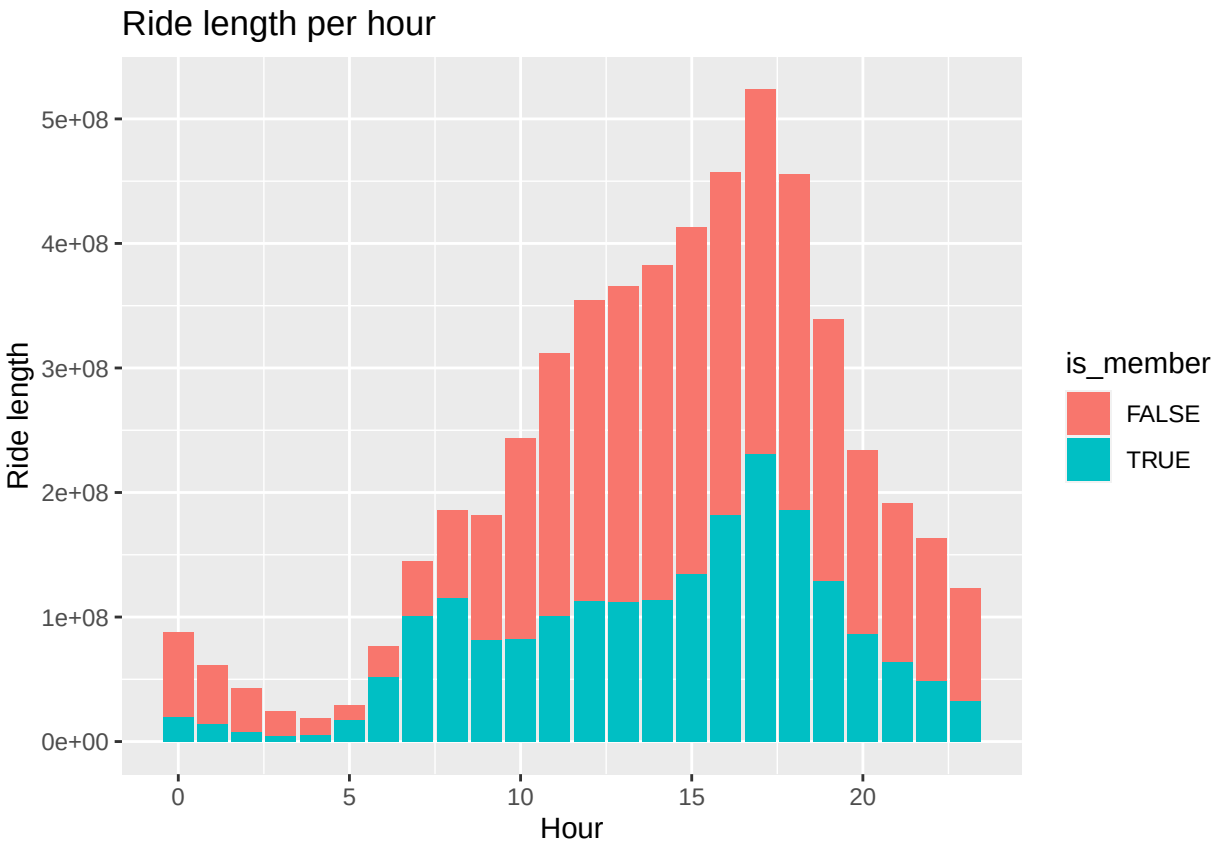
Rides with electric bikes are not only 21% faster, as we can see in the following table, but they also do an average of an extra 400m compared with clients that stick with classic bikes:

```
df %>%
    group_by(rideable_type) %>%
    summarize(mean = mean(ride_distance))
```

```
## # A tibble: 3 x 2
##   rideable_type  mean
##   <chr>         <dbl>
## 1 classic_bike   2.04
## 2 docked_bike    2.16
## 3 electric_bike  2.40
```

As for the length of rides, casual riders tend to spend more time than members. Here we can see the difference for each hour:

```
ggplot(data=df) +
  geom_bar(mapping = aes(x=hour, y=ride_length, fill=is_member), stat='identity') +
  labs(x="Hour", y="Ride length", title="Ride length per hour")
```



In general, members spend an average of 13 min for each ride and casual riders 28 min:

```
df_avg_length_rides <- df %>%
    group_by(is_member) %>%
    summarize(avg_ride_length_mins = mean(as.numeric(ride_length, units="mins")))

df_avg_length_rides
```

```
## # A tibble: 2 x 2
##   is_member avg_ride_length_mins
##   <lgl>                    <dbl>
## 1 FALSE                     28.4
## 2 TRUE                      12.8
```

Which are the favorite destinations of our clients? On the following map, we indicate the top 5 destination

stations:

```
stations_rides <- df %>%
  group_by(end_station_name) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
head(stations_rides, 10)
```

