PortfolioProject2

Yeonkyung Seo

2022-10-09

Bike share Case Study

Differnt bike use tendency between casual riders and members

1. Install and load the packages for the analysis

```
#install and load packages
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/User/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:WUsersWUserWAppDataWLocalWTempWRtmpeiVezOWdownloaded_packages
```

```
library(tidyverse)
```

```
## ── Attaching packages
## ──────────────────────────────────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.1
## ✓ readr   2.1.2     ✓ forcats 0.5.2
## ── Conflicts ──────────────────────────────────────────────────────────────────
── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
install.packages("lubridate", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/User/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'lubridate' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'lubridate'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:WUsersWUserWAppDataWLocalWRWwin-
## libraryW4.2WOOLOCKWlubridateWlibsWx64Wlubridate.dll to C:
## WUsersWUserWAppDataWLocalWRWwin-libraryW4.2WlubridateWlibsWx64Wlubridate.dll:
## Permission denied
```

```
## Warning: restored 'lubridate'
```

```
##
## The downloaded binary packages are in
##  C:WUsersWUserWAppDataWLocalWTempWRtmpeiVez0Wdownloaded_packages
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
install.packages("ggplot2", repos = "http://cran.us.r-project.org")
```

```
## Warning: package 'ggplot2' is in use and will not be installed
```

```
library(ggplot2)
```

2. Import the 12 months datasets and bind them into a complete dataset

```
#Import datasets from Sep 2021 to Aug 2022 and bind them all
filepath <- "data/202109-divvy-tripdata.csv"
new_dataset <- read.csv(filepath)
repeat{
 dateofdata <- as.integer(substr(filepath, 6, 11))+1
 substr(filepath, 6, 11) <- as.character(dateofdata)
 second_import <- read.csv(filepath)
 new_dataset <- rbind(new_dataset, second_import)
 if(dateofdata == 202112){
   filepath <- "data/202201-divvy-tripdata.csv"
   second_import <- read.csv(filepath)
   new_dataset <- rbind(new_dataset, second_import)
 }else if(dateofdata == 202208){
   break
 }
}
```

3. Split the dataset into two datasets(time, location)

```
#Prepare a dataset to inspect its datetime data
dt_dataset <- select(new_dataset, 'ride_id', 'started_at', 'ended_at', 'member_casual')

#Prepare a dataset to inspect geographic data
lc_dataset <- select(new_dataset, 'ride_id', 'start_station_name', 'end_station_name', 'member_casua
l')
```

4. Prepare the dataframes

- Convert the date columns from character to date type.
- Extract the month column and wday column from the date column.
- Reorder the weekdays, because it's not in the right order.
- Convert the started_at, ended_at columns to time type.
- Calculate the trip duration using the converted time columns.
- Remove bad data which contains negative time difference from trip duration column.
- Deal with missing values.
- Concatenate start and end station names to observe trip routes.

