Bike_Share

2022-09-20

Definition of riders

- Customer riders: purchase single-ride or full-day passes
- Member riders: purchase annual memberships

Business task

- How do annual members and customer riders use share_bikes differently?

Dataset

Public data set on https://divvy-tripdata.s3.amazonaws.com/index.html. Download 2019 full year data to use in this case study.

Data propare

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
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   1.0.10
## v tidyr   1.2.1      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(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
```

Data prepare

```
trips_df <- 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
## 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.
```

```
#View(trips_df)
#spec(trips_df)
#table(trips_df$usertype)
#is.numeric(trips_df$tripduration)
```

Data process

- Add week day column (Mon.-Sun.)

```
trips_df <- trips_df %>% mutate(weekday=weekdays(end_time))
colnames(trips_df)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
## [13] "weekday"
```

- Drop station related info, since there is no lat.& long. associated

```
trips_df <- trips_df %>%
  select(-c(from_station_id,from_station_name,to_station_id,to_station_name))
```

```
colnames(trips_df)
```

```
## [1] "trip_id"      "start_time"   "end_time"     "bikeid"       "tripduration"
## [6] "usertype"     "gender"       "birthyear"    "weekday"
```

```
dim(trips_df)
```

```
## [1] 365069      9
```

```
head(trips_df)
```

```
## # A tibble: 6 x 9
##    trip_id start_time          end_time            bikeid tripd~1 usert~2 gender
##      <dbl> <dttm>              <dttm>               <dbl>   <dbl> <chr>   <chr>
## 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07   2167     390 Subscr~ Male
## 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34   4386     441 Subscr~ Female
## 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12   1524     829 Subscr~ Female
## 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28    252    1783 Subscr~ Male
## 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56   1170     364 Subscr~ Male
## 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09   2437     216 Subscr~ Female
## # ... with 2 more variables: birthyear <dbl>, weekday <chr>, and abbreviated
## #   variable names 1: tripduration, 2: usertype
```

```
summary(trips_df)
```

```
##     trip_id           start_time
##  Min.   :21742443   Min.   :2019-01-01 00:04:37.00
##  1st Qu.:21848765   1st Qu.:2019-01-23 05:26:54.00
##  Median :21961829   Median :2019-02-25 07:52:56.00
##  Mean   :21960872   Mean   :2019-02-19 21:43:15.42
##  3rd Qu.:22071823   3rd Qu.:2019-03-17 16:52:47.00
##  Max.   :22178528   Max.   :2019-03-31 23:53:48.00
##
##     end_time                          bikeid      tripduration
##  Min.   :2019-01-01 00:11:07.00   Min.   :   1   Min.   :      61
##  1st Qu.:2019-01-23 05:49:40.00   1st Qu.:1777   1st Qu.:     326
##  Median :2019-02-25 08:03:50.00   Median :3489   Median :     524
##  Mean   :2019-02-19 22:00:11.91   Mean   :3429   Mean   :    1016
##  3rd Qu.:2019-03-17 17:16:16.00   3rd Qu.:5157   3rd Qu.:     866
##  Max.   :2019-06-17 16:04:35.00   Max.   :6471   Max.   :10628400
##
##    usertype            gender            birthyear       weekday
##  Length:365069      Length:365069      Min.   :1900    Length:365069
##  Class :character   Class :character   1st Qu.:1975    Class :character
##  Mode  :character   Mode  :character   Median :1985    Mode  :character
##                                        Mean   :1982
##                                        3rd Qu.:1990
##                                        Max.   :2003
##                                        NA's   :18023
```

- Check if all tripduration are and numeric and positive values

```
is.numeric(trips_df$tripduration)
```

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

- Check total categories of usertype

```
table(trips_df$usertype)
```

```
##
##   Customer Subscriber
##      23163     341906
```

Analysis

- Descriptive analysis on trip duration

```
summary(trips_df$tripduration)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##       61      326      524     1016      866 10628400
```

- Compare trip duration between members and casual users

```
aggregate(trips_df$tripduration ~ trips_df$usertype, FUN = mean)
```

```
##   trips_df$usertype trips_df$tripduration
## 1          Customer             3715.7376
## 2        Subscriber              833.4669
```

```
aggregate(trips_df$tripduration ~ trips_df$usertype, FUN = median)
```

```
##   trips_df$usertype trips_df$tripduration
## 1          Customer                  1401
## 2        Subscriber                   501
```

```
aggregate(trips_df$tripduration ~ trips_df$usertype, FUN = max)
```

```
##   trips_df$usertype trips_df$tripduration
## 1          Customer              10628400
## 2        Subscriber               6096430
```

