Analyze_trip_data

Juliette

12/03/2021

Analyzing cyclistic fictional trip data

This documents explores how different customers for a fictional company called Cyclistic are using their ride-share program in the last twelve months from August 2020 to July 2021. The data was stored in individual excel files by month. The data is owned and licensed by Motivate International Inc (https://divvy-tripdata.s3.amazonaws.com/index.html)

Load packages

Here I load the packages for analysis

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

Load and merge files

Here I load and merge the 12 excel files

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
setwd("C:/Users/julie/OneDrive/Documents/trip/Cyclist")
files <- list.files(pattern = ".csv")
data <- lapply(files, fread, sep= ",")
trip_data <- rbindlist(data)
```

Clean data

Here I start cleaning the data, removing empty columns, spaces, looking for duplicates

```
### remove N/A columns and empty columns
trip_data2 <- filter(trip_data, start_station_id != "NA" & end_station_id!="NA")
trip_data3 <-  filter(trip_data, start_station_name != "" & end_station_name!="")

### check for  duplicates
ride_id <- trip_data3 %>%
  count(ride_id) %>%
  filter(n > 1) # no duplicates found
```

Transform data

Here I start transforming the data for next step in analysis

Find time difference in sec between ended_at and started_at columns

Here I find ride length between start and end stations in minutes

```
trip_data3$ride_length <- difftime(
  trip_data3$ended_at,
  trip_data3$started_at,
  units = "mins"
)
```

Remove ride length < 0

Here I remove values less than zero that can negatively skew the results

```
trip_data3 <- trip_data3 %>%
  filter(!(ride_length < 0))
```

Create column for day of the week, month, year

Here I create three columns, one for the days of the week, month and year

```
trip_data3$day_of_week <- format(trip_data3$started_at,
                                 "%A")
trip_data3$day_of_week <- ordered(trip_data3$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wedn
trip_data3$month <- format(trip_data3$started_at, "%m")
trip_data3$year <- format(trip_data3$started_at, "%y")
trip_data3$hour <- format(trip_data3$started_at, "%H:%M:%S")
```

Select variables to work with

Here I select subset of variables to work with in next step of analysis

```
trip_variables <- trip_data3 %>%
  select(ride_id, rideable_type,started_at, start_station_name, end_station_name, member_casual,hour, r
```

Analyze members and casual riders

Here I start looking for similarities and differences among member and casual riders

Common start station trips for members

Here I create a data frame that includes common start station trips for members

```
trip_variables %>%
  select (start_station_name, member_casual) %>%
  group_by(start_station_name)%>%
  filter(member_casual == "member")%>%
  mutate(trips = n()) %>%
  distinct(start_station_name, .keep_all = TRUE)
```

```
## # A tibble: 714 x 3
## # Groups:   start_station_name [714]
##    start_station_name               member_casual trips
##    <chr>                            <chr>         <int>
##  1 Lake Shore Dr & Diversey Pkwy    member        10445
##  2 Marine Dr & Ainslie St           member         4951
##  3 Dearborn St & Erie St            member        17634
##  4 Wells St & Elm St                member        18256
##  5 Desplaines St & Kinzie St        member        15679
##  6 Wallace St & 35th St             member         1675
##  7 Broadway & Granville Ave         member         4713
##  8 Lake Park Ave & 56th St          member         3355
##  9 Clark St & 9th St (AMLI)         member         3593
## 10 Cottage Grove Ave & Oakwood Blvd member          925
## # ... with 704 more rows
```

Common end station trips for members

Here I create a data frame for common end station trips for members

```
 trip_variables %>%
  select (end_station_name, member_casual) %>%
  group_by(end_station_name)%>%
  filter(member_casual == "member")%>%
  mutate(trips = n()) %>%
  distinct(end_station_name, .keep_all = TRUE)
```

```
## # A tibble: 713 x 3
## # Groups:   end_station_name [713]
##    end_station_name                 member_casual trips
##    <chr>                            <chr>         <int>
##  1 Clark St & Lincoln Ave           member        14664
##  2 Sheridan Rd & Lawrence Ave       member         3550
##  3 Kingsbury St & Erie St           member        12260
##  4 State St & Van Buren St          member         3993
##  5 Cottage Grove Ave & Oakwood Blvd member          882
##  6 Broadway & Granville Ave         member         4855
##  7 Stony Island Ave & 71st St       member           80
##  8 Lake Park Ave & 56th St          member         3554
##  9 Wallace St & 35th St             member         1652
## 10 Clark St & Schiller St           member        12169
## # ... with 703 more rows
```

Common start station trips for casual riders

Here I create a data frame for common start station trips among casual riders

```
 trip_variables %>%
  select (start_station_name, member_casual)%>%
  group_by (start_station_name)%>%
  filter(member_casual == "casual")%>%
  mutate(trips = n()) %>%
  distinct(start_station_name, .keep_all= TRUE)
```

```
## # A tibble: 727 x 3
## # Groups:   start_station_name [727]
##    start_station_name              member_casual trips
##    <chr>                           <chr>         <int>
##  1 Michigan Ave & 14th St          casual         5068
##  2 Columbus Dr & Randolph St       casual        11467
##  3 Daley Center Plaza              casual         6052
##  4 Leavitt St & Division St        casual         1147
##  5 Cityfront Plaza Dr & Pioneer Ct casual         6494
##  6 Sheffield Ave & Fullerton Ave   casual         4918
##  7 Southport Ave & Wellington Ave  casual         4405
##  8 Theater on the Lake             casual        21812
##  9 Lakeview Ave & Fullerton Pkwy   casual        10840
## 10 Morgan St & Lake St             casual         6931
## # ... with 717 more rows
```

Common end station trips for casual riders

Here I create a data frame for common end stations trips for casual riders

