Google_Capstone_Project

Nkechi Ihewulezi

2022-09-14

Case Study: How Does a Bike-Share Navigate Speedy Success?

Scenario You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

Step 1: Load packages

Start by installing the required packages. If you have already installed and loaded `tidyverse`, `skimr`, and `janitor` in this session, feel free to skip the code chunks in this step. This may take a few minutes to run, and you may get a pop-up window asking if you want to proceed. Click yes to continue installing the packages. Once a package is installed, you can load it by running the `library()` function with the package name inside the parentheses:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(lubridate)
```

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

```
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
rm(list = ls())
```

Load the data into R

Import previous 12 months data. In this case data for year 2021 Get the file list in the directory and combine into one data frame.

```
df1 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202101-divvy-tripdata.cs
df2 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202102-divvy-tripdata.cs
df3 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202103-divvy-tripdata.cs
df4 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202104-divvy-tripdata.cs
df5 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202105-divvy-tripdata.cs
df6 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202106-divvy-tripdata.cs
df7 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202107-divvy-tripdata.cs
df8 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202108-divvy-tripdata.cs
df9 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202109-divvy-tripdata.cs
df10 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202110-divvy-tripdata.c
df11 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202111-divvy-tripdata.c
df12 <- read.csv("C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers/Bikers_data_2021/202112-divvy-tripdata.c
```

The files are of same number of variables and as such we can combine into one.

```
# combine the 12 data frames into one data frame
bike_trips_2021 <- rbind(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)
# remove empty rows/cols.
bike_trips_2021 <- janitor::remove_empty(bike_trips_2021, which = c("cols"))
bike_trips_2021 <- janitor::remove_empty(bike_trips_2021, which = c("rows"))
```

Data cleaning

check data structures to be sure they are in order

```
str(bike_trips_2021)
```

```
## 'data.frame':    5595063 obs. of  13 variables:
##  $ ride_id           : chr  "A3F8D895163BBB49" "0D139A3203274B87" "C7AE8E9CDB197A8E" "3097EF26414C70
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "01/01/2021 0:02" "01/01/2021 0:02" "01/01/2021 0:06" "01/01/2021 0:12"
##  $ ended_at          : chr  "01/01/2021 0:12" "01/01/2021 0:08" "01/01/2021 0:26" "01/01/2021 0:12"
##  $ start_station_name: chr  "" "State St & 33rd St" "Lakeview Ave & Fullerton Pkwy" "Montrose Harbor
##  $ start_station_id  : chr  "" "13216" "TA1309000019" "TA1308000012" ...
##  $ end_station_name  : chr  "" "MLK Jr Dr & 29th St" "Ritchie Ct & Banks St" "Montrose Harbor" ...
##  $ end_station_id    : chr  "" "TA1307000139" "KA1504000134" "TA1308000012" ...
##  $ start_lat         : num  42 41.8 41.9 42 42 ...
##  $ start_lng         : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  42 41.8 41.9 42 42 ...
##  $ end_lng           : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Changing to suitable data types

All data structures are in order except the start and end date that are in character format. This has to be corrected to date and time format.

```
bike_trips_2021$started_at = lubridate::dmy_hm(bike_trips_2021$started_at)
bike_trips_2021$ended_at = lubridate::dmy_hm(bike_trips_2021$ended_at)
# confirm Data structure
str(bike_trips_2021)
```

```
## 'data.frame':    5595063 obs. of  13 variables:
##  $ ride_id           : chr  "A3F8D895163BBB49" "0D139A3203274B87" "C7AE8E9CDB197A8E" "3097EF26414C70
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-01-01 00:02:00" "2021-01-01 00:02:00" ...
##  $ ended_at          : POSIXct, format: "2021-01-01 00:12:00" "2021-01-01 00:08:00" ...
##  $ start_station_name: chr  "" "State St & 33rd St" "Lakeview Ave & Fullerton Pkwy" "Montrose Harbor
##  $ start_station_id  : chr  "" "13216" "TA1309000019" "TA1308000012" ...
##  $ end_station_name  : chr  "" "MLK Jr Dr & 29th St" "Ritchie Ct & Banks St" "Montrose Harbor" ...
##  $ end_station_id    : chr  "" "TA1307000139" "KA1504000134" "TA1308000012" ...
##  $ start_lat         : num  42 41.8 41.9 42 42 ...
##  $ start_lng         : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  42 41.8 41.9 42 42 ...
##  $ end_lng           : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Data Manipulation

Adding calculated field(trip_duration), extracting start_date, end_date , start_time and end_time and weekday column.