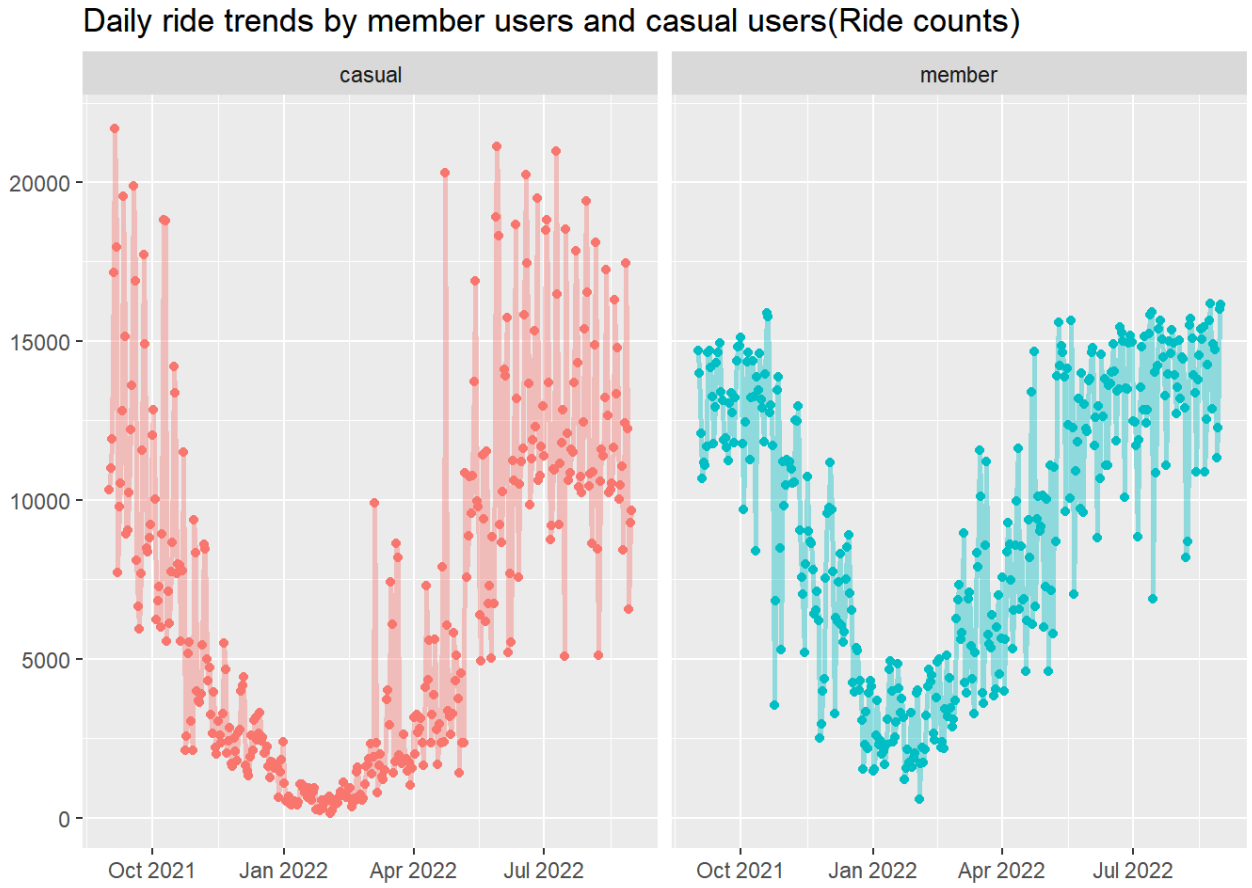
```
#Data cleaning for dt_dataset

#Convert Character to Date and format to Month and Weekday
dt_dataset$trip_date <- as.Date(dt_dataset$started_at)
dt_dataset$trip_month <- format(as.Date(dt_dataset$trip_date), "%m")
dt_dataset$trip_wday <- format(as.Date(dt_dataset$trip_date), "%a")

#Reorder the weekdays
dt_dataset$trip_wday <- factor(dt_dataset$trip_wday, levels= c("Sun", "Mon", "Tue", "Wed", "Thu", "Fr
i", "Sat"))

#Convert Datetime to Time datatype, remove rows with negative trip_duration
dt_dataset$trip_stime <- as.POSIXct(dt_dataset$started_at, tz='UTC')
dt_dataset$trip_etime <- as.POSIXct(dt_dataset$ended_at, tz='UTC')

dt_dataset$trip_duration <- difftime(dt_dataset$trip_etime, dt_dataset$trip_stime)
dt_dataset2 <- subset(dt_dataset, trip_duration>0)


#Data Cleaning for lc_dataset

#Deal with missing Values
lc_dataset <- lc_dataset %>%
  filter(start_station_name!=""&end_station_name!="")

#Create trip_route column by concatenating the start and end station names
lc_dataset$trip_route <- paste(lc_dataset$start_station_name, lc_dataset$end_station_name, sep=" - ")

#Create round trip column by inspecting whether start station name equals to end station name or not
lc_dataset$round_trip[lc_dataset$start_station_name==lc_dataset$end_station_name] <- "YES"
lc_dataset$round_trip[lc_dataset$start_station_name!=lc_dataset$end_station_name] <- "NO"
```

