Stat650 Midterm: Lyft Baywheels Data

Kelly Radimer, Section 2

Download, Unzip and Read the Data into R

```
library(pacman)
p_load(tidyverse,tictoc,ggmap,skimr,lubridate,forcats, Amelia)
```

First, we will download the files into the data directory by looping over the one value in the url and filename that changes.

```
for (i in 5:9) {
URL <- paste0("https://s3.amazonaws.com/baywheels-data/20190",i,"-baywheels-tripdata.csv.zip")
download.file(URL, destfile = paste0("C:/Users/snapd/Documents/R/STAT 650/Midterm/data/20190",i,"-baywhe
}

for (i in 0:2) {
URL <- paste0("https://s3.amazonaws.com/baywheels-data/20191",i,"-baywheels-tripdata.csv.zip")
download.file(URL, destfile = paste0("C:/Users/snapd/Documents/R/STAT 650/Midterm/data/20191",i,"-baywhe
}

for (i in 1:8) {
URL <- paste0("https://s3.amazonaws.com/baywheels-data/20200",i,"-baywheels-tripdata.csv.zip")
download.file(URL, destfile = paste0("C:/Users/snapd/Documents/R/STAT 650/Midterm/data/20200",i,"-baywhe
}
```

Next we will unzip downloaded files, again with a for loop.

```
for (i in 5:9) {
fn<-paste0("./data/20190",i,"-baywheels-data.csv.zip")
 unzip(fn, exdir = "./data")
}

for (i in 0:2) {
fn<-paste0("./data/20191",i,"-baywheels-data.csv.zip")
 unzip(fn, exdir = "./data")
}

for (i in 1:8) {
fn<-paste0("./data/20200",i,"-baywheels-data.csv.zip")
 unzip(fn, exdir = "./data")
}
```

Next, we clean up data directory.

```
for (i in 5:9) {
fn<-paste0("./data/20190",i,"-baywheels-data.csv.zip")
 if (file.exists(fn)) file.remove(fn)
}

for (i in 0:2) {
fn<-paste0("./data/20191",i,"-baywheels-data.csv.zip")
 if (file.exists(fn)) file.remove(fn)
}

for (i in 1:8) {
fn<-paste0("./data/20200",i,"-baywheels-data.csv.zip")
 if (file.exists(fn)) file.remove(fn)
}
```

Read the .csv files into data frames.

```
for (i in 5:9) {
  fn <- paste0("./data/20190",i,"-baywheels-tripdata.csv")
  nam <- paste("baywheels20190", i, sep = "")
  assign(nam, read_csv(file = fn))
}

for (i in 0:2) {
  fn <- paste0("./data/20191",i,"-baywheels-tripdata.csv")
  nam <- paste("baywheels20191", i, sep = "")
  assign(nam, read_csv(file = fn))
}

for (i in 1:8) {
  fn <- paste0("./data/20200",i,"-baywheels-tripdata.csv")
  nam <- paste("baywheels20200", i, sep = "")
  assign(nam, read_csv(file = fn))
}
```

