# Cyclistic Bike-share Analysis

Pius Mutuma Kimathi

2022-03-18

## Contents

#### 0.0.1 BACKGROUND

In 2016, Cyclistic launched a successful bike-share offering. The bikes can be unlocked from one station and returned to any other station in the system anytime.

Customers who purchase single-ride or full-day passes are referred to as Casual Riders. Customers who purchase annual memberships are Cyclistic member

#### 0.0.2 ASK

## 0.1 Key Objectives

1. Identify the business necessity: The company seeks to improve their earnings by converting their casual riders. Therefore, the company needs to analyze in what aspects the casual and the annual customers differ. This will enable the creation of a focused and successful marketing message to the casual customers that would convince them to change to the annual subscription.
2. Consider the main stakeholders: The team consists of the director of marketing, manager Lily Moreno, the rest of the marketing analytics team, and the Cyclistic executive team.
3. Main task: The company aims to convert casual riders into annual members by designing marketing strategies. The case study seeks to analyze the the Cyclistic historical bike trip data to identify trends.

Here, the question, ': How do annual members and casual riders use Cyclistic bikes differently?', seeks to be answered.

#### 0.1.1 PREPARE

1. First, the data is checked for credibility.

The data is public data attained from Cyclistic Bike-share company which is a bike sharing company. The data is organized monthly from the year 2020 until 2021. The naming of the columns and the overall data seems to be in good condition and its first hand data collected by the company itself with many entries and useful data.

2. Sorting and filtering the data:

The focus in this analysis is on the 2020-2021 period as it is a more relevant period to the business task, has Geo-location data, types of bike used, and rider types.

Here, the datasets are loaded for use.

# 1 Combining the data frames to one data frame and removing the empty columns and rows.

```
bike_rides <- rbind(Apr_04, May_05, Jun_06, Jul_07, Aug_08, Sep_09, Oct_10, Nov_11, Dec_12, Jan_01, Feb_
bike_rides <- janitor::remove_empty(bike_rides, which = c("cols"))
bike_rides <- janitor:: remove_empty(bike_rides, which = c("rows"))
```

### 1.0.1 PROCESS

Here, the data was cleaned and processed to ensure it was ready for analysis. Now that all the data is in one data frame, it should be cleaned to remove missing data. Also, new columns will be added from the existing columns in order to come up with a conclusive analysis.

# 2 Viewing the data and its structure

```
glimpse(bike_rides)
```

```
## Rows: 3,489,748
## Columns: 13
## $ ride_id            <chr> "A847FADBBC638E45", "5405B80E996FF60D", "5DD24A79A4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <chr> "2020-04-26 17:45:14", "2020-04-17 17:08:54", "2020~
## $ ended_at           <chr> "2020-04-26 18:12:03", "2020-04-17 17:17:03", "2020~
## $ start_station_name <chr> "Eckhart Park", "Drake Ave & Fullerton Ave", "McClu~
## $ start_station_id   <chr> "86", "503", "142", "216", "125", "173", "35", "434~
## $ end_station_name   <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszko Park", "~
## $ end_station_id     <chr> "152", "499", "255", "657", "323", "35", "635", "38~
## $ start_lat          <dbl> 41.8964, 41.9244, 41.8945, 41.9030, 41.8902, 41.896~
## $ start_lng          <dbl> -87.6610, -87.7154, -87.6179, -87.6975, -87.6262, -~
## $ end_lat            <dbl> 41.9322, 41.9306, 41.8679, 41.8992, 41.9695, 41.892~
## $ end_lng            <dbl> -87.6586, -87.7238, -87.6230, -87.6722, -87.6547, -~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

```
summary(bike_rides)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:3489748     Length:3489748     Length:3489748     Length:3489748
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_name start_station_id   end_station_name   end_station_id
##  Length:3489748     Length:3489748     Length:3489748     Length:3489748
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.64   Min.   :-87.87   Min.   :41.54   Min.   :-88.07
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.64   Mean   :41.90   Mean   :-87.64
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.08   Max.   :-87.52   Max.   :42.16   Max.   :-87.44
##                                   NA's   :4738    NA's   :4738
##  member_casual
##  Length:3489748
##  Class :character
##  Mode  :character
##
##
##
##
```

## 3 Parsing Time and creating a hour field

```
bike_rides$date <- as.Date(bike_rides$started_at)
bike_rides$started_at <- lubridate::ymd_hms(bike_rides$started_at)
bike_rides$ended_at <- lubridate::ymd_hms(bike_rides$ended_at)

bike_rides$start_hour <- lubridate::hour(bike_rides$started_at)
bike_rides$end_hour <- lubridate::hour(bike_rides$ended_at)
```

## 4 Creating date, Day, Year, Day of week, and Month column

```
bike_rides$Month <- format(as.Date(bike_rides$date), "%m")
bike_rides$Day <- format(as.Date(bike_rides$date), "%d")
bike_rides$Year <- format(as.Date(bike_rides$date), "%Y")
bike_rides$Day_of_week <- format(as.Date(bike_rides$date), "%A")

str(bike_rides)
```

```
## 'data.frame':    3489748 obs. of  20 variables:
##  $ ride_id           : chr  "A847FADBBC638E45" "5405B80E996FF60D" "5DD24A79A4E006F4" "2A59BBDF5CDBA7
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct, format: "2020-04-26 17:45:14" "2020-04-17 17:08:54" ...
##  $ ended_at          : POSIXct, format: "2020-04-26 18:12:03" "2020-04-17 17:17:03" ...
##  $ start_station_name: chr  "Eckhart Park" "Drake Ave & Fullerton Ave" "McClurg Ct & Erie St" "Calif
##  $ start_station_id  : chr  "86" "503" "142" "216" ...
##  $ end_station_name  : chr  "Lincoln Ave & Diversey Pkwy" "Kosciuszko Park" "Indiana Ave & Roosevelt
##  $ end_station_id    : chr  "152" "499" "255" "657" ...
```

```
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 42 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ date              : Date, format: "2020-04-26" "2020-04-17" ...
##  $ start_hour        : int  17 17 17 12 10 17 14 12 10 15 ...
##  $ end_hour          : int  18 17 18 13 11 18 14 13 10 15 ...
##  $ Month             : chr  "04" "04" "04" "04" ...
##  $ Day               : chr  "26" "17" "01" "07" ...
##  $ Year              : chr  "2020" "2020" "2020" "2020" ...
##  $ Day_of_week       : chr  "Sunday" "Friday" "Wednesday" "Tuesday" ...
```

# 5 Creating ride distance column

```
bike_rides$ride_distance <- distGeo(matrix(c(bike_rides$start_lng, bike_rides$start_lat), ncol = 2), ma
bike_rides$ride_distance <- bike_rides$ride_distance/1000
```

# 6 Converting ride_length to minutes and hours

```
bike_rides$ride_length_by_minutes <- difftime(bike_rides$ended_at, bike_rides$started_at, units = c("mi
bike_rides$ride_length_by_hour <- difftime(bike_rides$ended_at, bike_rides$started_at, units = c("hours

str(bike_rides)
```

```
## 'data.frame':    3489748 obs. of  23 variables:
##  $ ride_id               : chr  "A847FADBBC638E45" "5405B80E996FF60D" "5DD24A79A4E006F4" "2A59BBDF5C
##  $ rideable_type         : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at            : POSIXct, format: "2020-04-26 17:45:14" "2020-04-17 17:08:54" ...
##  $ ended_at              : POSIXct, format: "2020-04-26 18:12:03" "2020-04-17 17:17:03" ...
##  $ start_station_name    : chr  "Eckhart Park" "Drake Ave & Fullerton Ave" "McClurg Ct & Erie St" "C
##  $ start_station_id      : chr  "86" "503" "142" "216" ...
##  $ end_station_name      : chr  "Lincoln Ave & Diversey Pkwy" "Kosciuszko Park" "Indiana Ave & Roose
##  $ end_station_id        : chr  "152" "499" "255" "657" ...
##  $ start_lat             : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng             : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat               : num  41.9 41.9 41.9 41.9 42 ...
##  $ end_lng               : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual         : chr  "member" "member" "member" "member" ...
##  $ date                  : Date, format: "2020-04-26" "2020-04-17" ...
##  $ start_hour            : int  17 17 17 12 10 17 14 12 10 15 ...
##  $ end_hour              : int  18 17 18 13 11 18 14 13 10 15 ...
##  $ Month                 : chr  "04" "04" "04" "04" ...
##  $ Day                   : chr  "26" "17" "01" "07" ...
##  $ Year                  : chr  "2020" "2020" "2020" "2020" ...
##  $ Day_of_week           : chr  "Sunday" "Friday" "Wednesday" "Tuesday" ...
##  $ ride_distance         : num  3.98 0.98 2.98 2.14 9.12 ...
##  $ ride_length_by_minutes: 'difftime' num  26.8166666666667 8.15 14.3833333333333 12.2 ...
```

