Case Study: Analytics Divvy using R

Dinh Cuong

2024-06-02

Purpose

Answer the question: How do annual members and casual riders user Cyclistic bikes differently?

1. Install package "tidyverse"

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

2. Import data

```
library(readr)
all_trip_2019 <- read_csv("Divvy_Trips_2019_Q1 - Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 365069 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): start_time, end_time, from_station_name, to_station_name, usertype,...
## dbl (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num (1): tripduration
##
## 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.
```

3. Rename columns

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
all_trip_2019 <- rename(all_trip_2019
                   ,ride_id = trip_id
                   ,rideable_type = bikeid
                   ,started_at = start_time
                   ,ended_at = end_time
                   ,start_station_name = from_station_name
                   ,start_station_id = from_station_id
```

```
                   ,end_station_name = to_station_name
                   ,end_station_id = to_station_id
                   ,member_casual = usertype)
```

4. Check information all_trip_2019

```
str(all_trip_2019)
```

```
## spc_tbl_ [365,069 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:365069] 21742443 21742444 21742445 21742446 21742447 ...
##  $ started_at        : chr [1:365069] "2019-01-01 0:04:37" "2019-01-01 0:08:13" "2019-01-01 0:13:23"
##  $ ended_at          : chr [1:365069] "2019-01-01 0:11:07" "2019-01-01 0:15:34" "2019-01-01 0:27:12"
##  $ rideable_type     : num [1:365069] 2167 4386 1524 252 1170 ...
##  $ tripduration      : num [1:365069] 390 441 829 1783 364 ...
##  $ start_station_id  : num [1:365069] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:365069] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:365069] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:365069] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ member_casual     : chr [1:365069] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender            : chr [1:365069] "Male" "Female" "Female" "Male" ...
##  $ birthyear         : num [1:365069] 1989 1990 1994 1993 1994 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_character(),
##   ..   end_time = col_character(),
##   ..   bikeid = col_double(),
##   ..   tripduration = col_number(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   usertype = col_character(),
##   ..   gender = col_character(),
##   ..   birthyear = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

5. Change type of 2 columns ride_id, rideable_type to character

```
all_trip_2019 <- mutate(all_trip_2019, ride_id = as.character(ride_id)
                        ,rideable_type = as.character(rideable_type))
```

5. Create a new table name Data_trips_2019 just includes information is necessary

```
Data_trips_2019 <- all_trip_2019%>%
  select(ride_id, rideable_type, member_casual,
         start_station_id, start_station_name,
         end_station_id, end_station_name)
```

7. Check Data_trips_2019

```
colnames(Data_trips_2019) # Name of columns
```

```
## [1] "ride_id"            "rideable_type"      "member_casual"
## [4] "start_station_id"   "start_station_name" "end_station_id"
## [7] "end_station_name"
```

```
nrow(Data_trips_2019) # Number of rows
```

```
## [1] 365069
```

```
dim(Data_trips_2019) # Number of columms
```

```
## [1] 365069      7
```

```
summary(Data_trips_2019) # Information for each column (min, max, mean,...)
```

```
##    ride_id          rideable_type      member_casual      start_station_id
##  Length:365069      Length:365069      Length:365069      Min.   :  2.0
##  Class :character   Class :character   Class :character   1st Qu.: 76.0
##  Mode  :character   Mode  :character   Mode  :character   Median :170.0
##                                                           Mean   :198.1
##                                                           3rd Qu.:287.0
##                                                           Max.   :665.0
##  start_station_name end_station_id  end_station_name
##  Length:365069      Min.   :  2.0   Length:365069
##  Class :character   1st Qu.: 76.0   Class :character
##  Mode  :character   Median :168.0   Mode  :character
##                     Mean   :198.6
##                     3rd Qu.:287.0
##                     Max.   :665.0
```

8. Change data in member_casual column member -> Subcriber casual -> Customer

```
Data_trips_2019 <- mutate(Data_trips_2019,member_casual = recode(member_casual
                                                  ,"Subcriber" = "member"
                                                  ,"Customer" = "casual"))
```

9. Check data of member_casual column

```
unique(Data_trips_2019$member_casual)
```

```
## [1] "Subscriber" "casual"
```

10. Add date, month, day, year, day_of_week into Data_trips_2019 table

```
Data_trips_2019$date <- as.Date(all_trip_2019$started_at) #The default format is yyyy-mm-dd
Data_trips_2019$month <- format(as.Date(Data_trips_2019$date), "%m")
Data_trips_2019$day <- format(as.Date(Data_trips_2019$date), "%d")
Data_trips_2019$year <- format(as.Date(Data_trips_2019$date), "%Y")
Data_trips_2019$day_of_week <- format(as.Date(Data_trips_2019$date), "%A")
```

11. Add a column to calcuate ride_length

```
Data_trips_2019$ride_length <- difftime(all_trip_2019$ended_at,all_trip_2019$started_at)
```

12. Change type of ride_length to numberic

```
Data_trips_2019 <- mutate(Data_trips_2019,
                          ride_length = as.numeric(ride_length))
```

