MBTA Ridership

Ben Czekanski

3/7/2020

The first step in our exploration of MBTA ridership is to open the data and see what it looks like

```
import requests
import pandas as pd
import datetime as dt
import matplotlib.pyplot as plt

# this gives a warning
# 3e6 rows goes back to 2017
gses = pd.read_csv('/Users/Ben/Desktop/mbtaCV19/MBTA_Gated_Station_Entries.csv', nrows = 3e6)
gses.head()
```

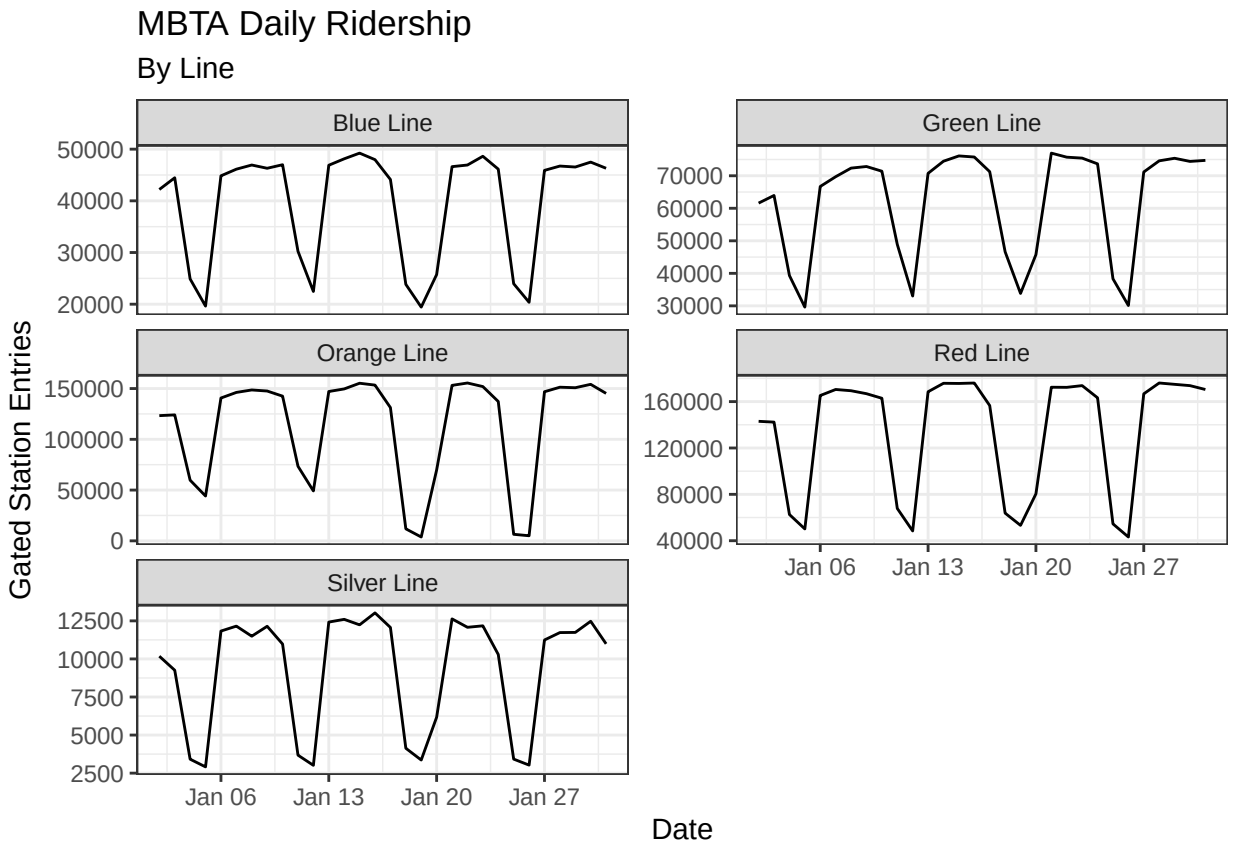
```
##                service_date  ...                                GlobalID
## 0  2020-01-30T05:00:00.000Z  ...  68328652-7a2d-43c6-aa88-df3fd24b6c31
## 1  2020-01-30T05:00:00.000Z  ...  990843a1-783e-43a9-95b4-411622aadc03
## 2  2020-01-31T05:00:00.000Z  ...  a90de5bb-8434-44d1-9c60-f73e714a4185
## 3  2020-01-30T05:00:00.000Z  ...  8987a6fd-bbe9-4455-b003-557b776d3aa8
## 4  2020-01-30T05:00:00.000Z  ...  47f42bef-49c4-4a62-a098-261822d7d3a5
##
## [5 rows x 8 columns]
```

