Bike - share Case Study

2025-12-03

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Here I load in data

```
bike_share2019 <- read.csv("bike_share2019.csv")
bike_share2020 <- read.csv("bike_share2020.csv")
```

```
library(lubridate)
```

```
##
## Attaching package: ’lubridate’

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

```
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
```

```
library(ggplot2)
```

Here I change start and end times to dates, calculate ride length and add a check to see if the bike is dropped off at the same location that it was picked up from.

```
bike_share2019 <- bike_share2019 %>%
  mutate(end_time=ymd_hms(end_time), start_time=ymd_hms(start_time))  %>%
  mutate(ride_length = as.numeric(end_time - start_time, units = "mins")) %>%
  mutate(loop = from_station_id == to_station_id) %>%
  mutate(day_of_week = weekdays(start_time))
```

this is what the data looks like after the initial edits

```
head(bike_share2019)
```

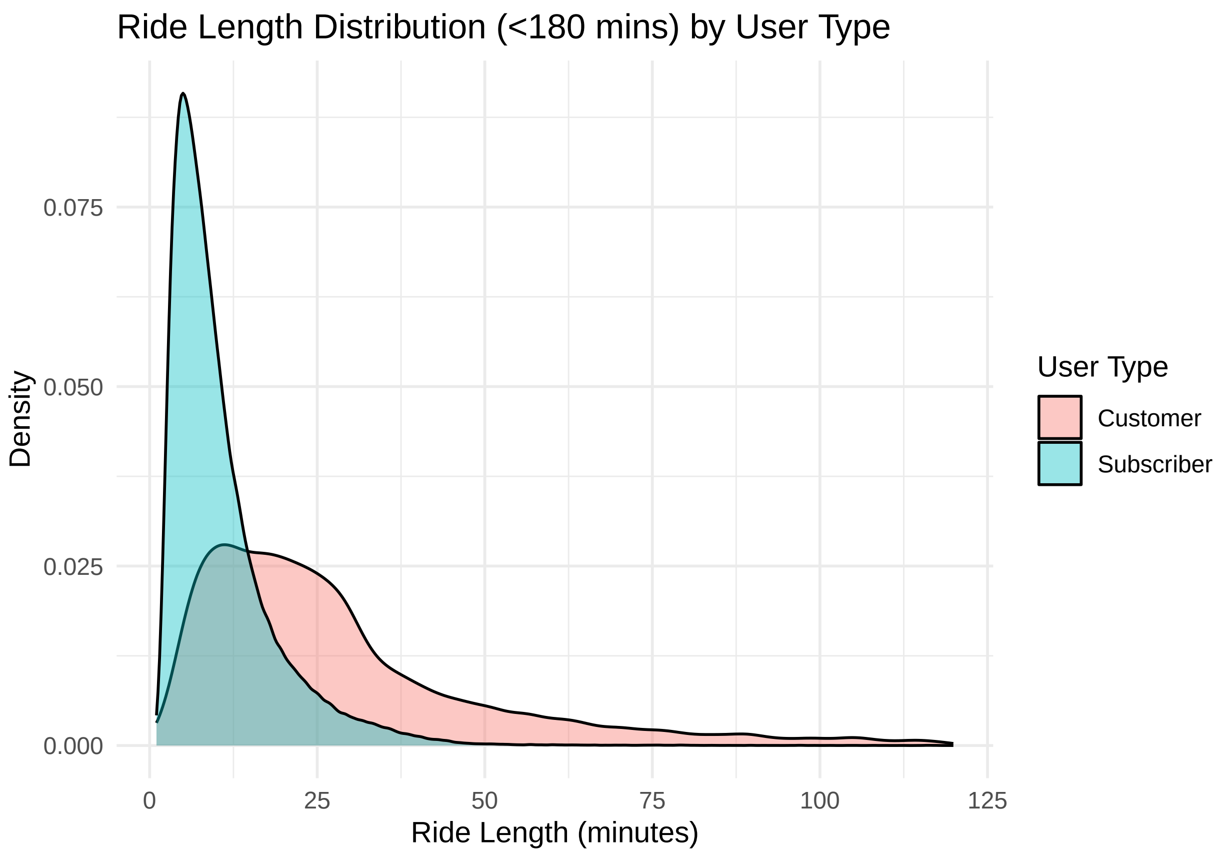
```
##    trip_id          start_time            end_time bikeid tripduration
## 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07   2167          390
## 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34   4386          441
## 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12   1524          829
## 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28    252     1,783.00