5. Explore and analyze the data

- Take a look at total ride trends on daily basis by user types.

```
#Comparison by the number of daily ride between user types
dt_dataset2 %>%
  group_by(trip_date, member_casual) %>%
  summarize(daily_ride_count=n_distinct(ride_id)) %>%
  ggplot(aes(x = trip_date, y = daily_ride_count, color = member_casual)) +
  geom_point() +
  geom_line(aes(col=member_casual), size=1, alpha=0.4) +
  facet_wrap(~member_casual) +
  labs(title="Daily ride trends by member users and casual users(Ride counts)") +
  theme(axis.title.x=element_blank(), axis.title.y=element_blank(), legend.position="none")
```

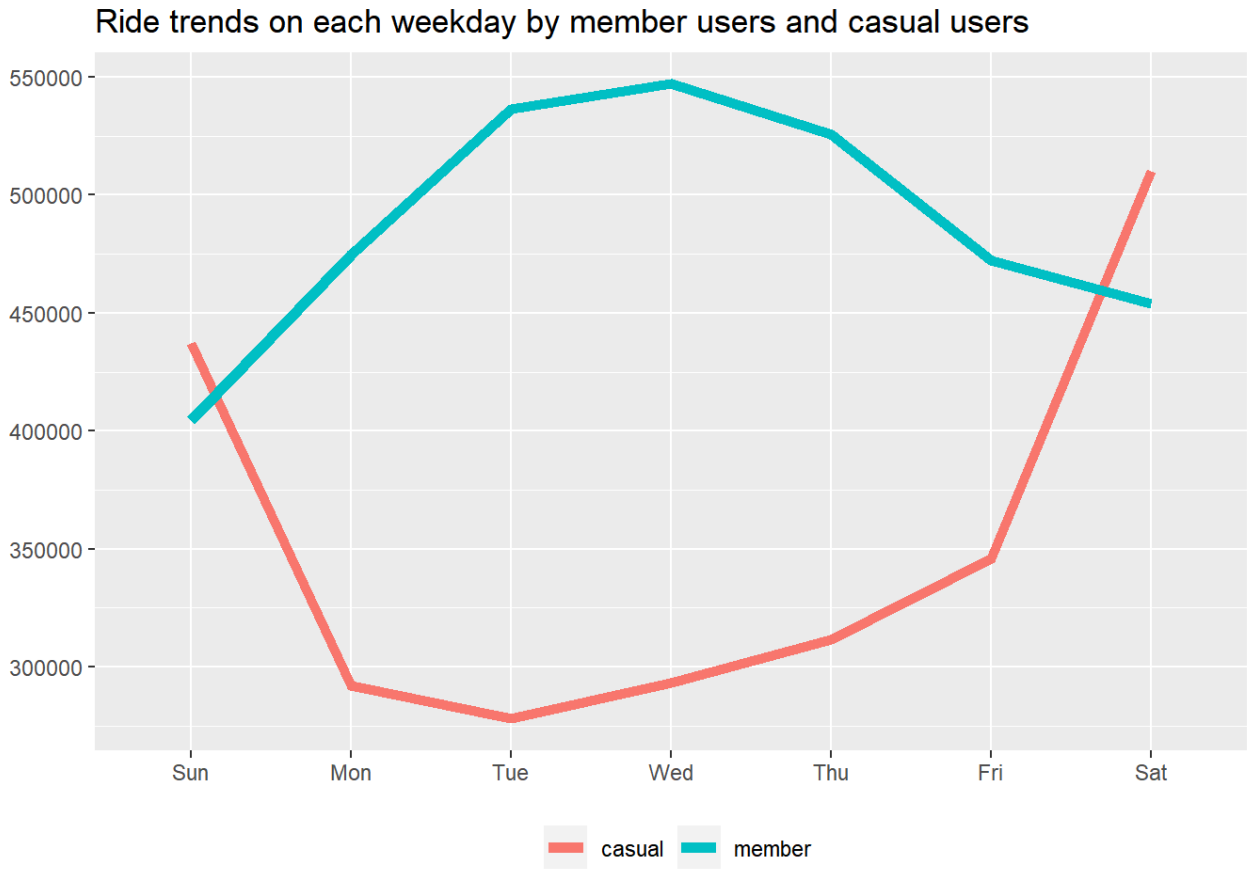
```
## `summarise()` has grouped output by 'trip_date'. You can override using the
## `.groups` argument.
```