```
##   ..- attr(*, "units")= chr "mins"
##  $ ride_length_by_hour   : 'difftime' num  0.446944444444444 0.135833333333333 0.239722222222222 0.2
##   ..- attr(*, "units")= chr "hours"
```

```
#The speed in Km/h
bike_rides$ride_speed = c(bike_rides$ride_distance)/as.numeric(c(bike_rides$ride_length_by_hour))
```

# 7 Removing bad data and NAs

```
#Removing bad data

bike_rides_2 <- bike_rides %>%
  filter (bike_rides$ride_length_by_minutes > 0) %>% drop_na()

str(bike_rides_2)
```

```
## 'data.frame':    3343689 obs. of  24 variables:
##  $ ride_id               : chr  "A847FADBBC638E45" "5405B80E996FF60D" "5DD24A79A4E006F4" "2A59BBDF5C
##  $ rideable_type         : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at            : POSIXct, format: "2020-04-26 17:45:14" "2020-04-17 17:08:54" ...
##  $ ended_at              : POSIXct, format: "2020-04-26 18:12:03" "2020-04-17 17:17:03" ...
##  $ start_station_name    : chr  "Eckhart Park" "Drake Ave & Fullerton Ave" "McClurg Ct & Erie St" "Ca
##  $ start_station_id      : chr  "86" "503" "142" "216" ...
##  $ end_station_name      : chr  "Lincoln Ave & Diversey Pkwy" "Kosciuszko Park" "Indiana Ave & Roosev
##  $ end_station_id        : chr  "152" "499" "255" "657" ...
##  $ start_lat             : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng             : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat               : num  41.9 41.9 41.9 41.9 42 ...
##  $ end_lng               : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual         : chr  "member" "member" "member" "member" ...
##  $ date                  : Date, format: "2020-04-26" "2020-04-17" ...
##  $ start_hour            : int  17 17 17 12 10 17 14 12 10 15 ...
##  $ end_hour              : int  18 17 18 13 11 18 14 13 10 15 ...
##  $ Month                 : chr  "04" "04" "04" "04" ...
##  $ Day                   : chr  "26" "17" "01" "07" ...
##  $ Year                  : chr  "2020" "2020" "2020" "2020" ...
##  $ Day_of_week           : chr  "Sunday" "Friday" "Wednesday" "Tuesday" ...
##  $ ride_distance         : num  3.98 0.98 2.98 2.14 9.12 ...
##  $ ride_length_by_minutes: 'difftime' num  26.8166666666667 8.15 14.3833333333333 12.2 ...
##   ..- attr(*, "units")= chr "mins"
##  $ ride_length_by_hour   : 'difftime' num  0.446944444444444 0.135833333333333 0.239722222222222 0.2
##   ..- attr(*, "units")= chr "hours"
##  $ ride_speed            : num  8.91 7.21 12.45 10.53 10.34 ...
```

### 7.0.1 ANALYZE

Here, we seek to identify trends and relationships. The data frame is available with the needed columns and will help understand the differences in behavior between the casual and member riders.

# 8 Creating a summary data frame of bike rides

```
bike_rides_summary <- bike_rides_2 %>%
  group_by(weekly = floor_date(date, "week"), start_hour, end_hour, member_casual) %>%
  summarize(
    Minutes = sum(ride_length_by_minutes),
    Mean = mean(ride_length_by_minutes),
    Median = median(ride_length_by_minutes),
    Max = max(ride_length_by_minutes),
    Min = min(ride_length_by_minutes),
    Count = n()
  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'weekly', 'start_hour', 'end_hour'. You can
## override using the `.groups` argument.
```

# 9 Creating a summary data frame of bike Types and plotting count of rides by rider type

```
bike_type <- bike_rides_2 %>%
  group_by(member_casual, rideable_type, weekly = floor_date(date, "week")) %>%
  summarize(
    Minutes = sum(ride_length_by_minutes),
    Mean_mins = mean(ride_length_by_minutes),
    mean_hour = mean(ride_length_by_hour),
    Median_mins = median(ride_length_by_minutes),
    Max_mins = max(ride_length_by_minutes),
    Min_mins = min(ride_length_by_minutes),
    mean_distance = mean(ride_distance),
    Count = n()
  ) %>%
  ungroup()
```

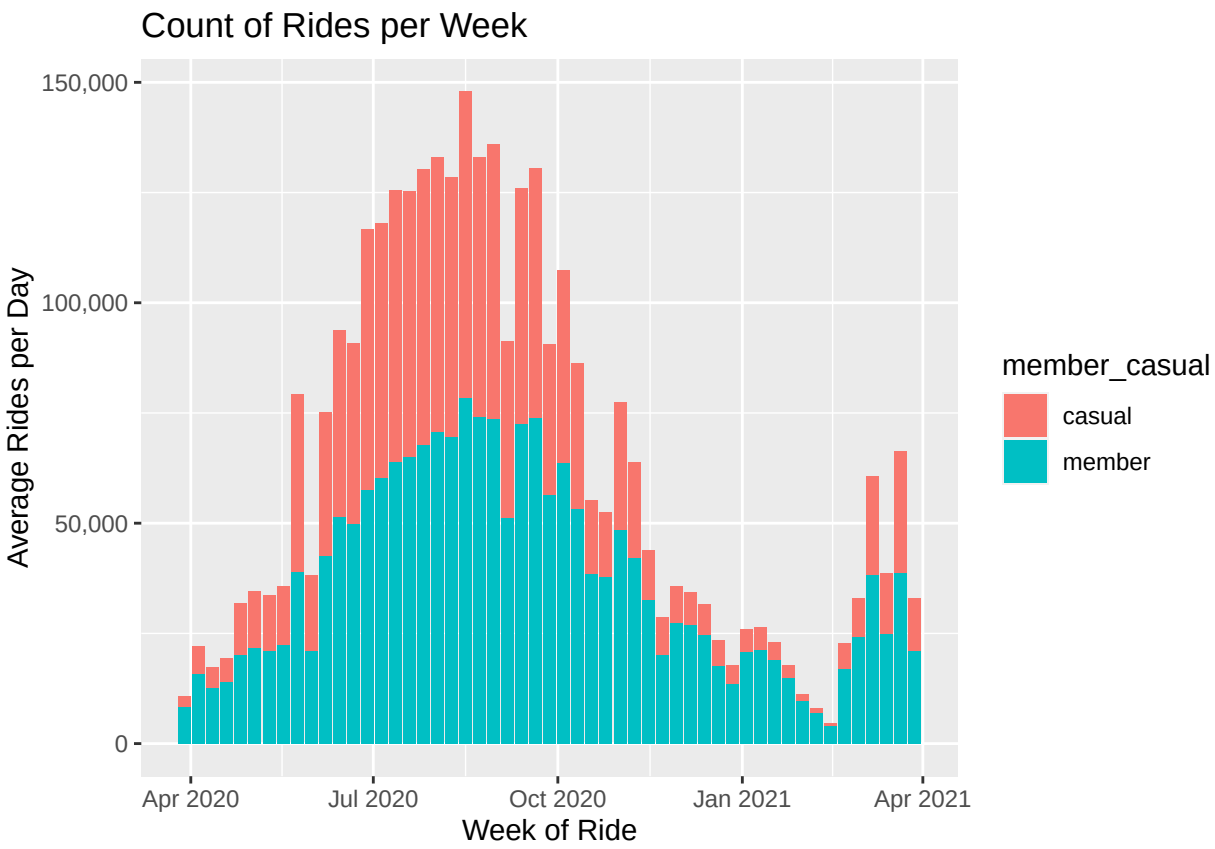
```
## `summarise()` has grouped output by 'member_casual', 'rideable_type'. You can
## override using the `.groups` argument.
```

# 10 Total count rides by week