## 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56   1170          364
## 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09   2437          216
##   from_station_id                   from_station_name to_station_id
## 1             199              Wabash Ave & Grand Ave            84
## 2              44              State St & Randolph St           624
## 3              15                Racine Ave & 18th St           644
## 4             123      California Ave & Milwaukee Ave           176
## 5             173 Mies van der Rohe Way & Chicago Ave            35
## 6              98          LaSalle St & Washington St            49
##                  to_station_name   usertype gender birthyear ride_length  loop
## 1      Milwaukee Ave & Grand Ave Subscriber   Male      1989    6.500000 FALSE
## 2 Dearborn St & Van Buren St (*) Subscriber Female      1990    7.350000 FALSE
## 3  Western Ave & Fillmore St (*) Subscriber Female      1994   13.816667 FALSE
## 4              Clark St & Elm St Subscriber   Male      1993   29.716667 FALSE
## 5        Streeter Dr & Grand Ave Subscriber   Male      1994    6.066667 FALSE
## 6        Dearborn St & Monroe St Subscriber Female      1983    3.600000 FALSE
##   day_of_week
## 1     Tuesday
## 2     Tuesday
## 3     Tuesday
## 4     Tuesday
## 5     Tuesday
## 6     Tuesday
```

I calculated the percentages of rides that are loops by user type