- Check out the difference between the user types by each weekday

```
#Compare the number of ride by each user type every weekdays
dt_dataset2 %>%
  group_by(trip_wday, member_casual) %>%
  summarise(ride_count_by_weekdays=n_distinct(ride_id)) %>%
  ggplot(aes(x=trip_wday, y=ride_count_by_weekdays, group=member_casual)) +
  geom_line(aes(color=member_casual), size=2) +
  labs(title="Ride trends on each weekday by member users and casual users") +
  theme(axis.title.x=element_blank(), axis.title.y=element_blank(), legend.position="bottom", legend.
title = element_blank())
```

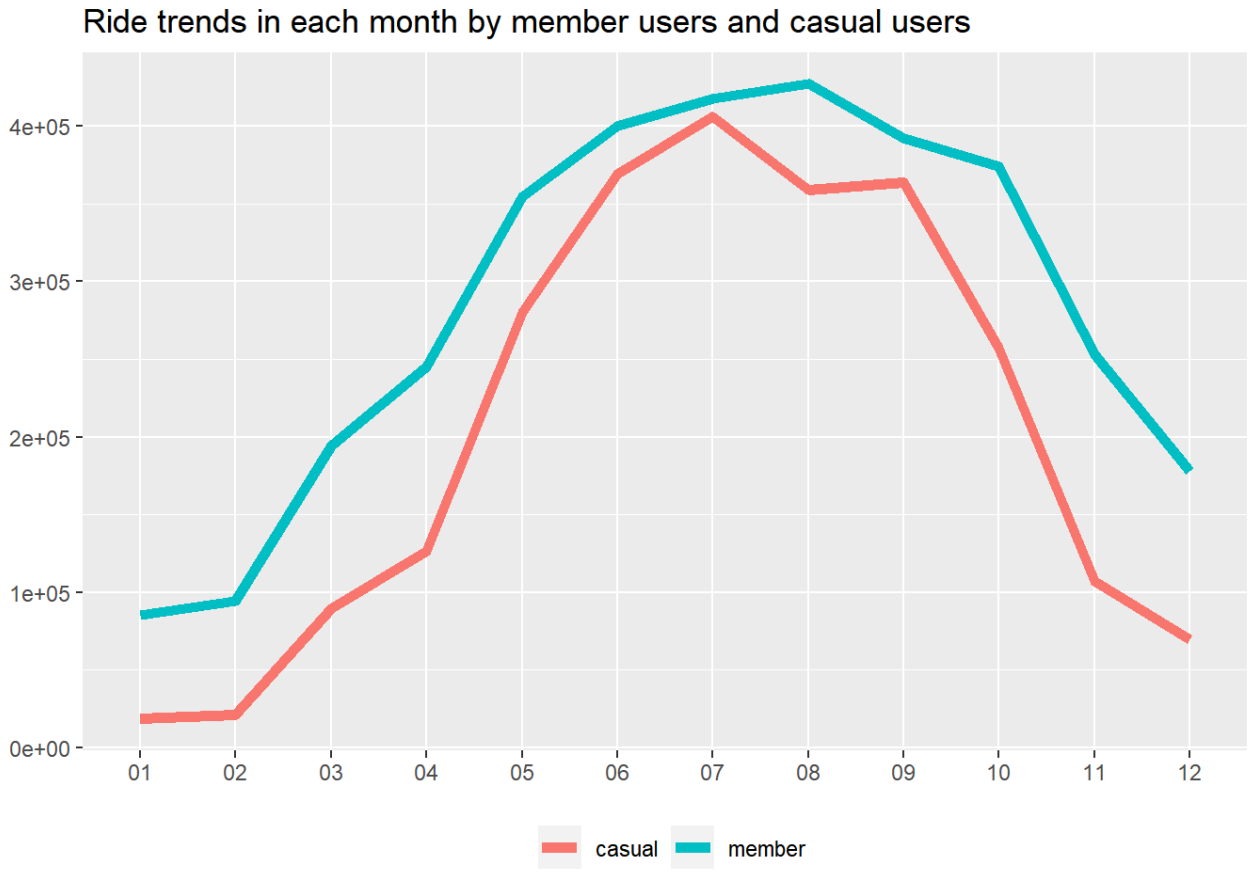
```
## `summarise()` has grouped output by 'trip_wday'. You can override using the
## `.groups` argument.
```