```
aggregate(trips_df$tripduration ~ trips_df$usertype, FUN = min)
```

```
##   trips_df$usertype trips_df$tripduration
## 1          Customer                    61
## 2        Subscriber                    61
```

- Calculate number of rides and average duration of each weekday by rider type

```
trips_df_summary <- trips_df %>%
  group_by(usertype, weekday) %>%
  summarise(number_of_rides = n(),average_duration = mean(tripduration)) %>%
  arrange(usertype, weekday)
```

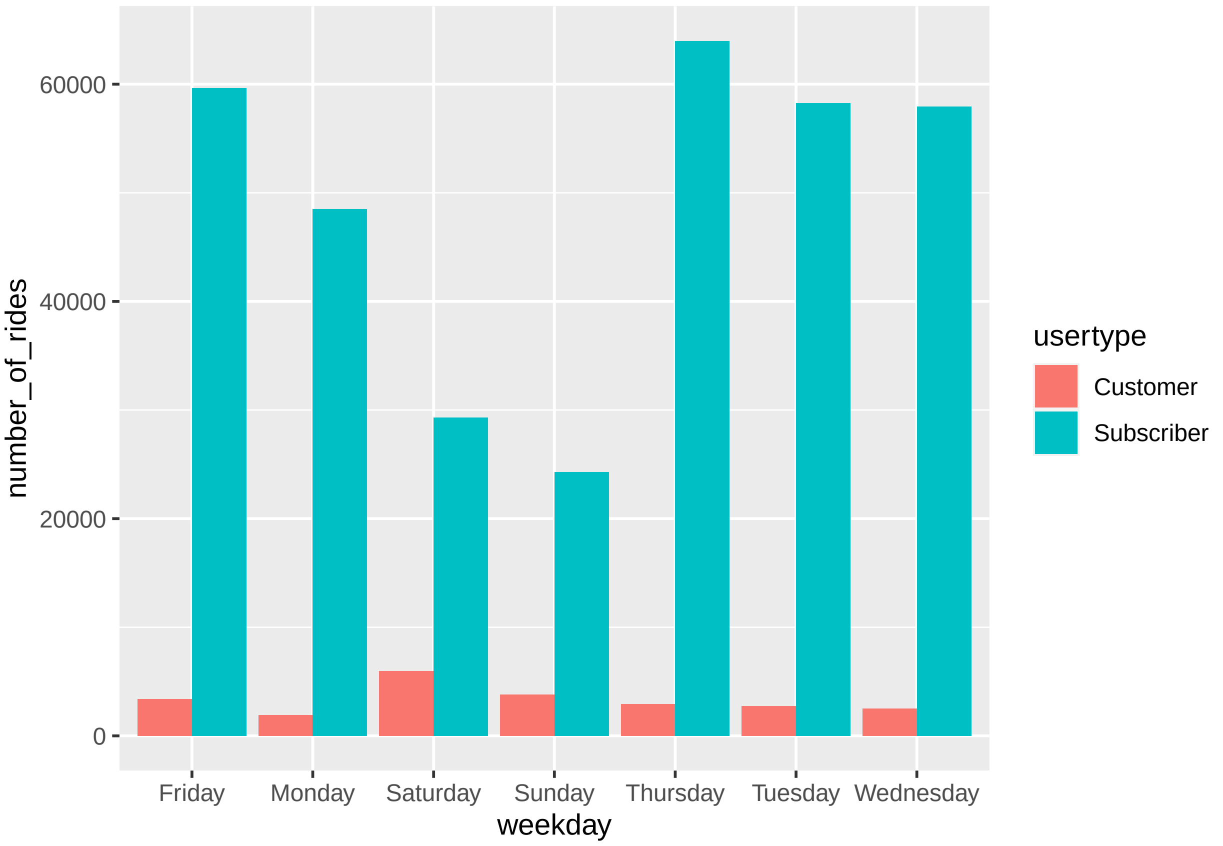
```
## `summarise()` has grouped output by 'usertype'. You can override using the
## `.groups` argument.
```

