bikers_GoogleCapstone

Daniel Fidalgo

11/30/2021

1. Ask

1.1 The objective

- How do annual members and casual members use bikes differently?
- How to convert casual riders into annual members?

2. Prepare

2.1 Download the required data

In this case, we are using the last 12 months of data provided by the stakeholders company.

```
#Load the data from the last 12 months
biker_10_2021 <- read_csv("bikers_data/202110-divvy-tripdata/202110-divvy-
tripdata.csv") # October 2021

## Rows: 631226 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.

biker_09_2021 <- read_csv("bikers_data/202109-divvy-tripdata/202109-divvy-
tripdata.csv") # September 2021

## Rows: 756147 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.
```

```
biker_08_2021 <- read_csv("bikers_data/202108-divvy-tripdata/202108-divvy-
tripdata.csv") # August 2021
```

```
## Rows: 804352 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.
```

```
biker_07_2021 <- read_csv("bikers_data/202107-divvy-tripdata/202107-divvy-
tripdata.csv") # July 2021
```

```
## Rows: 822410 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.
```

```
biker_06_2021 <- read_csv("bikers_data/202106-divvy-tripdata/202106-divvy-
tripdata.csv") # June 2021
```

```
## Rows: 729595 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.
```

```
biker_05_2021 <- read_csv("bikers_data/202105-divvy-tripdata/202105-divvy-
tripdata.csv") # May 2021
```

```
## Rows: 531633 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.
```

```
biker_04_2021 <- read_csv("bikers_data/202104-divvy-tripdata/202104-divvy-
tripdata.csv") # April 2021
```

```
## Rows: 337230 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.
```

```
biker_03_2021 <- read_csv("bikers_data/202103-divvy-tripdata/202103-divvy-
tripdata.csv") # March 2021
```

```
## Rows: 228496 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.

biker_02_2021 <- read_csv("bikers_data/202102-divvy-tripdata/202102-divvy-
tripdata.csv") # February 2021

## Rows: 49622 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.

biker_01_2021 <- read_csv("bikers_data/202101-divvy-tripdata/202101-divvy-
tripdata.csv") # January 2021

## Rows: 96834 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.

biker_12_2020 <- read_csv("bikers_data/202012-divvy-tripdata/202012-divvy-
tripdata.csv") # December 2020
```

```
## Rows: 131573 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.
```

```
biker_11_2020 <- read_csv("bikers_data/202011-divvy-tripdata/202011-divvy-
tripdata.csv") # November 2020
```

```
## Rows: 259716 Columns: 13

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name,
memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat,
e...
## 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.
```

```
#Pool the data into a single data.frame
bikers_pooled <- rbind(biker_10_2021, biker_09_2021, biker_08_2021,
biker_07_2021, biker_06_2021, biker_05_2021, biker_04_2021, biker_03_2021,
biker_02_2021,biker_01_2021, biker_12_2020, biker_11_2020)
```

2.2 Identify the how the data is organized

```
# Look at the data structure
glimpse(bikers_pooled)
```

```
## Rows: 5,378,834
## Columns: 13
## $ ride_id            <chr> "620BC6107255BF4C", "4471C70731AB2E45",
"26CA69D43D~
## $ rideable_type      <chr> "electric_bike", "electric_bike",
"electric_bike", ~
## $ started_at         <dttm> 2021-10-22 12:46:42, 2021-10-21 09:12:37,
2021-10-~
## $ ended_at           <dttm> 2021-10-22 12:49:50, 2021-10-21 09:14:14,
```

```
2021-10-~
## $ start_station_name <chr> "Kingsbury St & Kinzie St", NA, NA, NA, NA, NA,
NA,~
## $ start_station_id   <chr> "KA1503000043", NA, NA, NA, NA, NA, NA, NA, NA,
NA,~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA,~
## $ start_lat          <dbl> 41.88919, 41.93000, 41.92000, 41.92000,
41.89000, 4~
## $ start_lng          <dbl> -87.63850, -87.70000, -87.70000, -87.69000, -
87.710~
## $ end_lat            <dbl> 41.89000, 41.93000, 41.94000, 41.92000,
41.89000, 4~
## $ end_lng            <dbl> -87.63000, -87.71000, -87.72000, -87.69000, -
87.690~
## $ member_casual      <chr> "member", "member", "member", "member",
"member", "~

colnames(bikers_pooled)

##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"

head(bikers_pooled)

## # A tibble: 6 x 13
##   ride_id rideable_type started_at          ended_at
start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 620BC6~ electric_bike 2021-10-22 12:46:42 2021-10-22 12:49:50 Kingsbury
St & ~
## 2 4471C7~ electric_bike 2021-10-21 09:12:37 2021-10-21 09:14:14 <NA>
## 3 26CA69~ electric_bike 2021-10-16 16:28:39 2021-10-16 16:36:26 <NA>
## 4 362947~ electric_bike 2021-10-16 16:17:48 2021-10-16 16:19:03 <NA>
## 5 BB731D~ electric_bike 2021-10-20 23:17:54 2021-10-20 23:26:10 <NA>
## 6 717630~ electric_bike 2021-10-21 16:57:37 2021-10-21 17:11:58 <NA>
## # ... with 8 more variables: start_station_id <chr>, end_station_name
<chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>

# Look at missing data

# table.NA function

# This function automatically calculates the number and percentage of missing
```

```
values for each column in a data frame.

table_NA<- function(data){

  require(ggplot2)

  na.table<- matrix(NA,ncol(data),3)
  na.table[,1] <- colnames(data)
  na.table<- data.frame(na.table)
  colnames(na.table)<- c("Variable","n_missing","missing_percent")

  for (a in 1:(ncol(data))) {

    na.table[a,2]<- sum(is.na(data[,a]))
    na.table[a,3]<- paste(round((sum(is.na(data[,a]))/nrow(data)*100),1),"%")

  }

  return(table.NA = na.table)
}

table_NA(bikers_pooled)
```

```
##              Variable n_missing missing_percent
## 1             ride_id         0              0 %
## 2       rideable_type         0              0 %
## 3          started_at         0              0 %
## 4            ended_at         0              0 %
## 5  start_station_name    600479          11.2 %
## 6    start_station_id    600586          11.2 %
## 7    end_station_name    646471            12 %
## 8      end_station_id    646548            12 %
## 9           start_lat         0              0 %
## 10          start_lng         0              0 %
## 11            end_lat      4831           0.1 %
## 12            end_lng      4831           0.1 %
## 13      member_casual         0              0 %
```

