Cyclistic DA Report - with code

Maria Braeuner

Nov 2021

```r
library(tidyverse)
library(ggmap)
library(viridis)

CyclisticData <- read_csv("Data/CyclisticData.csv",
                          col_types = cols(
                            rideable_type = col_factor(),
                            start_station_id = col_character(),
                            end_station_id = col_character(),
                            member_casual = col_factor(),
                            dow = col_factor(),
                            season = col_factor(),
                            tod_char = col_factor()
                          ))

#colors for discretes (for some reason I cannot make the viridis work for some plots)

cols <- c("casual" = "#482677ff", "member" = "#b8de29ff")
```

Business task

Cyclistic is a (fictional) bike-share company based in Chicago. More than recruiting new customers to use Cyclistic, **there is a current interest in turning casual customers into annual members.** To support marketing strategies, we must first understand customer behavior. My task here is to gain insights and identify how casual riders and annual members use Cyclistic differently to support marketing decisions.

Description of data & cleaning process

The company provides the historical rides data. The downloadable public data set has been made available by Motivate International Inc under a non-exclusive, royalty-free, limited, perpetual license.

The data used in this report corresponds to the last 12 months (November 2020 to October 2021). The raw data is organized in separate csv files, each one corresponding to one month of data, which includes the following columns: X1, 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, date, month, day, year, dow, tod, ride_length, ride_length_min, season, tod_char.

I performed this entire task using R & RStudio. First I merged the 12 datasets into one data frame, corrected data types (e.g. casual_member from character to factor), & added columns that would help for further analysis (e.g. ride duration, time of day, day of the week, & season). Initially, there were several rows in which the company performed tests to the bikes, where "TEST" was part of the station_id or station names. These rows were removed. Detailed documentation of all cleaning & manipulation of data can be found here