Check the head() and tail() of a couple of the data.frames to make sure they look as we expect. We'll check an older month and a newer month to see if the variables match up.

```
head(baywheels202008)
```

```
## # A tibble: 6 x 13
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 6549E1~ electric_bike 2020-08-14 09:41:03 2020-08-14 10:03:45 <NA>
## 2 B7F273~ electric_bike 2020-08-13 18:43:00 2020-08-13 18:52:52 <NA>
## 3 33B224~ electric_bike 2020-08-14 09:13:54 2020-08-14 09:20:29 23rd St at Tenn~
## 4 053D5F~ electric_bike 2020-08-14 11:26:54 2020-08-14 11:29:17 Broderick St at~
## 5 B3BDEC~ electric_bike 2020-08-14 08:37:37 2020-08-14 08:52:20 Broderick St at~
## 6 EB7FCC~ electric_bike 2020-08-14 11:03:06 2020-08-14 11:17:29 <NA>
## # ... with 8 more variables: start_station_id <dbl>, end_station_name <chr>,
## #   end_station_id <dbl>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>
```

```
tail(baywheels201905)
```

```
## # A tibble: 6 x 14
##   duration_sec start_time          end_time            start_station_id
##          <dbl> <dttm>              <dttm>                         <dbl>
## 1          404 2019-05-01 00:08:51 2019-05-01 00:15:35               20
## 2          193 2019-05-01 00:11:19 2019-05-01 00:14:32              121
## 3          145 2019-05-01 00:10:52 2019-05-01 00:13:17              253
## 4          173 2019-05-01 00:08:35 2019-05-01 00:11:28              120
## 5          305 2019-05-01 00:05:48 2019-05-01 00:10:53              243
## 6          121 2019-05-01 00:08:20 2019-05-01 00:10:21              180
## # ... with 10 more variables: start_station_name <chr>,
## #   start_station_latitude <dbl>, start_station_longitude <dbl>,
## #   end_station_id <dbl>, end_station_name <chr>, end_station_latitude <dbl>,
## #   end_station_longitude <dbl>, bike_id <dbl>, user_type <chr>,
## #   bike_share_for_all_trip <chr>
```

The variables do not match. Also, the number of variables in my data frames ranges from 13 to 15. Let's figure out why.

```
col2008 <- colnames(baywheels202008)
col1911 <- colnames(baywheels201911)
intersect(col2008, col1911)
```

```
## [1] "start_station_name" "start_station_id"   "end_station_name"
## [4] "end_station_id"
```

Only four variables match between these two data frames, so it looks like they've renamed several variables in addition to dropping and adding some variables.

Old variable names:

```
col1911
```

```
##  [1] "duration_sec"            "start_time"
##  [3] "end_time"                "start_station_id"
##  [5] "start_station_name"      "start_station_latitude"
##  [7] "start_station_longitude" "end_station_id"
##  [9] "end_station_name"        "end_station_latitude"
## [11] "end_station_longitude"   "bike_id"
## [13] "user_type"               "bike_share_for_all_trip"
## [15] "rental_access_method"
```

New variable names:

```
col2008
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

It looks like the new system no longer uses the variables duration_sec, bike_share_for_all_trip, rental_access_method and bike_id, so we should remove these variables.

Variables start_time and end_time should change to started_at and ended_at, start_station_latitude should become start_lat (likewise for end and longitude).

The number of variables fluctuates from month to month, so let's have a look at some more variable names in order to find the cause for these fluctuations and determine when they made the big switch.

```
col1905<-colnames(baywheels201905)
col1906<-colnames(baywheels201906)
intersect(col1906,col1911)
```

```
##  [1] "duration_sec"            "start_time"
##  [3] "end_time"                "start_station_id"
##  [5] "start_station_name"      "start_station_latitude"
##  [7] "start_station_longitude" "end_station_id"
##  [9] "end_station_name"        "end_station_latitude"
## [11] "end_station_longitude"   "bike_id"
## [13] "user_type"               "bike_share_for_all_trip"
## [15] "rental_access_method"
```

June and November 2019 match in all variables.

```
intersect(col1905,col1906)
```

```
##  [1] "duration_sec"            "start_time"
##  [3] "end_time"                "start_station_id"
##  [5] "start_station_name"      "start_station_latitude"
##  [7] "start_station_longitude" "end_station_id"
##  [9] "end_station_name"        "end_station_latitude"
## [11] "end_station_longitude"   "bike_id"
## [13] "user_type"               "bike_share_for_all_trip"
```

May is just missing rental_access_method relative to June 2019.

```
col1910<-colnames(baywheels201910)
intersect(col1905,col1910)
```

```
##  [1] "duration_sec"            "start_time"
##  [3] "end_time"                "start_station_id"
##  [5] "start_station_name"      "start_station_latitude"
##  [7] "start_station_longitude" "end_station_id"
##  [9] "end_station_name"        "end_station_latitude"
## [11] "end_station_longitude"   "bike_id"
## [13] "user_type"               "bike_share_for_all_trip"
```

May and October 2019 match.