- Ride comparison between the user types by each month

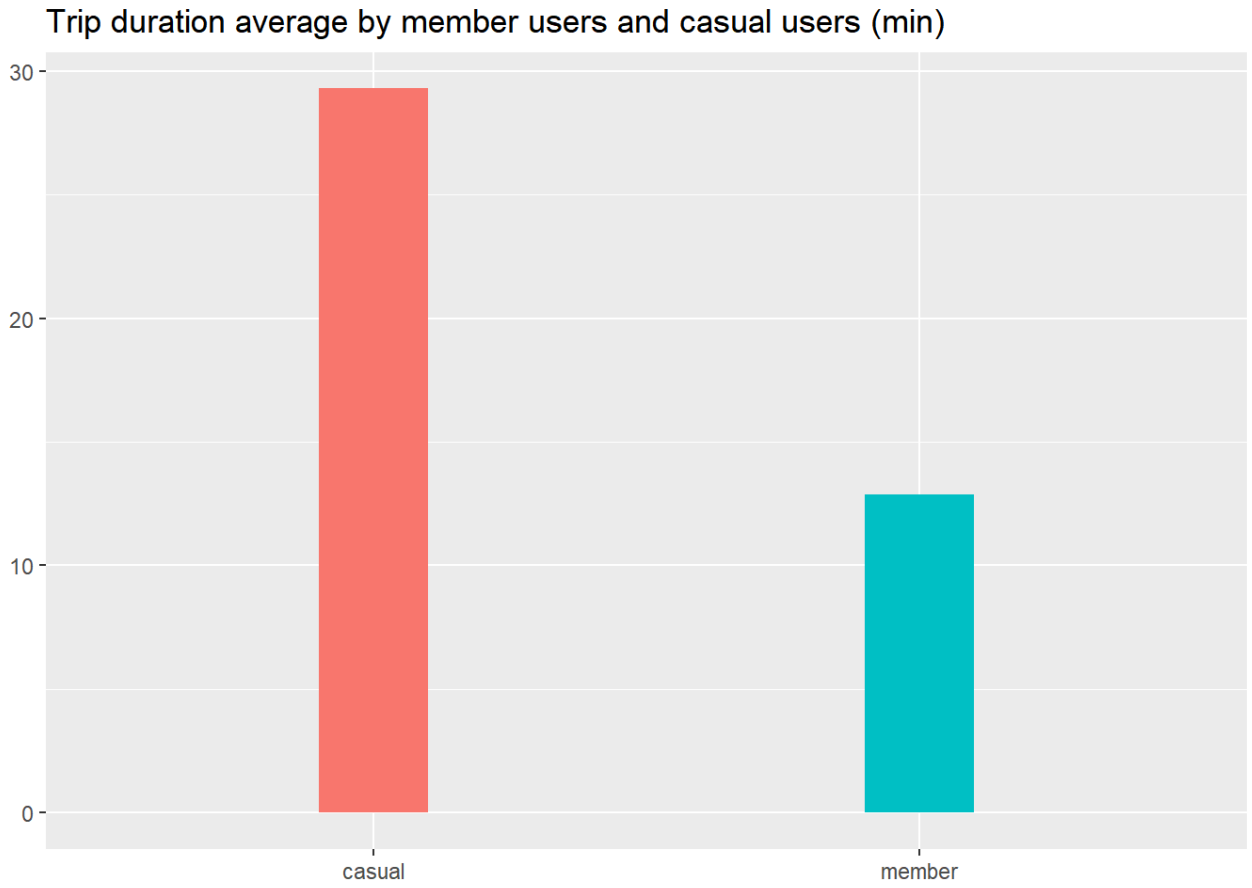
```
#Compare the number of ride by each user type every months
dt_dataset2 %>%
  group_by(trip_month, member_casual) %>%
  summarise(ride_count_by_months=n_distinct(ride_id)) %>%
  ggplot(aes(x=trip_month, y=ride_count_by_months, group=member_casual)) +
  geom_line(aes(color=member_casual), size=2) +
  labs(title="Ride trends in each month by member users and casual users") +
  theme(axis.title.x=element_blank(), axis.title.y=element_blank(), legend.position="bottom", legend.
title = element_blank())
```