```
bike_trips_2021$ride_length <- difftime(bike_trips_2021$ended_at, bike_trips_2021$started_at, units = c

bike_trips_2021$start_date <- as.Date(bike_trips_2021$started_at)
```

```
bike_trips_2021$end_date <- as.Date(bike_trips_2021$ended_at)

bike_trips_2021$start_hour <- lubridate::hour(bike_trips_2021$started_at)
bike_trips_2021$end_hour <- lubridate::hour(bike_trips_2021$ended_at)


bike_trips_2021$day_of_week <- wday(bike_trips_2021$started_at, label = TRUE)
#drop rows with ride_length <= 0 and columns not needed
df <- bike_trips_2021 %>% filter(ride_length > 0) %>% drop_na() %>%
  select(-ride_id, -end_station_name, -end_station_id)
bike_trips_2021 <- df
```

view Cleaned Data

```
head(bike_trips_2021)
```

```
##   rideable_type          started_at            ended_at
## 1 electric_bike 2021-01-01 00:02:00 2021-01-01 00:12:00
## 2  classic_bike 2021-01-01 00:02:00 2021-01-01 00:08:00
## 3  classic_bike 2021-01-01 00:06:00 2021-01-01 00:26:00
## 4 electric_bike 2021-01-01 00:12:00 2021-01-01 00:20:00
## 5 electric_bike 2021-01-01 00:12:00 2021-01-01 00:43:00
## 6 electric_bike 2021-01-01 00:13:00 2021-01-01 00:33:00
##              start_station_name start_station_id start_lat start_lng  end_lat
## 1                                                  41.98000 -87.65000 41.98000
## 2            State St & 33rd St            13216  41.83473 -87.62581 41.84205
## 3 Lakeview Ave & Fullerton Pkwy     TA1309000019  41.92586 -87.63897 41.90687
## 4    Kedzie Ave & Milwaukee Ave            13085  41.92953 -87.70790 41.92000
## 5       Western Ave & Howard St              527  42.01886 -87.69002 41.92468
## 6               Montrose Harbor     TA1308000012  41.96390 -87.63821 41.98401
##     end_lng member_casual ride_length start_date   end_date start_hour end_hour
## 1 -87.66000        member     10 mins 2021-01-01 2021-01-01          0        0
## 2 -87.61700        member      6 mins 2021-01-01 2021-01-01          0        0
## 3 -87.62622        member     20 mins 2021-01-01 2021-01-01          0        0
## 4 -87.72000        member      8 mins 2021-01-01 2021-01-01          0        0
## 5 -87.68933        member     31 mins 2021-01-01 2021-01-01          0        0
## 6 -87.65234        casual     20 mins 2021-01-01 2021-01-01          0        0
##   day_of_week
## 1         Fri
## 2         Fri
## 3         Fri
## 4         Fri
## 5         Fri
## 6         Fri
```

```
#**Arrange Data via ride_length to see the min ride_length**
head(arrange(bike_trips_2021, (bike_trips_2021$ride_length)))
```

```
##   rideable_type          started_at            ended_at
## 1 electric_bike 2021-01-01 01:04:00 2021-01-01 01:05:00
```

```
## 2 electric_bike 2021-01-01 02:16:00 2021-01-01 02:17:00
## 3   docked_bike 2021-01-01 02:20:00 2021-01-01 02:21:00
## 4  classic_bike 2021-01-01 02:29:00 2021-01-01 02:30:00
## 5  classic_bike 2021-01-01 06:26:00 2021-01-01 06:27:00
## 6  classic_bike 2021-01-01 09:45:00 2021-01-01 09:46:00
##           start_station_name start_station_id start_lat start_lng  end_lat
## 1                                               41.90000 -87.64000 41.90000
## 2                                               41.90000 -87.69000 41.90000
## 3 LaSalle St & Washington St            13006  41.88266 -87.63253 41.88266
## 4       McClurg Ct & Erie St     KA1503000041  41.89450 -87.61785 41.89450
## 5   Wilton Ave & Belmont Ave     TA1307000134  41.94018 -87.65304 41.94018
## 6            Millennium Park            13008  41.88103 -87.62408 41.88103
##     end_lng member_casual ride_length start_date   end_date start_hour end_hour
## 1 -87.64000        casual      1 mins 2021-01-01 2021-01-01          1        1
## 2 -87.69000        member      1 mins 2021-01-01 2021-01-01          2        2
## 3 -87.63253        casual      1 mins 2021-01-01 2021-01-01          2        2
## 4 -87.61785        casual      1 mins 2021-01-01 2021-01-01          2        2
## 5 -87.65304        member      1 mins 2021-01-01 2021-01-01          6        6
## 6 -87.62408        member      1 mins 2021-01-01 2021-01-01          9        9
##   day_of_week
## 1         Fri
## 2         Fri
## 3         Fri
## 4         Fri
## 5         Fri
## 6         Fri
```

##Create summary data frame