Summary of analysis & visualizations

```
glimpse(CyclisticData)
```

```
## Rows: 5,376,195
## Columns: 24
## $ X1                 <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ ride_id            <chr> "BD0A6FF6FFF9B921", "96A7A7A4BDE4F82D", "C61526D...
## $ rideable_type      <fct> electric_bike, electric_bike, electric_bike, ele...
## $ started_at         <dttm> 2020-11-01 13:36:00, 2020-11-01 10:03:26, 2020-...
## $ ended_at           <dttm> 2020-11-01 13:45:40, 2020-11-01 10:14:45, 2020-...
## $ start_station_name <chr> "Dearborn St & Erie St", "Franklin St & Illinois...
## $ start_station_id   <chr> "110", "672", "76", "659", "2", "72", "76", NA, ...
## $ end_station_name   <chr> "St. Clair St & Erie St", "Noble St & Milwaukee ...
## $ end_station_id     <chr> "211", "29", "41", "185", "2", "76", "72", NA, "...
## $ start_lat          <dbl> 41.89418, 41.89096, 41.88098, 41.89550, 41.87650...
## $ start_lng          <dbl> -87.62913, -87.63534, -87.61675, -87.68201, -87....
## $ end_lat            <dbl> 41.89443, 41.90067, 41.87205, 41.91774, 41.87645...
## $ end_lng            <dbl> -87.62338, -87.66248, -87.62955, -87.69139, -87....
## $ member_casual      <fct> casual, casual, casual, casual, casual, casual, ...
## $ date               <date> 2020-11-01, 2020-11-01, 2020-11-01, 2020-11-01,...
## $ month              <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, ...
## $ day                <chr> "01", "01", "01", "01", "01", "14", "14", "14", ...
## $ year               <dbl> 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020, ...
## $ dow                <fct> Sunday, Sunday, Sunday, Sunday, Sunday, Saturday...
## $ tod                <time> 13:36:00, 10:03:26, 00:34:05, 00:45:16, 15:43:2...
## $ ride_length        <dbl> 580, 679, 1741, 555, 2007, 2961, 934, 918, 1645,...
## $ ride_length_min    <dbl> 9.6666667, 11.3166667, 29.0166667, 9.2500000, 33...
## $ season             <fct> Autumn, Autumn, Autumn, Autumn, Autumn, Autumn, ...
## $ tod_char           <fct> Afternoon, Morning, Night, Night, Afternoon, Aft...
```

```
#Order some vars for better display

##order dow by dow instead of alphabetically:
CyclisticData$dow <- ordered(CyclisticData$dow,
                             levels=c("Monday",
                                      "Tuesday",
                                      "Wednesday",
                                      "Thursday",
                                      "Friday",
                                      "Saturday",
                                      "Sunday"))

#order season
CyclisticData$season <- ordered(CyclisticData$season,
                                levels=c("Winter",
                                         "Spring",
                                         "Summer",
                                         "Autumn"))

#order tod_char
CyclisticData$tod_char <- ordered(CyclisticData$tod_char,
                                  levels=c("Morning",
                                           "Afternoon",
                                           "Evening",
                                           "Night"))

#order month
CyclisticData$month <- ordered(CyclisticData$month,
                                  levels=c("11","12","1",
                                           "2","3","4",
                                           "5","6","7",
                                           "8","9","10"))

CyclisticData <- CyclisticData %>%
  mutate(Month = ifelse(month == 11, "Nov20",
                    ifelse(month == 12, "Dec20",
                       ifelse(month == 1, "Jan21",
                          ifelse(month == 2, "Feb21",
                             ifelse(month == 3, "Mar21",
                                ifelse(month == 4, "Apr21",
                                   ifelse(month == 5, "May21",
                                      ifelse(month == 6, "Jun21",
                                         ifelse(month == 7, "Jul21",
                                            ifelse(month == 8, "Aug21",
                                               ifelse(month == 9, "Sep21",
                                                  ifelse(month == 10, "Oct21", "ERR"))))))))))))

CyclisticData$Month <- ordered(CyclisticData$Month,
                               levels=c("Nov20", "Dec20", "Jan21",
                                        "Feb21", "Mar21", "Apr21",
                                        "May21", "Jun21", "Jul21",
                                        "Aug21", "Sep21", "Oct21"))


CyclisticData <- CyclisticData %>%
  mutate(hour = format(strptime(tod, "%H:%M:%S"), '%H'))
```

```
#Data Viz

#Number of rides per month & bike preference by customer

NoRidesMonth <- CyclisticData %>%
  group_by(member_casual, rideable_type, Month) %>%
  summarise(no_rides = n(),
            avg_ride_duration = mean(ride_length_min)) %>%
  ggplot() +
  geom_col(mapping = aes(x = Month, y = no_rides, fill=member_casual),
           color = "black") +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "1. Number of rides by type of customer and bike used",
       subtitle = "from November 2020 to October 2021",
       x = "Month",
       y = "Number of rides") +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = cols) +
#  scale_color_viridis(discrete=T, option="viridis") +
  facet_grid(rideable_type ~ member_casual)


#Ride duration histogram for rides less than 24 hours long (3677 rows still over 24h)
# hist_lab is for the geom_text
hist_lab <- CyclisticData %>%
  filter(ride_length_min <= 60) %>%
  group_by(member_casual) %>%
  summarise(n = paste("n = ", n()),
            mean = paste("mean = ", round(mean(ride_length_min), digits=2)))

RideDuration <- CyclisticData %>%
  filter(ride_length_min <= 60) %>%
  ggplot() +
  geom_histogram(mapping = aes(x = ride_length_min, fill=member_casual),
                 binwidth = 5,
                 color = "black") +
  scale_fill_manual(values = cols) +
  theme_minimal() +
  theme(legend.position = "none") +
  labs(title = "2. Histogram of number of rides by their duration for each type of customer",
       subtitle = "from November 2020 to October 2021",
       caption = "Only depicting rides under 60 minutes \n (5min bins)",
       x = "minutes",
       y = "Number of rides") +
  scale_y_continuous(labels = scales::comma) +
  facet_grid(~ member_casual) +
  geom_text(data = hist_lab,
            mapping = aes(x = 50, y = 755500, label=n),
            size = 3.5)

#Traffic by day

DayTraffic <- CyclisticData %>%
  group_by(member_casual, dow, hour) %>%
  summarise(no_rides = n(),
            avg_ride_duration = mean(ride_length_min)) %>%
