Bike analysis Q4

Daniele Favale

2024-01-22

Contents

0.1 Tools used:

For this analysis I decided to use R Studio due to its ability to manipulate big datasets

0.2 A description of all data sources used:

This file only takes into consideration the data from the last quarter (Q4) of 2019. There are further files with data from the other quarters (Q1, Q2, Q3). The following analysis only concerns the last four months and it is not possible to analyze monthly trends. For a more complete analysis and to be able to analyze seasonal trends it is necessary to also analyze the other quarters and combine the data to get a more complete picture.

0.3 What is the problem I'm trying to solve?

The objective is to understand how customers and subscribers use the bike share service differently. The results of this analysis can help understand how to convert occasional customers into annual subscribers.

———Library used———

```r
library(tidyverse)
library(ggplot2)
library(tidyr)
library(dplyr)
library(skimr)
library(lubridate)
library(forcats)
```

####################################–EXPLORE–#######################
###

```r
bike_original_4 <- read.csv("Divvy_Trips_2019_Q4.csv", sep = ",")
```

```r
head(bike_original_4)
```

```
##    trip_id          start_time            end_time bikeid tripduration
## 1 25223640 2019-10-01 00:01:39 2019-10-01 00:17:20   2215        940.0
## 2 25223641 2019-10-01 00:02:16 2019-10-01 00:06:34   6328        258.0
## 3 25223642 2019-10-01 00:04:32 2019-10-01 00:18:43   3003        850.0
## 4 25223643 2019-10-01 00:04:32 2019-10-01 00:43:43   3275      2,350.0
## 5 25223644 2019-10-01 00:04:34 2019-10-01 00:35:42   5294      1,867.0
## 6 25223645 2019-10-01 00:04:38 2019-10-01 00:10:51   1891        373.0
##   from_station_id              from_station_name to_station_id
## 1              20   Sheffield Ave & Kingsbury St           309
## 2              19 Throop (Loomis) St & Taylor St           241
## 3              84      Milwaukee Ave & Grand Ave           199
## 4             313  Lakeview Ave & Fullerton Pkwy           290
## 5             210      Ashland Ave & Division St           382
## 6             156      Clark St & Wellington Ave           226
##               to_station_name   usertype gender birthyear
## 1   Leavitt St & Armitage Ave Subscriber   Male      1987
## 2         Morgan St & Polk St Subscriber   Male      1998
## 3      Wabash Ave & Grand Ave Subscriber Female      1991
## 4      Kedzie Ave & Palmer Ct Subscriber   Male      1990
## 5 Western Ave & Congress Pkwy Subscriber   Male      1987
## 6    Racine Ave & Belmont Ave Subscriber Female      1994
```

```
glimpse(bike_original_4)
```

```
## Rows: 704,054
## Columns: 12
## $ trip_id           <int> 25223640, 25223641, 25223642, 25223643, 25223644, 25~
## $ start_time        <chr> "2019-10-01 00:01:39", "2019-10-01 00:02:16", "2019-~
## $ end_time          <chr> "2019-10-01 00:17:20", "2019-10-01 00:06:34", "2019-~
## $ bikeid            <int> 2215, 6328, 3003, 3275, 5294, 1891, 1061, 1274, 6011~
## $ tripduration      <chr> "940.0", "258.0", "850.0", "2,350.0", "1,867.0", "37~
## $ from_station_id   <int> 20, 19, 84, 313, 210, 156, 84, 156, 156, 336, 77, 19~
## $ from_station_name <chr> "Sheffield Ave & Kingsbury St", "Throop (Loomis) St ~
## $ to_station_id     <int> 309, 241, 199, 290, 382, 226, 142, 463, 463, 336, 50~
## $ to_station_name   <chr> "Leavitt St & Armitage Ave", "Morgan St & Polk St", ~
## $ usertype          <chr> "Subscriber", "Subscriber", "Subscriber", "Subscribe~
## $ gender            <chr> "Male", "Male", "Female", "Male", "Male", "Female", ~
## $ birthyear         <int> 1987, 1998, 1991, 1990, 1987, 1994, 1991, 1995, 1993~
```

```
# to see the unique values
unique(bike_original_4$usertype)
```

```
## [1] "Subscriber" "Customer"
```

```
# how many Customers and Subscribers
table(bike_original_4$usertype)
```

```
##
##   Customer Subscriber
##     106194     597860
```

```
# quick analysis
skim_without_charts(bike_original_4)
```