```
trip_variables %>%
  select (end_station_name, member_casual)%>%
  group_by (end_station_name)%>%
  filter(member_casual == "casual")%>%
  mutate(trips = n()) %>%
  distinct(end_station_name, .keep_all= TRUE)
```

```
## # A tibble: 727 x 3
## # Groups:   end_station_name [727]
##    end_station_name          member_casual trips
##    <chr>                     <chr>         <int>
##  1 Michigan Ave & 14th St    casual         5075
##  2 State St & Randolph St    casual         9099
##  3 State St & Kinzie St      casual         9482
##  4 Leavitt St & Division St  casual         1016
##  5 Dearborn St & Monroe St   casual         5588
##  6 Southport Ave & Roscoe St casual         7609
##  7 Halsted St & Dickens Ave  casual         6646
##  8 Western Ave & Division St casual         1948
##  9 Broadway & Belmont Ave    casual         6870
## 10 LaSalle St & Jackson Blvd casual         2166
## # ... with 717 more rows
```

Summarize number of trips by month

Here I create a data frame that summarizes number of trips by month

```
trip_variables %>%
  select(
    member_casual,
    month) %>%
  group_by(member_casual,) %>%
  mutate( trips = n()) %>%
  distinct(
    month,
    member_casual,
    .keep_all = TRUE
  )
```

```
## # A tibble: 24 x 3
## # Groups:   member_casual [2]
##    member_casual month   trips
##    <chr>         <chr>   <int>
##  1 member        08    2333635
##  2 casual        08    1826594
##  3 casual        09    1826594
##  4 member        09    2333635
##  5 casual        10    1826594
##  6 member        10    2333635
##  7 casual        11    1826594
##  8 member        11    2333635
##  9 member        12    2333635
## 10 casual        12    1826594
## # ... with 14 more rows
```

Dataframe: summarize number of trips by year

Here I create a data frame that summarizes number of trips by year

```
 trip_variables %>%
  select(
    member_casual, year) %>%
  group_by(member_casual) %>%
  mutate(trips = n()) %>%
  distinct(
    year,
    member_casual,
    .keep_all = TRUE
  )
```

```
## # A tibble: 4 x 3
## # Groups:   member_casual [2]
##   member_casual year    trips
##   <chr>         <chr>   <int>
## 1 member        20    2333635
```

```
## 2 casual        20    1826594
## 3 member        21    2333635
## 4 casual        21    1826594
```

Visualization

Summarize member casual ride length, group by member_casual

Here I summarize ride length by customer type

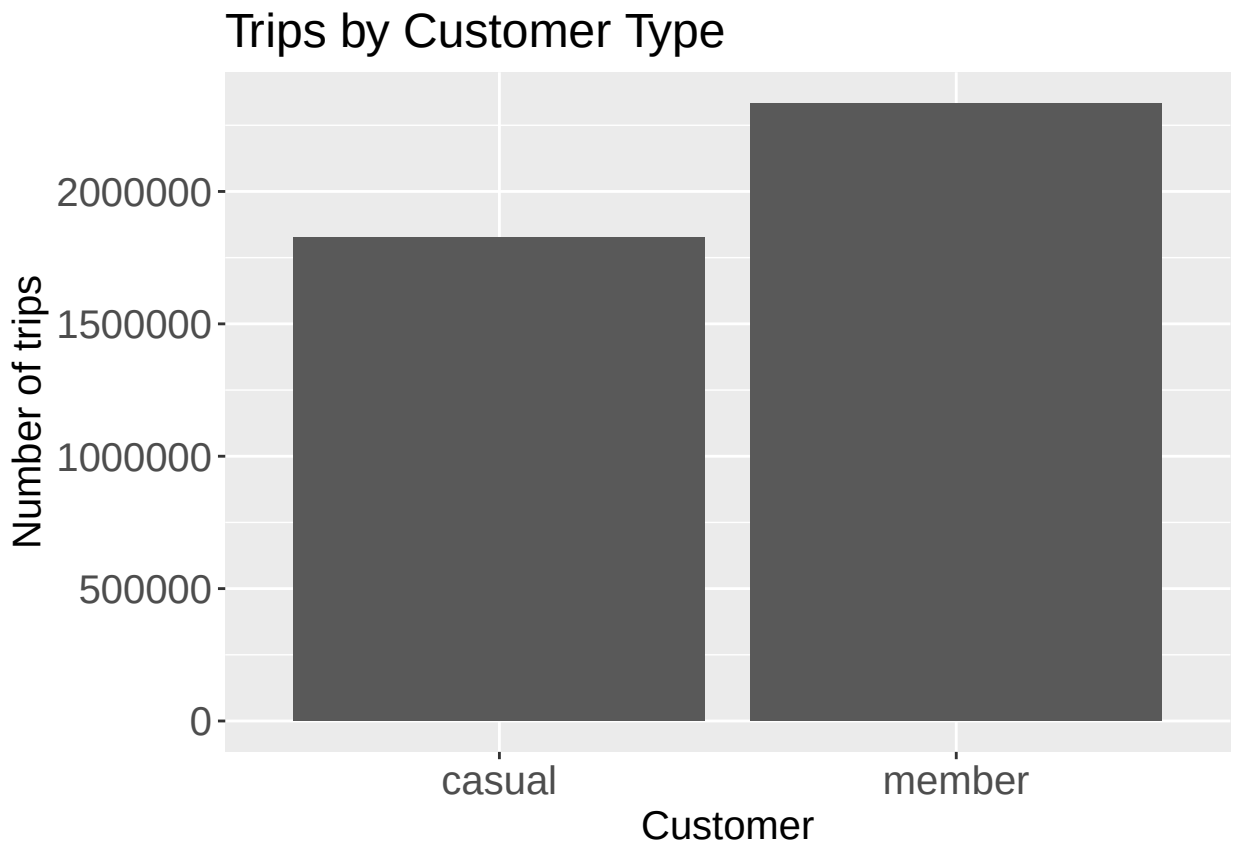
```
trip_variables %>%
  select(member_casual, ride_length)%>%
  group_by(member_casual) %>%
  summarize(mean(ride_length), sd(ride_length)) # casual members ride longer than members
```