```
summary(bike_rides_summary$Count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       4     224     103    6361
```

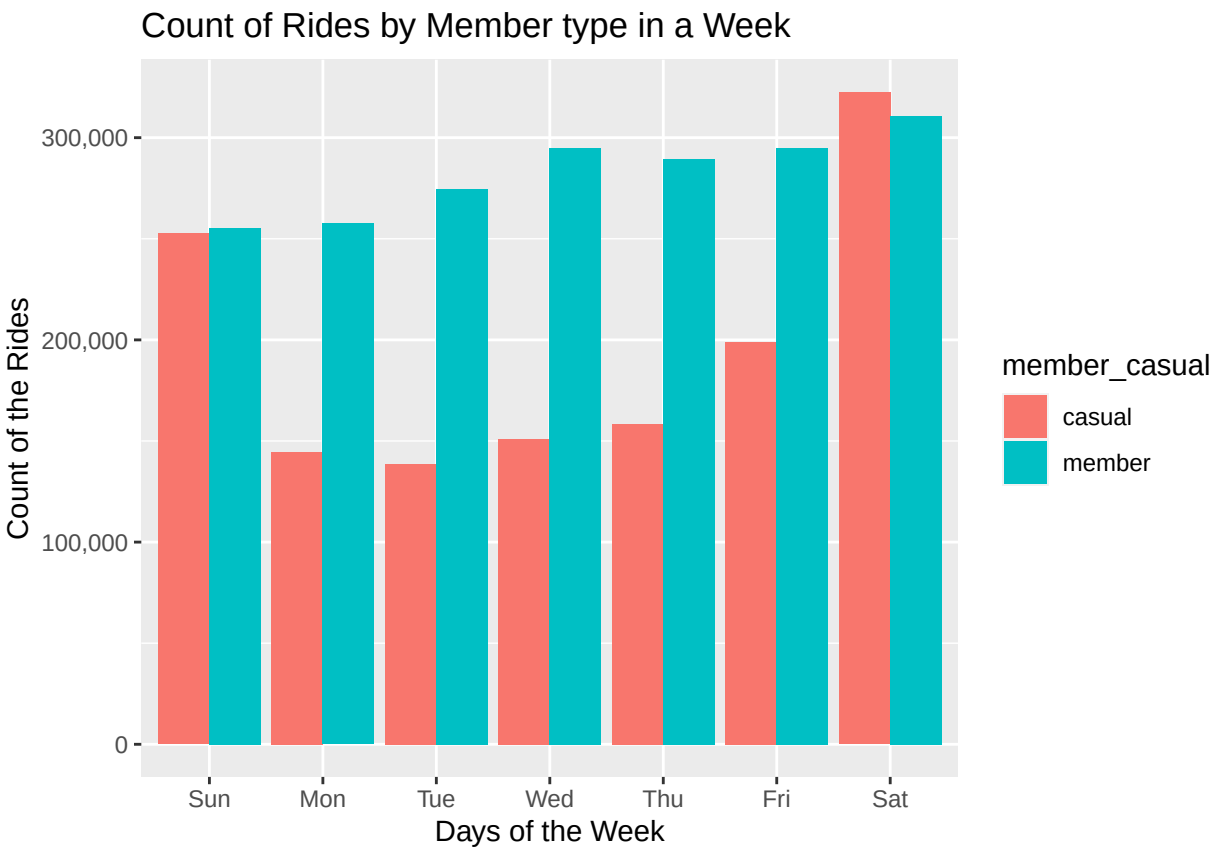
```
bike_rides_summary$Monthly <- lubridate::month(bike_rides_summary$weekly)