Next, let's aggregate the total number of boardings by date and line and seewhat this looks like over time. For now, we are doing the data analysis in Python and the plotting in R because I am not sure that matplotlib allows for as much easy customization as ggplot.

```
gses['date'] = pd.to_datetime(gses['service_date'])

gses_agg_line = gses[['date', 'station_name', 'route_or_line', 'gated_entries']].groupby(['date', 'route
```

```
# Specify order
# py$gses_agg_line$route_or_line <- factor(py$gses_agg_line$route_or_line, ordered = TRUE)
## Just looking at 2020 for now
ggplot(py$gses_agg_line[py$gses_agg_line$date > as.POSIXct("2020-01-01"),],
       aes(x = date, y = entries)) + # , color = route_or_line, group = route_or_line)) +
    geom_line() +
    facet_wrap(.~route_or_line, scales = "free_y", ncol = 2) +
    # scale_color_manual(values = c("blue", "forestgreen", "darkorange", "red", "grey")) +
    labs(title = "MBTA Daily Ridership", subtitle = "By Line",
         x = "Date", y = "Gated Station Entries", color = "Line") +
    theme_bw()
```

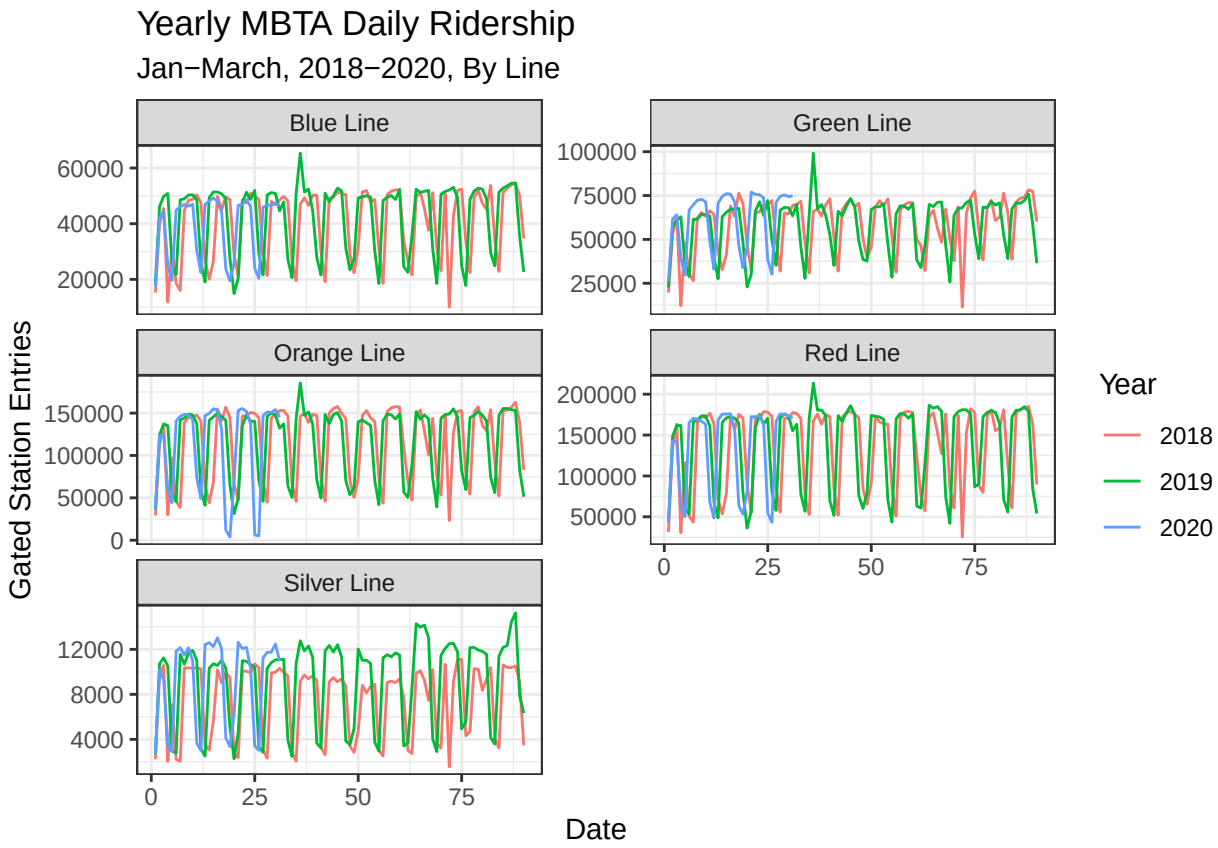


Let's next look at how ridership in the first 3 months of the year compares over the last few years

```
gses_agg_line['month'] =  gses_agg_line['date'].dt.month
gses_agg_line['year'] =  gses_agg_line['date'].dt.year
gses_agg_line['doy'] = gses_agg_line['date'].dt.dayofyear

f3mtoplot = gses_agg_line[(gses_agg_line['month'] <= 3) & (gses_agg_line['year'] >= 2018)]
```

```
ggplot(py$f3mtoplot, aes(x = doy, y = entries, color = as.factor(year), group = as.factor(year))) +