```
bike_trips_2021_summ <- bike_trips_2021 %>%
  group_by(member_casual,rideable_type,start_date,day_of_week) %>%
                summarise(Total_ride_length = sum(ride_length),
                  Average_ride_length = mean(ride_length),
                  Median = median(ride_length),
                  Max_ride_length = max(ride_length),
                  Min_ride_length = min(ride_length),
                  Count = n()
                ) %>% ungroup()
```

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

```
head(bike_trips_2021_summ)
```

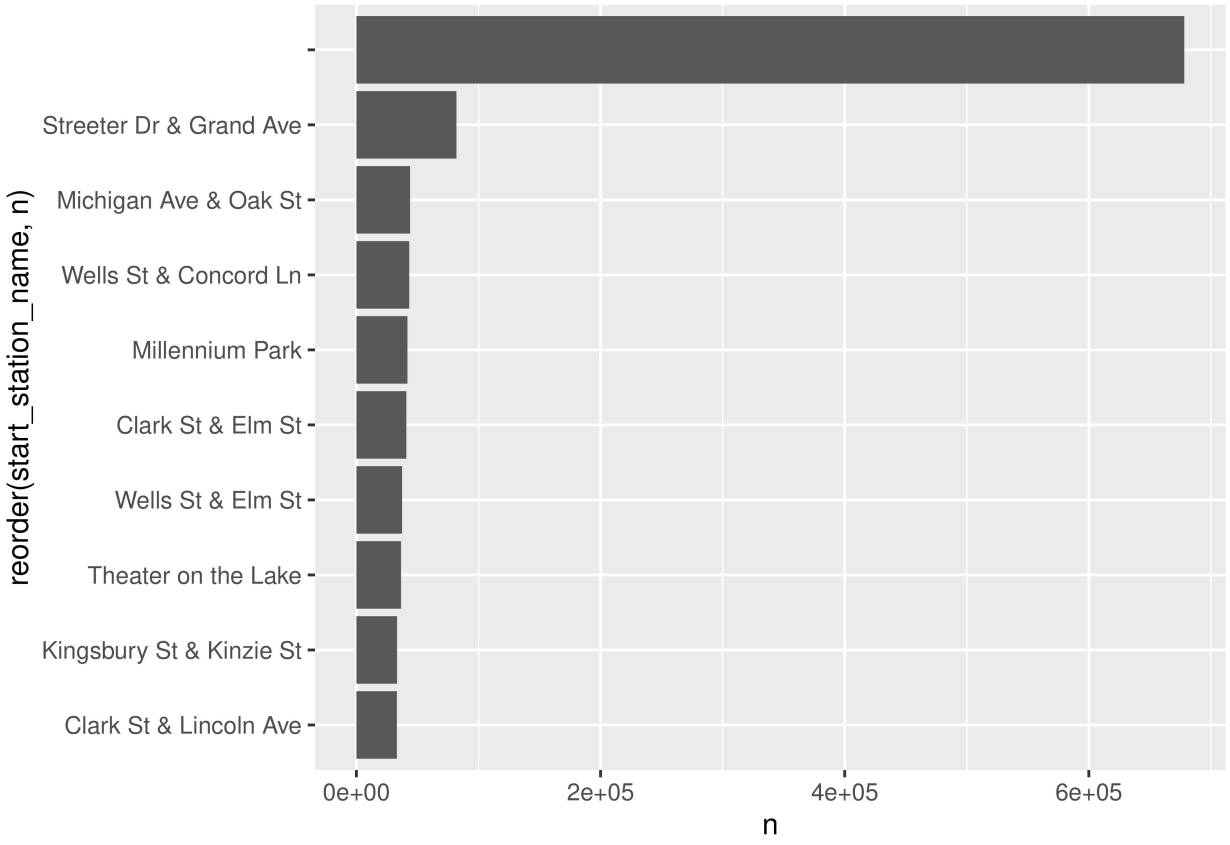
```
## # A tibble: 6 x 10
##   member_cas~1 ridea~2 start_date day_o~3 Total~4 Avera~5 Median Max_r~6 Min_r~7
##   <chr>        <chr>   <date>     <ord>   <drtn>  <drtn>  <drtn> <drtn>  <drtn>
## 1 casual       classi~ 2021-01-01 Fri     2345 m~ 24.175~ 11 mi~  810 m~ 1 mins
## 2 casual       classi~ 2021-01-02 Sat     6441 m~ 25.764~ 17 mi~  202 m~ 2 mins
## 3 casual       classi~ 2021-01-03 Sun     7388 m~ 25.388~ 15 mi~  190 m~ 1 mins
## 4 casual       classi~ 2021-01-04 Mon     3937 m~ 17.420~ 13 mi~  101 m~ 1 mins
## 5 casual       classi~ 2021-01-05 Tue     5346 m~ 18.183~ 12 mi~  222 m~ 1 mins
```