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



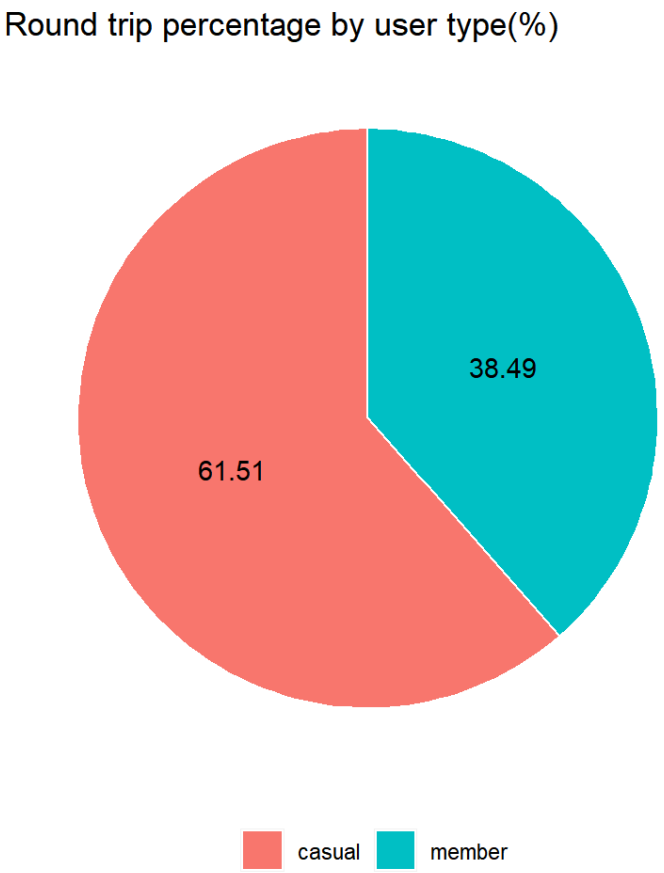
- Compare the average trip duration

```
#Compare the average trip duration between the two user types
dt_dataset2 %>%
  group_by(member_casual) %>%
  summarise(trip_duration_average=mean(as.integer(trip_duration)/60)) %>%
  ggplot(aes(x=member_casual, y=trip_duration_average, fill=member_casual)) +
  geom_bar(stat = "identity", width=0.2) +
  labs(title="Trip duration average by member users and casual users (min)") +
  theme(axis.title.x=element_blank(), axis.title.y=element_blank(), legend.position="none")
```