  geom_line() +
  facet_wrap(.~route_or_line, scales = "free_y", ncol = 2) +
  labs(title = "Yearly MBTA Daily Ridership", subtitle = "Jan-March, 2018-2020, By Line",
         x = "Date", y = "Gated Station Entries", color = "Year") +
    theme_bw()
```



This is messy because the weekends don't line up, so next we can try to either line up the weekends or aggregate by week. However, this is cool because we can see the Patriots' 2019 Super Bowl parade showing up here. In this chart we can also see Silver Line ridership increasing, possibly due to continuing development in the Seaport, and possibly due to to the 2018 launch of SL3 service to Chelsea in 2018. We also see a relatively low ridership day in early March, perhaps due to a snowstorm.

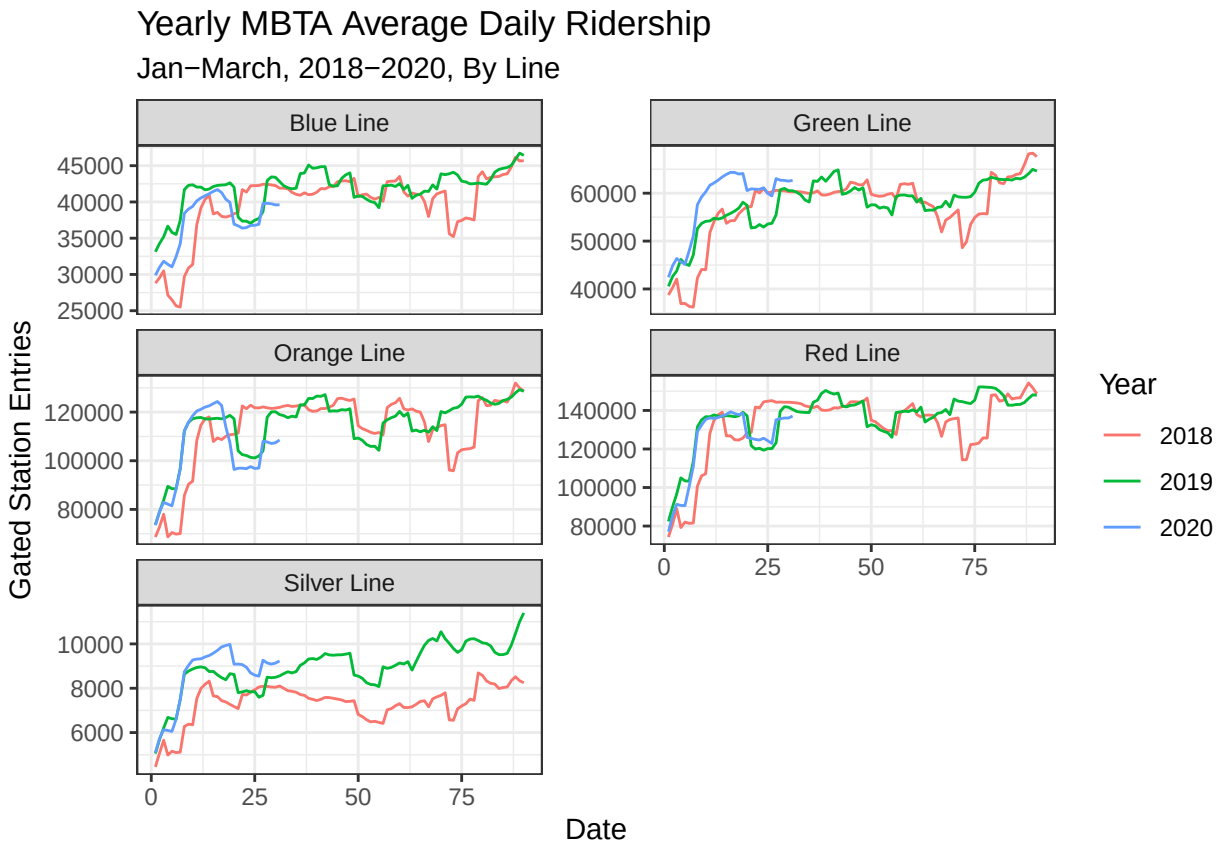
What we will do is calculate a seven-day average ridership to smooth the weekday/weekend noise.

```
# order first
# check calculation
gses_agg_line['weekly_avg'] = gses_agg_line.groupby('route_or_line')['entries'].rolling(window = 7, cen

# maybe also do weekdays only w recent weekend shutdowns

f3mtoplot2 = gses_agg_line[(gses_agg_line['month'] <= 3) & (gses_agg_line['year'] >= 2018)]
```

```
ggplot(py$f3mtoplot2, aes(x = doy, y = weekly_avg, color = as.factor(year), group = as.factor(year))) +
  geom_line() +
  facet_wrap(.~route_or_line, scales = "free_y", ncol = 2) +
  labs(title = "Yearly MBTA Average Daily Ridership", subtitle = "Jan-March, 2018-2020, By Line",
         x = "Date", y = "Gated Station Entries", color = "Year") +
    theme_bw()
```