```
## # A tibble: 2 x 3
##   member_casual `mean(ride_length)` `sd(ride_length)`
##   <chr>         <drtn>                          <dbl>
## 1 casual        37.56549 mins                   335.
## 2 member        14.38951 mins                    51.2
```

Visualize member and casual riders

Here I visualize member and casual riders

```
ggplot(data = trip_variables) +
  geom_bar(mapping = aes(x = member_casual))+
labs(title = "Trips by Customer Type" ) +
  xlab("Customer") + ylab("Number of trips") +
  theme(text = element_text(size=15), axis.text = element_text(size=15),legend.text=element_text(size=15
```



Group by rideable_type

Here I group customer by type of bikes used

```
trip_variables %>%
  group_by(member_casual, rideable_type) %>%
  summarize(mean(ride_length))
```

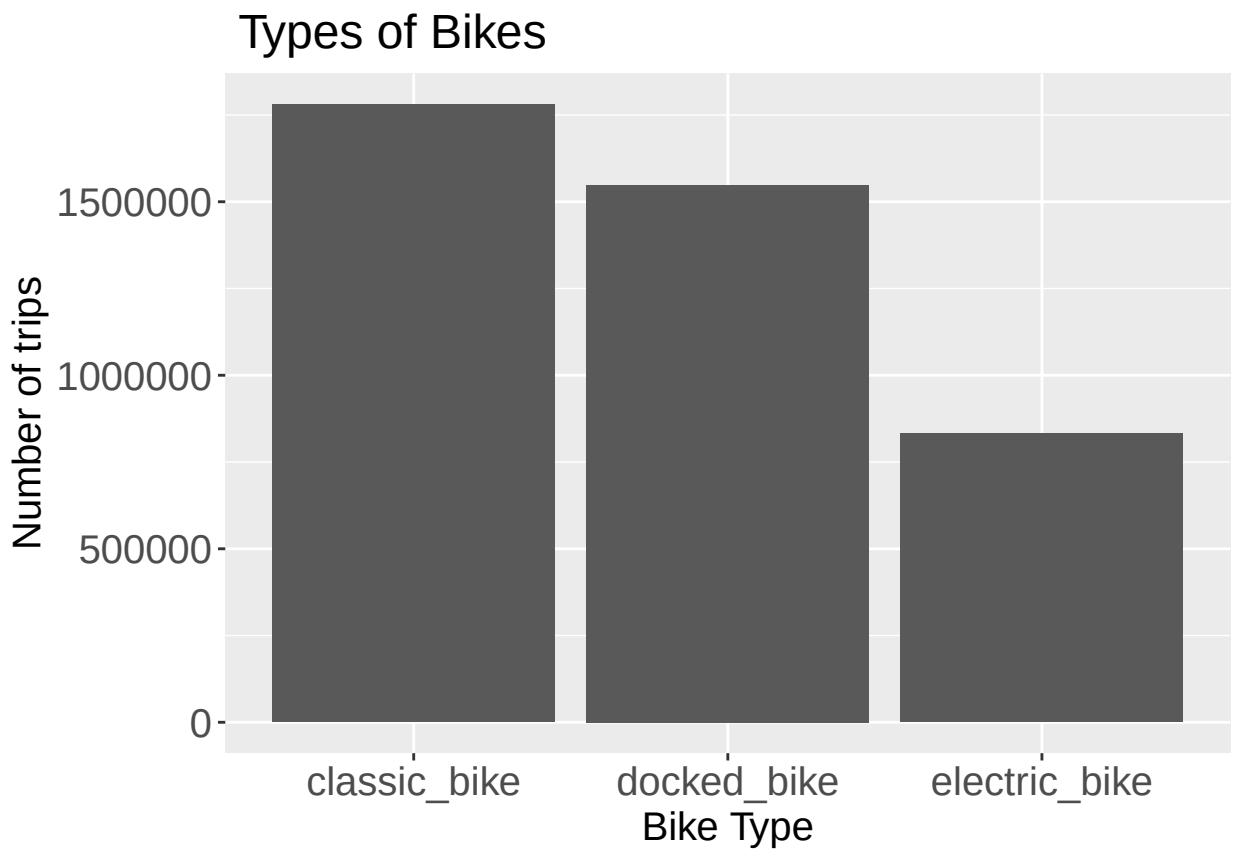
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
## # A tibble: 6 x 3
## # Groups:   member_casual [2]
##   member_casual rideable_type `mean(ride_length)`
##   <chr>         <chr>         <drtn>
## 1 casual        classic_bike  27.39723 mins
## 2 casual        docked_bike   54.92282 mins
## 3 casual        electric_bike 21.40788 mins
## 4 member        classic_bike  14.11236 mins
## 5 member        docked_bike   15.54795 mins
## 6 member        electric_bike 13.03680 mins
```