13. Create new table name Data_trips_2019_v2 to remove bad data

```
Data_trips_2019_v2 <- Data_trips_2019 %>%
  filter(ride_length > 0 | start_station_name == "HQ QR")
```

14. Sort day_of_week column

```r
Data_trips_2019_v2$day_of_week <- ordered(Data_trips_2019_v2$day_of_week, levels=c("Sunday", "Monday"
        ,"Tuesday", "Wednesday"
        ,"Thursday", "Friday", "Saturday"))
```

15. Calculate mean ride_length group by member_casual and day_of_week

```r
aggregate(Data_trips_2019_v2$ride_length ~ Data_trips_2019_v2$member_casual + Data_trips_2019_v2$day_of_
          FUN = mean)
```

```
##    Data_trips_2019_v2$member_casual Data_trips_2019_v2$day_of_week
## 1                            casual                         Sunday
## 2                        Subscriber                         Sunday
## 3                            casual                         Monday
## 4                        Subscriber                         Monday
## 5                            casual                        Tuesday
## 6                        Subscriber                        Tuesday
## 7                            casual                      Wednesday
## 8                        Subscriber                      Wednesday
## 9                            casual                       Thursday
## 10                       Subscriber                       Thursday
## 11                           casual                         Friday
## 12                       Subscriber                         Friday
## 13                           casual                       Saturday
## 14                       Subscriber                       Saturday
##    Data_trips_2019_v2$ride_length
## 1                        41.58239
## 2                        16.79934
## 3                        44.45613
## 4                        14.63459
## 5                        40.45482
## 6                        14.36700
## 7                        51.95724
## 8                        12.09533
## 9                       133.79221
## 10                       12.00980
## 11                       59.89280
## 12                       13.88976
## 13                       60.32985
## 14                       16.98921
```

16. Calculate number of rides by type ride

```r
library(lubridate) # to use function wday()
```

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

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

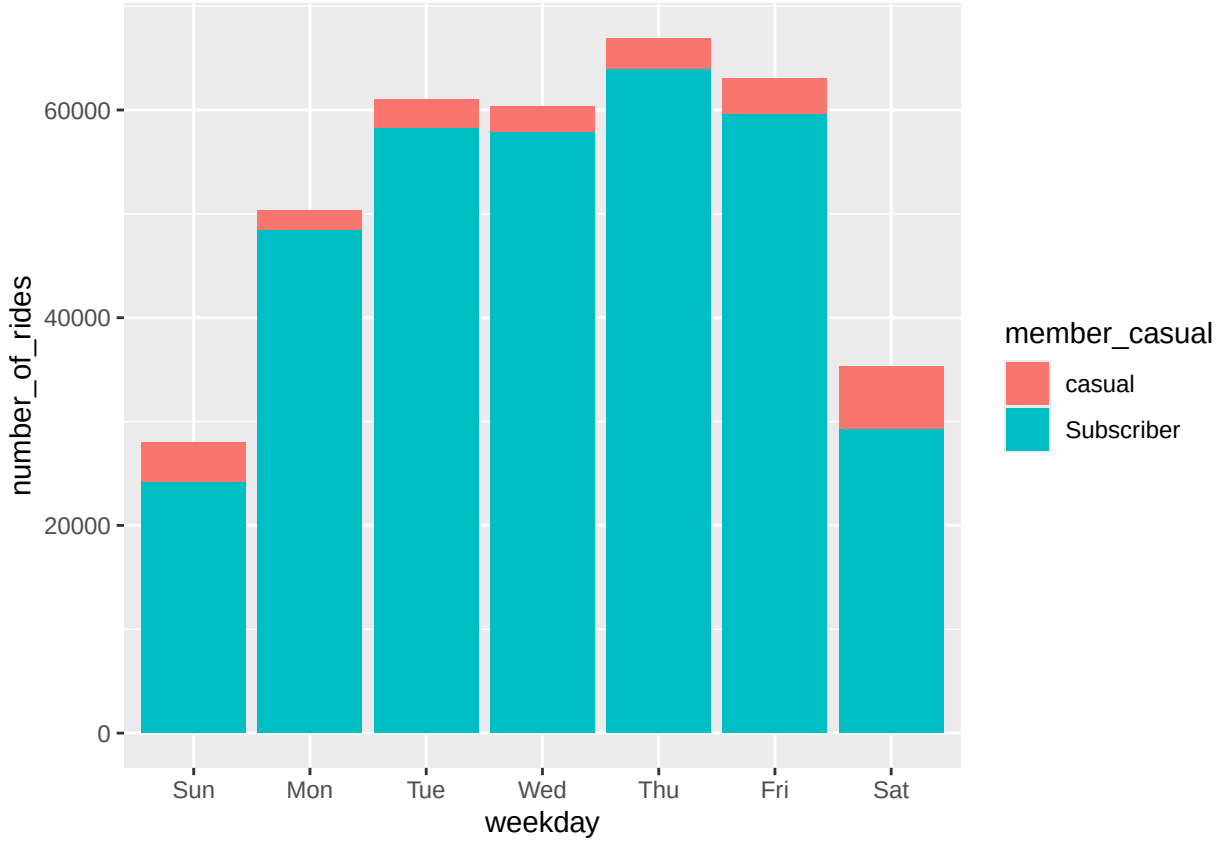
```r
number_of_rides_by_type <- Data_trips_2019_v2 %>%
  mutate(weekday = wday(date, label = TRUE)) %>% #creates weekday field using wday()
group_by(member_casual, weekday) %>% #groups by usertype and weekday
  summarise(number_of_rides = n() #calculates the number of rides and average duration
            ,average_duration = mean(ride_length)) %>% # calculates the average duration
arrange(member_casual, weekday) #sort
```

