

Ridership Trends Across North American Transit Systems

Alex Pan

February 2, 2018

Contents

Transit Ridership	1
Set up	1
Ridership by Region	1
Edmonton	1
Exploratory Plots	2
Chicago	5
Data Manipulation	5
Plots (Monthly ridership)	7
Plots (Daily Ridership)	8
Maryland	12
Data Manipulation	12
Exploratory Plots	16

Transit Ridership

Set up

```
library(tidyverse)
library(jsonlite)
library(lubridate)
library(ggplot2)
library(gganimate)
theme_set(theme_bw())
```

Ridership by Region

Edmonton

From <https://dashboard.edmonton.ca/Dashboard/Transit-Ridership/q4c4-5fu4> (updated January 9, 2018)

This file contains the annual total transit system ridership based on the last 12 months of fare revenue.

```
edmonton <- read_csv('Data/edmonton_ridership.csv')
```

```
head(edmonton)
```

```
## # A tibble: 6 x 7
##       ID      DateTime YEAR REPORT_PERIOD MONTH_RIDERSHIP
##   <int>      <chr> <int>      <chr>          <int>
## 1     1 01/01/2005 12:00:00 AM   2005      05-Jan      4762992
## 2     2 02/01/2005 12:00:00 AM   2005      05-Feb      4741322
```

```
## 3      3 03/01/2005 12:00:00 AM 2005      05-Mar      5081793
## 4      4 04/01/2005 12:00:00 AM 2005      05-Apr      4579192
## 5      5 05/01/2005 12:00:00 AM 2005      05-May      4119390
## 6      6 06/01/2005 12:00:00 AM 2005      05-Jun      4099550
## # ... with 2 more variables: `LAST 12 MONTHS` <int>, `CHANGE_%` <chr>
```

DateTime is being read as a factor, which will have to be converted to dates

Change_ is also being read as a factor, but I'm not immediately interested in that variable.

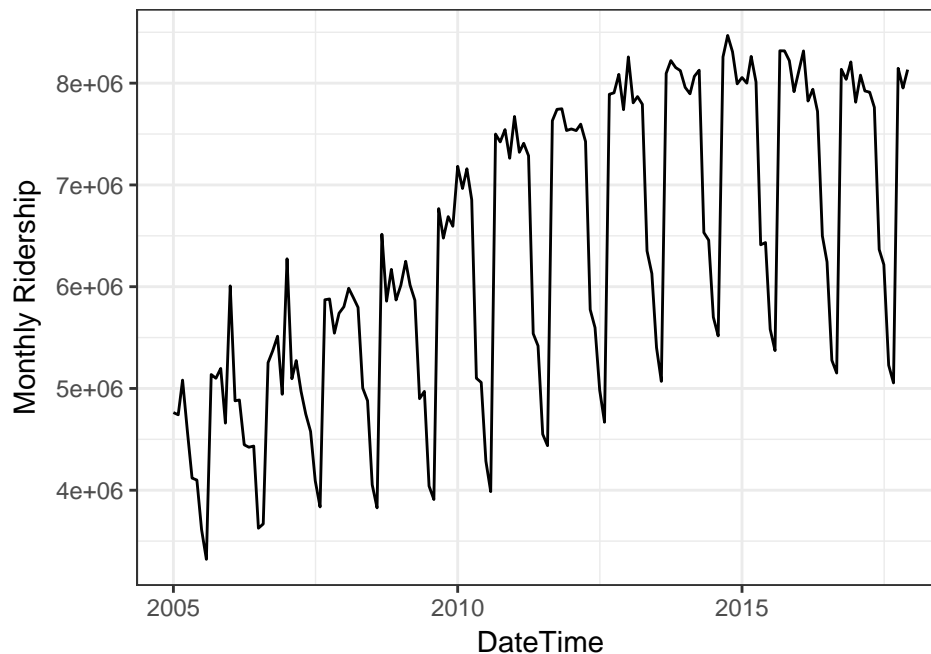
*# In addition to converting `DateTime` into a date object, I'm also going to
pull the month from that date object into a new column.*

```
edmonton <- edmonton %>%
  mutate(DateTime = mdy_hms(as.character(DateTime)),
         Month = month(DateTime)) %>%
  rename(Year = YEAR)
```

Exploratory Plots

A simple ridership vs. time scatterplot

```
ggplot(edmonton) +
  aes(x = DateTime, y = MONTH_RIDERSHIP) +
  geom_line() +
  ylab('Monthly Ridership')
```



This is an interesting plot that suggests a very strong seasonality.