```
head(trips_df_summary, n=14)
```

```
## # A tibble: 14 x 4
## # Groups:   usertype [2]
##    usertype   weekday   number_of_rides average_duration
##    <chr>      <chr>               <int>            <dbl>
##  1 Customer   Friday               3362            3563.
##  2 Customer   Monday               1895            8696.
##  3 Customer   Saturday             5968            2315.
##  4 Customer   Sunday               3804            2955.
##  5 Customer   Thursday             2909            4089.
##  6 Customer   Tuesday              2730            3233.
##  7 Customer   Wednesday            2495            4743.
##  8 Subscriber Friday              59650             710.
##  9 Subscriber Monday              48507             795.
## 10 Subscriber Saturday            29302             883.
## 11 Subscriber Sunday              24279             881.
## 12 Subscriber Thursday            63977             822.
## 13 Subscriber Tuesday             58263             936.
## 14 Subscriber Wednesday           57928             858.
```

- Add data visualization for number of rides of each weekday by rider type

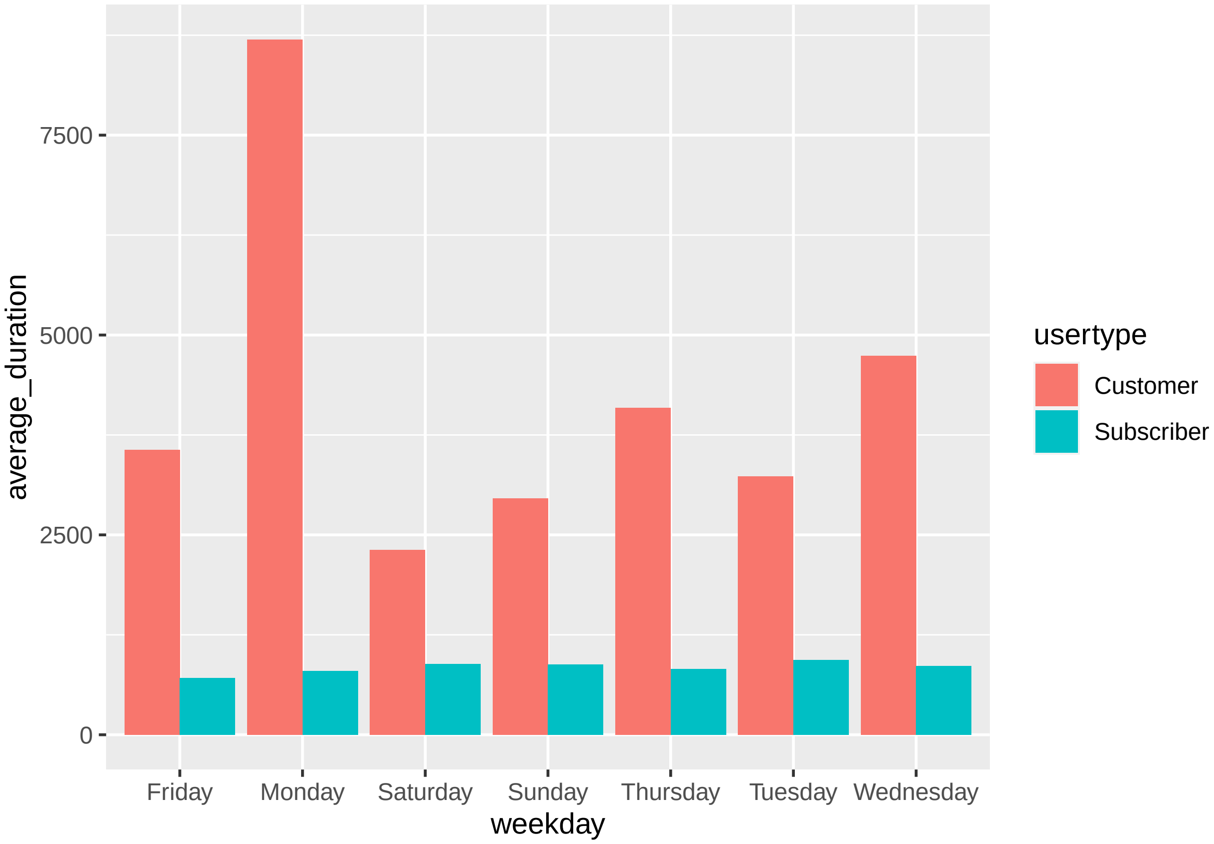
```
trips_df_summary %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = usertype)) +
  geom_col(position = "dodge")
```



*

Add data visualization for average duration of each weekday by rider type

```
trips_df_summary %>%
  ggplot(aes(x = weekday, y = average_duration, fill = usertype)) +
  geom_col(position = "dodge")
```



Export summary file for further analysis

write.csv(trips_df_summary, path='C:\Users\trips_df_summary.csv')

Key findings

- Customer riders has longer trip duration than member riders by each weekday
- Customer riders has less trips than member riders by each weekday

Recommendataions

- Design marketing strategies to convert casual riders into annual members.
- Run survey to collect data about why casual riders would buy a membership.