```
sub_loop <- (sum(bike_share2019$usertype == "Subscriber" & bike_share2019$loop == TRUE))

cus_loop <- (sum(bike_share2019$usertype == "Customer" & bike_share2019$loop == TRUE))

P_sub_loop <- sub_loop/(sum(bike_share2019$usertype == "Subscriber")) *100

p_cus_loop <- cus_loop/(sum(bike_share2019$usertype == "Customer")) *100
```

Here i calculated the ride length by user type. I cut the data from rides over 3 hours as it seemed people had forgotten to end rides with the max length being over 100 days

```
ggplot(bike_share2019 %>% filter(ride_length < 120),
       aes(x = ride_length, fill = usertype)) +
  geom_density(linewidth = .5, alpha =.4) +
  labs(
    title = "Ride Length Distribution (<180 mins) by User Type",
    x = "Ride Length (minutes)",
    y = "Density",
    fill = "User Type"
  ) +
  theme_minimal()
```



I found the day of the week that the ride took place on, as well as ordered the days of the week in the proper order

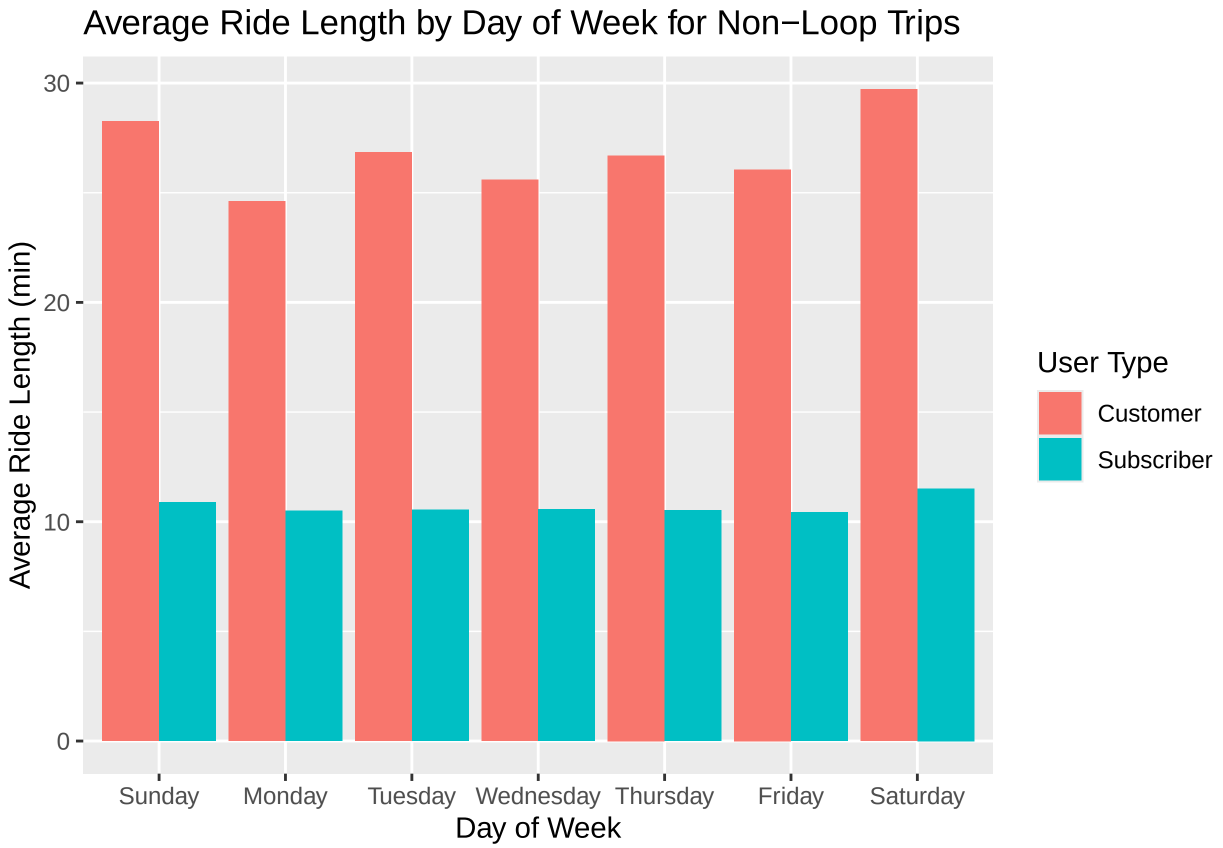
```
bike_share2019$day_of_week <- factor(
  bike_share2019$day_of_week,
  levels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
)
```

Here we see the ride length by day of the week for subscribers and non-subscribers. Iam looking for data on commuting so the loop trips are removed

```
avg_by_day <- bike_share2019 %>%
  filter(ride_length < 120,loop == 0)%>%
  group_by(day_of_week, usertype) %>%
  summarise(avg_ride_length = mean(ride_length))
```

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