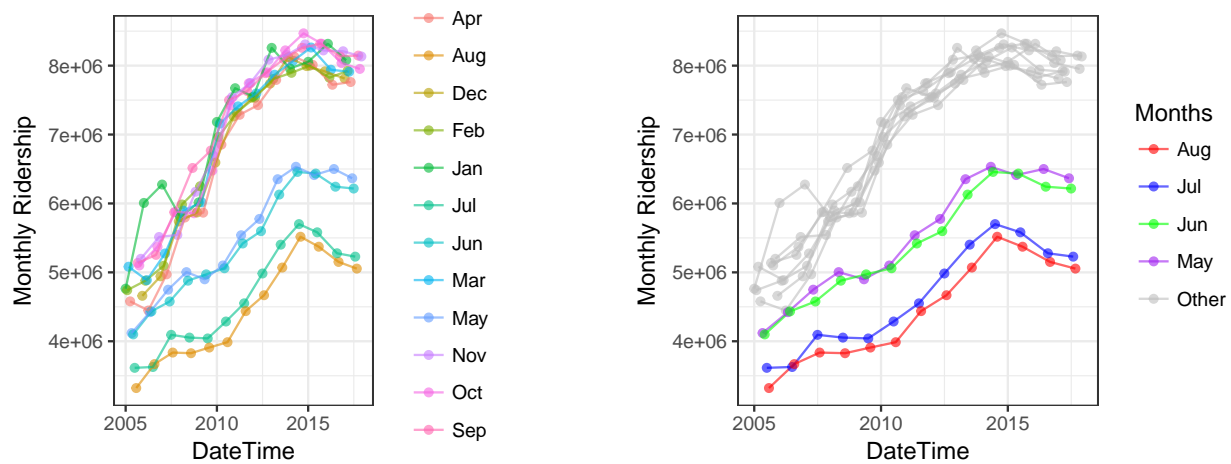
Split the plot by month

```
summer_months <- as.factor(
  ifelse(!(edmonton$Month %in% seq(5,8)), 'Other', month.abb[edmonton$Month]))
palette <- c('red', 'blue', 'green', 'purple', 'grey')

# All months
p1 <- ggplot(edmonton) +
  aes(x = DateTime, y = MONTH_RIDERSHIP, col = month.abb[Month]) +
  geom_point(alpha = 0.6) +
  geom_line(aes(group = month.abb[Month]), alpha = 0.6) +
  ylab('Monthly Ridership')

# Highlight summer months
p2 <- ggplot(edmonton) +
  aes(x = DateTime, y = MONTH_RIDERSHIP) +
  geom_line(aes(group = Month, col = summer_months), alpha = 0.6) +
  scale_colour_manual('Months', values = palette) +
  geom_point(aes(col = summer_months), alpha = 0.6) +
  ylab('Monthly Ridership')

gridExtra::grid.arrange(p1, p2, ncol = 2)
```



Here is another way to look at this:

```
p1 <- ggplot(edmonton) +
  geom_line(aes(x = Month, y = MONTH_RIDERSHIP / 1000000, group = Year, col = Year,
    cumulative = TRUE, frame = Year),
    alpha = 0.6) +
  geom_point(aes(x = Month, y = MONTH_RIDERSHIP / 1000000, group = Year, col = Year,
    cumulative = TRUE, frame = Year),
    alpha = 0.6) +
  scale_x_continuous(breaks = seq(1, 12)) +
  ylab('Monthly Ridership (millions)') +
  ggtitle('Edmonton Transit Service Monthly Ridership')

plot_animation <- gganimate(p1, interval = 1, 'edmonton_by_year.gif')
```

We can see that the summer months have much lower transit usage than the other months. This shoots way back up in September. Blame the university students?