```
## 6 casual       classi~ 2021-01-06 Wed     7542 m~ 24.329~ 13 mi~ 1162 m~ 1 mins
## # ... with 1 more variable: Count <int>, and abbreviated variable names
## #   1: member_casual, 2: rideable_type, 3: day_of_week, 4: Total_ride_length,
## #   5: Average_ride_length, 6: Max_ride_length, 7: Min_ride_length
```

Top Ten (10) Start Stations

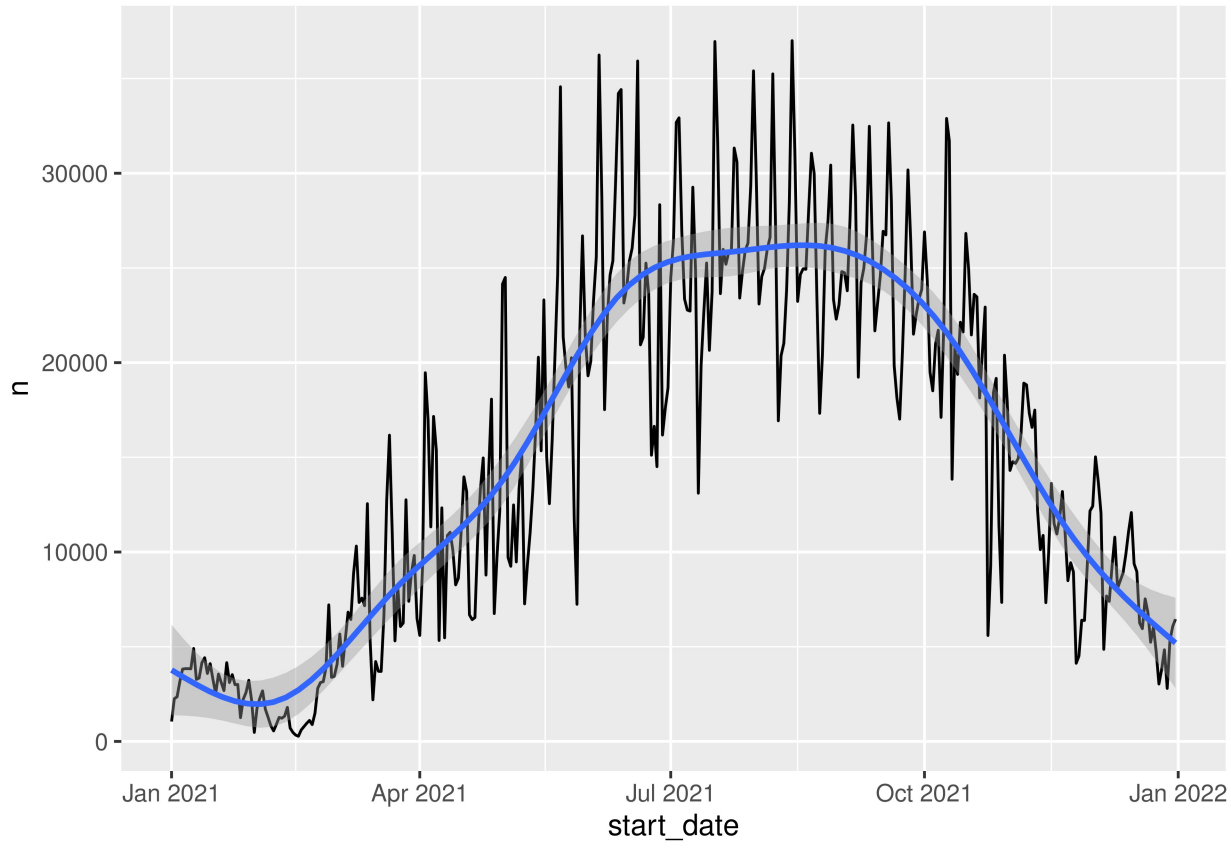
```
 bike_trips_2021 %>%  count(start_station_name,SORT = TRUE) %>% top_n(10) %>%
 ggplot() + geom_col(aes(x=reorder(start_station_name,n),y=n)) + coord_flip()
```