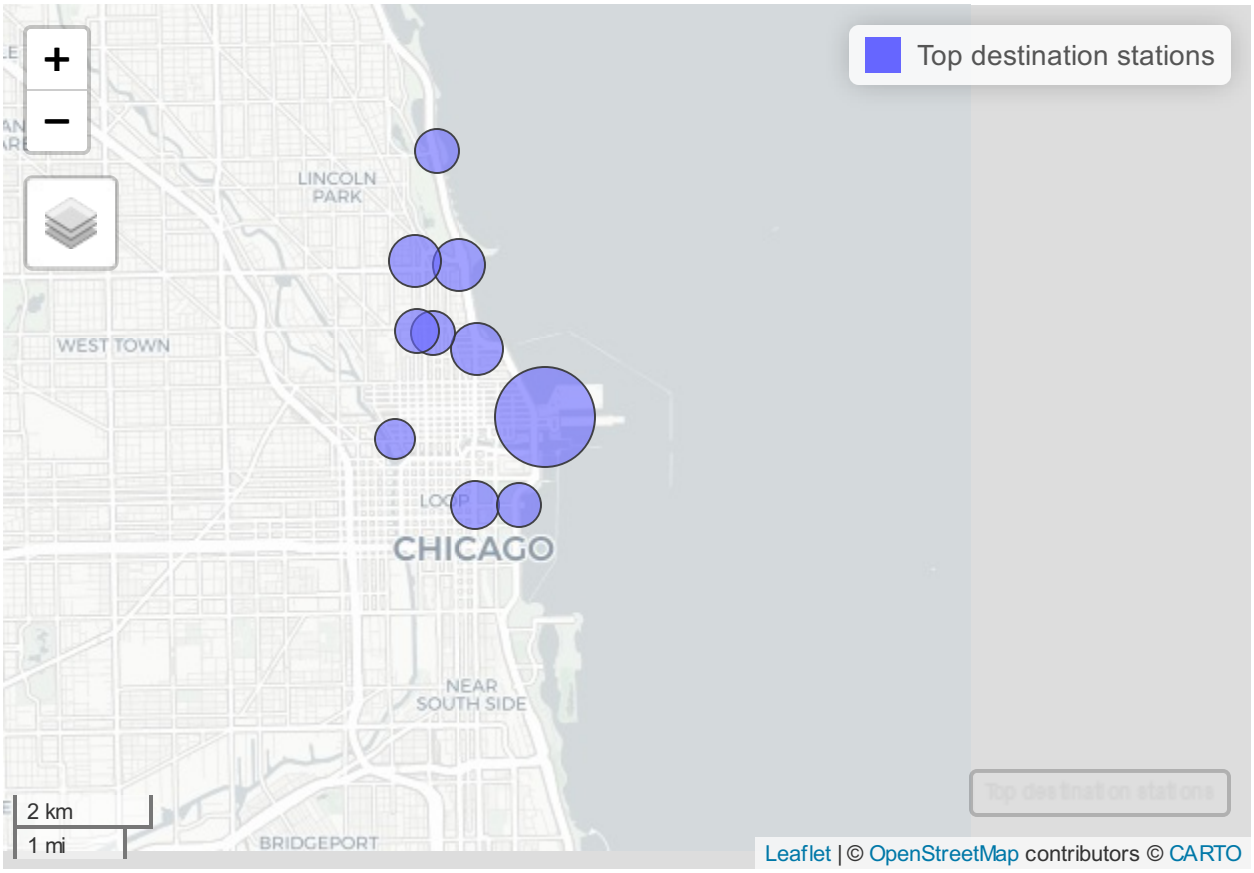
```
## # A tibble: 10 x 2
##    end_station_name                   count
##    <chr>                              <int>
##  1 Streeter Dr & Grand Ave            79972
##  2 DuSable Lake Shore Dr & North Blvd 42684
##  3 Michigan Ave & Oak St              42003
##  4 Wells St & Concord Ln              40758
##  5 Millennium Park                    38841
##  6 Clark St & Elm St                  36744
##  7 DuSable Lake Shore Dr & Monroe St  35175
##  8 Theater on the Lake                34852
##  9 Wells St & Elm St                  34414
## 10 Kingsbury St & Kinzie St           33172
```

```
first_10 <- head(stations_rides, 10)
```

For a more visual experience, we can plot each station on a map, mapping its importance with the size of dots:

```
require(data.table)
setDT(df); setDT(first_10) # convert to data.tables by reference

# join to get GPS coord from the top end stations with the number of rides
join <- df[first_10, mult = "first", on = "end_station_name", nomatch=0L]
join$rides_per_station <- first_10$count
# selection of the features to plot
df_end_stations <- select(join, c("end_station_name", "end_lng", "end_lat","rides_per_station"))
# resize for point size on map
df_end_stations$rides_per_station <- (df_end_stations$rides_per_station / max(df_end_stations$rides_per_
# map plot
mapview(df_end_stations, xcol = "end_lng", ycol = "end_lat", crs = 4269,
        cex = df_end_stations$rides_per_station, grid = FALSE, layer.name = 'Top destination stations')
```



It seems that the Navy Pier is the most visited area of Chicago.

But we can also check the different behavior between members and casual riders. Here we see the most used pair of stations (start and end) for members:

```
df_members <- filter(df, is_member == TRUE)

df_members$comb_station_names <- paste(df_members$start_station_name, " - " ,df_members$end_station_name
tail(names(sort(table(df_members$comb_station_names))), 5)
```

```
## [1] "Calumet Ave & 33rd St  -  State St & 33rd St"
## [2] "Ellis Ave & 55th St  -  Ellis Ave & 60th St"
## [3] "Ellis Ave & 60th St  -  Ellis Ave & 55th St"
## [4] "University Ave & 57th St  -  Ellis Ave & 60th St"
## [5] "Ellis Ave & 60th St  -  University Ave & 57th St"
```

And the same with casual riders:

```
df_casuals <- filter(df, is_member == FALSE)

df_casuals$comb_station_names <- paste(df_casuals$start_station_name, " - " ,df_casuals$end_station_name
tail(names(sort(table(df_casuals$comb_station_names))), 5)
```

```
## [1] "Millennium Park  -  Millennium Park"
## [2] "DuSable Lake Shore Dr & Monroe St  -  Streeter Dr & Grand Ave"
## [3] "Michigan Ave & Oak St  -  Michigan Ave & Oak St"
## [4] "DuSable Lake Shore Dr & Monroe St  -  DuSable Lake Shore Dr & Monroe St"
## [5] "Streeter Dr & Grand Ave  -  Streeter Dr & Grand Ave"
```