2.3 Sort and filter the data

We are dropping all rows with any missing values. After removing them, we lost 16.47% of the rows, however a very large portion of the dataset is still intact.

```
# filter and drop NAs
bikers_clean <- bikers_pooled%>%
  select(member_casual,rideable_type,started_at,ended_at,start_station_name,
         end_station_name)%>%
  drop_na()
```

```r
# Percentage of the dataset that was removed
((nrow(bikers_clean)-nrow(bikers_pooled))/nrow(bikers_pooled)*100)
```

```
## [1] -16.47123
```

```r
# Arrange the data by started date
bikers_clean<- bikers_clean%>%
  arrange(started_at)

# Change characters to factors and check for naming errors
bikers_clean$member_casual<- as.factor(bikers_clean$member_casual)
levels(bikers_clean$member_casual)
```

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

```r
bikers_clean$rideable_type<- as.factor(bikers_clean$rideable_type)
levels(bikers_clean$rideable_type)
```

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

```r
bikers_clean$start_station_name <- as.factor(bikers_clean$start_station_name)
nlevels(bikers_clean$start_station_name)
```

```
## [1] 807
```

```r
bikers_clean$end_station_name <- as.factor(bikers_clean$end_station_name)
nlevels(bikers_clean$end_station_name)
```

```
## [1] 804
```

```r
# Make sure all dates are in Year-month-day_hours_minutes_seconds
bikers_clean$started_at<- ymd_hms(bikers_clean$started_at)
bikers_clean$ended_at<- ymd_hms(bikers_clean$ended_at)

# Clean the column names for possible inconsistencies
bikers_clean<- clean_names(bikers_clean)
```

3. Process

3.1 Transform the data

After this are going to manipulate the data to create some more variables:

- A column for the day of the week each ride was taken.
- A column for the month each ride was taken.

```r
# Create a column for day of the week, another for month, and another for
time
bikers_clean <- bikers_clean%>%
  mutate(hour_start = hour(started_at),
         week_day = wday(started_at,label = TRUE, abbr = FALSE),
         month = month(started_at, label = TRUE,),
```

```
         ride_length_mins = as.numeric(abs(round(difftime(started_at,
ended_at, unit="mins"),1))))

#Remove rides whose length (in minutes) is greater than the mean plus two
times the standard deviation

mean_ride_length<- mean(bikers_clean$ride_length_mins)
sd_ride_length<- sd((bikers_clean$ride_length_mins))

outlier.index<-
which(bikers_clean$ride_length_mins>mean_ride_length+sd_ride_length*2)

bikers_clean<- bikers_clean[-outlier.index,]

# A function to calculate the mode for a given vector
# This function does not for for entire data.frames, only single vectors.

mode<- function(vector){

  #transfor the vector into a factor
  vector<- as.factor(vector)
  #Use the table function to count each of the factor
  table_vector<- table(vector)
  #Which factor repeats itself the most
  max_index<- max(table(vector))
  #print the name of the factor
  result<- names(which(table_vector==max_index))

  return(result)
}
```

3.3 Summarize data

```
# According to membership
membership<- bikers_clean%>%
  group_by(member_casual)%>%
  summarize(N = n(),
            average_ridelength_mins = mean(ride_length_mins),
            sd_ridelength = sd(ride_length_mins),
            max_ridelength = max(ride_length_mins),
            mode_week = mode(week_day),
            mode_start_station = mode(start_station_name),
            mode_end_station = mode(end_station_name))%>%
  ungroup()

# According to membership AND type of bike
membership_biketype<-
  bikers_clean%>%
  group_by(member_casual, rideable_type)%>%
  summarize(N = n(),
            average_ridelength_mins = mean(ride_length_mins),
```

```
            sd_ridelength = sd(ride_length_mins),
            max_ridelength = max(ride_length_mins),
            mode_week = mode(week_day),
            mode_start_station = mode(start_station_name),
            mode_end_station = mode(end_station_name))%>%
  ungroup()
```

```
## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.
```

```
# According to membership AND Hours of the day
membership_hours<-
  bikers_clean%>%
  group_by(member_casual, hour_start)%>%
  summarize(N = n(),
            average_ridelength_mins = mean(ride_length_mins),
            sd_ridelength = sd(ride_length_mins),
            max_ridelength = max(ride_length_mins),
            mode_week = mode(week_day),
            mode_start_station = mode(start_station_name),
            mode_end_station = mode(end_station_name))%>%
  ungroup()
```

```
## `summarise()` has grouped output by 'member_casual', 'hour_start'. You can
override using the `.groups` argument.
```

```
# Acording to membership AND days of the week AND months
membership_week_month<- bikers_clean%>%
  group_by(member_casual,week_day,month) %>%
  summarize(N = n(),
            average_ridelength_mins = mean(ride_length_mins),
            sd_ridelength = sd(ride_length_mins),
            max_ridelength = max(ride_length_mins),
            mode_week = mode(week_day),
            mode_start_station = mode(start_station_name),
            mode_end_station = mode(end_station_name))%>%
  ungroup()
```

```
## `summarise()` has grouped output by 'member_casual', 'week_day', 'month'.
You can override using the `.groups` argument.
```