  ggplot() +
  geom_col(mapping = aes(x = hour, y = no_rides, fill=member_casual)) +
  scale_fill_manual(values = cols) +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Number of rides by time of day",
       subtitle = "from November 2020 to October 2021",
       x = "hour",
       y = "Number of rides") +
  scale_y_continuous(labels = scales::comma) +
  facet_grid(dow ~ member_casual)


DurationTraffic <- CyclisticData %>%
  group_by(member_casual, dow, hour) %>%
  summarise(no_rides = n(),
            avg_ride_duration = mean(ride_length_min)) %>%
  ggplot() +
  geom_col(mapping = aes(x = hour, y = avg_ride_duration, fill=member_casual)) +
  scale_fill_manual(values = cols) +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "fix",
       subtitle = "from November 2020 to October 2021",
       x = "hour",
       y = "Number of rides",
       fill = "Type of Customer") +
  scale_y_continuous(labels = scales::comma) +
  facet_grid(dow ~ member_casual)


#map stations

chicagomap <- get_stamenmap(bbox = c(left = -87.8,
                              bottom = 41.77,
                              right = -87.5,
                              top = 42),
                     zoom = 11)

ForMapping <- CyclisticData %>%
  mutate(start_lat = round(start_lat, digits=3),
         start_lng = round(start_lng, digits=3),
         end_lat = round(end_lat, digits=3),
         end_lng = round(end_lng, digits=3)) %>%
  group_by(start_lat, start_lng, end_lat, end_lng, member_casual) %>%
  summarise(count = n())

#Map: filter routes used <50 times (for map clarity)
formapfiltered <- ForMapping %>%
  filter(count > 365)

map365 <- ggmap(chicagomap) +
  geom_segment(formapfiltered,
               mapping = aes(x = start_lng,
                             y = start_lat,
                             xend = end_lng,
                             yend = end_lat,
                             color = count,
                             alpha = count,
                             size = count)) +
  geom_point(formapfiltered,
             mapping = aes(x = start_lng, y = start_lat),
             size = 1,
             alpha = 0.5,
             color = "#809848") +
  geom_point(formapfiltered,
             mapping = aes(x = end_lng, y = end_lat),
             size = 1,
             alpha = 0.5,
             color = "#4f759b") +
  facet_grid(~ member_casual) +
  labs(x = NULL, y = NULL, fill = NULL,
       title = "3. Customer travel routes") +
  theme_minimal() +
  scale_colour_viridis(option = "viridis") +
  theme(legend.position = "none")

#Top routes
toproutes_casual <- formapfiltered %>%
  filter(member_casual == "casual") %>%
  arrange(desc(count)) %>%
  head(100)

toproutes_member <- formapfiltered %>%
  filter(member_casual == "member") %>%
  arrange(desc(count)) %>%
  head(100)


chicagozoom <- get_stamenmap(bbox = c(left = -87.7,
                              bottom = 41.78,
                              right = -87.5,
                              top = 41.96),
                     zoom = 11)


casual100 <- ggmap(chicagozoom) +
  geom_segment(toproutes_casual,
               mapping = aes(x = start_lng,
                             y = start_lat,
                             xend = end_lng,
                             yend = end_lat,
                             color = count,
                             alpha = count,
                             size = count)) +
  geom_point(toproutes_casual,
             mapping = aes(x = start_lng, y = start_lat),
             size = 1,
             alpha = 0.5,
             color = "#809848") +
  geom_point(toproutes_casual,
             mapping = aes(x = end_lng, y = end_lat),
             size = 1,
             alpha = 0.5,
             color = "#4f759b") +
  labs(x = NULL, y = NULL, fill = NULL,
       title = "4. Casual customer 100 most used travel routes") +
  theme_minimal() +
  scale_colour_viridis(option = "viridis") +
  theme(legend.position = "none")
```

```r
member100 <- ggmap(chicagozoom) +
  geom_segment(toproutes_member,
               mapping = aes(x = start_lng,
                             y = start_lat,
                             xend = end_lng,
                             yend = end_lat,
                             color = count,
                             alpha = count,
                             size = count)) +
  geom_point(toproutes_member,
             mapping = aes(x = start_lng, y = start_lat),
             size = 1,
             alpha = 0.5,
             color = "#809848") +
  geom_point(toproutes_member,
             mapping = aes(x = end_lng, y = end_lat),
             size = 1,
             alpha = 0.5,
             color = "#4f759b") +
  labs(x = NULL, y = NULL, fill = NULL,
       title = "5. Members 100 most used travel routes") +
  theme_minimal() +
  scale_colour_viridis(option = "viridis") +
  theme(legend.position = "none")


TopStartStation <- CyclisticData %>%
  filter(!is.na(start_station_id)) %>%
  group_by(member_casual, start_station_id) %>%
  summarise(count = n()) %>%
  group_by(member_casual) %>%
  filter(count == max(count)) %>%
  ungroup()

TopEndStation <- CyclisticData %>%
  filter(!is.na(end_station_id)) %>%
  group_by(member_casual, end_station_id) %>%
  summarise(count = n()) %>%
  group_by(member_casual) %>%
  filter(count == max(count)) %>%
  ungroup()


#tile viz

MoDowTile <- CyclisticData %>%
  group_by(member_casual, Month, dow) %>%
  summarise(no_rides = n()) %>%
  ggplot() +
  geom_tile(mapping = aes(x = Month,
                          y = dow,
                          fill = no_rides)) +
  facet_grid(rows = vars(member_casual)) +
  theme_minimal() +
  labs(title = "6. Number of rides by day and month by each type of customer",
       subtitle = "from Nov. 2020 to Oct. 2021",
       x = "month",
       y = NULL,
       fill = "Number of rides") +
  scale_fill_viridis(option = "viridis") +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 5.5),
        axis.text.y = element_text(hjust=1))

hodtiles <- CyclisticData %>%
  group_by(member_casual, dow, hour) %>%
  summarise(no_rides = n()) %>%
  ggplot() +
  geom_tile(mapping = aes(x = hour,
                          y = dow,
                          fill = no_rides)) +
  facet_grid(rows = vars(member_casual)) +
  theme_minimal() +
  labs(title = "7. Number of rides by hour of each day by each type of customer",
       subtitle = "from Nov. 2020 to Oct. 2021",
       x = "hour",
       y = NULL,
       fill = "Number of rides") +
  scale_fill_viridis(option = "viridis") +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 5.5),
        axis.text.y = element_text(hjust=1))
```