Types of bikes riders use

Here I plot types of bikes used by riders

```
ggplot(data = trip_variables) +
  geom_bar(mapping = aes(x = rideable_type))+
labs(title = " Types of Bikes" ) +
  xlab("Bike Type") + ylab("Number of trips") +
  theme(text = element_text(size=15), axis.text = element_text(size=15),legend.text=element_text(size=15
```

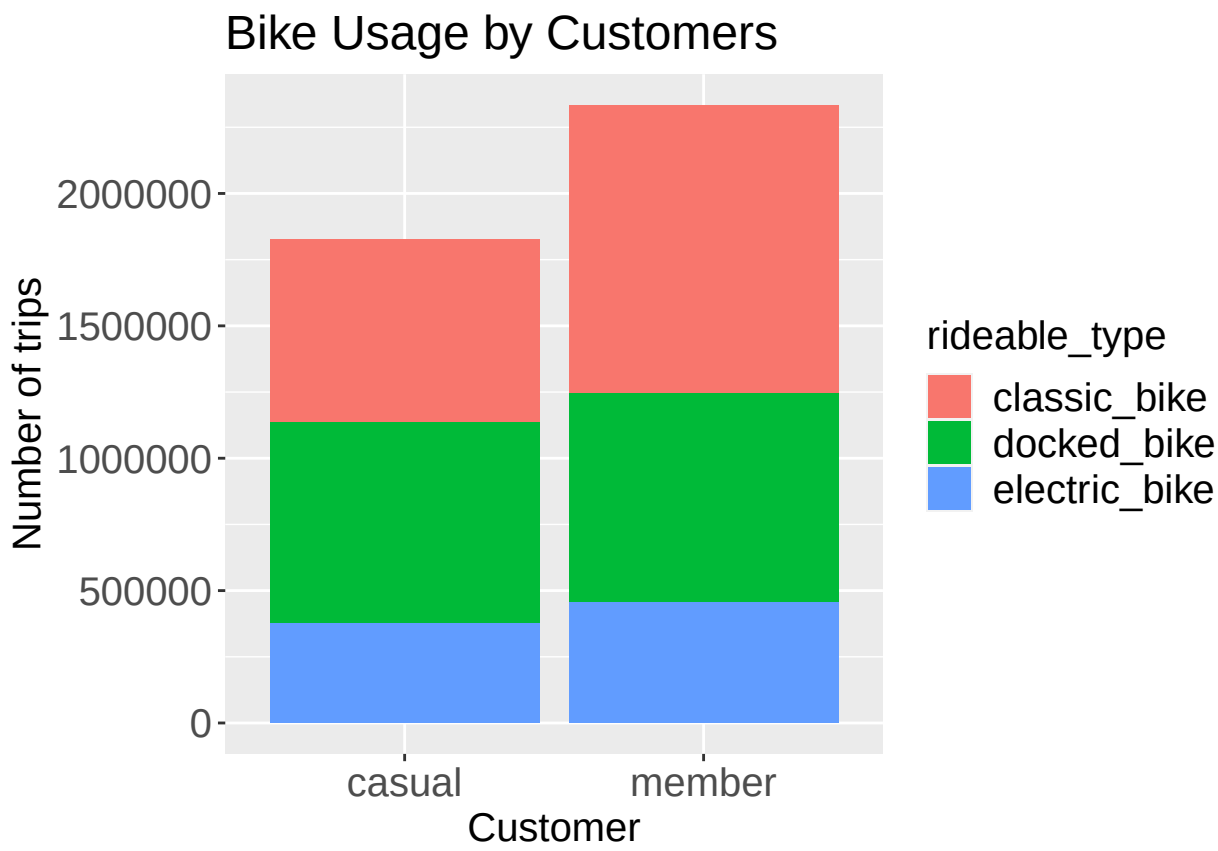



```
# most popular bike is the classic bike followed, by the docked bike, and lastly the electric bike
```

Compare member and casual riders to types of bikes used

Here I compare member and casual riders to types of bikes used

```
ggplot(data = trip_variables) +
  geom_bar(mapping = aes(x = member_casual, fill=rideable_type))+
labs(title = "Bike Usage by Customers" ) +
  xlab("Customer") + ylab("Number of trips") +
  theme(text = element_text(size=15), axis.text = element_text(size=15),legend.text=element_text(size=15
```