This is useful, and we can see how holidays holidays have fallen on different points of the year but I am concerned that this may not be very comparable year over year because there have been weekend shutdowns that will show up in these series, but do not really represent a decrease in ridership.

Next we can look at a similar series of charts by stop

```
gses_agg_stn = gses[['date', 'station_name', 'route_or_line', 'gated_entries']].groupby(['date', 'static

gses_agg_stn['month'] =  gses_agg_stn['date'].dt.month
gses_agg_stn['year'] =  gses_agg_stn['date'].dt.year
gses_agg_stn['doy'] = gses_agg_stn['date'].dt.dayofyear
```

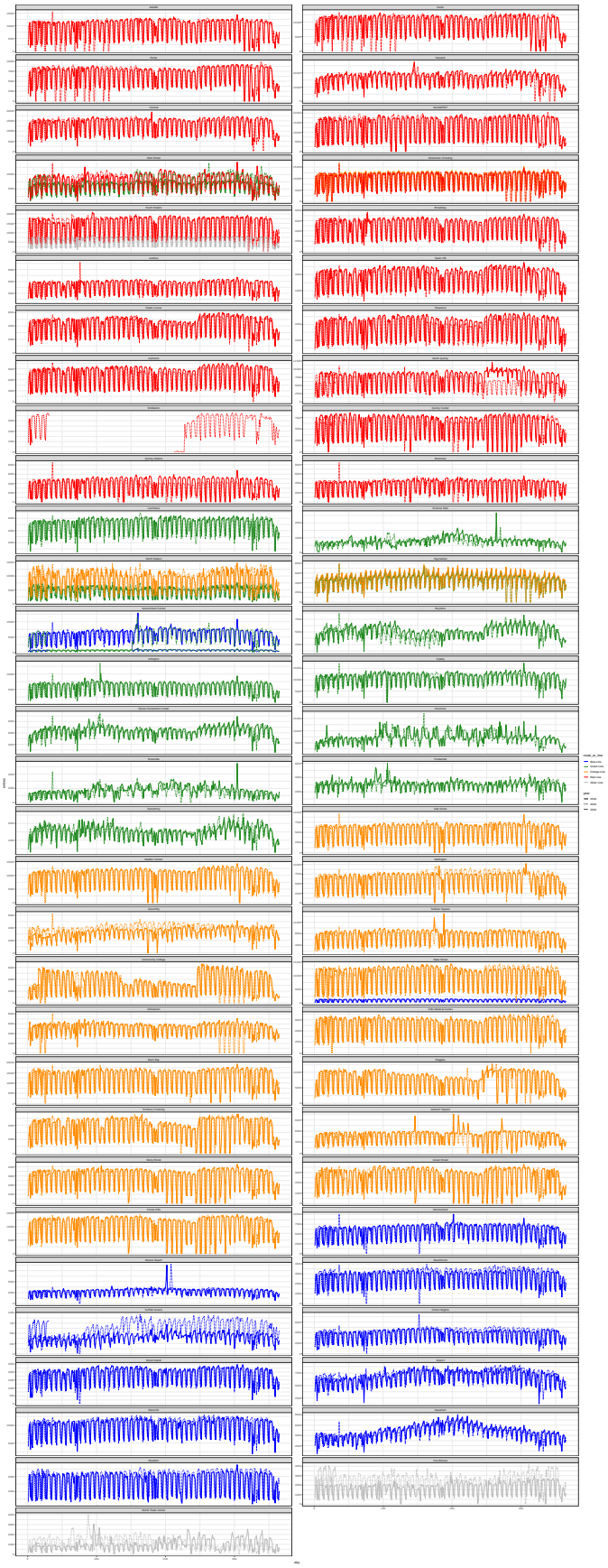
```
# order first
# check calculation
gses_agg_stn['weekly_avg'] = gses_agg_stn.groupby('station_name')['entries'].rolling(window = 7, center

# maybe also do weekdays only w recent weekend shutdowns

f3mtoplot3 = gses_agg_stn[(gses_agg_stn['month'] <= 3) & (gses_agg_stn['year'] >= 2018)]
```

```
# Charles MGH missing & Mass ave?
py$gses_agg_stn$station_name <- factor(py$gses_agg_stn$station_name, levels = c(
  "Alewife", "Davis", "Porter", "Harvard", "Central", "Kendall/MIT", "Park Street", "Downtown Crossing"
  "Lechmere", "Science Park", "North Station", "Haymarket", "Government Center", "Boylston", "Arlington
  "Oak Grove", "Malden Center", "Wellington", "Assembly", "Sullivan Square", "Community College", "Stat
  "Wonderland", "Revere Beach", "Beachmont", "Suffolk Downs", "Orient Heights", "Wood Island", "Airport
  "Courthouse", "World Trade Center"), ordered = TRUE)
py$gses_agg_stn$route_or_line <- as.factor(py$gses_agg_stn$route_or_line)
py$gses_agg_stn$year <- as.factor(py$gses_agg_stn$year)
```

```r
ggplot(py$gses_agg_stn[py$gses_agg_stn$date > as.POSIXct("2018-01-01"),],