```
col2003 <- colnames(baywheels202003)
col2004 <- colnames(baywheels202004)
intersect(col2003,col2004)
```

```
## [1] "start_station_id"   "start_station_name" "end_station_id"
## [4] "end_station_name"
```

It looks like the change happened in April 2020, so let's bind together all the data frames before that change and then modify the variables.

```
old_var<-bind_rows(baywheels201905,baywheels201906,baywheels201907, baywheels201908,baywheels201909,bayw
```

```
old_var <- old_var %>% select(-duration_sec) %>%
  rename(started_at = start_time,
         ended_at = end_time,
         start_lat = start_station_latitude,
         start_lng = start_station_longitude,
         end_lat = end_station_latitude,
         end_lng = end_station_longitude
         )
```

The Baywheels website says User Type (Subscriber or Customer – "Subscriber" = Member or "Customer" = Casual), so we can rename these in the old data set, while getting rid of a few more variables that are no longer used.

```
old_var <- old_var %>%
  select(-bike_share_for_all_trip, -rental_access_method, -bike_id) %>%
  mutate(member_casual = ifelse(user_type=="Customer", "casual", "member"))
```

```
old_var <- old_var %>%
  select(-user_type)
```

Now we can merge the old and new data.

```
lyft <- bind_rows(old_var, baywheels202004, baywheels202005, baywheels202006, baywheels202007, baywheels
glimpse(lyft)
```

```
## Rows: 3,229,177
## Columns: 14
## $ started_at         <dttm> 2019-05-31 20:34:56, 2019-05-31 19:43:56, 2019-...
## $ ended_at           <dttm> 2019-06-01 10:09:34, 2019-06-01 08:48:06, 2019-...
## $ start_station_id   <dbl> 321, 246, 149, 186, 34, 50, 50, 324, 22, 211, 3,...
## $ start_station_name <chr> "5th St at Folsom", "Berkeley Civic Center", "Em...
## $ start_lat          <dbl> 37.78015, 37.86906, 37.83128, 37.80132, 37.78399...
## $ start_lng          <dbl> -122.4031, -122.2706, -122.2856, -122.2626, -122...
## $ end_station_id     <dbl> 60, 266, 149, 186, 368, 6, 6, 50, 81, 181, 336, ...
## $ end_station_name   <chr> "8th St at Ringold St", "Parker St at Fulton St"...
## $ end_lat            <dbl> 37.77452, 37.86246, 37.83128, 37.80132, 37.78543...
## $ end_lng            <dbl> -122.4094, -122.2648, -122.2856, -122.2626, -122...
## $ member_casual      <chr> "casual", "member", "casual", "casual", "member"...
## $ ride_id            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ rideable_type      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ is_equity          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
```

Questions to Answer

1. Explain what the GBFS is.

GBFS stands for General Bikeshare Feed Specification. It tells what variables are optional and required for data that bikeshare companies upload into the feed, what type they should be, etc. It has been adopted by hundreds of bikeshare companies worldwide to share real-time read-only data. Per https://nabsa.net/resources/gbfs/, it is intended to:

- Provide the status of the system at this moment
- Do not provide information whose primary purpose is historical
- The data in the specification is intended for consumption by clients intending to provide real-time (or semi-real-time) transit advice and is designed as such.

2. Explain any difficulties you encountered getting the code to work.

I initially made a typo in the URL for the lyftbaywheels data site, so it seemed at first to be working, it made a bunch of .zip files that were named what I expected them to be named, but then when I tried to unzip them I encountered an error. I tried manually opening the files and discovered that they were empty. When I fixed the URL, the code worked.

3. The analysis is to work with the data since Lyft BayWheels started, start with the data since May 2019. Modify the code to download the data to be analyzed. How many bike rentals were there before the COVID-19 lockdown in CA? How many bike rentals were there after the lockdown? How many bike rentals have there been since the beginning of Lyft BayWheels?

The Bay Area lockdown began March 17, 2020, so for pre-lockdown bike rentals we will include May 2019-March 16, 2020:

```
lyft %>%
  filter(started_at < as.Date("2020-03-17 00:00:00")) %>%
  summarise(n=n())
```

```
## # A tibble: 1 x 1
##         n
##     <int>
## 1 2504999
```

Pre-lockdown, there was a total of 2,504,999 rides.