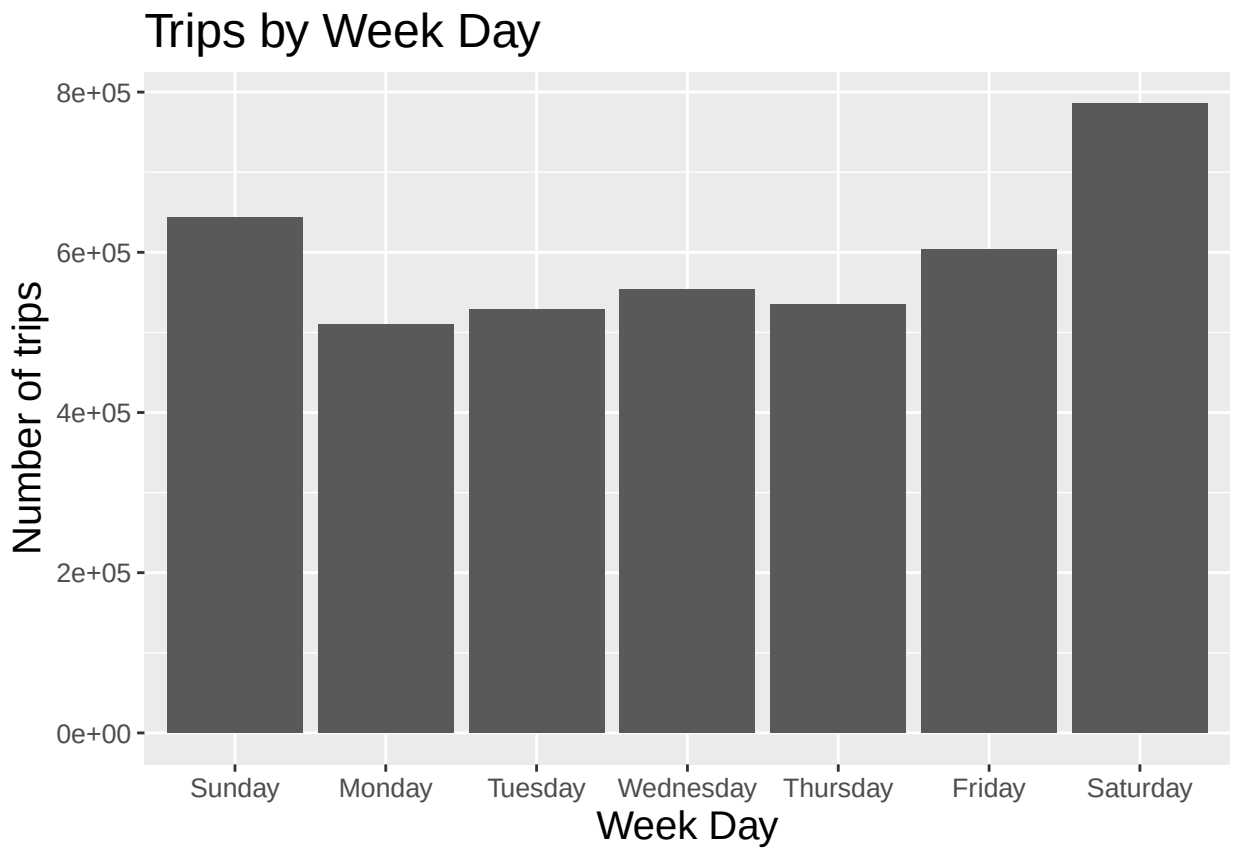



```
# members take more trips with the classic bike and docked bike compared to casual customers
```

Plot day of week

Here I plot the week days to find trends

```
ggplot(data = trip_variables
       ) +
  geom_bar(mapping = aes(x = day_of_week))+ # saturday and sunday has most riders
labs(title = "Trips by Week Day" )+
  xlab("Week Day") + ylab("Number of trips")+
  theme(text = element_text(size=15), axis.text = element_text(size=10),legend.text=element_text(size=15
```



Average ride length per day based on customer type

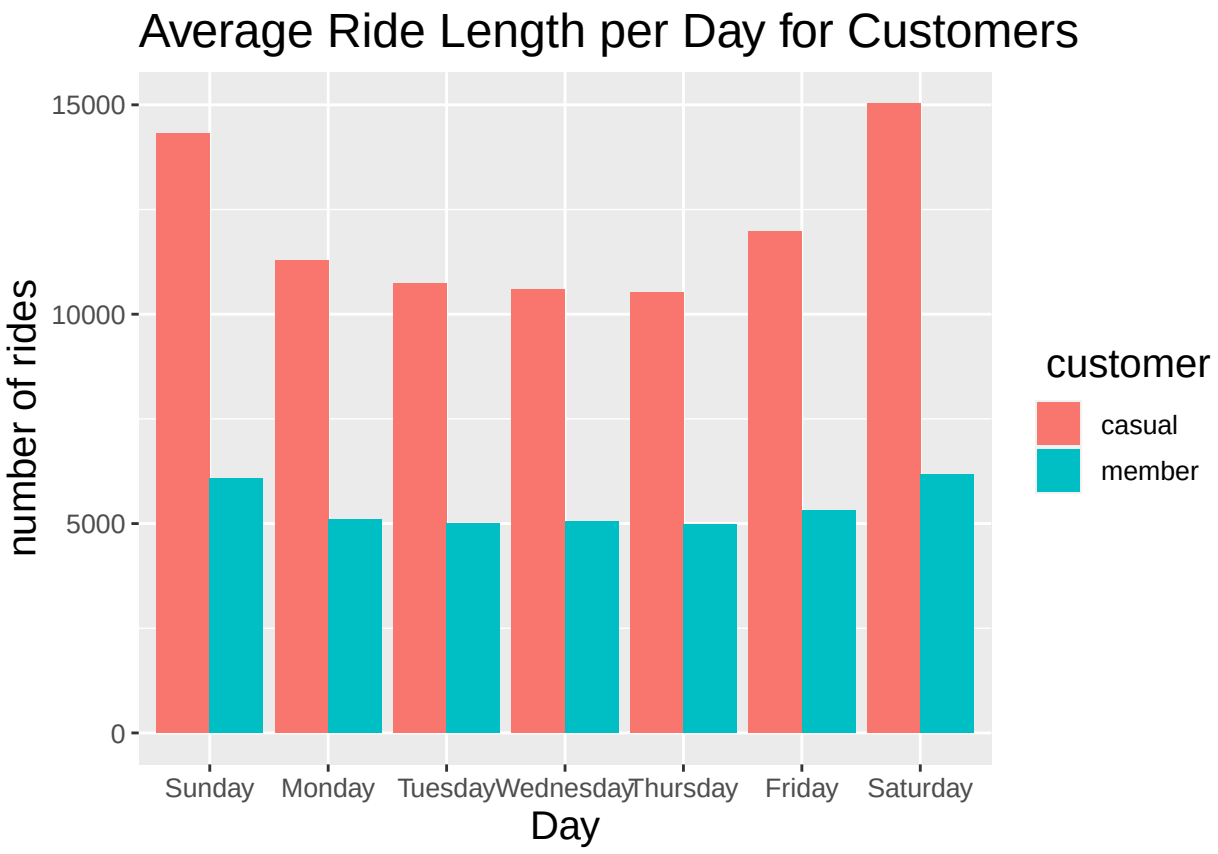
Here I compare the average ride length per day based on customer type

```
all_trips_v2 <- trip_variables %>%
  select(day_of_week, member_casual, ride_length) %>%
  group_by(member_casual, day_of_week, ride_length) %>%
  summarise(trips = n())%>%
  arrange(member_casual, day_of_week)
```

```
## `summarise()` has grouped output by 'member_casual', 'day_of_week'. You can override using the `.grou
```

```
all_trips_v2 %>%
select(day_of_week, member_casual, ride_length)%>%
  group_by(member_casual,day_of_week)%>%
  summarise(trips = n()
            ,average_duration = mean(ride_length))%>%
  arrange(member_casual, day_of_week)%>%
  ggplot(aes(x =day_of_week, y = trips, fill = member_casual))+
  geom_col(position = "dodge") +
  labs(title = "Average Ride Length per Day for Customers" , fill=" customer")+
  xlab("Day") + ylab("number of rides")+
  theme(text = element_text(size=15), axis.text = element_text(size=10),legend.text=element_text(size=1
```