Table 1: Data summary

Name	bike_original_4
Number of rows	704054
Number of columns	12
Column type frequency:	
character	7
numeric	5

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
start_time	0	1	19	19	0	633380	0
end_time	0	1	19	19	0	632834	0
tripduration	0	1	4	11	0	10401	0
from_station_name	0	1	10	43	0	610	0
to_station_name	0	1	10	43	0	608	0
usertype	0	1	8	10	0	2	0
gender	0	1	0	6	66591	3	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
trip_id	0	1.00	25592222.05	213569.66	25223640	25407380	25590864	25777172	25962904
bikeid	0	1.00	3396.36	1913.95	1	1724	3473	5065	6946
from_station_id	0	1.00	203.96	157.78	2	77	174	291	673
to_station_id	0	1.00	203.95	157.94	2	77	174	291	673
birthyear	61681	0.91	1983.81	11.10	1899	1978	1987	1992	2003

```
# check for NA values
colSums(is.na(bike_original_4))
```

```
##          trip_id        start_time          end_time            bikeid
##                0                 0                 0                 0
##      tripduration   from_station_id from_station_name     to_station_id
##                0                 0                 0                 0
##   to_station_name          usertype            gender         birthyear
##                0                 0                 0             61681
```

```
# check for NA or null
colSums(is.na(bike_original_4) | bike_original_4 == "")
```

```
##          trip_id        start_time          end_time            bikeid
##                0                 0                 0                 0
##      tripduration   from_station_id from_station_name     to_station_id
##                0                 0                 0                 0
##   to_station_name          usertype            gender         birthyear
##                0                 0             66591             61681
```

################################–MANIPULATE–#######################
##

```
# convert "start_time" into a POSIXct class object
bike_original_4$start_time <- as.POSIXct(bike_original_4$start_time, format="%Y-%m-%d %H:%M:%S")
```

```
# now we can create 3 new variables
bike_original_4$month <- month(bike_original_4$start_time, label = TRUE, abbr = FALSE)
bike_original_4$day <- weekdays(bike_original_4$start_time)
```

```r
# create the variable "hour" and convert into a POSIXct class object in order to modify it later
bike_original_4$hour <- as.POSIXct(bike_original_4$start_time, format="%I:%M %p", tz = "UTC")
```

```r
# round up hours to see only 24 unique values inside variable (format is still 2019-10-01 00:05:00)
bike_original_4$hour_rounded <- floor_date(bike_original_4$hour, unit = "hour")
```

```r
# change format to 12:00 AM, ecc...
bike_original_4$hour_rounded <- strftime(bike_original_4$hour_rounded, format="%I:%M %p", tz = "UTC")
```

```r
# remove commas and decimals from variable "tripduration"
bike_original_4$tripduration <- gsub(",", "", bike_original_4$tripduration)
bike_original_4$tripduration <- as.numeric(gsub("\\..*", "", bike_original_4$tripduration))
```

```r
# create a new variable "trip_d_minutes" expressed in minutes as integer value
bike_original_4 <- bike_original_4 %>%
  mutate(trip_d_minutes = tripduration %/% 60)
```

```r
# create a smaller dataframe by filtering only necessaries values for the analysis
tab_4 <- bike_original_4[c("usertype", "gender", "birthyear", "month", "day", "hour_rounded", "trip_d_mi
```

