Case Study

Prakhar

2023-05-26

```
library(tidyverse)  #helps wrangle data
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)  #helps wrangle date attributes
library(ggplot2)  #helps visualize data
setwd("C:/Users/sriva/OneDrive/Desktop/COLLEGE/google data analytics/tab/Track 1")
```

```
df_2019_q1 <- read_csv("Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 365069 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
df_2019_q2 <- read_csv("Divvy_Trips_2019_Q2.csv")
```

```
## Rows: 1108163 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): 03 - Rental Start Station Name, 02 - Rental End Station Name, User...
## dbl  (5): 01 - Rental Details Rental ID, 01 - Rental Details Bike ID, 03 - R...
## num  (1): 01 - Rental Details Duration In Seconds Uncapped
## dttm (2): 01 - Rental Details Local Start Time, 01 - Rental Details Local En...
```

```
##
## 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.
```

```
df_2019_q3 <- read_csv("Divvy_Trips_2019_Q3.csv")
```

```
## Rows: 1640718 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
df_2019_q4 <- read_csv("Divvy_Trips_2019_Q4.csv")
```

```
## Rows: 704054 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
(df_2019_q2 <-  rename(df_2019_q2,
                    trip_id = "01 - Rental Details Rental ID"
                   ,bikeid = "01 - Rental Details Bike ID"
                   ,tripduration = "01 - Rental Details Duration In Seconds Uncapped"
                   ,start_time = "01 - Rental Details Local Start Time"
                   ,end_time = "01 - Rental Details Local End Time"
                   ,from_station_name = "03 - Rental Start Station Name"
                   ,from_station_id = "03 - Rental Start Station ID"
                   ,to_station_name = "02 - Rental End Station Name"
                   ,to_station_id = "02 - Rental End Station ID"
                   ,usertype = "User Type"
                   ,gender = "Member Gender"
                   ,birthyear = "05 - Member Details Member Birthday Year" ))
```

```
## # A tibble: 1,108,163 x 12
##     trip_id start_time          end_time            bikeid tripduration
##       <dbl> <dttm>              <dttm>               <dbl>        <dbl>
##  1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48   6251          446
##  2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30   6226         1048
##  3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19   5649          252
##  4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58   4151          357
##  5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13   3270         1007
```

```
##  6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56   3123          257
##  7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41   6418          548
##  8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11   4513          383
##  9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44   3280         2137
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39   5534         2120
## # i 1,108,153 more rows
## # i 7 more variables: from_station_id <dbl>, from_station_name <chr>,
## #   to_station_id <dbl>, to_station_name <chr>, usertype <chr>, gender <chr>,
## #   birthyear <dbl>
```

```
all_data<- bind_rows(df_2019_q1,df_2019_q2,df_2019_q3,df_2019_q4)
```

```
all_data<-all_data %>%
  select(-c(gender, birthyear )) %>%
  mutate(trip_id = as.character(trip_id), bikeid=as.character(bikeid), from_station_id = as.character(f
```

```
all_data<- rename(all_data, member_casual = "usertype")
```

```
# Reassign to the desired values
all_data <-  all_data %>%
  mutate(member_casual = recode(member_casual
                           ,"Subscriber" = "member"
                           ,"Customer" = "casual"))
```

```
#Add columns that list the date, month, day, and year of each ride
all_data$date <- as.Date(all_data$start_time) #The default format is yyyy-mm-dd
all_data$month <- format(as.Date(all_data$date), "%m")
all_data$day <- format(as.Date(all_data$date), "%d")
all_data$year <- format(as.Date(all_data$date), "%Y")
all_data$day_of_week <- format(as.Date(all_data$date), "%A")


all_data$ride_length <- difftime(all_data$end_time,all_data$start_time)
```

```
all_data$ride_length <- as.numeric(as.character(all_data$ride_length))
is.numeric(all_data$ride_length)
```

```
## [1] TRUE
```

```
all_data <- all_data[!(all_data$ride_length<0),]
```

```
aggregate(all_data$ride_length ~ all_data$member_casual, FUN = mean)
```

```
##   all_data$member_casual all_data$ride_length
## 1                 casual             57.01802
## 2                 member             14.32780
```

```
aggregate(all_data$ride_length ~ all_data$member_casual, FUN = median)
```

```
##   all_data$member_casual all_data$ride_length
## 1                 casual             25.83333
## 2                 member              9.80000
```

```
aggregate(all_data$ride_length ~ all_data$member_casual, FUN = max)
```

```
##   all_data$member_casual all_data$ride_length
## 1                 casual             177200.4
## 2                 member             150943.9
```

```
aggregate(all_data$ride_length ~ all_data$member_casual, FUN = min)
```

```
##   all_data$member_casual all_data$ride_length
## 1                 casual             1.016667
## 2                 member             1.016667
```

```
all_data$day_of_week <- ordered(all_data$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesda
```

```
aggregate(all_data$ride_length ~ all_data$member_casual + all_data$day_of_week, FUN = mean)
```

```
##    all_data$member_casual all_data$day_of_week all_data$ride_length
## 1                  casual               Sunday             56.18519
## 2                  member               Sunday             15.40290
## 3                  casual               Monday             54.49989
## 4                  member               Monday             14.24928
## 5                  casual              Tuesday             57.41328
## 6                  member              Tuesday             14.15259
## 7                  casual            Wednesday             60.33407
## 8                  member            Wednesday             13.80984
## 9                  casual             Thursday             59.95112
## 10                 member             Thursday             13.77979
## 11                 casual               Friday             60.17561
## 12                 member               Friday             13.89748
## 13                 casual             Saturday             54.06111
## 14                 member             Saturday             16.30271
```

```
# analyze ridership data by type and weekday
all_data %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                              #calculates the number of rides and average
  ,average_duration = mean(ride_length)) %>%        # calculates the average duration
  arrange(member_casual, weekday)
```