4. Analyze

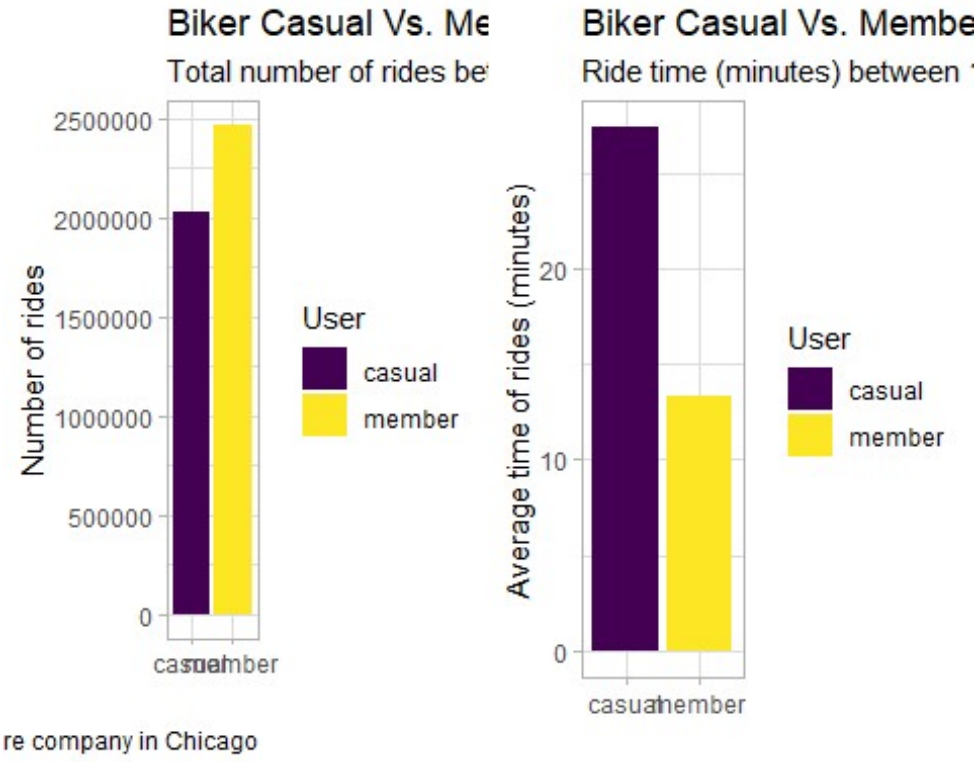
4.1 Are there differences in biking time and number of rides between members and casuals over the last year

Over the last year, casual members have ride a higher amount of time, but members lead the number of rides.

```
knitr::kable(membership)
```

member_casual	N	average_ridelengthh_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
casual	2026898	27.35564	34.18203	631.3	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
member	2461277	13.33710	13.42991	630.9	Wednesday	Clark St & Elm St	Clark St & Elm St

```
p1<- ggplot(membership, aes(x= member_casual,y=N, fill=member_casual))+
  geom_col()+
  labs(title = "Biker Casual Vs. Members",subtitle = "Total number of rides
between 11-2020 and 10-2021",x="",y="Number of rides", caption = "data
provided by Cyclistic, a bike-share company in Chicago",fill = "User")+
  scale_fill_viridis_d()

p2<- ggplot(membership, aes(x= member_casual,y=average_ridelength_mins,
fill=member_casual))+
  geom_col()+
  labs(title = "Biker Casual Vs. Members",subtitle = "Ride time (minutes)
between 11-2020 and 10-2021",x="",y="Average time of rides (minutes)",fill =
"User")+
  scale_fill_viridis_d()

grid.arrange(p1, p2, nrow = 1)
```



4.2 Does bike type influences the length or number of rides between members and casuals?

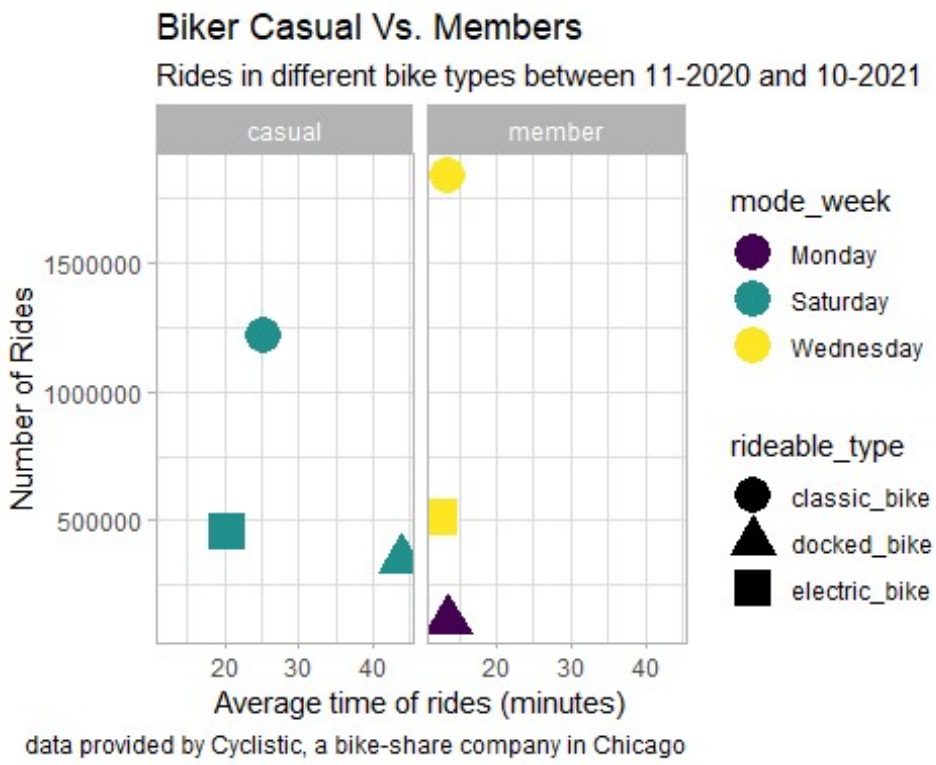
- Casual members use bikes mostly on a Saturday independently of bike type.
- Classic bikes have the highest amount of rides independent of membership.
- Docked bikes have the lowest number of rides per type of bike independent of membership. But casuals using docked bikes the longest rides on average.
- Electric bikes always have the shortest rides, and their usage is between classic and docked bikes.

```
knitr::kable(membership_biketype)
```