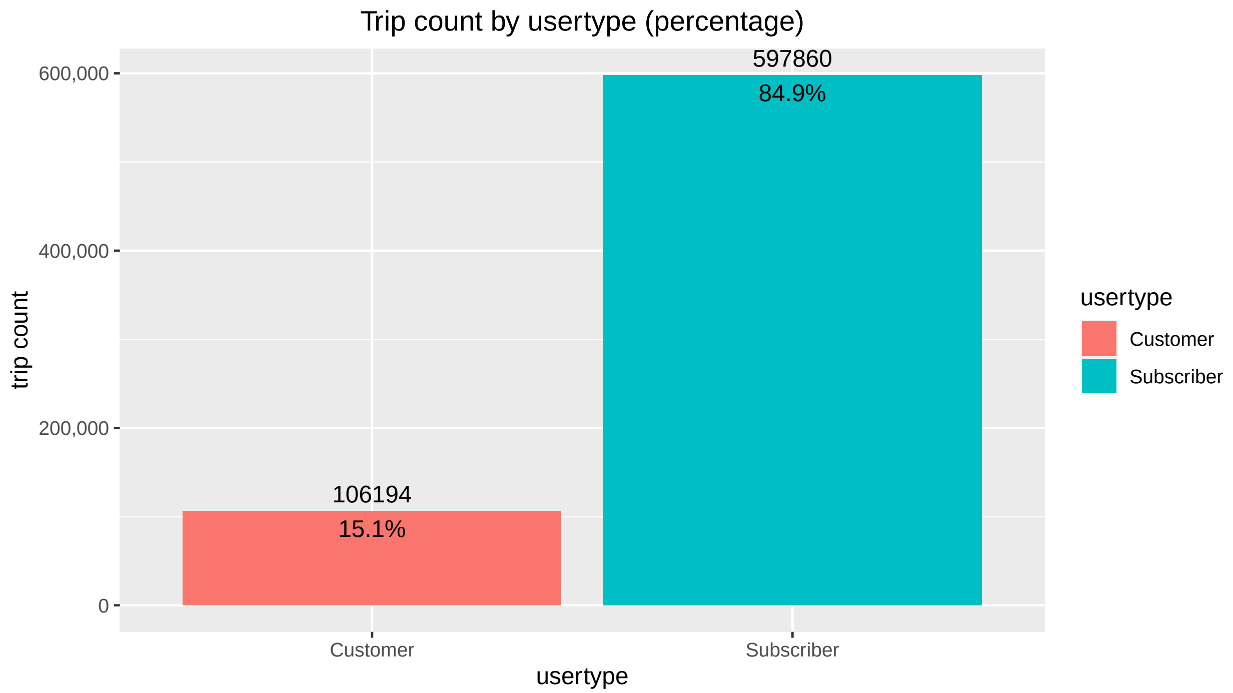
###################################–CLEAN–##########################
##

```r
# replace null values with "NA"
tab_4$gender[tab_4$gender == ""] <- "NA"
```

###############################-VISUALIZE & ANALIZE–#####################
##

```r
# 1) TRIP COUNT: add percentage values on top of the bar
ggplot(data=tab_4, aes(x=usertype, fill=usertype)) +
  geom_bar() +
  geom_text(stat='count', aes(label=..count..), vjust=-0.5) +
  geom_text(stat='count', aes(label=sprintf("%.1f%%", (..count..)/sum(..count..)*100)), vjust=1.5) +
  ggtitle("Trip count by usertype (percentage)") +
  ylab("trip count") +
  scale_y_continuous(labels = scales::comma_format()) +
  theme(plot.title = element_text(hjust = 0.5))
```

