Cleaning Process and Plots

Nick Fitzgerald

9/17/2021

## Setup Environment and Data

You can download the raw data here: [Divvy Data](https://divvy-tripdata.s3.amazonaws.com/index.html). The ones I used were Divvy\_Trips\_2019\_Q2 thru Q4, and Divvy\_Trips\_2020\_Q1, since it’s the most recent data of the last four quarters.

After importing and inspecting the data, we can see that since that data is from separate years, the column names and data types are different. In 2020, a few columns are also omitted from being tracked.

There were quite a lot of inspection functions used during the process. For simplicity, I omitted them and focused on the portions used during the cleaning, organizing, and plotting process.

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.1

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.1.1

##   
## Attaching package: 'lubridate'

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

library(ggplot2)  
library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

q2\_2019 <- read.csv("/Users/Nick/Desktop/Divvy\_2019/Data/Divvy\_Trips\_2019\_Q2.csv")  
q3\_2019 <- read.csv("/Users/Nick/Desktop/Divvy\_2019/Data/Divvy\_Trips\_2019\_Q3.csv")  
q4\_2019 <- read.csv("/Users/Nick/Desktop/Divvy\_2019/Data/Divvy\_Trips\_2019\_Q4.csv")  
q1\_2020 <- read.csv("/Users/Nick/Desktop/Divvy\_2019/Data/Divvy\_Trips\_2020\_Q1.csv")  
  
*#renaming columns so they are all uniform.*  
q4\_2019 <- rename(q4\_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)  
  
q3\_2019 <- rename(q3\_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)  
  
q2\_2019 <- rename(q2\_2019,  
 ride\_id = X01...Rental.Details.Rental.ID,  
 rideable\_type = X01...Rental.Details.Bike.ID,  
 started\_at = X01...Rental.Details.Local.Start.Time,  
 ended\_at = X01...Rental.Details.Local.End.Time,  
 start\_station\_name = X03...Rental.Start.Station.Name,  
 start\_station\_id = X03...Rental.Start.Station.ID,  
 end\_station\_name = X02...Rental.End.Station.Name,  
 end\_station\_id = X02...Rental.End.Station.ID,  
 member\_casual = User.Type)  
  
*#converting value types to match Q4\_2020 so when combined, the columns stack correctly.*  
q4\_2019 <- mutate(q4\_2019, ride\_id = as.character(ride\_id),  
 rideable\_type = as.character(rideable\_type))  
  
q3\_2019 <- mutate(q3\_2019, ride\_id = as.character(ride\_id),  
 rideable\_type = as.character(rideable\_type))  
  
q2\_2019 <- mutate(q2\_2019, ride\_id = as.character(ride\_id),  
 rideable\_type = as.character(rideable\_type))  
  
*#Merges the 4 data frames into 1.*  
all\_trips <- bind\_rows(q2\_2019, q3\_2019, q4\_2019, q1\_2020)  
  
*#Removing columns that didn't carry over from 2019 to 2020 and other data not in main objective.*  
all\_trips <- all\_trips %>%   
 select(-c(start\_lat, start\_lng, end\_lat, end\_lng, birthyear, gender,  
 X05...Member.Details.Member.Birthday.Year, Member.Gender,  
 X01...Rental.Details.Duration.In.Seconds.Uncapped, tripduration))  
  
*#For the member\_casual column, there are four classifications, when there should only be two. This renames Subscriber and Customer to member and casual to fix it.*  
all\_trips <- all\_trips %>%   
 mutate(member\_casual = recode(member\_casual,  
 "Subscriber" = "member",  
 "Customer" = "casual"))  
  
*#adding columns that list date/month/year/day of week*  
all\_trips$date <- as.Date(all\_trips$started\_at)  
all\_trips$month <- format(as.Date(all\_trips$date), "%m")  
all\_trips$day <- format(as.Date(all\_trips$date), "%d")  
all\_trips$year <- format(as.Date(all\_trips$date), "%Y")  
all\_trips$day\_of\_week <- format(as.Date(all\_trips$date), "%A")  
  
*#adding ride length column.*  
all\_trips$ride\_length <- difftime(all\_trips$ended\_at, all\_trips$started\_at)  
  
*#convert ride\_length from factor to numeric.*  
all\_trips$ride\_length <- as.numeric(as.character(all\_trips$ride\_length))  
  
*#creating new data frame since we are dropping rows: Bikes taken out of service during the year and the HQ QR stations.*  
all\_trips\_v2 <- all\_trips[!(all\_trips$start\_station\_name == "HQ QR" | all\_trips$ride\_length < 0),]  
  
*#Renaming numeric months to string names for easier visual labeling in Tableau*  
all\_trips\_v2 <- all\_trips\_v2 %>%   
 mutate(month = recode(month,  
 "01" = "January",  
 "02" = "February",  
 "03" = "March",  
 "04" = "April",  
 "05" = "May",  
 "06" = "June",  
 "07" = "July",  
 "08" = "August",  
 "09" = "September",  
 "10" = "October",  
 "11" = "November",  
 "12" = "December"))

## Analysis

Taking a quick look at casual vs members. Also wanted to take a look at the most popular stations.