member_casual	rideable_type	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
casual	classic_bike	1220247	25.23462	30.78375	631.2	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	docked_bike	347613	43.88586	50.02930	631.3	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	electric_bike	459038	20.47614	21.87459	480.0	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
member	classic_bike	1836733	13.57031	13.47103	630.9	Wednesday	Clark St & Elm St	Clark St & Elm St
member	docked_bike	113179	13.59348	13.40060	613.1	Monday	Clark St & Elm St	Clark St & Elm St
member	electric_bike	511365	12.44269	13.24969	478.0	Wednesday	Wells St & Concord Ln	Dearborn St & Erie St

```
p3<- ggplot(membership_biketype, aes(x= average_ridelength_mins, y= N, shape= rideable_type, color= mode_week))+
  geom_point(size=6)+
  labs(title = "Biker Casual Vs. Members",subtitle = "Rides in different bike types between 11-2020 and 10-2021",x="Average time of rides (minutes)",y="Number of Rides",caption = "data provided by Cyclistic, a bike-share company in Chicago",fill = "User")+
  facet_wrap(~member_casual)+
  scale_color_viridis_d()
```

```
p3
```



4.3 Are the differences during the day that we should account for?

- Compared to Casuals, Members have the highest number of rides throughout the day, except during night time (approximately between 20.00h and 04.00h).
- Members always have the shortest rides throughout the day.
- In both cases, there is a spike in the number of rides during the afternoon, and a big decrease during the night.

```
knitr::kable(membership_hours)
```

member_casual	hour_start	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
casual	0	41312	25.51452	42.93504	631.2	Sunday	Wells St & Concord Ln	Wells St & Concord Ln
casual	1	29797	25.07527	44.12535	631.3	Sunday	Clark St & Elm St	Wabash Ave & Grand Ave
casual	2	188	25.52538	48.108	626.3	Sunda	Clark St &	Ashland

member_casual	hour_start	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
		49		65		y	Elm St	Ave & Division St
casual	3	9875	26.95957	53.12948	626.7	Sunday	Clark St & Elm St	Wabash Ave & Grand Ave
casual	4	6498	23.06919	43.61580	626.4	Sunday	Winthrop Ave & Lawrence Ave	Southport Ave & Wavelan d Ave
casual	5	8440	20.62052	37.34067	600.7	Sunday	Indiana Ave & Roosevelt Rd	St. Clair St & Erie St
casual	6	18881	18.33346	30.55585	594.3	Tuesday	Kingsbury St & Erie St	St. Clair St & Erie St
casual	7	34751	19.13813	30.29851	615.0	Wednesday	Clark St & Elm St	Franklin St & Monroe St
casual	7	34751	19.13813	30.29851	615.0	Wedn	St. Clair St & Erie St	Franklin St & Monroe St
casual	8	48358	22.16271	32.88442	625.9	Saturday	Michigan Ave & Oak St	Streeter Dr & Grand Ave
casual	9	59989	28.10847	37.68076	627.0	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	10	84599	31.52452	39.27907	631.2	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	11	111949	32.04375	39.15804	626.5	Saturday	Streeter Dr & Grand	Streeter Dr & Grand

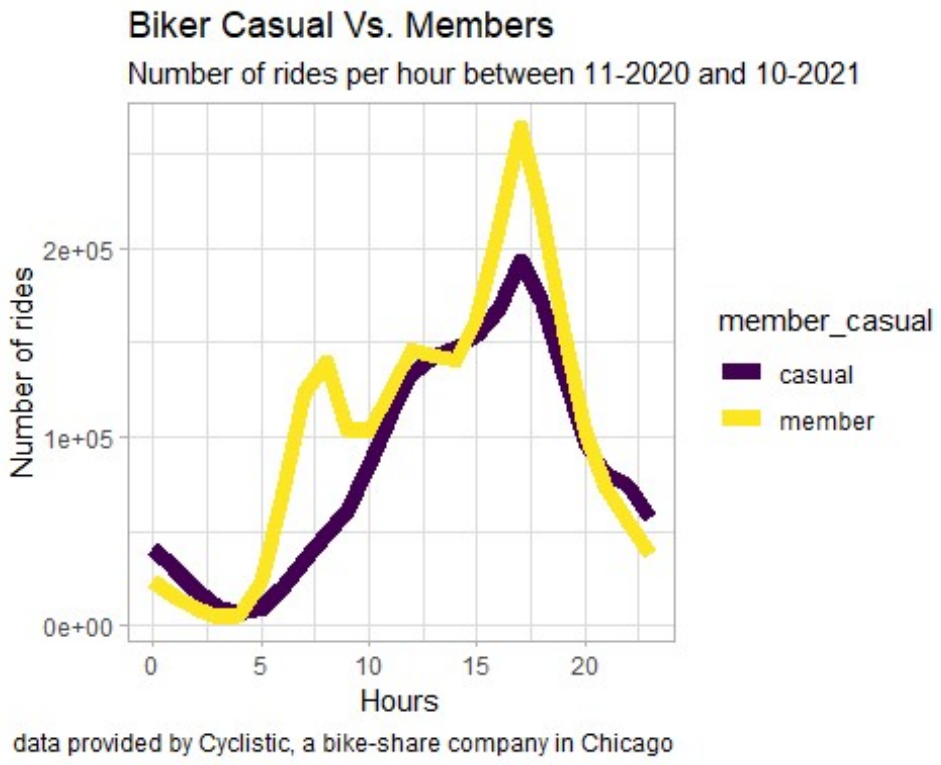
member_casual	hour_start	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
							Ave	Ave
casual	12	133369	31.21265	36.66849	628.1	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	13	142659	32.09043	37.10496	626.6	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	14	147006	31.52289	35.45404	628.1	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	15	153670	29.83408	33.10108	567.1	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	16	167420	27.58348	31.14048	594.3	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	17	193350	25.26339	28.66270	627.7	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	18	173569	24.26470	27.33329	605.7	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	19	132873	24.69389	28.79210	615.5	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	20	96540	25.05298	30.47831	628.2	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	21	81072	24.45615	31.72411	630.3	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave

member_casual	hour_start	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
casual	22	74684	23.94015	33.00681	629.9	Saturday	Streeter Dr & Grand Ave	Millennium Park
casual	23	57388	24.51782	37.81784	631.0	Saturday	Wells St & Concord Ln	Millennium Park
member	0	23730	12.54491	18.14706	622.8	Sunday	Wells St & Elm St	Clark St & Elm St
member	1	15289	13.18611	20.51706	628.8	Sunday	Halsted St & Roscoe St	Clark St & Elm St
member	2	8472	13.26549	21.05645	566.2	Sunday	Clark St & Elm St	Clark St & Elm St
member	3	4788	13.45541	21.52032	522.5	Sunday	Broadway & Waveland Ave	Clark St & Lincoln Ave
member	3	4788	13.45541	21.52032	522.5	Sunday	Broadway & Waveland Ave	Racine Ave & Fullerton Ave
member	4	5759	11.98790	17.92796	537.9	Sunday	Desplaines St & Jackson Blvd	St. Clair St & Erie St
member	5	24010	10.84337	11.81297	512.5	Tuesday	Columbus Dr & Randolph St	St. Clair St & Erie St
member	6	68809	11.85968	12.33547	580.5	Tuesday	Clinton St & Washington Blvd	St. Clair St & Erie St
member	7	123657	12.02017	11.49526	621.6	Tuesday	Clark St & Elm St	St. Clair St & Erie St
member	8	139501	11.96851	12.16147	622.1	Wednesday	Clinton St & Madison	Clark St & Randolph

member_casual	hour_start	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
							St	St
member	9	104290	12.59207	13.24105	605.8	Saturday	Kingsbury St & Kinzie St	University Ave & 57th St
member	10	103196	13.43348	14.43406	530.6	Saturday	Kingsbury St & Kinzie St	Michigan Ave & Oak St
member	11	125280	13.65760	14.19688	613.3	Saturday	Wells St & Concord Ln	Kingsbury St & Kinzie St
member	12	145718	13.47574	13.56394	600.9	Saturday	Wells St & Concord Ln	Wells St & Concord Ln
member	13	143198	13.82327	14.28531	626.8	Saturday	Kingsbury St & Kinzie St	Theater on the Lake
member	14	141344	14.16858	14.00049	627.5	Saturday	Theater on the Lake	Wells St & Concord Ln
member	15	162554	13.99325	13.21143	531.2	Sunday	St. Clair St & Erie St	Clinton St & Washington Blvd
member	16	211104	13.88413	13.03311	597.6	Wednesday	St. Clair St & Erie St	Clark St & Elm St
member	17	263927	13.86428	12.61197	605.1	Wednesday	Kingsbury St & Kinzie St	Clark St & Elm St
member	18	221497	13.66997	12.55825	576.4	Wednesday	Clark St & Elm St	Wells St & Elm St
member	19	156465	13.47879	12.82753	611.8	Wednesday	Clark St & Elm St	Clark St & Elm St
member	20	102303	13.30192	13.45624	630.9	Wednesday	Wells St & Concord Ln	Clark St & Elm St
member	21	73382	12.93562	13.81306	627.3	Wednesday	Wells St & Concord Ln	Clark St & Elm St