Key Findings

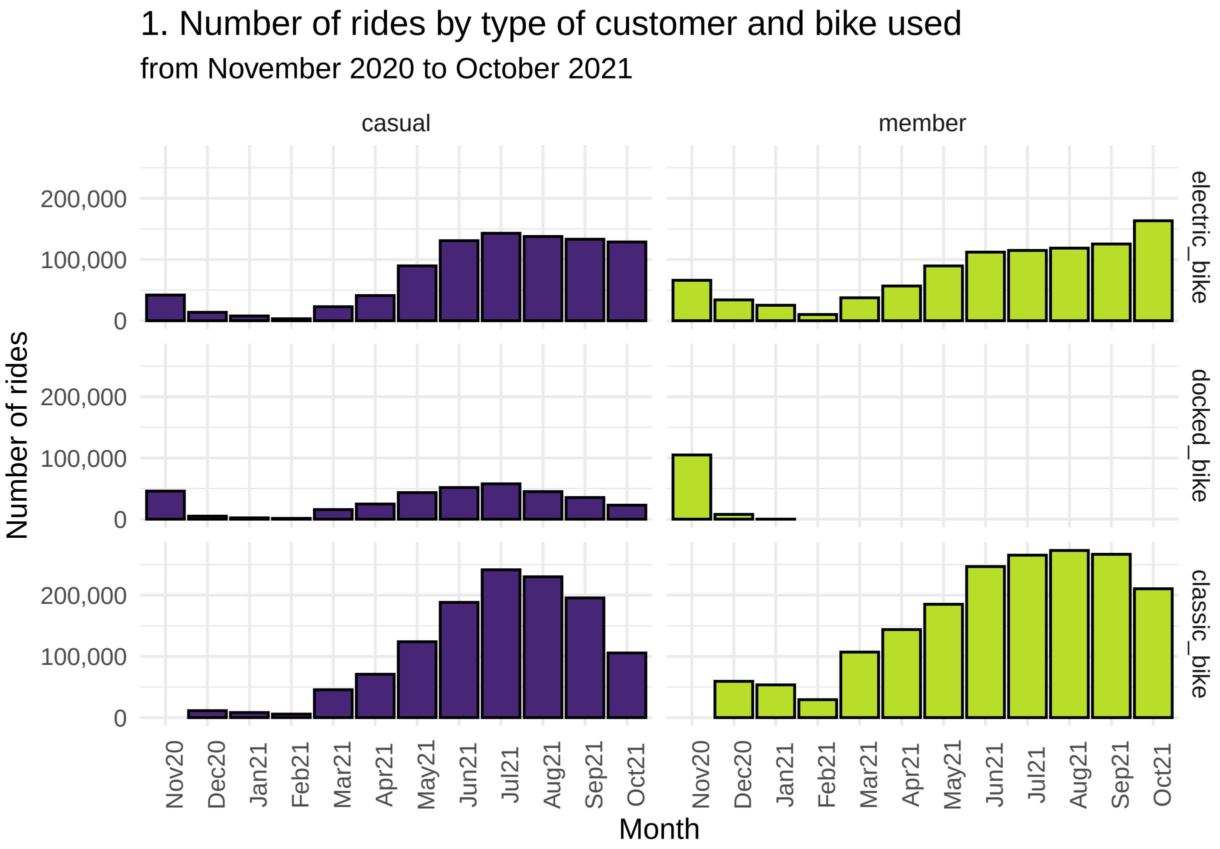
- Generally speaking, casual riders seem to use the bike rental service more for leisure while annual members use it as a daily commute to/from work.

```
mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

SumStats <- CyclisticData %>%
  group_by(member_casual) %>%
  summarise(no_rides = n(),
            avg_duration_minutes = mean(ride_length_min),
            TopHour = mode(hour),
            TopDay = mode(dow),
            TopMonth = mode(Month),
            TopSeason = mode(season))

SumStats
```

```
## # A tibble: 2 x 7
##   member_casual no_rides avg_duration_minut~ TopHour TopDay   TopMonth TopSeason
## * <fct>            <int>               <dbl> <chr>   <ord>    <ord>    <ord>
## 1 casual          2469725                 31.8 17       Saturday Jul21     Summer
## 2 member          2906470                 14.0 17       Wednesd~ Sep21     Summer
```