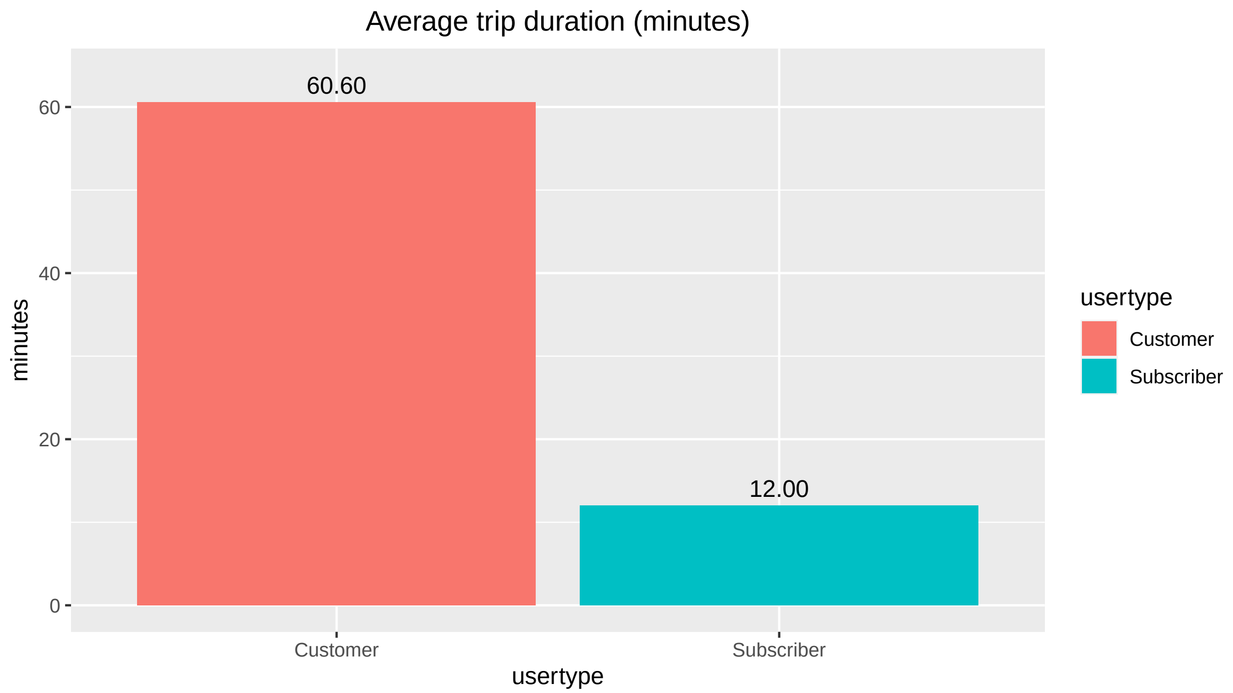
```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



0.3.1 Graph number 1 highlights that Subscribers takes the majority of trips with 85% of the total. Customer: 106194 Subscriber: 597860

0.3.2 We cannot establish the real number of customers because we do not have a unique ID that identifies them. We can only know how many trips Customers make compared to Subscribers but we cannot know the precise number of individual customers. For this reason there may be customers who travel once a month, others who travel every day, and others who travel several times a day.

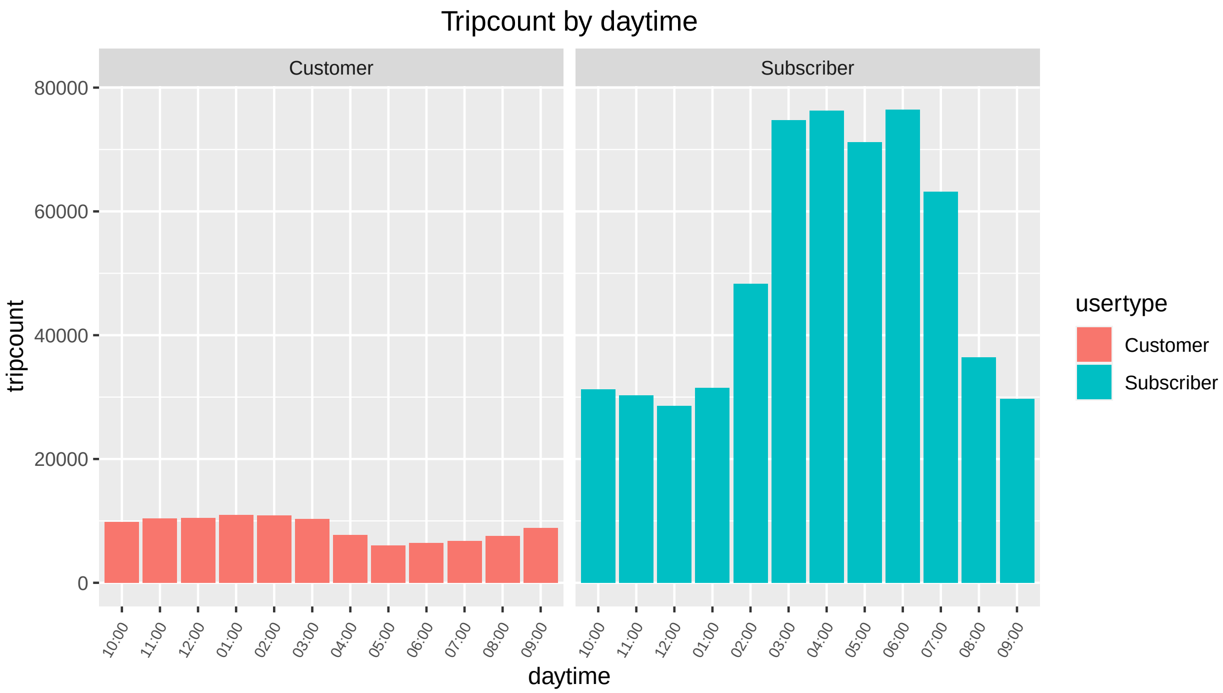
```r
# 2) TRIP AVERAGE BY USERTYPE: calculate the average trip duration (minutes) by usertype
ggplot(data=tab_4, aes(x=usertype, y=trip_d_minutes, fill=usertype)) +