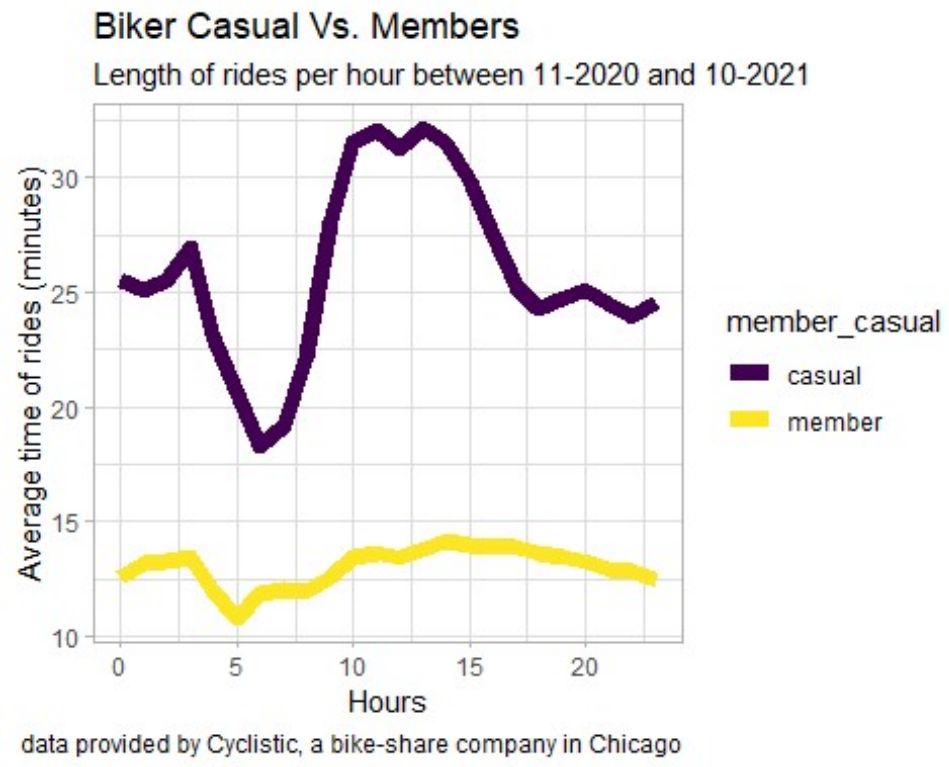
member_casual	hour_start	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
member	22	55270	12.81217	15.14352	630.2	Saturday	Wells St & Concord Ln	Clark St & Elm St
member	23	37734	12.52541	15.36043	629.1	Saturday	Wells St & Concord Ln	Clark St & Elm St

```
p4<- ggplot(membership_hours,aes(x=hour_start,y=N,color=member_casual))+
  geom_line(size=3)+
    labs(title = "Biker Casual Vs. Members",subtitle = "Number of rides per hour between 11-2020 and 10-2021",x="Hours",y="Number of rides", caption = "data provided by Cyclistic, a bike-share company in Chicago",fill = "User")+
  scale_color_viridis_d()

p4
```




```
p5<-
ggplot(membership_hours,aes(x=hour_start,y=average_ridelength_mins,color=member_casual))+
  geom_line(size=3)+
    labs(title = "Biker Casual Vs. Members",subtitle = "Length of rides per hour between 11-2020 and 10-2021",x="Hours",y="Average time of rides (minutes)", caption = "data provided by Cyclistic, a bike-share company in Chicago",fill = "User")+
```

```
  scale_color_viridis_d()

p5
```



4.4 Are the differences during the week days that we should account for?

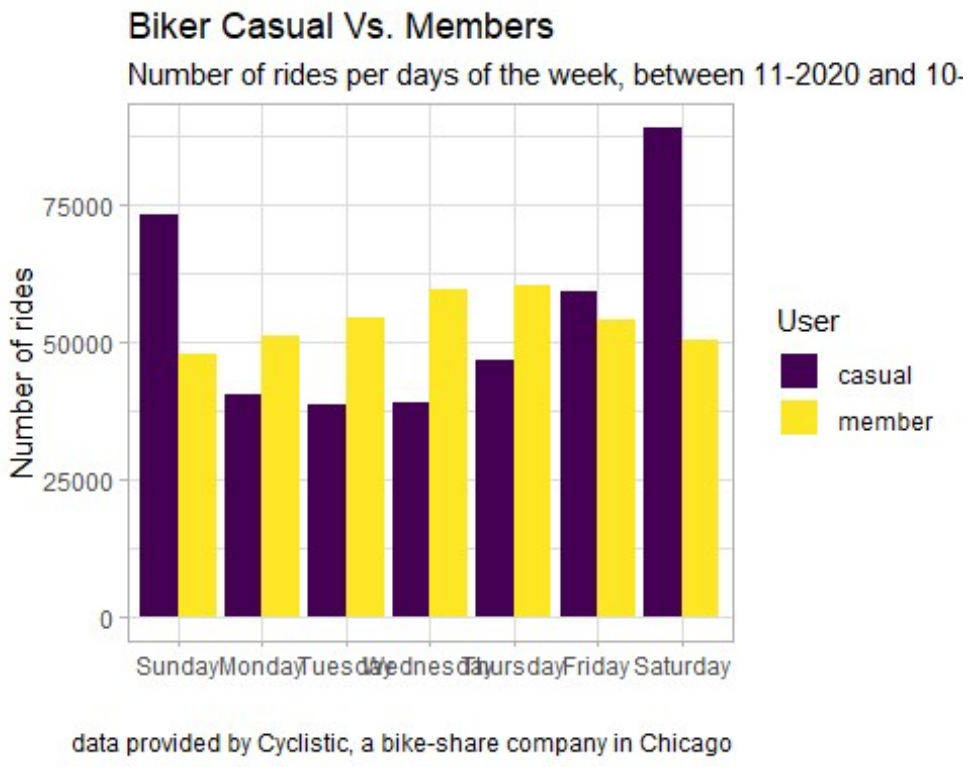
- Although members have the overall highest number of rides, casuals surpass it on Fridays, Saturdays and Sundays.
- Casuals always have the longest bike rides in every day of the week.

```
knitr::kable(head(membership_week_month))
```