- Both classic and electronic bikes seem to be the preferred ride type for both customer groups (figure 1).

```
#Las 12 months trend in user type and number of rides.

NoRidesMonth
```

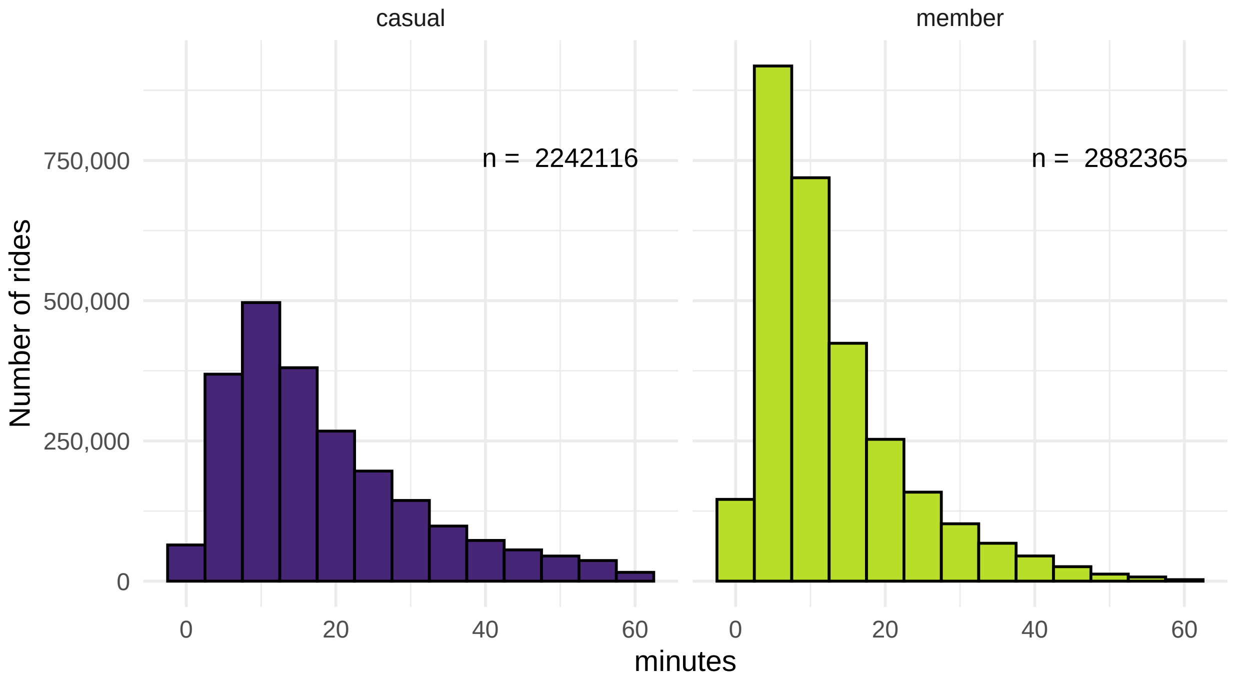


- Casual users seem to take on average longer rides when compared to annual members (table 1; figure 2); casual users are also mostly heading to/from stations near the Navy Pier (figures 3L & 4), while annual members take shorter routes within the city and outskirts (figures 3R & 5).

```
RideDuration
```