  geom_bar(stat="summary", fun = "mean", position = "dodge") +
  geom_text(stat="summary", aes(label=sprintf("%.2f",..y..)), position=position_dodge(width=0.9), vjust=
  ggtitle("Average trip duration (minutes)") +
  ylab("minutes") +
  theme(plot.title = element_text(hjust = 0.5))
```



0.3.3 Graph number 2 analyzes the average of trips. Occasional customers use the bike share service for longer than Subscribers. The average duration of a trip for Subscribers is 12 minutes, while it reaches 60 minutes for occasional customers.

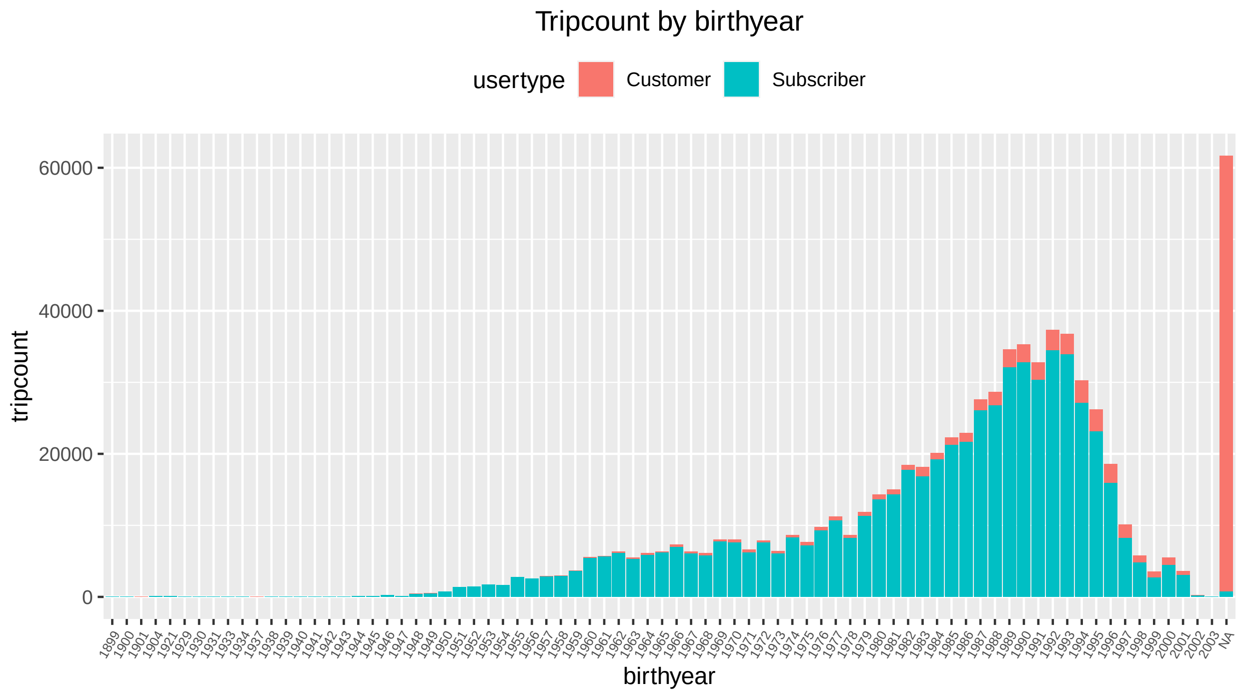
```r
# create a variable 'unique_levels' which contains unique leves from variable 'hour_rounded'. This is n
unique_levels <- unique(tab_4$hour_rounded)
tab_4$hour_rounded <- factor(tab_4$hour_rounded, levels = unique_levels)