#Count of Rides Per week
bike_rides_summary %>% ggplot() + geom_col(aes(x = weekly, y = Count, fill = member_casual)) +
  scale_y_continuous(labels = comma) +
  labs(title = "Count of Rides per Week",
       x = 'Week of Ride',
       y = "Average Rides per Day")
```




The casual riders have a higher ride count than the annual members. It can be assumed that the casual riders utilize the bikes for leisure than the annual members.

## 11 Number of rides by Rider Type in a Week

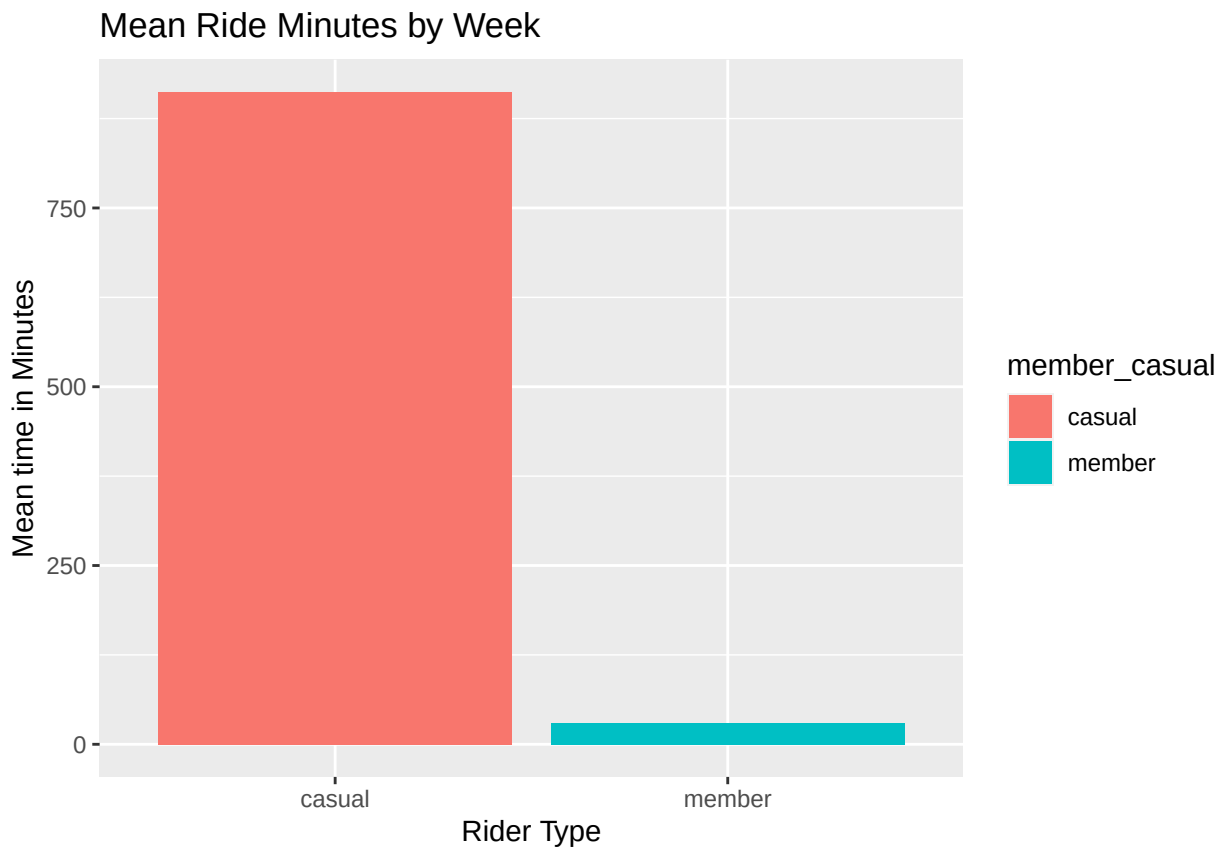
```
bike_rides_2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(num_of_rides = n(), .groups = 'drop') %>%
  ggplot(aes(x = weekday, y = num_of_rides, fill = member_casual)) +
  geom_col(position = 'dodge') +
  scale_y_continuous(labels = comma) +
  labs(title = "Count of Rides by Member type in a Week",
       x = 'Days of the Week',
       y = "Count of the Rides")
```




1. Here, the annual members have a higher daily usage of the rides except on Saturday.
2. The casual riders tend to use the bikes for leisure as their rides peak more on weekends than on weekdays while annual members use the bikes as a formal type of transport i.e. to work.

## 12 Mean Travel time by Member Type

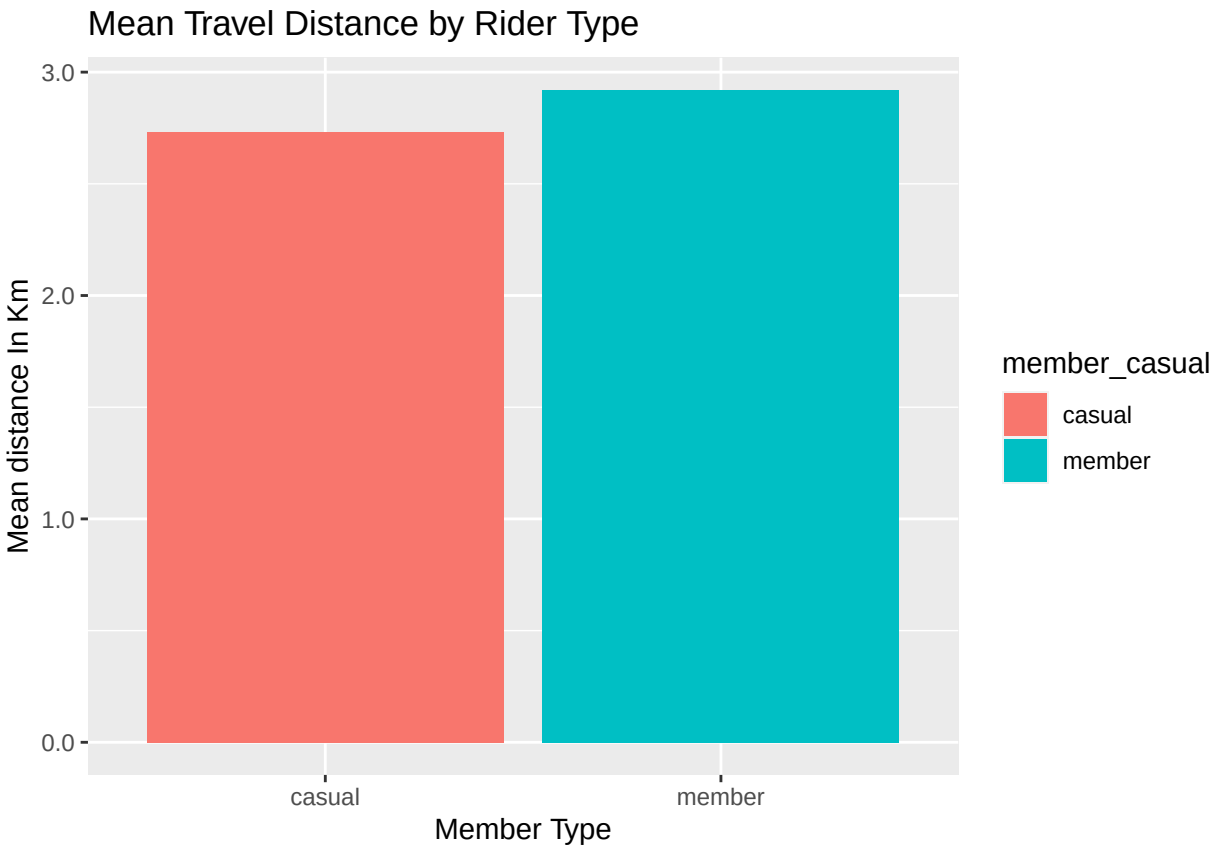
```
ggplot(bike_type, aes(x = member_casual, y = Mean_mins, fill = member_casual)) +
geom_col(position = 'dodge') +
scale_y_continuous(labels = comma) +
labs(title = "Mean Ride Minutes by Week",
     x="Rider Type",y = "Mean time in Minutes")
```




The casual members had a higher mean ride in minutes weekly which denotes their high usage.

## 13 Mean Travel distance by Member Type

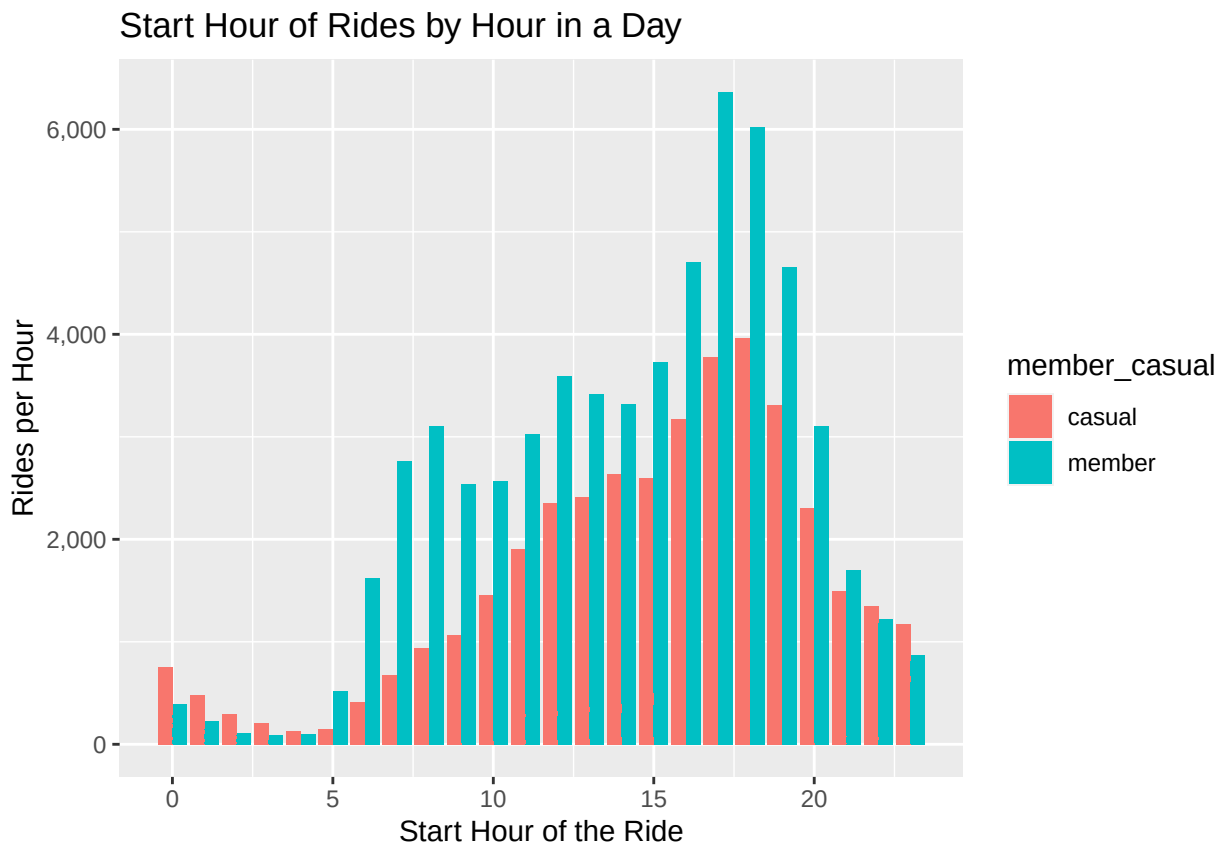
```
ggplot(bike_type, aes(x = member_casual, y = mean_distance, fill = member_casual)) +
geom_col(position = 'dodge') +
scale_y_continuous(labels = comma) +
labs(title = 'Mean Travel Distance by Rider Type', x = "Member Type", y = "Mean distance In Km")