```
lyft %>%
  filter(started_at >= as.Date("2020-03-17 00:00:00")) %>%
  summarise(n=n())
```

```
## # A tibble: 1 x 1
##        n
##    <int>
## 1 724178
```

Post-lockdown, there were a total of 724178 rides.

Adding these together will give us the total number of rides since the beginning of Lyft Baywheels.

```
724178+2504999
```

```
## [1] 3229177
```

All together there have been 3229177 renals since the beginning of Lyft Baywheels. This is good because it matches the number of rows in our lyft data frame.

4. There is a part of the code that uses the as.integer() function for some reason. Explain what this function is being used for in the code.

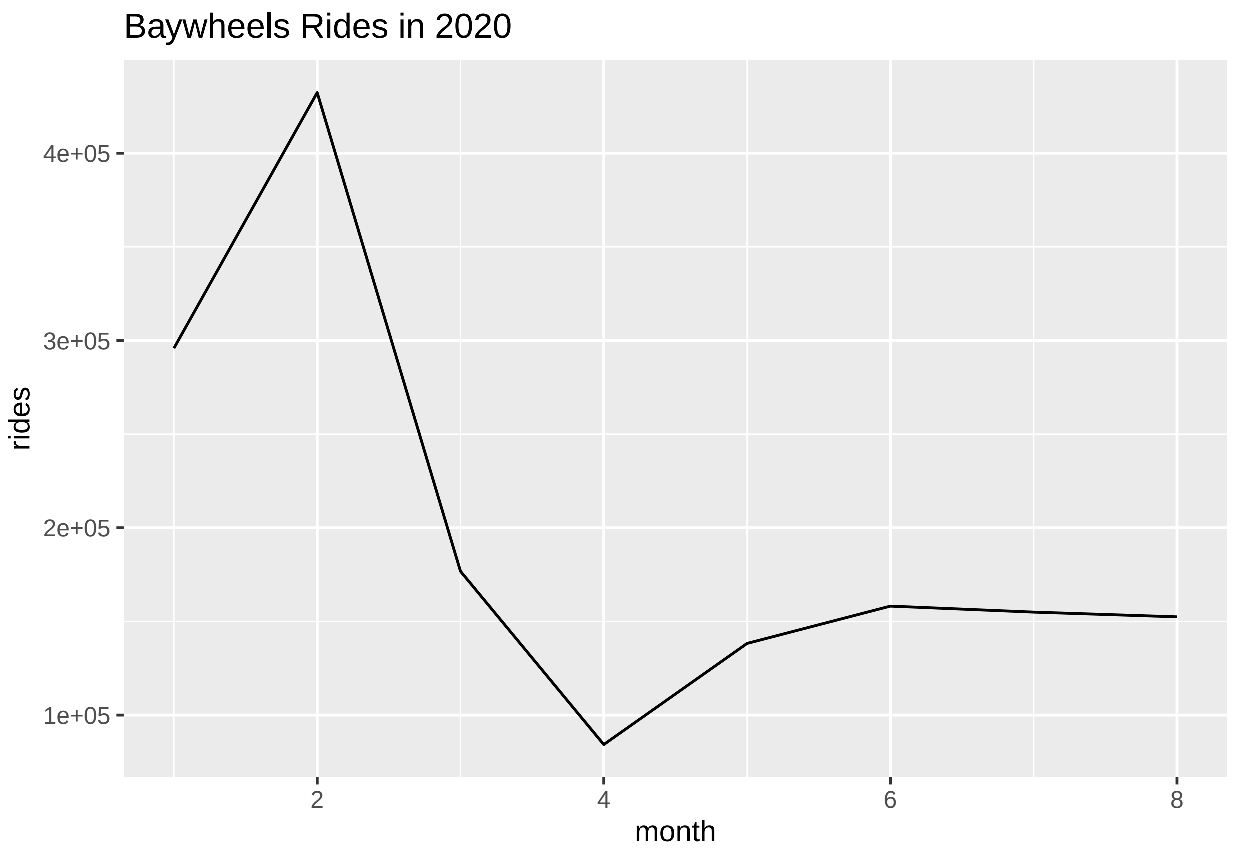
It's changing the variable type of the station id's to integer. The FordGoBike station id's were not all stored as integer type, which presented a problem when trying to merge the months together. This wasn't a problem for the Lyft data, as all the station ids were all already integers.

5. In 2020, what month had the highest number of riders? What month had the lowest number of riders? Interpret any seasonal patterns.