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

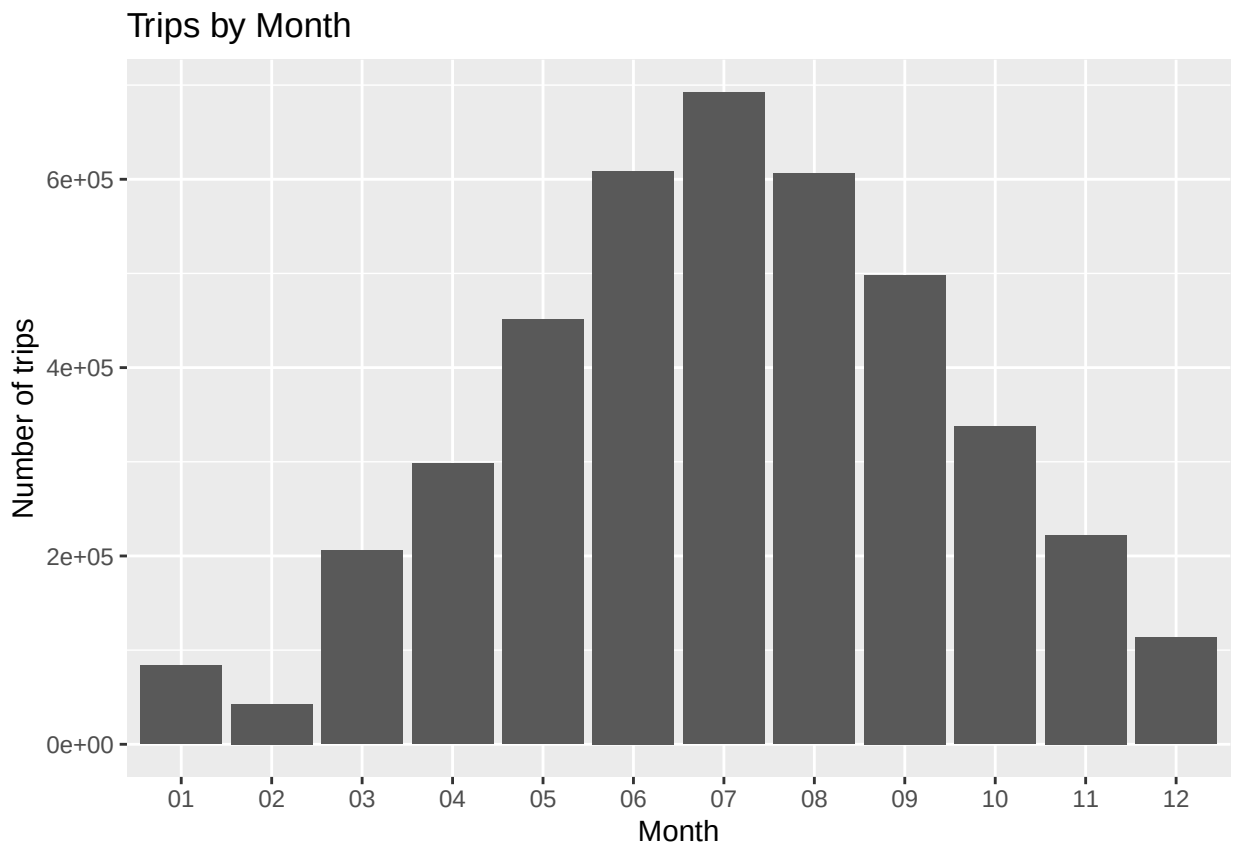



```
# Average ride length per day is higher for casual riders compared to members
```