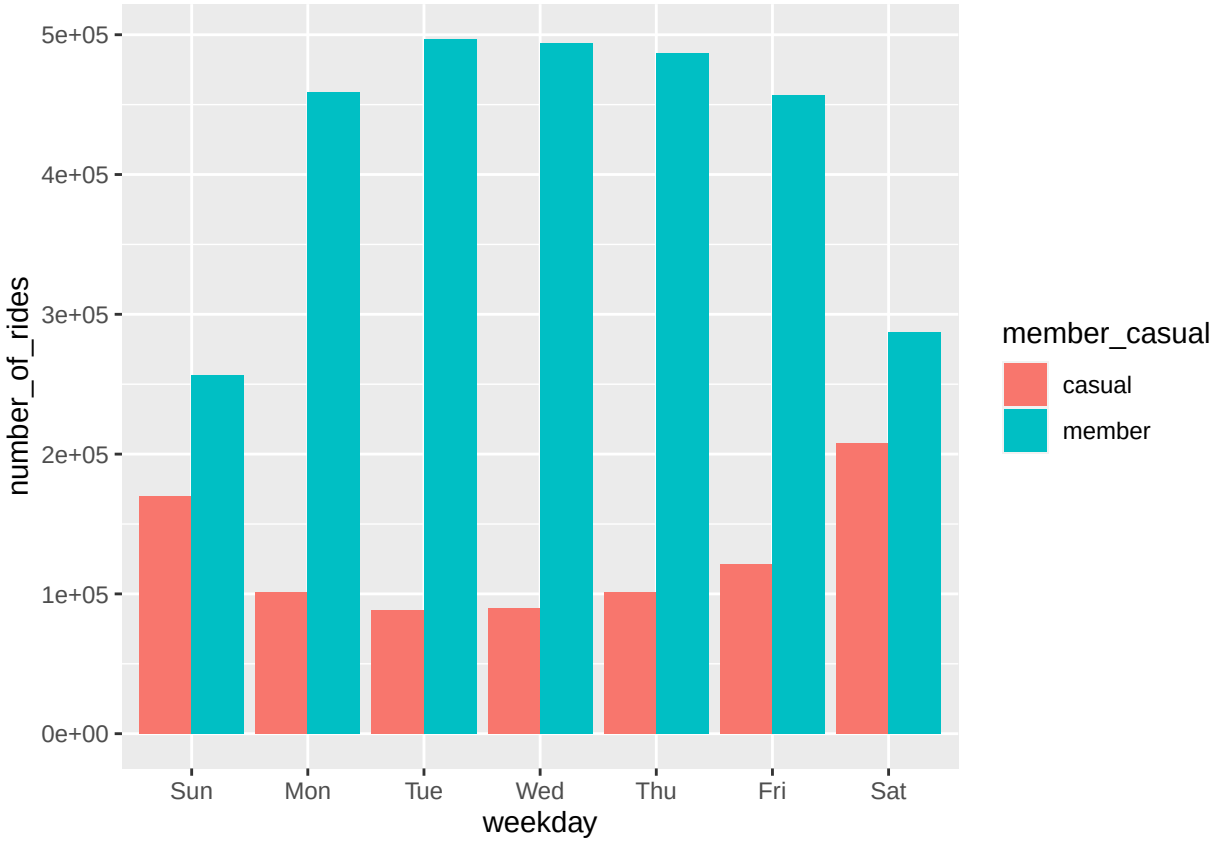
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              170173             56.2
##  2 casual        Mon              101489             54.5
##  3 casual        Tue               88655             57.4
```

```
##  4 casual        Wed              89745            60.3
##  5 casual        Thu             101372            60.0
##  6 casual        Fri             121141            60.2
##  7 casual        Sat             208056            54.1
##  8 member        Sun             256234            15.4
##  9 member        Mon             458780            14.2
## 10 member        Tue             497025            14.2
## 11 member        Wed             494277            13.8
## 12 member        Thu             486915            13.8
## 13 member        Fri             456966            13.9
## 14 member        Sat             287163            16.3
```

```r
# Let's visualize the number of rides by rider type
all_data %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

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



```r
# Let's create a visualization for average duration
all_data %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

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