       aes(x = doy, y = entries, color = route_or_line,
           group = interaction(route_or_line, year), linetype = year)) +
    geom_line() +
    facet_wrap(~station_name, drop = TRUE, ncol = 2, scale = "free_y") +
    scale_color_manual(values = c("blue", "forestgreen", "darkorange", "red", "grey")) +
    # labs(title = "MBTA Daily Ridership", subtitle = "By Line",
    #      x = "Date", y = "Gated Station Entries", color = "Line") +
    theme_bw()
```



```r
# Charles MGH missing & Mass ave?
py$f3mtoplot3$station_name <- factor(py$f3mtoplot3$station_name, levels = c(
  "Alewife", "Davis", "Porter", "Harvard", "Central", "Kendall/MIT", "Park Street", "Downtown Crossing"
  "Lechmere", "Science Park", "North Station", "Haymarket", "Government Center", "Boylston", "Arlington
  "Oak Grove", "Malden Center", "Wellington", "Assembly", "Sullivan Square", "Community College", "Stat
  "Wonderland", "Revere Beach", "Beachmont", "Suffolk Downs", "Orient Heights", "Wood Island", "Airport
  "Courthouse", "World Trade Center"), ordered = TRUE)
py$f3mtoplot3$route_or_line <- as.factor(py$f3mtoplot3$route_or_line)
py$f3mtoplot3$year <- as.factor(py$f3mtoplot3$year)

ggplot(py$f3mtoplot3[py$f3mtoplot3$date > as.POSIXct("2018-01-01"),],
       aes(x = doy, y = weekly_avg, color = route_or_line,
           group = interaction(route_or_line, year), linetype = year)) +
    geom_line() +
    facet_wrap(~station_name, drop = TRUE, ncol = 2, scale = "free_y") +
    scale_color_manual(values = c("blue", "forestgreen", "darkorange", "red", "grey")) +
    # labs(title = "MBTA Daily Ridership", subtitle = "By Line",
    #      x = "Date", y = "Gated Station Entries", color = "Line") +
    theme_bw()
```