```




The annual members had a higher mean travel distance than the casual members.

## 14 Count of Start of Rides by Hours

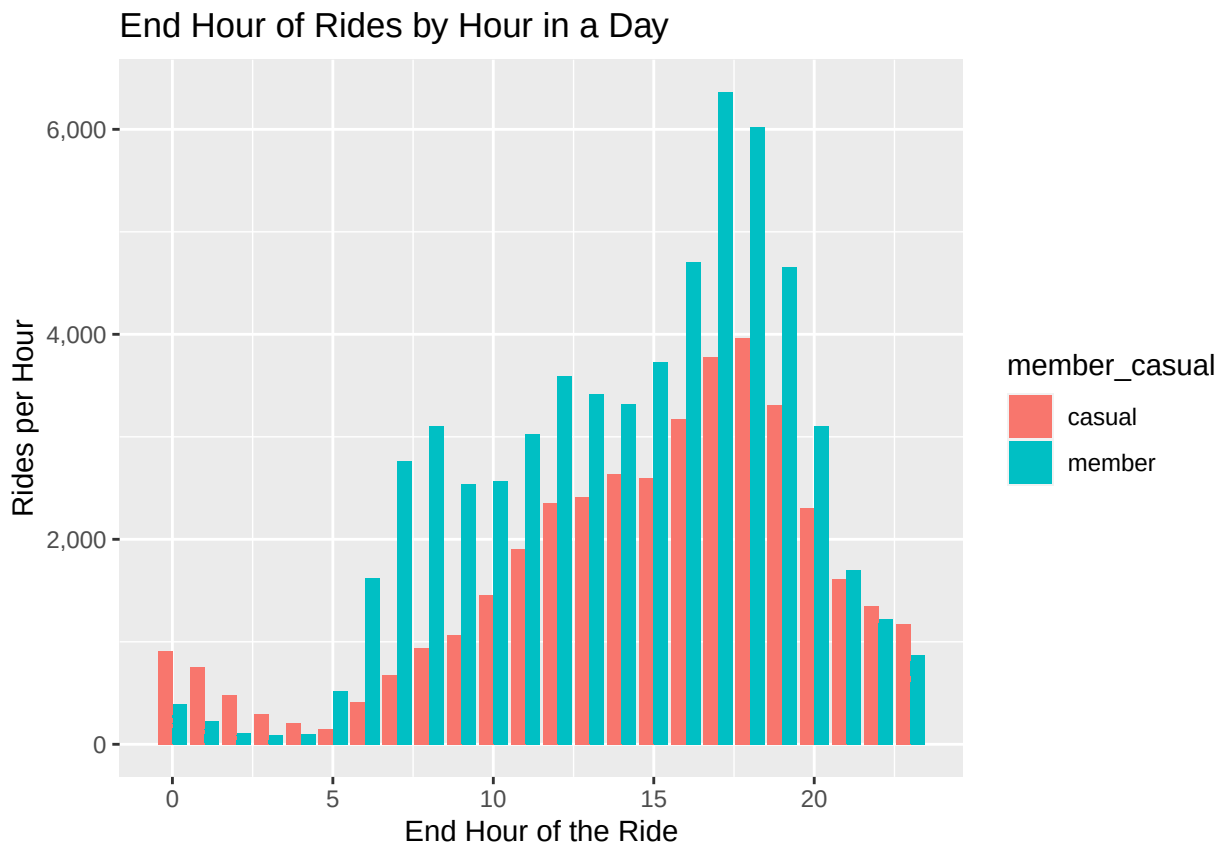
```
ggplot(bike_rides_summary, aes(x = start_hour, y = Count, fill = member_casual)) + geom_col(position =
  scale_y_continuous(labels = comma) +
  labs(title = "Start Hour of Rides by Hour in a Day",
       x = 'Start Hour of the Ride',
       y = "Rides per Hour")
```




The rides peak at 5 PM by start hour for all types of riders.

## 15 Count of End of Rides by Hours

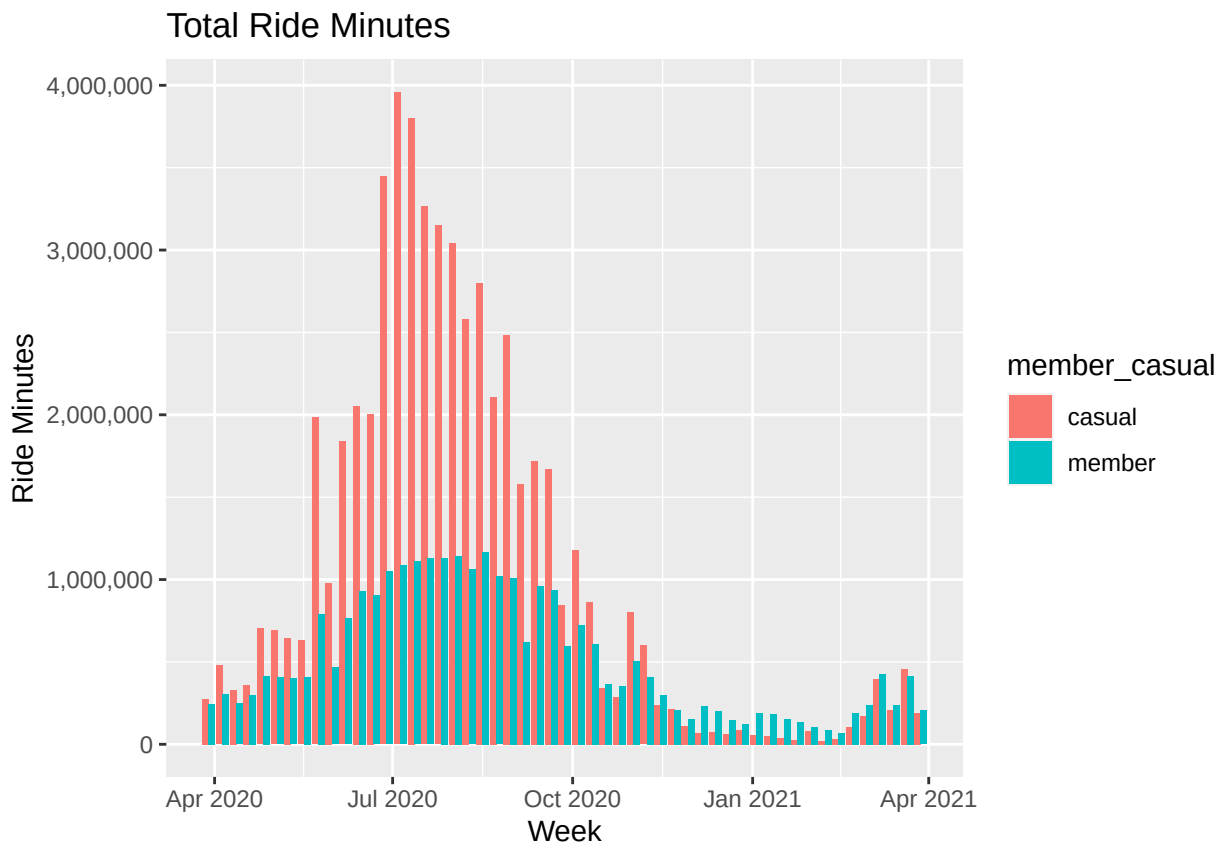
```
ggplot(bike_rides_summary, aes(x = end_hour, y = Count, fill = member_casual)) + geom_col(position = 'd
  scale_y_continuous(labels = comma) +
  labs(title = "End Hour of Rides by Hour in a Day",
       x = 'End Hour of the Ride',
       y = "Rides per Hour")
```




The rides end at 5 PM by end hour for all types of riders.

## 16 Total Ride Minutes by Week per member type

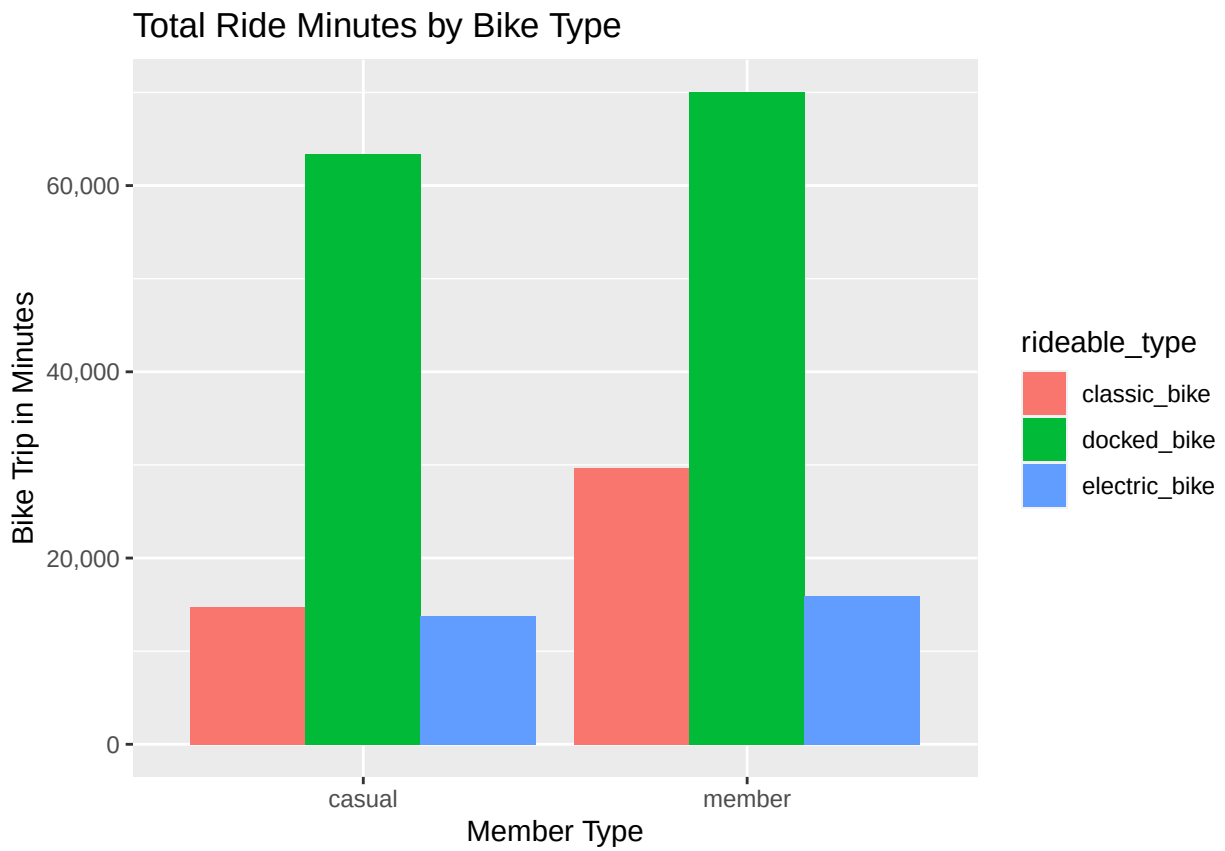
```
ggplot(bike_type, aes(x = weekly, y = Minutes, fill = member_casual)) +
  geom_col(position = 'dodge') +
  scale_y_continuous(labels = comma) +
  labs(title = "Total Ride Minutes",
       x = 'Week',
       y = 'Ride Minutes')
```




The casual riders had a higher count of rides by minutes which showcased their high usage for the bikes than the annual members.

## 17 Total Ride Minutes by Bike Type

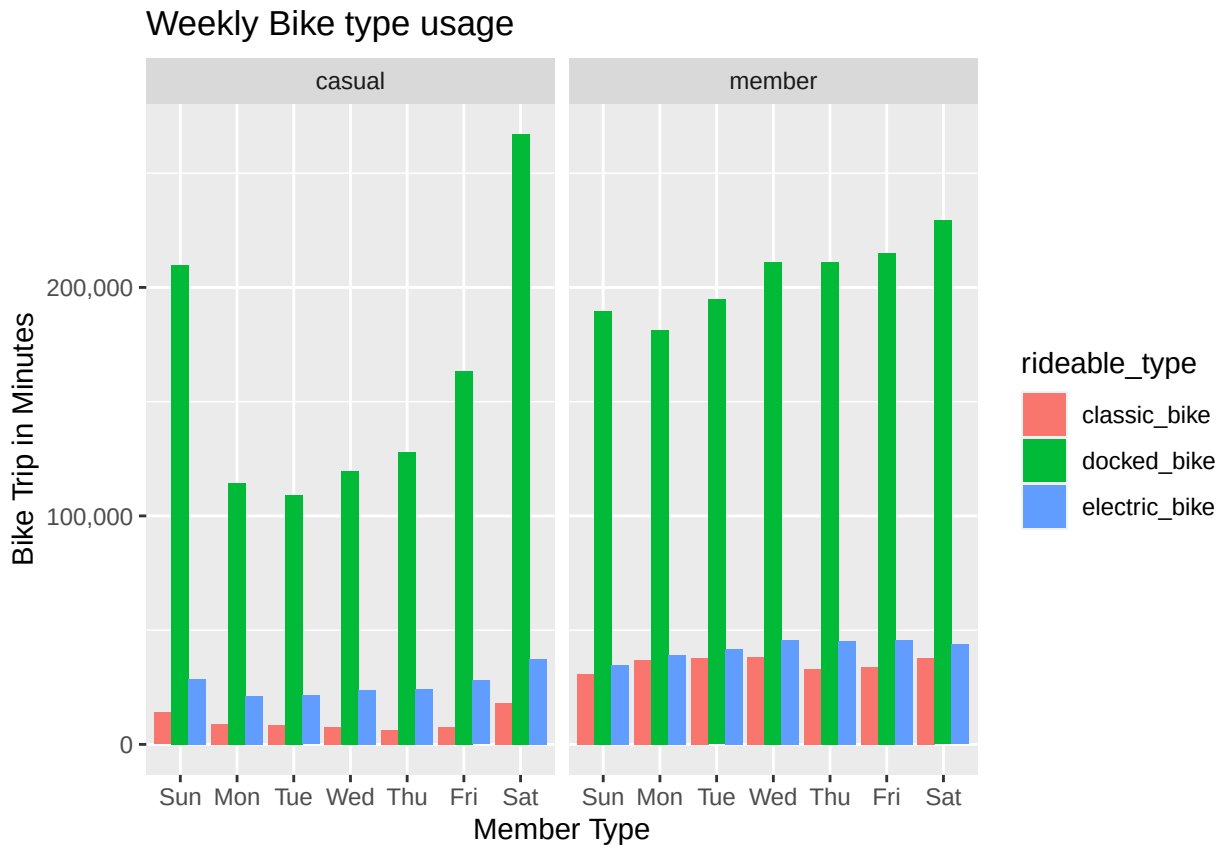
```
ggplot(bike_type, aes(x = member_casual, y = Count, fill = rideable_type)) +
  geom_col(position = 'dodge') +
  scale_y_continuous(labels = comma) +
  labs(title = "Total Ride Minutes by Bike Type",
       x = 'Member Type',
       y = "Bike Trip in Minutes")
```




Both types of members had similar preferences of Bike type with the preferred type of bike being the docked.

## 18 Weekly Bike Type usage