Plot Month

Here I plot month to find trends

```
ggplot(data = trip_variables) +
  geom_bar(mapping = aes(x = month))+ # summer months most popular time of year to ride
labs(title = "Trips by Month" ) +
  xlab("Month") + ylab("Number of trips")
```



Average Trip Duration for August 2020-July 2021

Here I find the average ride length for customers in from August 2020-July 2021

```
ride_length_stat_2020_2021 <- trip_variables %>%
    select(year, month,day_of_week, member_casual, ride_length) %>%
    group_by(year, month,day_of_week, member_casual) %>%
    summarise(trips = n()
              ,average_duration = round( mean(ride_length) ,digits=1)
              ,sum_duration = round( sum(ride_length), digits =1))%>%
  arrange(year, month, day_of_week, member_casual)
```

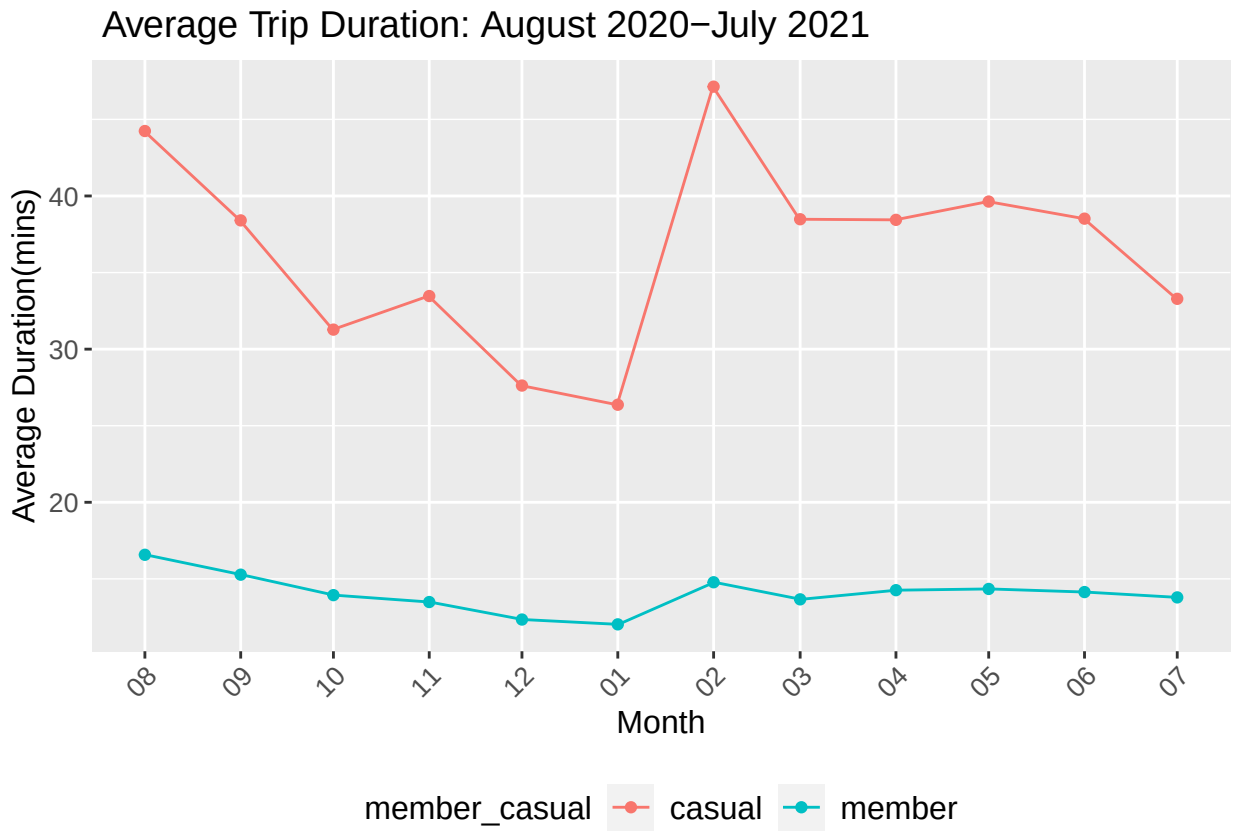
```
## `summarise()` has grouped output by 'year', 'month', 'day_of_week'. You can override using the `.grou
```

```
  ride_length_stat_2020_2021 %>%
   mutate(year_month = as.Date(paste(year,"-",month,"-1",sep="")) ) %>%
    group_by(member_casual, year_month) %>%
    summarise(sum_of_rides = sum(trips )
              ,average_duration = sum(sum_duration) / sum(trips) ) %>%
    arrange(member_casual, year_month)  %>%
    ggplot(aes(x = year_month, y = average_duration, group = member_casual, color = member_casual)) +
    geom_line( position = position_dodge(width = 0.9) ) +
    geom_point() +
    labs(title = " Average Trip Duration: August 2020-July 2021" ) +
```

```
    xlab("Month") + ylab("Average Duration(mins)") +
    scale_x_date( labels = date_format("%m"), breaks = "1 month", minor_breaks ="1 month") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position="bottom", text = element_t
```