- Round trip proportion by user type

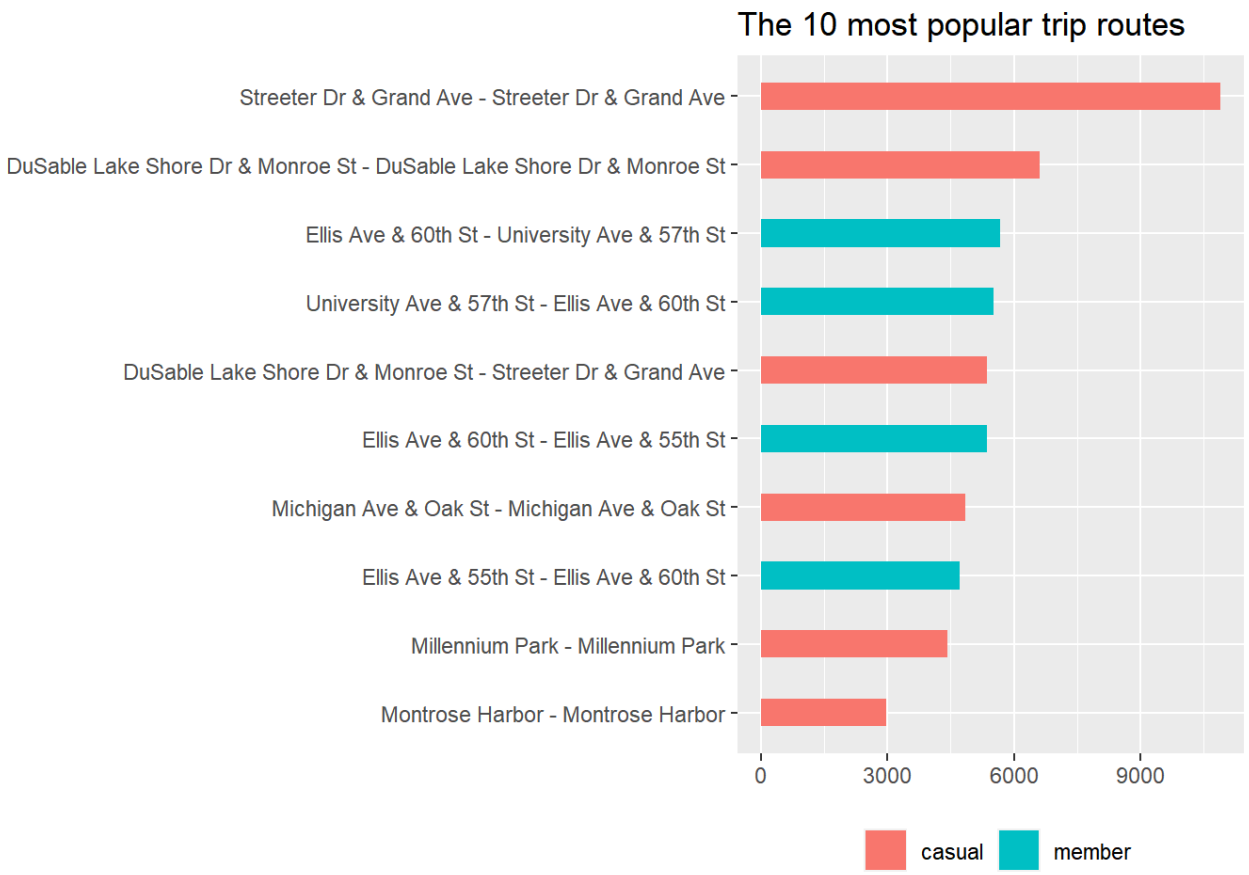
```
#Create the pie chart to see the proportion of each user type from all the round trips
lc_dataset %>%
  filter(round_trip=="YES") %>%
  group_by(member_casual) %>%
  summarize(count=n_distinct(ride_id)) %>%
  mutate(percent=count/sum(count)*100) %>%
  ggplot(aes(x = 2, y = percent, fill = member_casual)) +
  geom_bar(stat = "identity", color = "white") +
  coord_polar(theta = "y", start = 0) +
  labs(title="Round trip percentage by user type(%)") +
  geom_text(aes(label = round(percent,2)), position = position_stack(vjust = 0.5), color = "black") +
  theme(axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        panel.background = element_blank(),
        plot.background = element_blank(),
        legend.position="bottom",
        legend.title = element_blank())
```



- Check out the 10 most popular trip routes for each user type

```
#Check the 10 most popular trip routes
lc_dataset %>%
  group_by(member_casual, trip_route) %>%
  summarise(count=n_distinct(ride_id)) %>%
  arrange(desc(count), trip_route) %>%
  head(n=10) %>%
  ggplot(aes(x=count, y=reorder(trip_route, count), fill=member_casual)) +
  geom_bar(stat = "identity", width=0.4) +
  labs(title="The 10 most popular trip routes") +
  theme(axis.title.x=element_blank(), axis.title.y=element_blank(), legend.position="bottom", legend.
title = element_blank())
```