# 3) TRIPCOUNT BY DAYTIME: create a visualization depicting the relationship between trip duration and
ggplot(data = tab_4, aes(x = factor(hour_rounded, levels = unique_levels), fill = usertype)) +
  geom_bar(stat = "count") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 7)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("tripcount") +
  xlab("daytime") +
  labs(title = "Tripcount by daytime") +
  facet_grid(~usertype)
```



0.3.4 Graph number 3 highlights how Customers and Subscribers use the Bike Share service differently during the hours of the day. The graph takes into consideration the number of daily trips, divided by usertype. We can see that Subscribers concentrate their trips from 7:00 AM to 8:00 AM, and from 4:00 PM to 6:00 PM. These times correspond to the times you start work in the morning and return home in the afternoon. Subscribers probably use the Bike Share service more to go to work. Customers, on the other hand, tend not to follow the same trend. Customers increase their usage gradually starting from the morning hours until reaching the peak at 3:00 PM, and then gradually decreasing until the evening hours. This suggests a use that does not correspond to working hours.

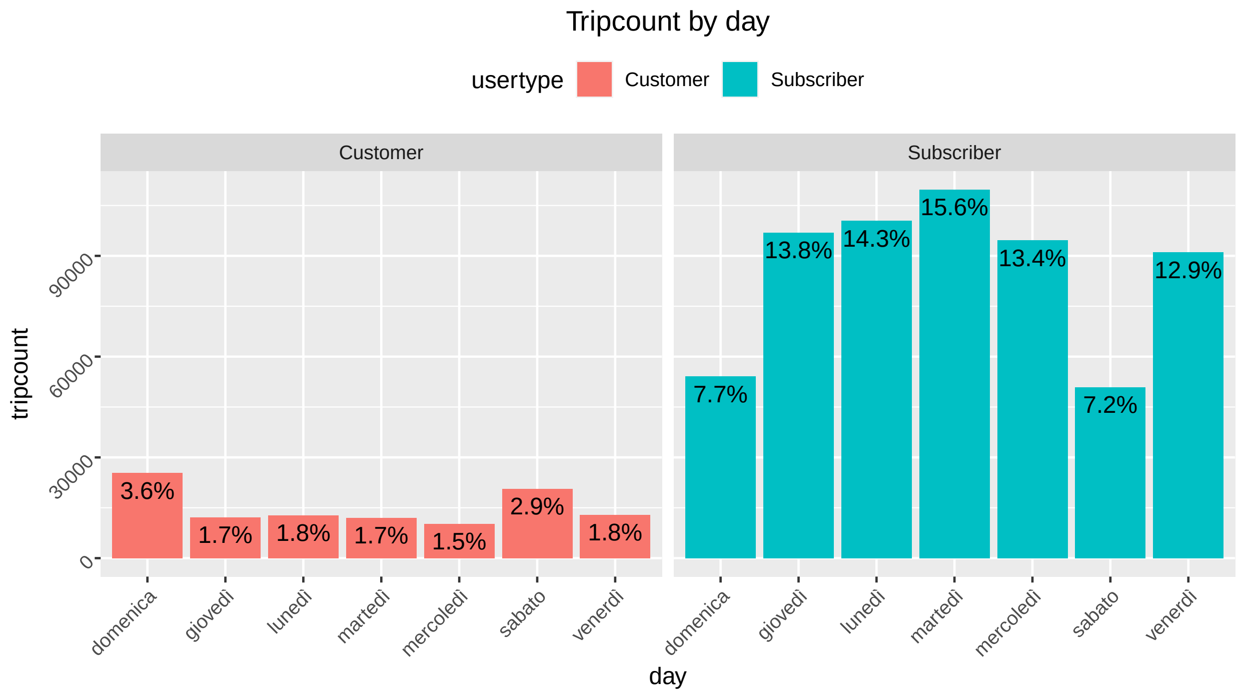
```
# 4) TRIPCOUNT BY BIRTHYEAR: this highlights the relationship between birthyear and tripduration groupe
tab_4$birthyear_factor <- factor(tab_4$birthyear, levels = c(levels(factor(tab_4$birthyear)), "NA"))
ggplot(data = tab_4, aes(x = birthyear_factor, fill = usertype)) +
  geom_bar(stat = "count") +
  theme(legend.position = "top") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 6)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("tripcount") +
  xlab("birthyear") +
  labs(title = "Tripcount by birthyear")
```



0.3.5 Graph number 4 relates the number of trips to the year of birth of the customers, and then divides them by color based on the usertype. The data highlights how there is a greater concentration of use for customers born approximately between the years 80' and 96'. Another data to take into consideration is the "NA" data. Within the dataset there are many missing values in the "gender" vector and in the "birthyear" vector. The color of the usertypes shows us a rather high lack of information regarding the year of birth of the Customers, while it is more negligible for the Subscribers. This may be due to the type of Customer registration. Maybe Customers don't fill out the form with their data correctly.

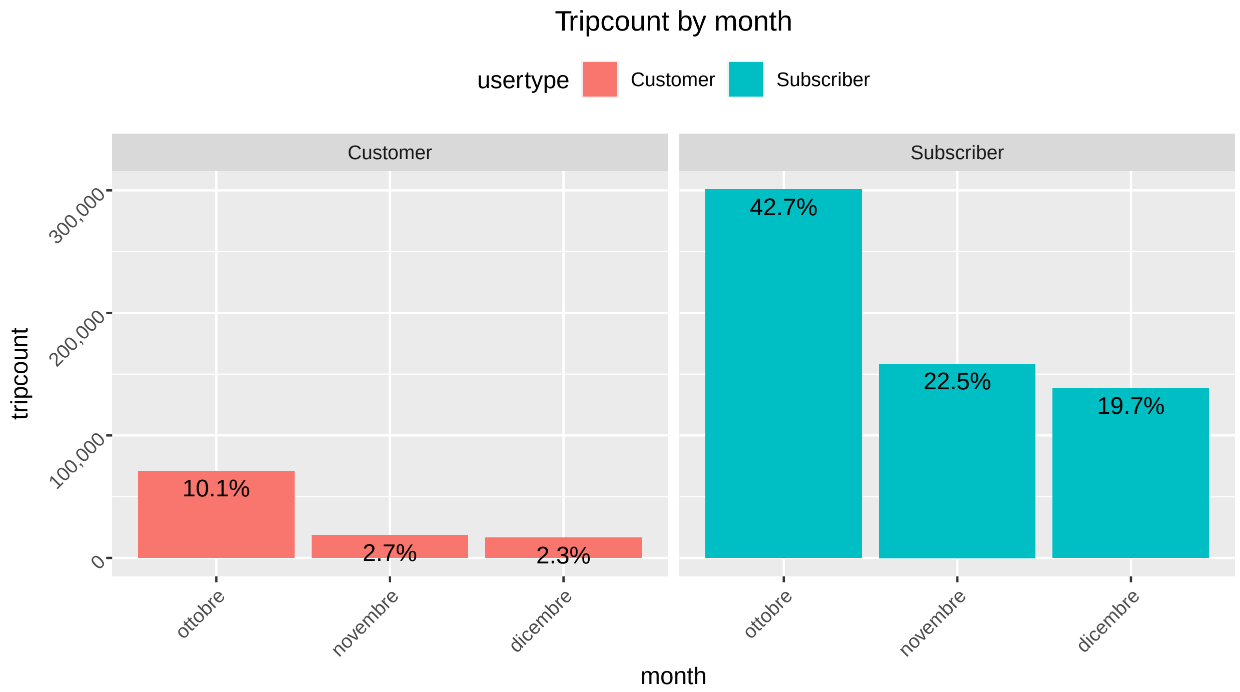
```r
# 5) TRIPCOUNT BY DAY: tripduration percentage by weekday
ggplot(data=tab_4, aes(x = fct_relevel(factor(day), "Monday","Tuesday","Wednesday","Thursday","Friday",
  geom_bar(stat = "count") +
  geom_text(stat='count', aes(label=sprintf("%.1f%%", after_stat(count)/sum(..count..)*100)), vjust=1.5
  theme(legend.position = "top") +
  theme(axis.text = element_text(angle = 45, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("tripcount") +
  xlab("day") +
  labs(title = "Tripcount by day") +
  facet_wrap(~usertype)
```