```
ggplot(avg_by_day, aes(x = day_of_week, y = avg_ride_length, fill = usertype)) +
  geom_col(position = "dodge") +
  labs(
    title = "Average Ride Length by Day of Week for Non-Loop Trips",
    x = "Day of Week",
    y = "Average Ride Length (min)",
    fill = "User Type"
  )
```



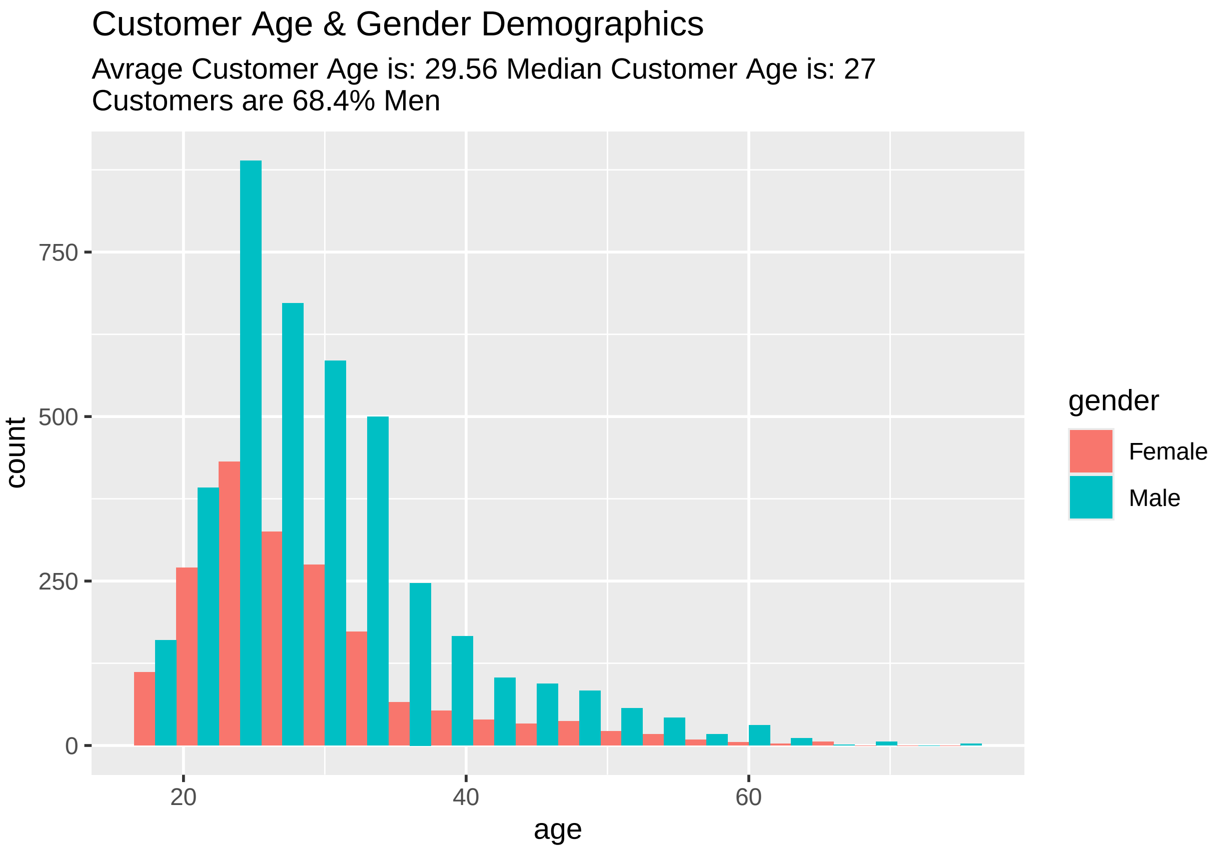
Here i add and calculate additional customer demographics

```
cust_demographics <- bike_share2019 %>%
  filter(usertype == "Customer", birthyear >= 1930, gender != "") %>%
  reframe(age = 2019 - birthyear, gender)

cust_mean_age <- round(mean(cust_demographics$age),2)
cust_med_age <- median(cust_demographics$age)
cust_pct_male <- round(mean(cust_demographics$gender == "Male"),4) * 100


sub_demographics <- bike_share2019 %>%
  filter(usertype == "Subscriber", birthyear >= 1930, gender != "") %>%
  reframe(age = 2019 - birthyear, gender)