```
## Selecting by n
```

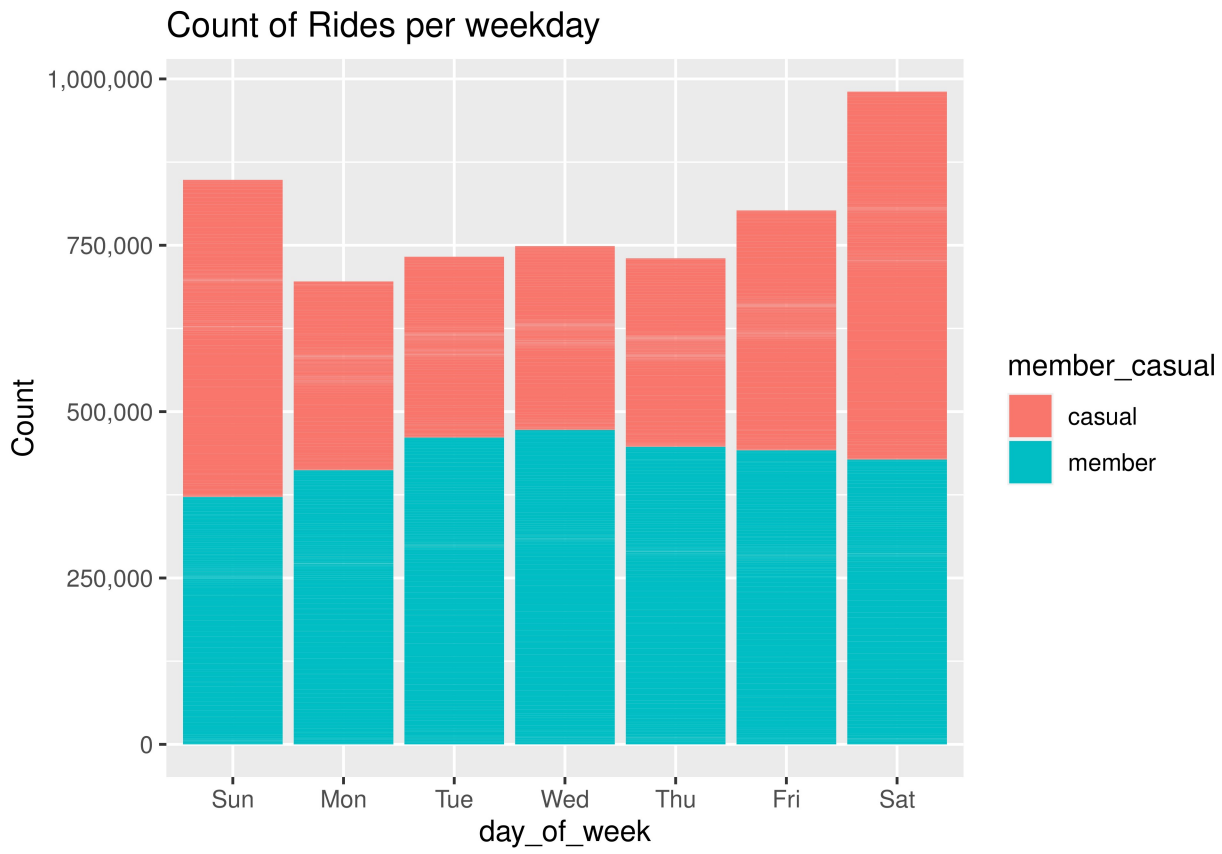



```
#**Bike Trips Per month**
 bike_trips_2021 %>% count(start_date,sort = TRUE) %>%
  ggplot() + geom_line(aes(x=start_date,n)) +
  geom_smooth(aes(x=start_date,y=n),method = "gam")
```

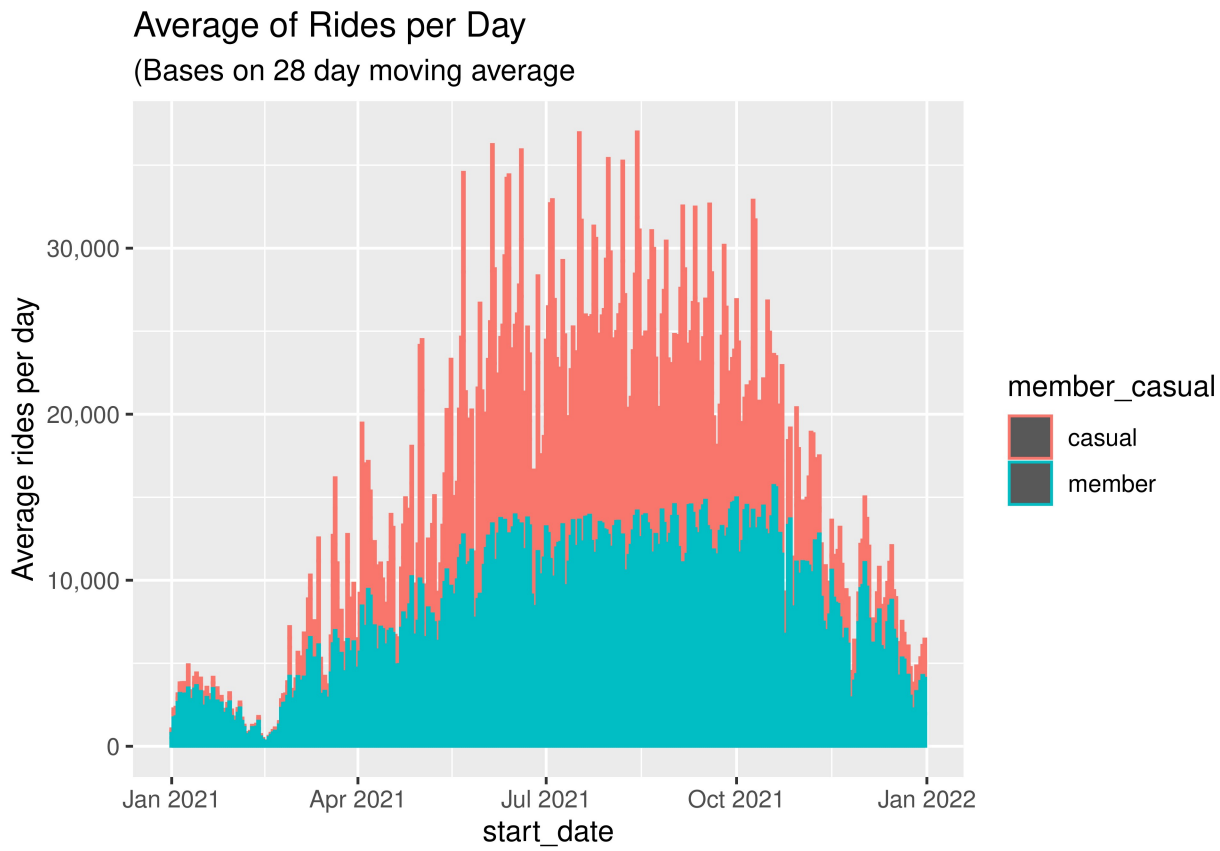
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```




```
#**Count by Bike Trips per day_of_Week**
ggplot(bike_trips_2021_summ) + geom_col(aes(x=day_of_week,y=Count,fill=member_casual)) +
  scale_y_continuous(labels = comma) +
  labs(title="Count of Rides per weekday")
```




```
bike_trips_2021_summ %>% ggplot() + geom_col(aes(x=start_date,y=Count, color=member_casual)) +
  scale_y_continuous(labels = comma) +
  labs(title = "Average of Rides per Day",
       subtitle = "(Bases on 28 day moving average",
       y="Average rides per day")
```




```
#**Export clean Data**
write.csv(bike_trips_2021_summ,"C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers\\bike_trips_2021_summ.csv"
write.csv(bike_trips_2021,"C:/Users/Nkechi Ihewulezi/Downloads/R/Bikers\\bike_trips_2021.csv",  row.nam
```