```
bike_rides_2 %>%
  mutate(Weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, rideable_type, Weekday) %>%
  summarise(Total = n(), .groups = 'drop') %>%
  ggplot(aes(x = Weekday, y = Total, fill = rideable_type)) +
  geom_col(position = 'dodge') +
  facet_wrap(~ member_casual) +
  scale_y_continuous(labels = comma) +
  labs(title = "Weekly Bike type usage",
       x = 'Member Type',
       y = "Bike Trip in Minutes")
```



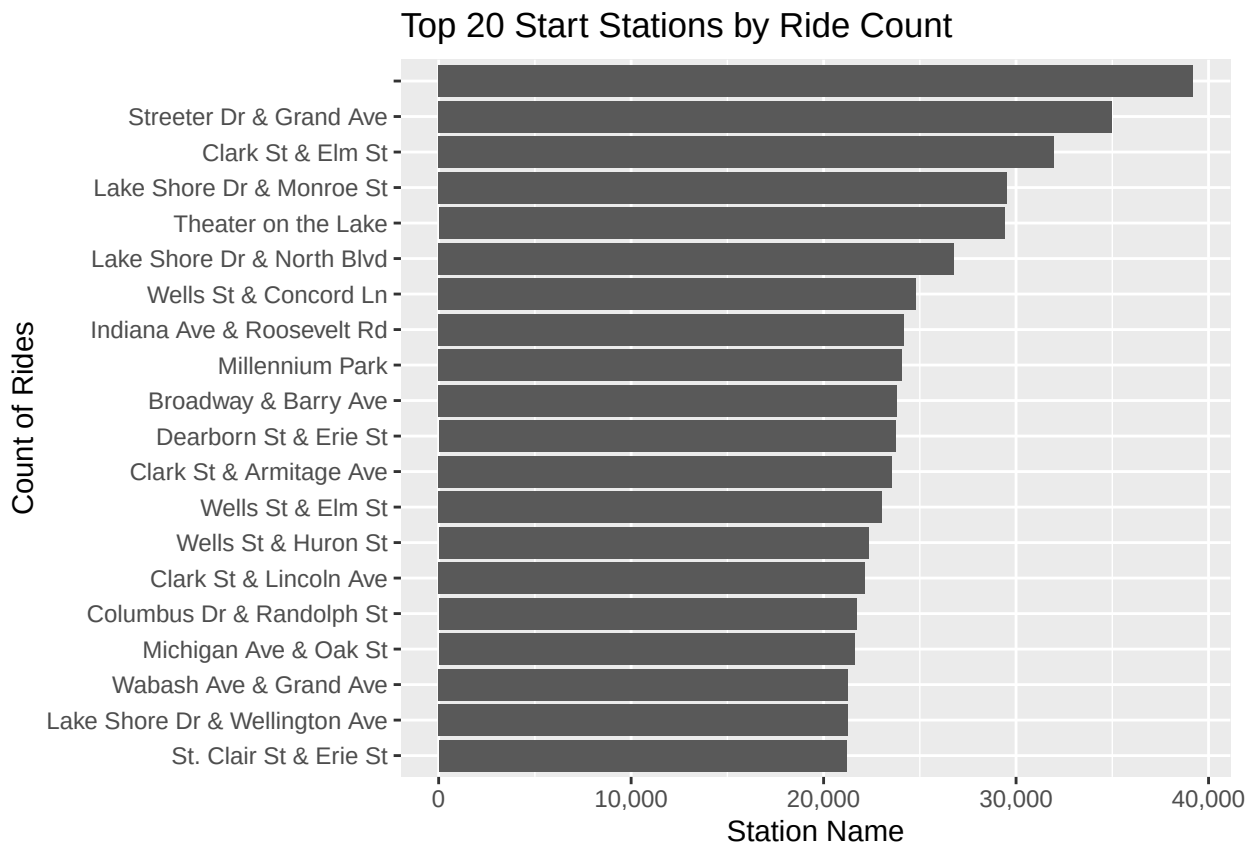

Both types of members maintained their preference of a docked bike.

Casual riders peaked their usage of the docked bike on weekends compared to the annual members who maintained their usage of the docked bike over the week peaking on Saturdays.

# 19 Top 20 Start Station Name by Ride Count