sub_mean_age <- round(mean(sub_demographics$age),2)
```

```
sub_med_age <- median(sub_demographics$age)
sub_pct_male <- round(mean(sub_demographics$gender == "Male"),4) * 100
```

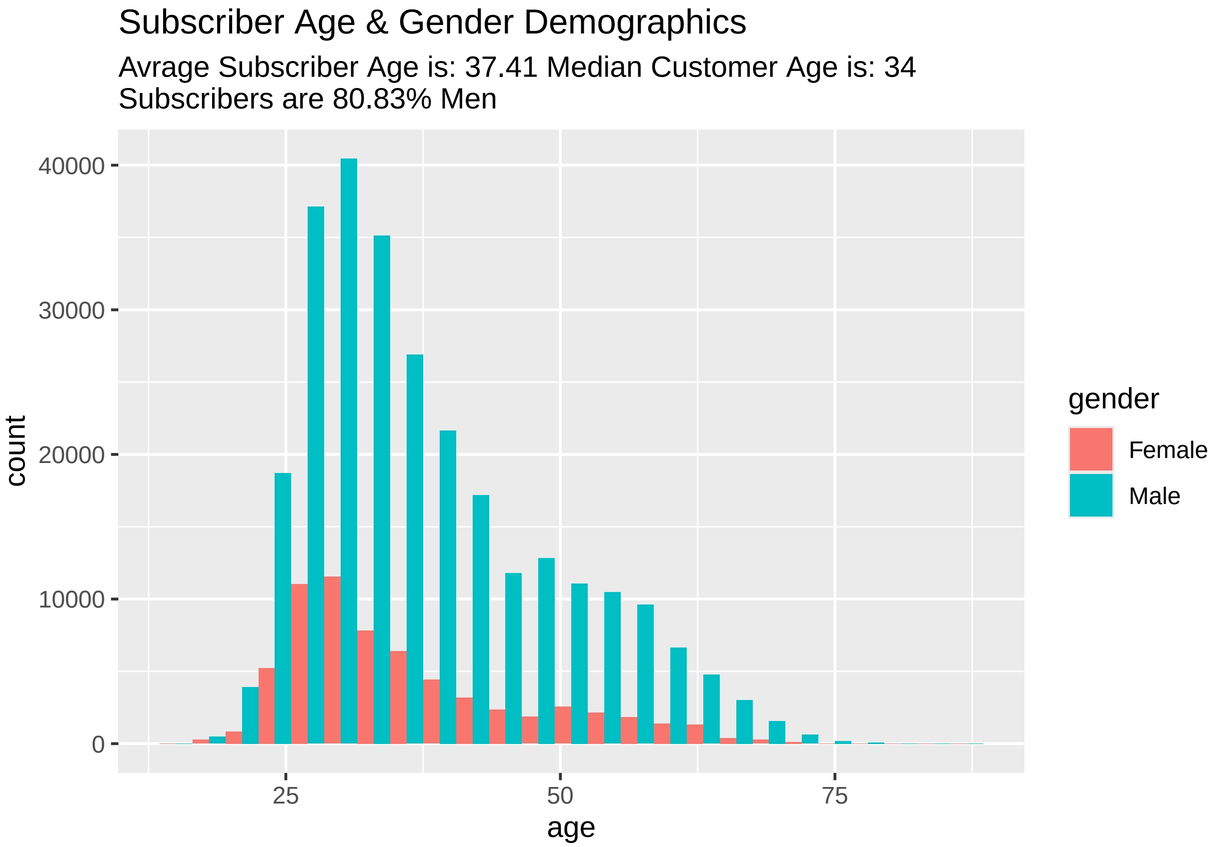
This uses the previous calculations as well as a graph to see better demographics

```
ggplot(data = cust_demographics, aes(age, fill = gender)) +
  geom_histogram(binwidth = 3,position = "dodge") +
  labs(title = "Customer Age & Gender Demographics", subtitle = paste0("Avrage Customer Age is: ", cust_
Customers are ",cust_pct_male,"% Men"))
```



Here is the same graphic for subscribers to the service

```
ggplot(data = sub_demographics, aes(age, fill = gender)) +
  geom_histogram(binwidth = 3,position = "dodge") +
  labs(title = "Subscriber Age & Gender Demographics", subtitle = paste0("Avrage Subscriber Age is: ", s
Subscribers are ",sub_pct_male,"% Men"))
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.