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

Create visual

```
library(ggplot2)
ggplot(data = number_of_rides_by_type)+
 geom_col(mapping = aes(x = weekday, y = number_of_rides, fill = member_casual, position = "dodge"))
```

```
## Warning in geom_col(mapping = aes(x = weekday, y = number_of_rides, fill =
## member_casual, : Ignoring unknown aesthetics: position
```



17.

Create a visualization for average duration

```
average_duration <- Data_trips_2019_v2 %>%
  mutate(weekday = wday(date, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)
```

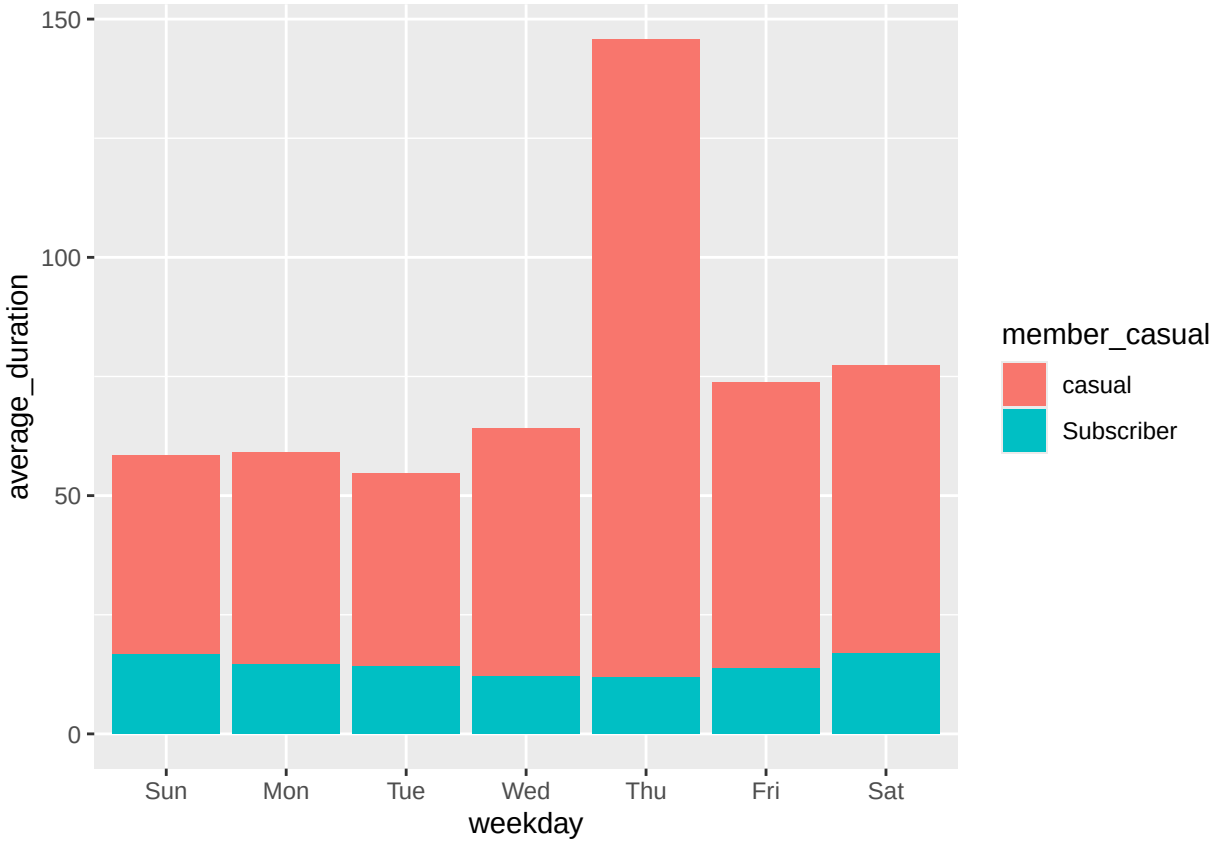
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

Visual

```
ggplot(data = average_duration)+
geom_col(mapping = aes(x = weekday, y = average_duration, fill = member_casual, position = "dodge"))
```

```
## Warning in geom_col(mapping = aes(x = weekday, y = average_duration, fill =
```

```
## member_casual, : Ignoring unknown aesthetics: position
```



Comment

- Number of subscriber is higher than casual on day of week
- Duration using of casual is higher than subscriber