```
bike_rides_2 %>%
  count(start_station_name, sort = TRUE) %>%
  top_n(20) %>%
  ggplot() + geom_col(aes(x = fct_reorder(start_station_name,n), y = n)) +
  coord_flip() +
  labs(title = "Top 20 Start Stations by Ride Count",
       y = "Station Name",
       x = "Count of Rides") +
  scale_y_continuous(labels = comma)
```

```
## Selecting by n
```




## 20 Filtering the most popular routes and creating a table and boundary coordinates

```
#Most Popular Routes
Routes_table <- bike_rides_2 %>%
  filter(start_lng != end_lng & start_lat != end_lat) %>%
  group_by(start_lng, start_lat, end_lng, end_lat, member_casual, rideable_type) %>%
  summarise(total = n(),.groups="drop") %>%
  filter(total > 250)

#Creating tables for the user types
casual_riders <- Routes_table %>%
  filter(member_casual == 'casual')
member_riders <- Routes_table %>%
  filter(member_casual == 'member')

#Creating the boundary coordinates for the ggmap
bouding_b_cord <- c(left = -87.700424,
                    bottom = 41.790769,
                    right = -87.554855,
                    top = 41.990119
                    )
```

```
#Storing the stamen map of the area
Area_stamen <- get_stamenmap(
  bbox = bouding_b_cord,
  zoom = 10,
  maptype = "toner-lite")
```

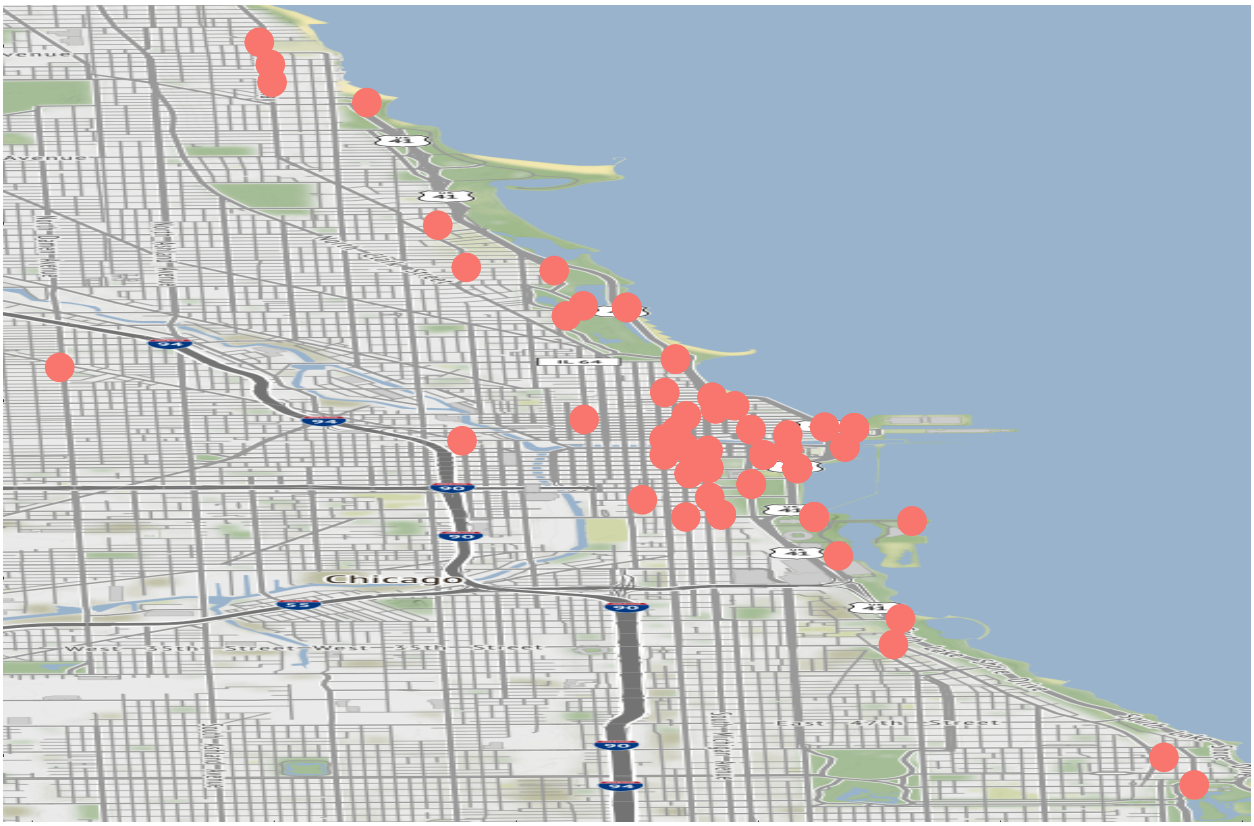
```
## Source : http://tile.stamen.com/toner-lite/10/262/380.png
```