```
twentytwenty <- tibble(month = c(1,2,3,4,5,6,7,8),
                       rides = c(dim(baywheels202001)[1], dim(baywheels202002)[1], dim(baywheels202003)
twentytwenty
```

```
## # A tibble: 8 x 2
##   month  rides
##   <dbl>  <int>
## 1     1 295854
## 2     2 432354
## 3     3 176799
## 4     4  84259
## 5     5 138251
## 6     6 158168
## 7     7 154967
## 8     8 152446
```

```
ggplot(twentytwenty, aes(month, rides)) +
  geom_line()+
  ggtitle("Baywheels Rides in 2020")
```



In 2020, February had the highest number of riders. April had the lowest number of riders. This makes it appear that more people rent bikes in the winter than in the sping and summer, but I don't think this would be typical in a non-pandemic world. Another confounding factor is that Lyft changed their membership and pricing structure, raising rates and allowing less rental return flexibility, on March 2, so part of the decline in membership and rentals might also be caused by this price increase.

6. What start station had the highest number of rides? That is, which start station was used most to start rides?

```
lyft %>% select(start_station_id,start_station_name) %>%
  group_by(start_station_id) %>%
  summarise(n=n()) %>%
  arrange(desc(n))
```

```
## # A tibble: 469 x 2
##    start_station_id      n
##               <dbl>  <int>
##  1               NA 807633
##  2               58  41721
##  3               81  39100
##  4               30  37885
##  5               15  32424
##  6                3  30788
##  7               16  29867
##  8               21  29729
##  9               22  27524
## 10                  5  24687
## # ... with 459 more rows
```

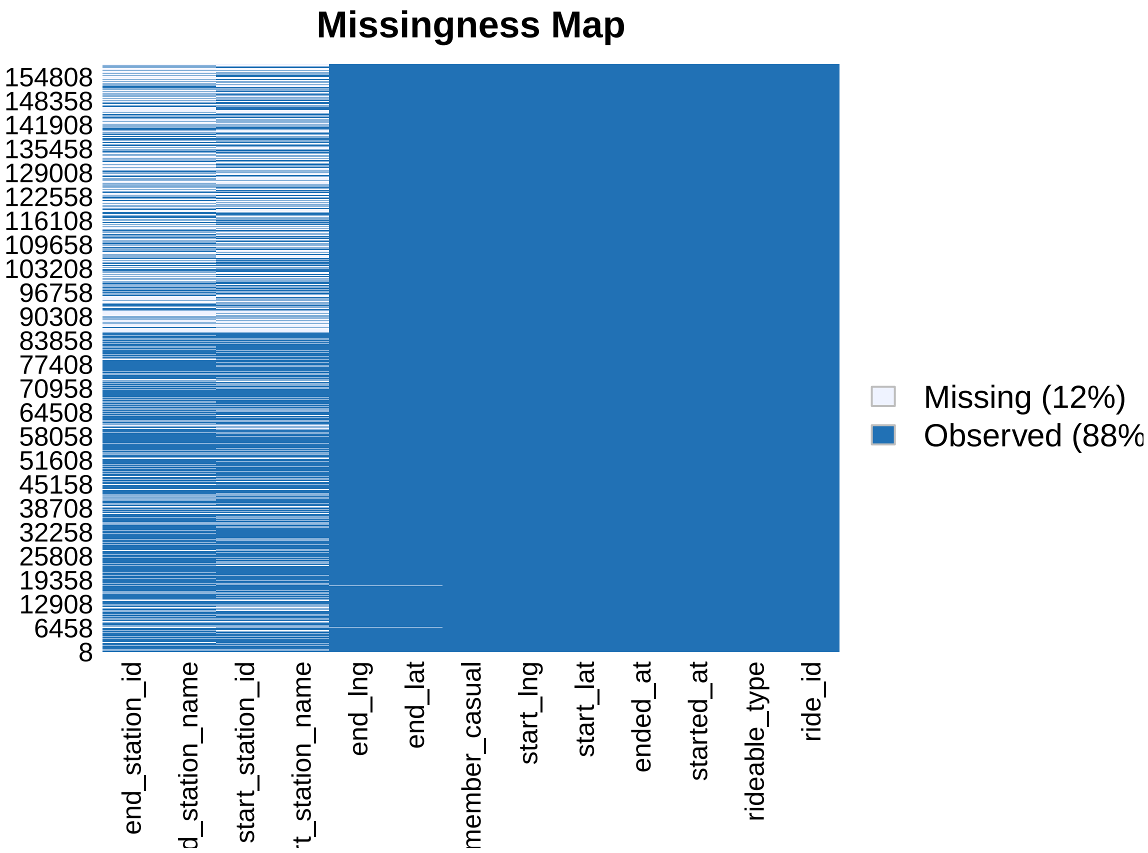
```
lyft %>% select(start_station_id, start_station_name) %>%
  filter(start_station_id == 58) %>%
  distinct()
```