```
## Warning: 7 unknown levels in `f`: Monday, Tuesday, Wednesday, Thursday, Friday,
## Saturday, and Sunday
## 7 unknown levels in `f`: Monday, Tuesday, Wednesday, Thursday, Friday,
## Saturday, and Sunday
## 7 unknown levels in `f`: Monday, Tuesday, Wednesday, Thursday, Friday,
## Saturday, and Sunday
```



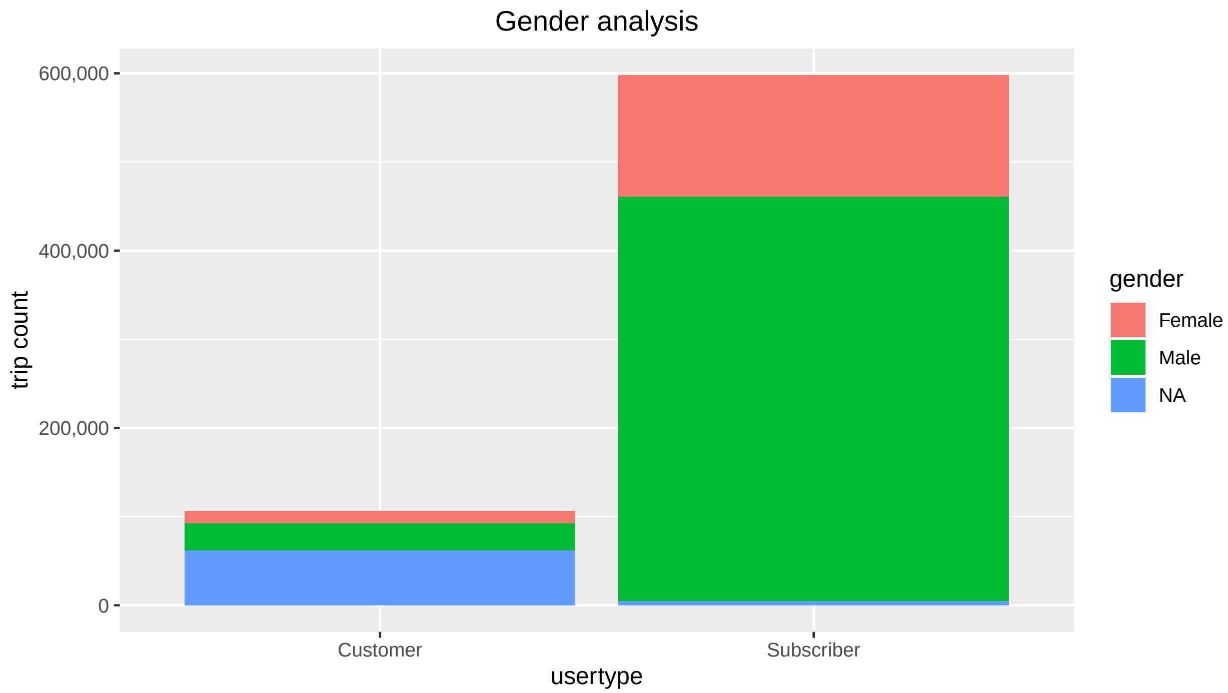
0.3.6 Graph number 5 shows us how Subscribers tend to use the Bike Share service during working days, from Monday to Friday. Customers, on the other hand, use the service more during the weekend.