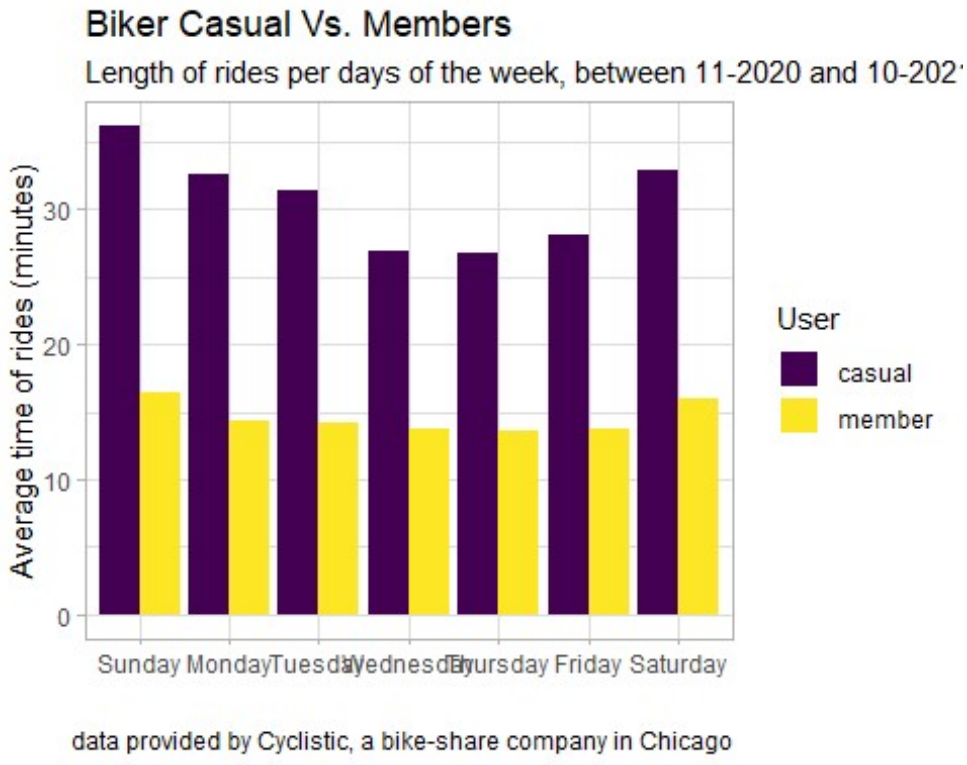
member_casual	week_day	month	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
casual	Sunday	Jan	2362	21.95174	26.91423	592.5	Sunday	Wells St & Elm St	Lake Shore Dr & Monroe St
casual	Sunday	Feb	1207	28.51939	35.29023	480.2	Sunday	Millennium Park	Millennium Park
casual	Sunday	Mar	1587 3	35.23982	37.48976	612.3	Sunday	Lake Shore Dr &	Lake Shore Dr &

member_casual	week_day	month	N	average_ridelength_mins	sd_ridelength	max_ridelength	mode_week	mode_start_station	mode_end_station
								Monroe St	Monroe St
casual	Sunday	Apr	22813	34.90732	39.13392	625.8	Sunday	Lake Shore Dr & Monroe St	Lake Shore Dr & Monroe St
casual	Sunday	May	53954	36.14129	40.52872	616.6	Sunday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave
casual	Sunday	Jun	58666	32.81691	40.37263	626.3	Sunday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave

```
p6<- ggplot(membership_week_month,aes(x=week_day,y=N,fill=member_casual))+
  geom_col(size=3,position = position_dodge())+
    labs(title = "Biker Casual Vs. Members",subtitle = "Number of rides per
days of the week, between 11-2020 and 10-2021",x="",y="Number of rides",
caption = "data provided by Cyclistic, a bike-share company in Chicago",fill
= "User")+
  scale_fill_viridis_d()

p6
```



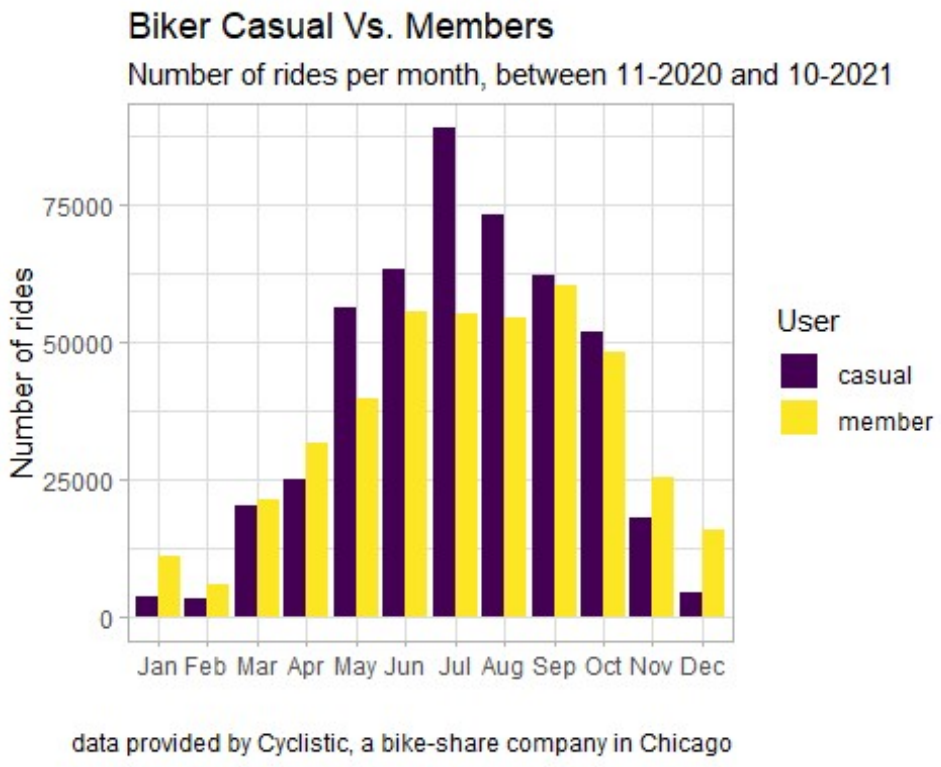

```
p7<-
ggplot(membership_week_month,aes(x=week_day,y=average_ridelength_mins,fill=me
mber_casual))+
  geom_col(size=3,position = position_dodge())+
    labs(title = "Biker Casual Vs. Members",subtitle = "Length of rides per
days of the week, between 11-2020 and 10-2021",x="",y="Average time of rides
(minutes)", caption = "data provided by Cyclistic, a bike-share company in
Chicago",fill = "User")+
  scale_fill_viridis_d()

p7
```