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

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
  # Average ride duration peaked in August and February for both casual and member riders
```

Start station average trip

Here I analyze station trips

Group data by start station

Here I first create a data frame that includes start station and number of trips

```
trip <- trip_variables%>%
  group_by(start_station_name)%>%
  mutate(trips = n())
```

Assign variable for mean of trips

Here I assign a variable to find mean of trips

```
x <- trip %>%
   summarise( trp = mean(trips))
```

Select top 10 start stations

Here I assign a variable to find top 10 start stations

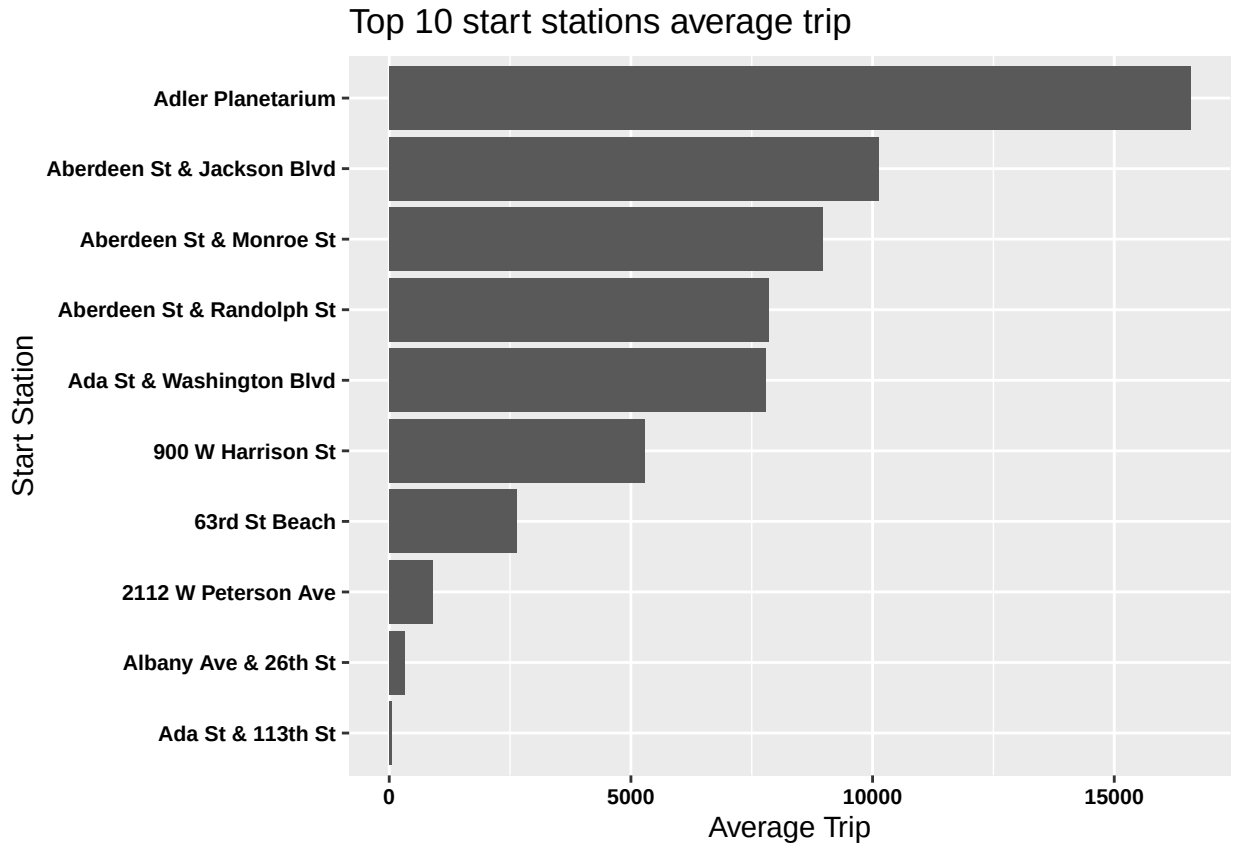
```
 y <- head( x, 10)
```

Create horizontal bar chart for top 10 start stations with average trip

Here I create a horizontal bar chart for the top 10 start stations

```
options(repr.plot.width=8, repr.plot.height=3)
 ggplot(y, aes(x = reorder(start_station_name, trp), start_station_name, y = trp)) +
   geom_bar(stat = "identity") +
   coord_flip() + scale_y_continuous(name="Average Trip") +
   scale_x_discrete(name="Start Station")+
   labs(title = "Top 10 start stations average trip")+

    theme(axis.text.x = element_text(face="bold", color="black",
                                     size=8, angle=0),
          axis.text.y = element_text(face="bold", color="black",
                                     size=8, angle=0))
```



Summary of Findings

- Saturday and Sunday has most riders compared to other days of the week
- Average ride length per day is higher for casual riders compared to members
- Average ride length from August 2020 to July 2021 was higher for casual riders compared to members
- Summer months most popular time of year to ride
- Average ride duration peaked in August and February for both casual and member customers
- The most popular bike is the classic bike, followed by the docked bike and last is the electric bike for all customer types
- Members take more trips with the classic bike and docked bike compared to casual customers

Recommendations

- Members take more trips but the average ride length is less compared to casual riders and vice versa. It would be best to target casual riders who take more trips during the week to convert them to members
- Best to target casual customers year around, but especially during summer months
- Average ride duration peaked in August and February for all customer types, best to target casual riders to convert to member status during these months