2. Histogram of number of rides by their duration for each type of custon

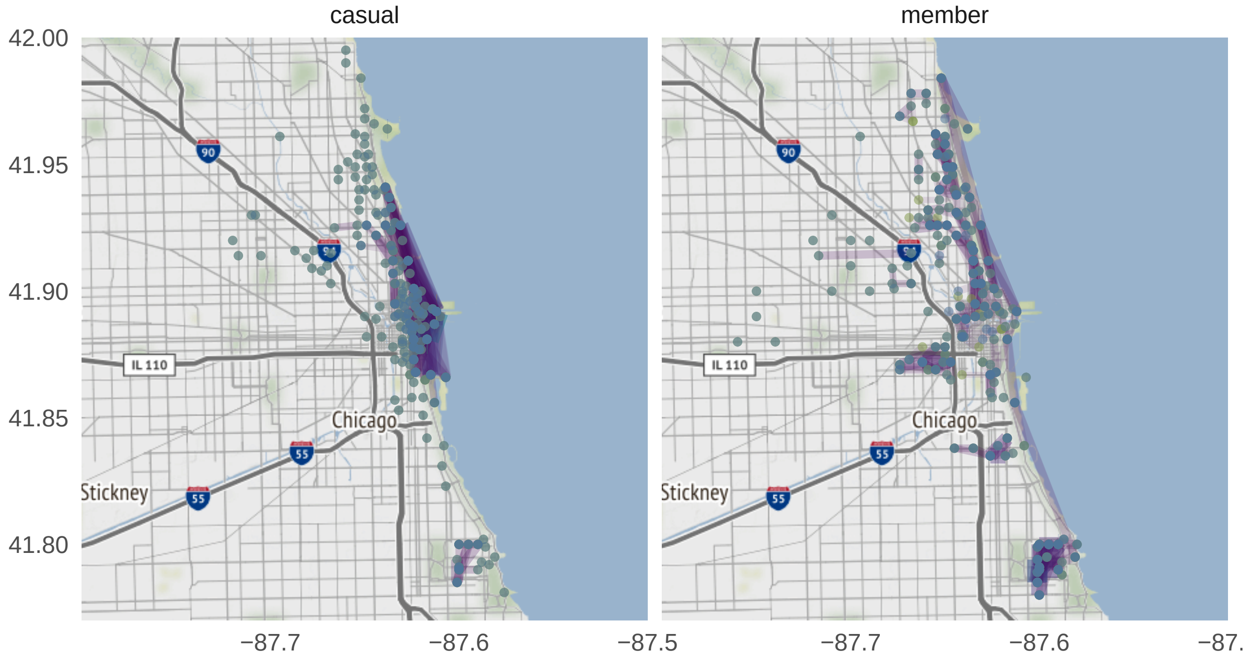
from November 2020 to October 2021



Only depicting rides under 60 minutes
(5min bins)