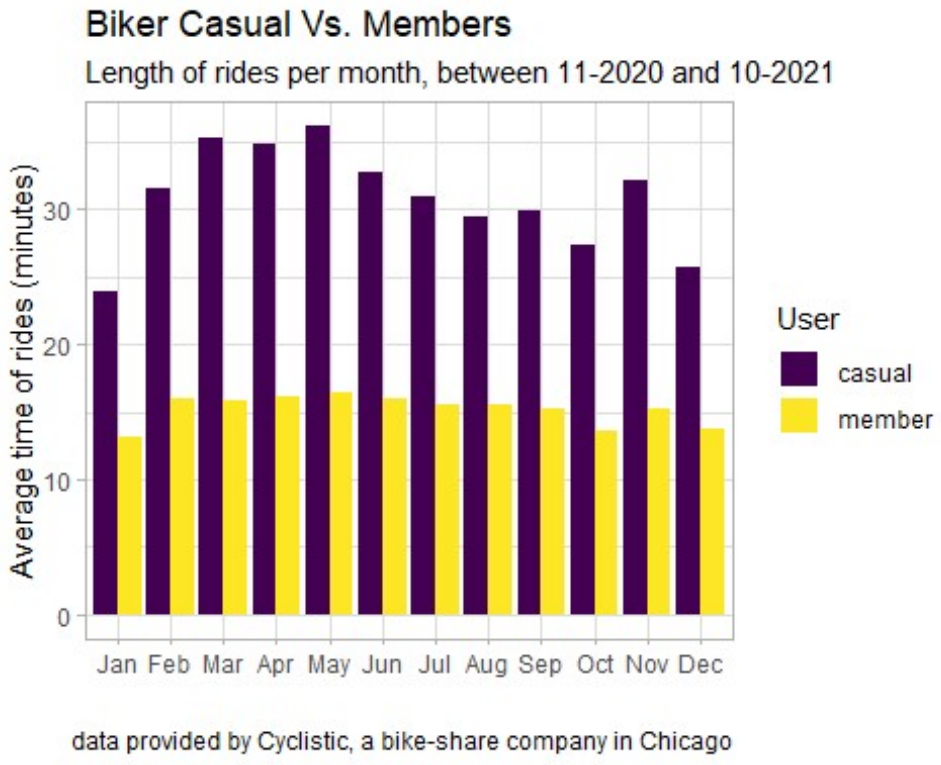
4.5 Are the differences during the months that we should account for?

- Number of rides fluctuate significantly during the year for both subscriptions. Both have a significant decrease in the number of rides between november 2020 and february 2021. Number of rides increase from march to july, and then start decreasing.
- For casuals, the number of rides peak in July.
- For members, the peak is in September, although this maximum is still lower than the number of casual rides.
- As for the length of the rides, the monthly average is always bigger for casual rather than members.
- The length of the rides is stable throuhout the year for members, while casuals have a higher fluctuation.

```
p8<- ggplot(membership_week_month,aes(x=month,y=N,fill=member_casual))+
  geom_col(size=3,position = position_dodge())+
    labs(title = "Biker Casual Vs. Members",subtitle = "Number of rides per
month, between 11-2020 and 10-2021",x="",y="Number of rides", caption = "data
provided by Cyclistic, a bike-share company in Chicago",fill = "User")+
  scale_fill_viridis_d()

p8
```




```
p9<-
ggplot(membership_week_month,aes(x=month,y=average_ridelength_mins,fill=membe
r_casual))+
  geom_col(size=3,position = position_dodge())+
    labs(title = "Biker Casual Vs. Members",subtitle = "Length of rides per
month, between 11-2020 and 10-2021",x="",y="Average time of rides (minutes)",
caption = "data provided by Cyclistic, a bike-share company in Chicago",fill
= "User")+
  scale_fill_viridis_d()

p9
```



5. Share

The results of this capstone project are now available at https://github.com/danfid15.

6. Act

6.1 Insights

In sum, the results of this analysis suggest the following profile for each type of consumer:

- **Casuals**: They have lower number of rides, but the longest ones (sometimes >20 minutes) when compared to members of this service. Nonetheless, their number of rides surpass Members during warmer months (March=August). There is a high fluctuation both on the length and number of rides over the year, low between November 2020 and March 2021, increases from April to July, and then progressively decreases. They use bikes mostly on Fridays, Saturdays and Sundays. Their peak of usage during the day is the afternoon, and prefer classic bikes over docker/ electric bikes.
- **Members**: They have the highest amount of rides, but these are usually shorter by comparison. Their usage of the service fluctuates significantly during the year, with barely use of bikes in February, and their peak of usage is in September. Nevertheless, unlike casuals, their ride length does not fluctuate significantly over the year. They use bikes throughout the week, and not just during weekends. Their peak of usage during the day is also in the afternoon, and they also prefer classic bikes over docker/ electric bikes.

6.2 Next steps

Design marketing strategies aimed at converting casual riders into annual members:

- Promote annual benefits for members during the weekends, which seems to be the favorite days for casuals.
- Focus these promotions between April and August, since its the peak of usage for both type of consumers.
- Focus the advertisement at Streeter Dr & Grand Ave, which the station most casuals use to both start and end most of the rides during the afternoon.

6.3 Additional data for future studies

- The trajectory of each ride.
- The cost of each ride for casuals.
- The cost of an annual subscription.