```
# 6) TRIPCOUNT BY MONTH
ggplot(data=tab_4, aes(x = month, fill = usertype)) +
  geom_bar(stat = "count") +
  geom_text(stat='count', aes(label=sprintf("%.1f%%", after_stat(count)/sum(..count..)*100)), vjust=1.5
  theme(legend.position = "top") +
  theme(axis.text = element_text(angle = 45, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("tripcount") +
  xlab("month") +
  scale_y_continuous(labels = scales::comma_format()) +
  labs(title = "Tripcount by month") +
  facet_wrap(~usertype)
```



0.3.7 Graph number 6 analyzes the usage trend over the months. This dataset contains data only on October, November, December. For both Customers and Subscribers, October is the month of greatest use.

```
# 7) GENDER ANALYSIS
ggplot(data = tab_4, aes(x = usertype, fill = gender)) +
  geom_bar(stat = "count") +
  ggtitle("Gender analysis") +
  ylab("trip count") +
  scale_y_continuous(labels = scales::comma_format()) +
  theme(plot.title = element_text(hjust = 0.5))
```



0.3.8 Graph number 7 analyzes the gender of customers. For both Customers and Subscribers, customers are mostly male. Also in this case there are missing data, especially for occasional customers who have the highest number of "NA".

0.3.9 Recommendations:

0.3.10 1) It is recommended to review your customer registration process to avoid NA values, especially one-time customers. The missing values of gender and birthyear are quite high and prevent in-depth analysis.

0.3.11 2) To convert occasional customers into subscribers, the average number of trips must also be considered. An occasional customer travels for 60 minutes on average, which is approximately 5 times more than subscribers who travel on average only 12 minutes. For an occasional customer it might be more convenient to sign up for a subscription rather than renting the bike for such long trips.

0.3.12 3) It would be useful to record a user ID within the dataset for subsequent customers. With a unique user ID it would be possible to do in-depth analysis. For example: how many trips does each occasional customer make compared to a season ticket holder? A unique ID would allow relationships to be found based on daily and monthly usage. With current data, however, there may be customers who only make one trip a month, while others may travel every day. This question cannot be answered because we only know whether the customers are Customers or Subscribers.