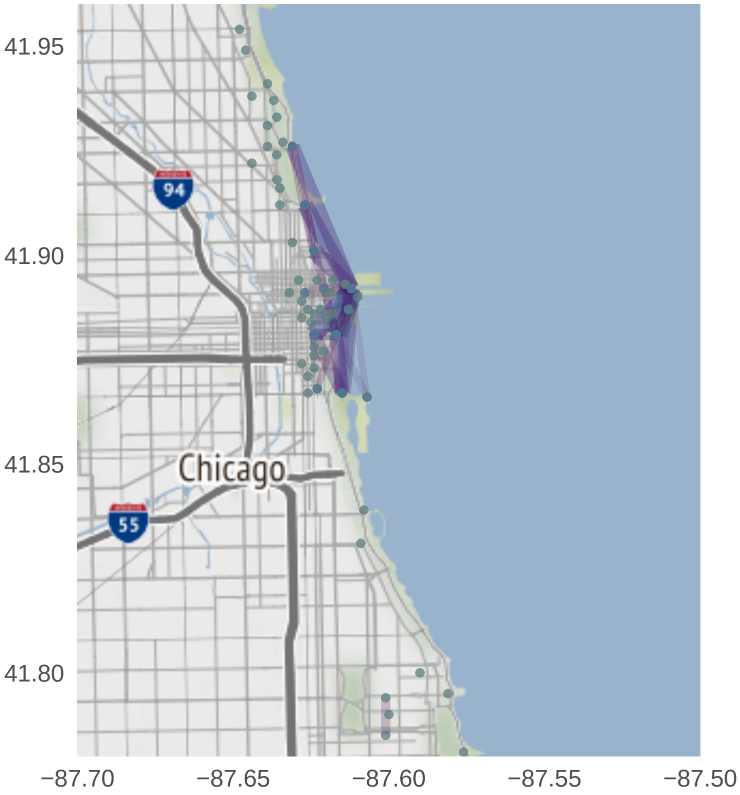
```
map365
```

3. Customer travel routes

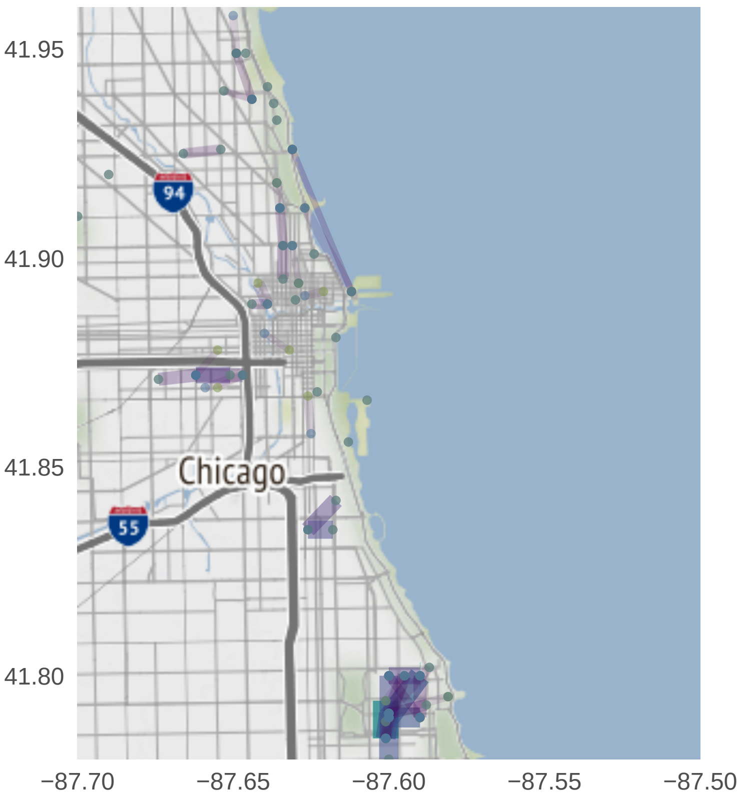


```
casual100
```

4. Casual customer 100 most used travel routes



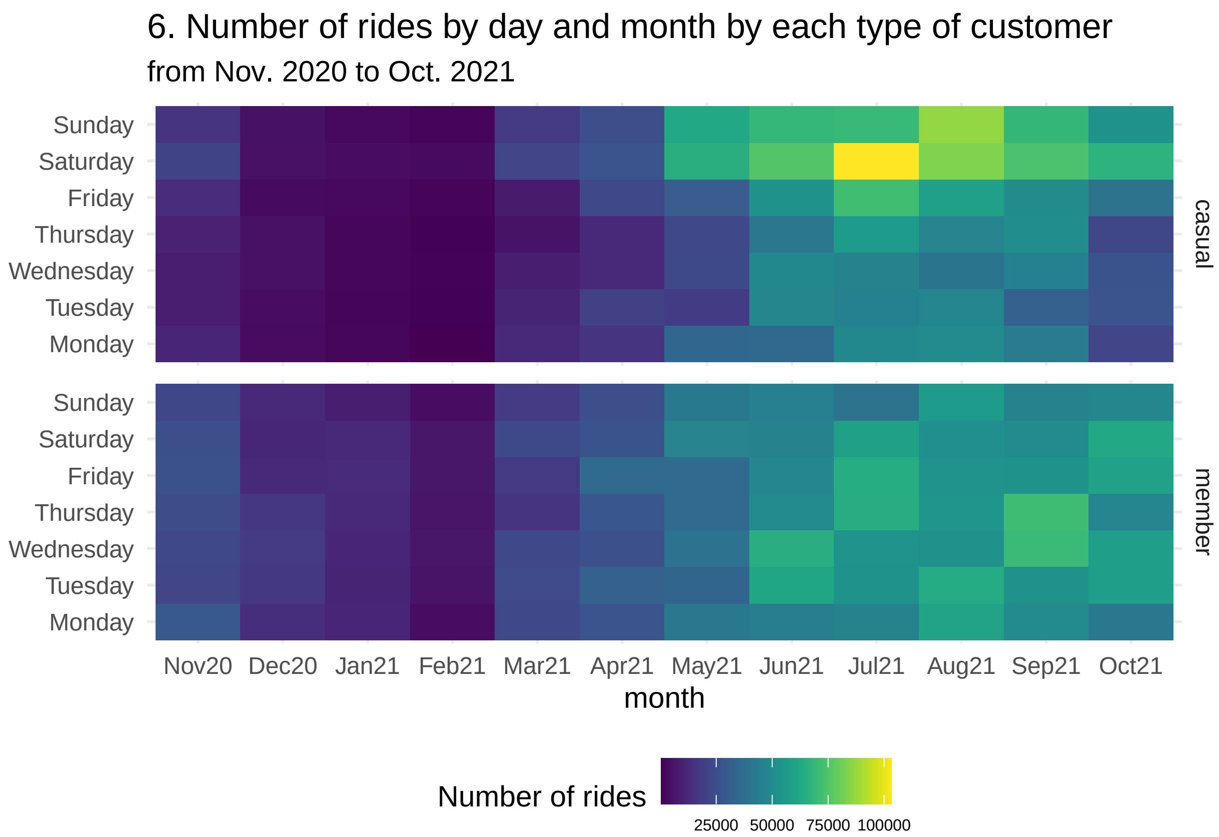
```
member100
```



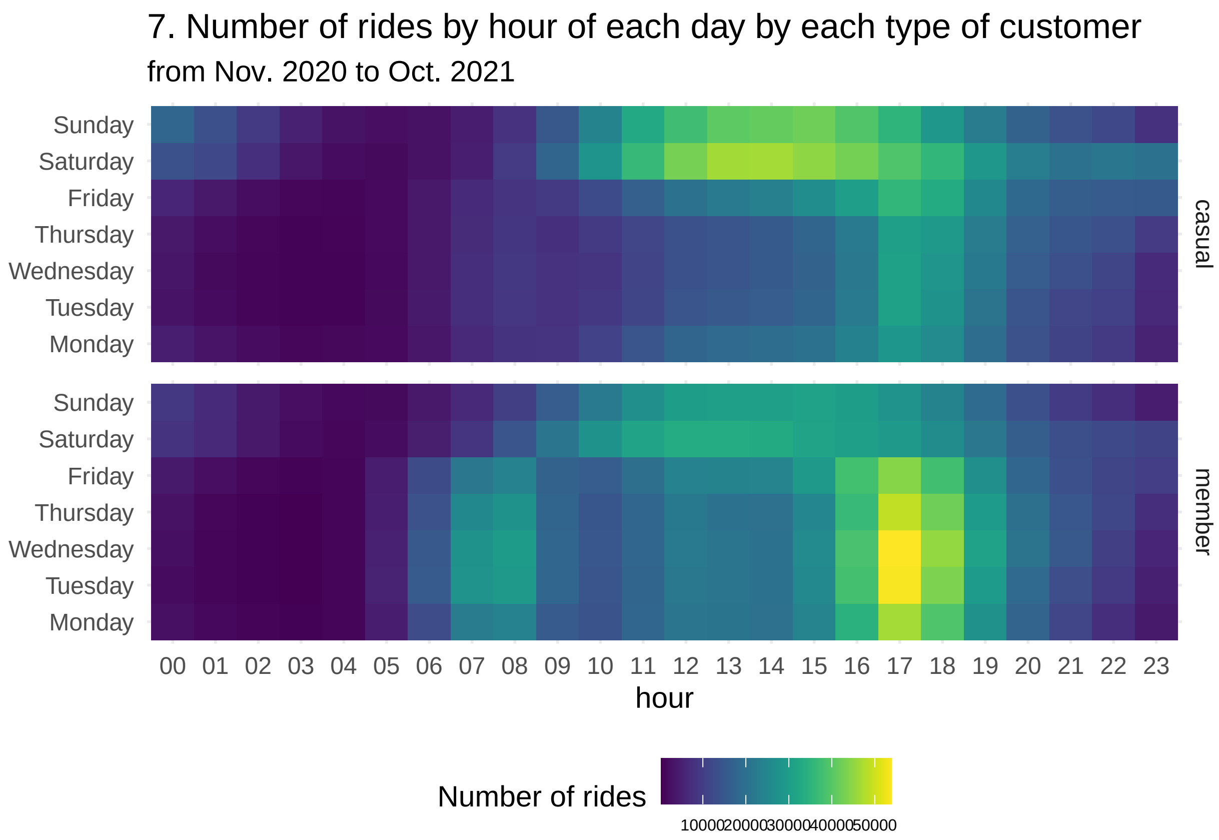
5. Members 100 most used travel routes

- For both casual users and members Winter time is when there is less use of the bike rental services, which starts to increase from early Spring and peaks during Summer (table 1; figure 6). Casual riders are using the bikes more on the weekends, particularly weekend afternoons, while annual members seem to use it regularly on the weekdays between 06:00-09:00 and 16:00-18:00 (figure 7).

```
MoDowTile
```


6. Number of rides by day and month by each type of customer
from Nov. 2020 to Oct. 2021
Sunday
Saturday
Friday
Thursday
Wednesday
Tuesday
Monday
casual
member
Nov20
Dec20
Jan21
Feb21
Mar21
Apr21
May21
Jun21
Jul21
Aug21
Sep21
Oct21
month
Number of rides
25000
50000
75000
100000

```
hodtiles
```



7. Number of rides by hour of each day by each type of customer

from Nov. 2020 to Oct. 2021

Top three recommendations

Based on the previous analysis, it seems safe to assume that casual users are using the service more for leisure and annual members as daily (weekdays) commute. To turn casual customers into annual members, the following could be recommended to the marketing team:

- In case casual users haven't considered a non-touristic use of the service, start a late-Winter or early-Spring marketing campaign that addresses bike usage as commute for health and environmental reasons.
- Summer and/or weekend promotions around the Navy Pier since casual riders are more active there and during the weekends.
- Offering member-only discount options based on trip length/duration could incentivize casual users to opt for a membership.