```
## # A tibble: 1 x 2
##   start_station_id start_station_name
##              <dbl> <chr>
## 1                 58 Market St at 10th St
```

The station with the most rides was Market St at 10th St, which had 41,721 rides.

7. Using the Amelia R package and the missmap() function determine the rate of missing data in the month of June 2020. Or try the visdat package and the vis_miss() function. Or check out the the naniar R package. (This might not work on your computer if you have too little RAM.) If you cannot get your code to run, sample the data first.

```
missmap(baywheels202006)
```



June 2020 is missing 12% of its data. The variables most commonly missing are start and end station names and ids.

8. What Type of rider uses the Lyft BayWheels more? Subscribers or Customers?

```
lyft %>%
  group_by(member_casual) %>%
  summarise(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   member_casual       n
##   <chr>           <int>
## 1 casual        1139665
## 2 member        2089512
```

Overall, since its inception, members have used Lyft Baywheels more than non-members, making up 2089512/3229177 of rides, or:

```
2089512/3229177
```

```
## [1] 0.6470726
```

64.7% of rides were taken by members.

Because of the change in subscription policy on March 2, 2020, I think it would be interesting to look at the prevalence of subscribers before and after this change. We will first find out what percent of rides were taken by members before the policy change:

```
lyft %>%
  filter(started_at < as.Date("2020-03-02 00:00:00")) %>%
  group_by(member_casual) %>%
  summarise(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   member_casual       n
##   <chr>           <int>
## 1 casual         652942
## 2 member        1720968
```

```
1720968/(1720968+652942)
```

```
## [1] 0.7249508
```

Next, we will see what percent of rides were taken by members after the policy change.

```
lyft %>%
  filter(started_at >= as.Date("2020-03-02 00:00:00")) %>%
  group_by(member_casual) %>%
  summarise(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
## # A tibble: 2 x 2
##   member_casual      n
##   <chr>          <int>
## 1 casual        486723
## 2 member        368544
```

```
368544/(368544+486723)
```

```
## [1] 0.430911
```

Before the price change, 72.5% of the rides were taken by members, whereas after the price change, only 43.1% were taken by members, so it seems that the price change did coincide with a drastic reduction in the percentage of rides taken by members, which may or may not be causal.