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$member\_casual, FUN = mean)

## all\_trips\_v2$member\_casual all\_trips\_v2$ride\_length  
## 1 casual 3552.7941  
## 2 member 850.0783

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$member\_casual, FUN = median)

## all\_trips\_v2$member\_casual all\_trips\_v2$ride\_length  
## 1 casual 1546  
## 2 member 589

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$member\_casual, FUN = max)

## all\_trips\_v2$member\_casual all\_trips\_v2$ride\_length  
## 1 casual 9383424  
## 2 member 9056634

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$member\_casual, FUN = min)

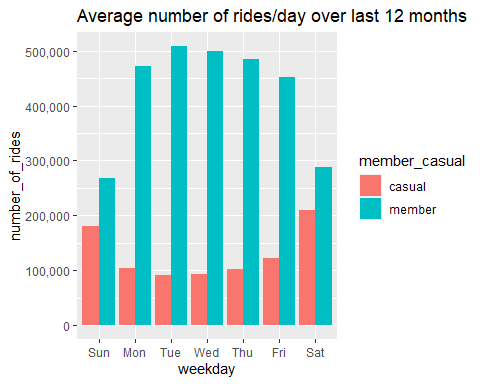
## all\_trips\_v2$member\_casual all\_trips\_v2$ride\_length  
## 1 casual 2  
## 2 member 1

*#Creating the count for total trips at each station, for ending, and starting stations.*  
count\_station\_id <- table(all\_trips\_v2$start\_station\_id)  
count\_stop\_id<- table(all\_trips\_v2$end\_station\_id)  
  
*#exporting .csv to re-upload as data frames*   
write.csv(count\_station\_id, "C:/Users/Nick/Desktop/Divvy\_2019/Saved/count\_station\_id.csv")  
write.csv(count\_stop\_id, "C:/Users/Nick/Desktop/Divvy\_2019/Saved/count\_stop\_id.csv")  
  
*#exported all\_stations and manipulated data further in excel, re-uploaded as station\_id\_totals*  
all\_stations <- bind\_rows(count\_station\_id, count\_stop\_id)  
  
station\_id\_totals <- read.csv("/Users/Nick/Desktop/Divvy\_2019/Saved/station\_id\_totals.csv")  
  
*#isolating top 10 busiest stations for graphing*  
df <- station\_id\_totals %>%   
 arrange(desc(total\_combined)) %>%   
 slice(1:10)   
  
*#attempted to make a few charts with that data, but it was coming out too difficult to understand/read because of the continuous scaling for the station id's. Transformed station id's to characters to fix that.*  
df\_v2 <- transform(df, station\_id = as.character(station\_id)) %>%   
 arrange(desc(total\_combined))

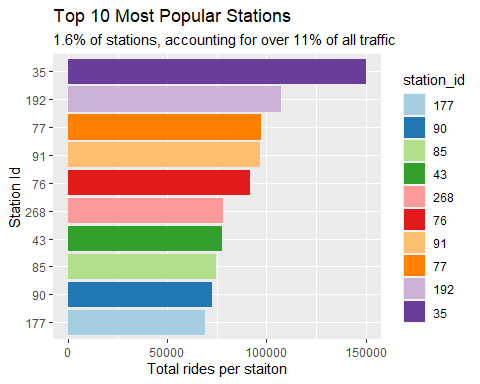
## Plots

*#A look at Casual vs Members vs Time spent on bikes.*  
all\_trips\_v2 %>%   
 mutate(weekday = wday(started\_at, 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') +  
 scale\_y\_continuous(labels = comma) +  
 labs(title = "Average number of rides/day over last 12 months")

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



*#Top 10 busiest stations*  
df\_v2 %>%   
 mutate(station\_id = fct\_reorder(station\_id, total\_combined)) %>%   
 ggplot(aes(x = station\_id, y = total\_combined, fill = station\_id)) +  
 geom\_col() +  
 coord\_flip() +  
 scale\_fill\_brewer(type = "seq", palette = "Paired", direction = 1) +  
 labs( y = "Total rides per station",  
 x = "Station Id",  
 title = "Top 10 Most Popular Stations",  
 subtitle = "1.6% of stations, accounting for over 11% of all traffic")



## Insights

Based on the graphs we can see a clear difference between Casual riders and Members:

* Members have a much higher frequency of rides on any given day, but especially on weekdays.
  + This supports a hypothesis that members use the service for work related purposes.
* Out of over 600 stations, these 10 account for over 11% of all bike rentals.
* Average ride length for a casual rider is nearly an hour, while a member’s average is around 15 minutes.