```
# Visualizing by casual riders routes
qmplot(x = start_lng, y = start_lat, xend = end_lng, yend = end_lat,
       data = casual_riders, maptype = 'terrain', geom = 'point',
       color = rideable_type, size = 0.5) +
  coord_cartesian() +
  labs(title = "The popular routes used by Casual Riders",x = NULL,y = NULL, color="User type") +
  theme(legend.position = "none")
```

```
## Using zoom = 13...

## Source : http://tile.stamen.com/terrain/13/2100/3040.png

## Source : http://tile.stamen.com/terrain/13/2101/3040.png

## Source : http://tile.stamen.com/terrain/13/2102/3040.png

## Source : http://tile.stamen.com/terrain/13/2103/3040.png

## Source : http://tile.stamen.com/terrain/13/2100/3041.png

## Source : http://tile.stamen.com/terrain/13/2101/3041.png

## Source : http://tile.stamen.com/terrain/13/2102/3041.png

## Source : http://tile.stamen.com/terrain/13/2103/3041.png

## Source : http://tile.stamen.com/terrain/13/2100/3042.png

## Source : http://tile.stamen.com/terrain/13/2101/3042.png

## Source : http://tile.stamen.com/terrain/13/2102/3042.png

## Source : http://tile.stamen.com/terrain/13/2103/3042.png

## Source : http://tile.stamen.com/terrain/13/2100/3043.png

## Source : http://tile.stamen.com/terrain/13/2101/3043.png

## Source : http://tile.stamen.com/terrain/13/2102/3043.png
```

```
## Source : http://tile.stamen.com/terrain/13/2103/3043.png

## Source : http://tile.stamen.com/terrain/13/2100/3044.png

## Source : http://tile.stamen.com/terrain/13/2101/3044.png

## Source : http://tile.stamen.com/terrain/13/2102/3044.png

## Source : http://tile.stamen.com/terrain/13/2103/3044.png

## Source : http://tile.stamen.com/terrain/13/2100/3045.png

## Source : http://tile.stamen.com/terrain/13/2101/3045.png

## Source : http://tile.stamen.com/terrain/13/2102/3045.png

## Source : http://tile.stamen.com/terrain/13/2103/3045.png

## Source : http://tile.stamen.com/terrain/13/2100/3046.png

## Source : http://tile.stamen.com/terrain/13/2101/3046.png

## Source : http://tile.stamen.com/terrain/13/2102/3046.png

## Source : http://tile.stamen.com/terrain/13/2103/3046.png

## Source : http://tile.stamen.com/terrain/13/2100/3047.png

## Source : http://tile.stamen.com/terrain/13/2101/3047.png

## Source : http://tile.stamen.com/terrain/13/2102/3047.png

## Source : http://tile.stamen.com/terrain/13/2103/3047.png

## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

## The popular routes used by Casual Riders



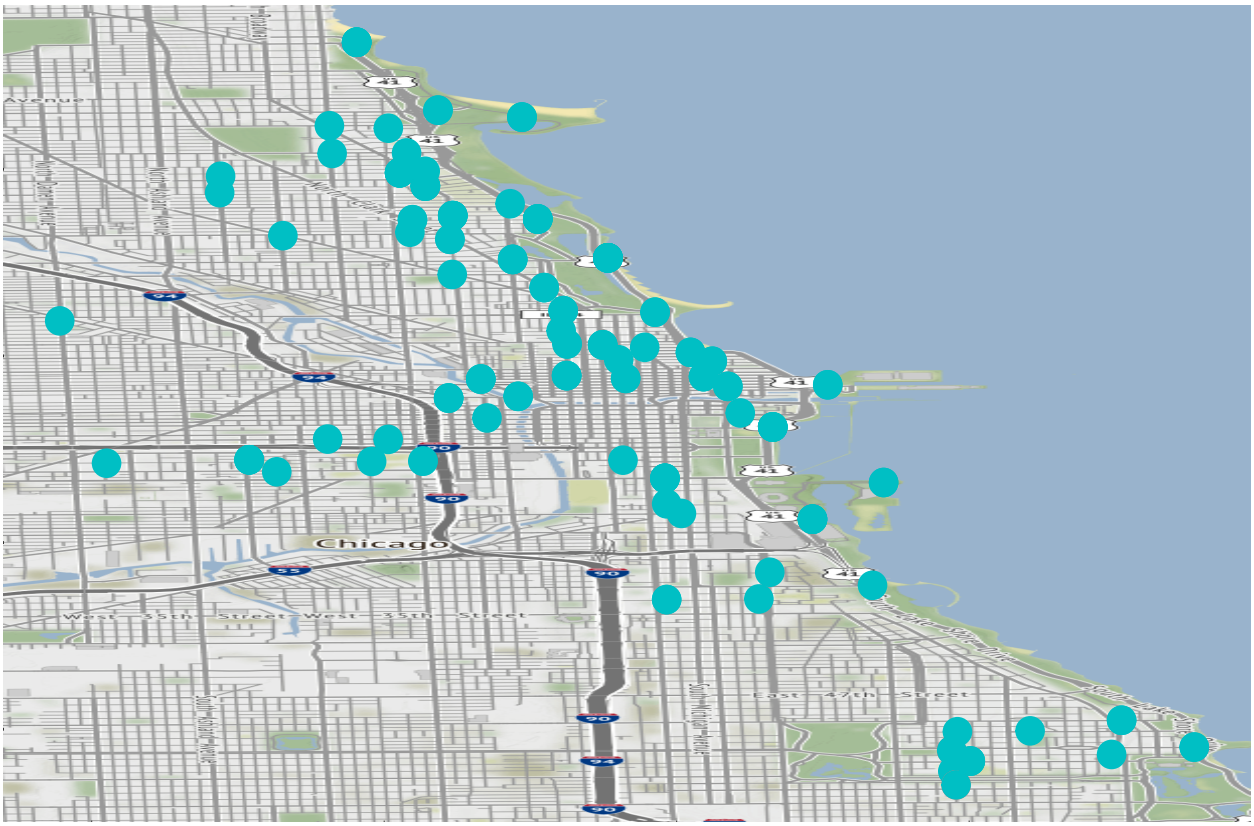

The Casual riders seem to be closely located in one area since almost all their rides are around one area. This indicates they use their rides for leisure and seem to ride in one area mostly.

```
#Visualizing by annual members routes
qmplot(x = start_lng, y = start_lat, xend = end_lng, yend = end_lat,
       data = member_riders, maptype = 'terrain', geom = 'point',
       color = rideable_type, size = 0.5) +
  coord_cartesian() +
  labs(title = "The popular routes used by Annual members",x = NULL,y = NULL, color="User type") +
  theme(legend.position = "none")
```

```
## Using zoom = 13...
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

## The popular routes used by Annual members




The annual riders are widespread to the outskirts of the city. This suggests that they use their rides for more than just leisure. The riders can be seen to use the bikes for activities such as commuting to work in the city daily.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.