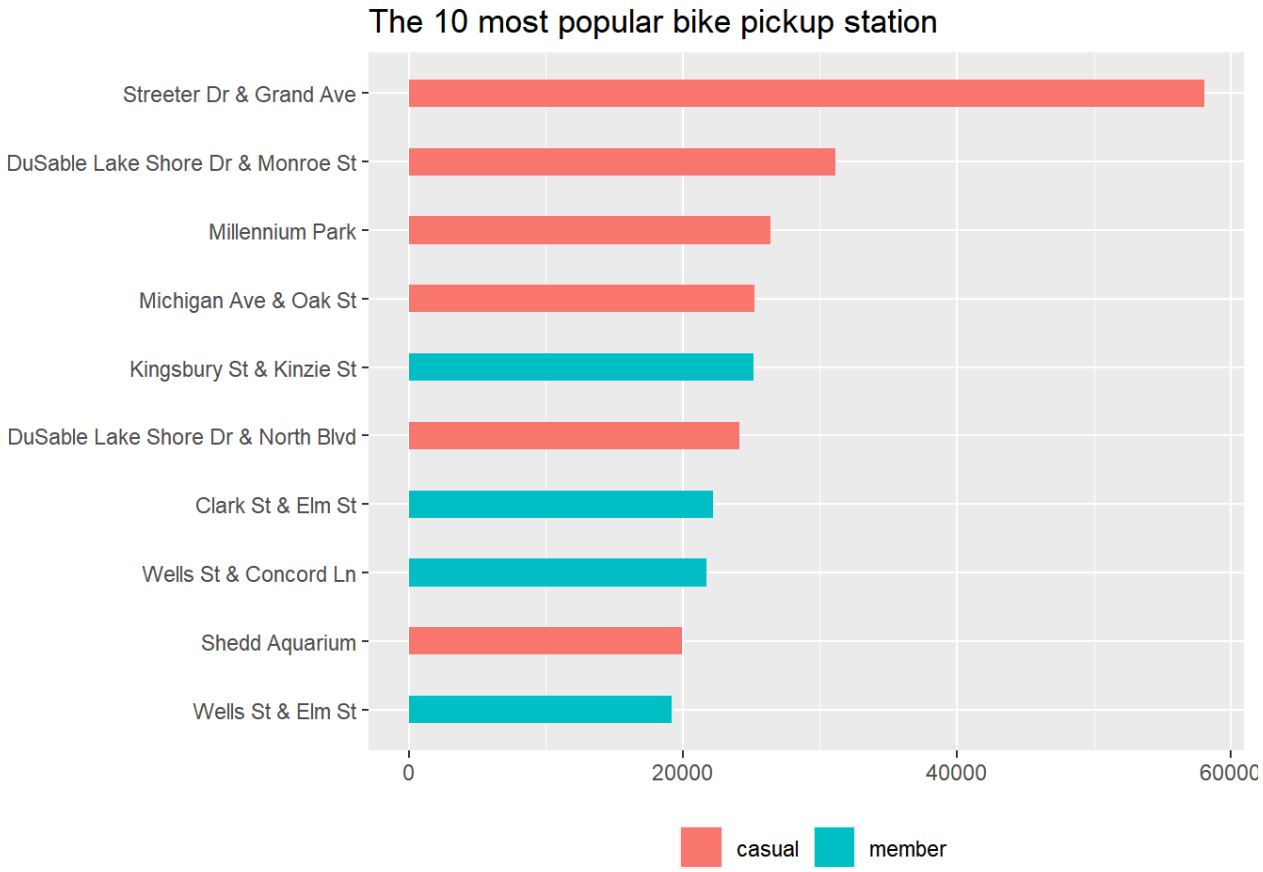
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



- The 10 most popular bike pickup(start) station

```
#Create the bar chart to see the top 10 popular station to start the trip
lc_dataset %>%
  group_by(start_station_name, member_casual) %>%
  summarize(ride_count=n_distinct(ride_id)) %>%
  arrange(desc(ride_count)) %>%
  head(n=10) %>%
  ggplot(aes(x=ride_count, y=reorder(start_station_name, ride_count), fill=member_casual)) +
  geom_bar(stat = "identity", width=0.4) +
  labs(title="The 10 most popular bike pickup station") +
  theme(axis.title.x=element_blank(), axis.title.y=element_blank(), legend.position="bottom", legend.
title = element_blank())
```

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


The 10 most popular bike pickup station
Streeter Dr & Grand Ave
DuSable Lake Shore Dr & Monroe St
Millennium Park
Michigan Ave & Oak St
Kingsbury St & Kinzie St
DuSable Lake Shore Dr & North Blvd
Clark St & Elm St
Wells St & Concord Ln
Shedd Aquarium
Wells St & Elm St
0
20000
40000
60000
casual
member