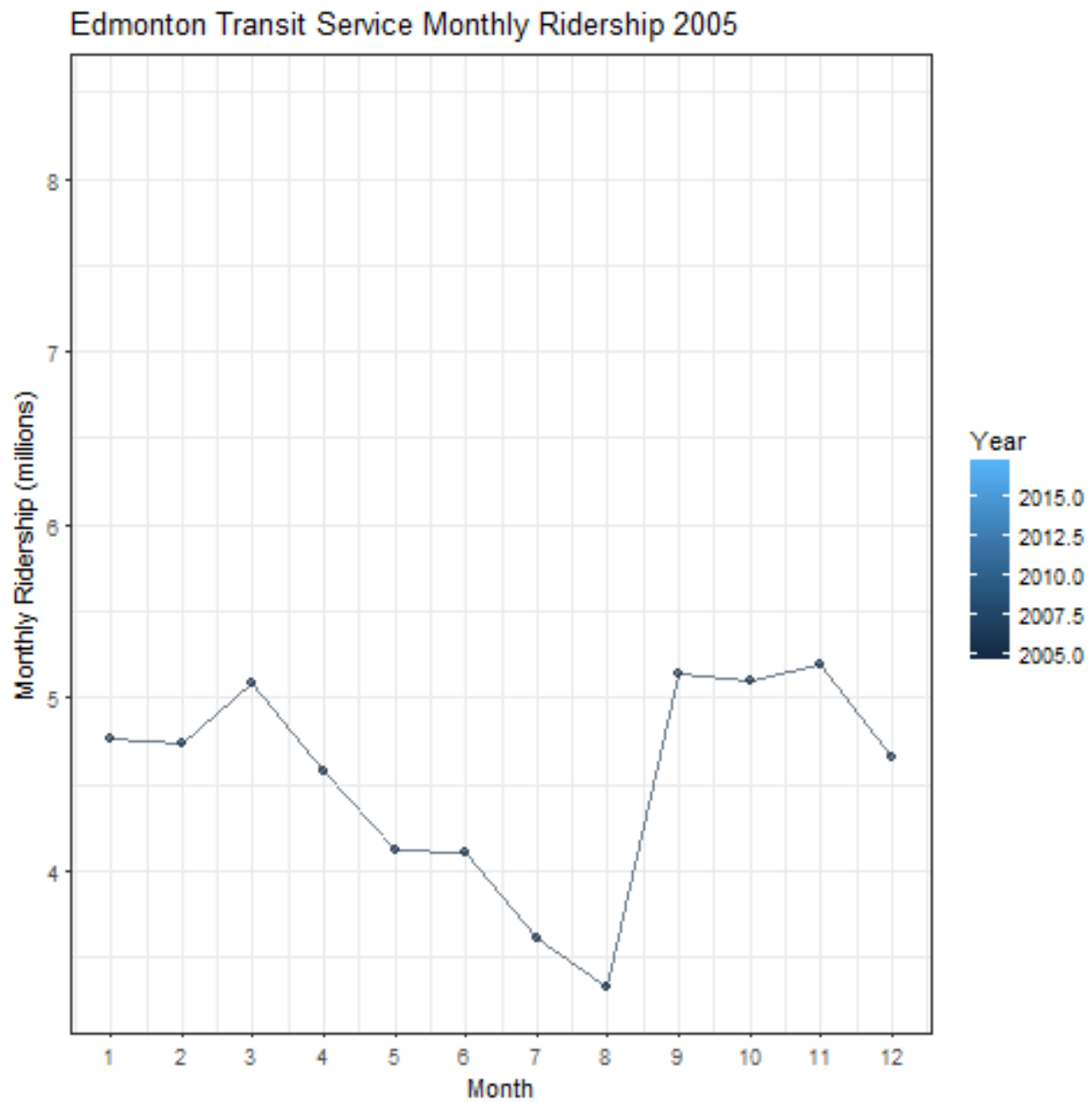


Figure 1:

How big is this difference?

```
temp <- edmonton %>%
  group_by(summer = ifelse(Month %in% seq(5,8), 'summer', 'other')) %>%
  summarize(mean = mean(MONTH_RIDERSHIP))

print(temp)

## # A tibble: 2 x 2
##   summer    mean
##   <chr>    <dbl>
## 1  other 6972246
## 2  summer 5010399

temp$mean[1] - temp$mean[2]

## [1] 1961847
```

On average, summer months have 1,961,847 fewer rides monthly than other times of the year.

Back to top

Chicago

From: <https://catalog.data.gov/dataset/cta-ridership-l-station-entries-monthly-day-type-averages-totals-26ba4>

These are data for the Chicago 'L' station riderships. This dataset lists monthly station entry averages, by day type (Weekday, Saturday or Sunday/Holiday), as well as monthly totals, beginning in 2001. Note that some stations (such as on the Cermak Branch—now Pink Line) and Skokie did not have Saturday and/or Sunday/holiday service until more recent years, although, in cases where weekday service ran past midnight, late evening fares may appear as part of Saturday tallies.

Unlike the other transit systems, this data set only includes data from train stations (and not buses)

Data Manipulation

```
chicago <- read_csv('Data/chicago_ridership.csv')

head(chicago)

## # A tibble: 6 x 7
##   station_id stationname month_beginning avg_weekday_rides
##   <int>      <chr>      <chr>          <dbl>
## 1    40900    Howard    01/01/2001    6233.9
## 2    41190    Jarvis    01/01/2001    1489.1
## 3    40100    Morse     01/01/2001    4412.5
## 4    41300    Loyola    01/01/2001    4664.5
## 5    40760    Granville 01/01/2001    3109.8
## 6    40880    Thorndale 01/01/2001    2977.7
## # ... with 3 more variables: avg_saturday_rides <dbl>,
## # `avg_sunday-holiday_rides` <dbl>, monthtotal <int>
```

Transform the dates to a usable format

```
chicago <- chicago %>%
  separate(month_beginning, into = c('Month', 'Day', 'Year'),
```

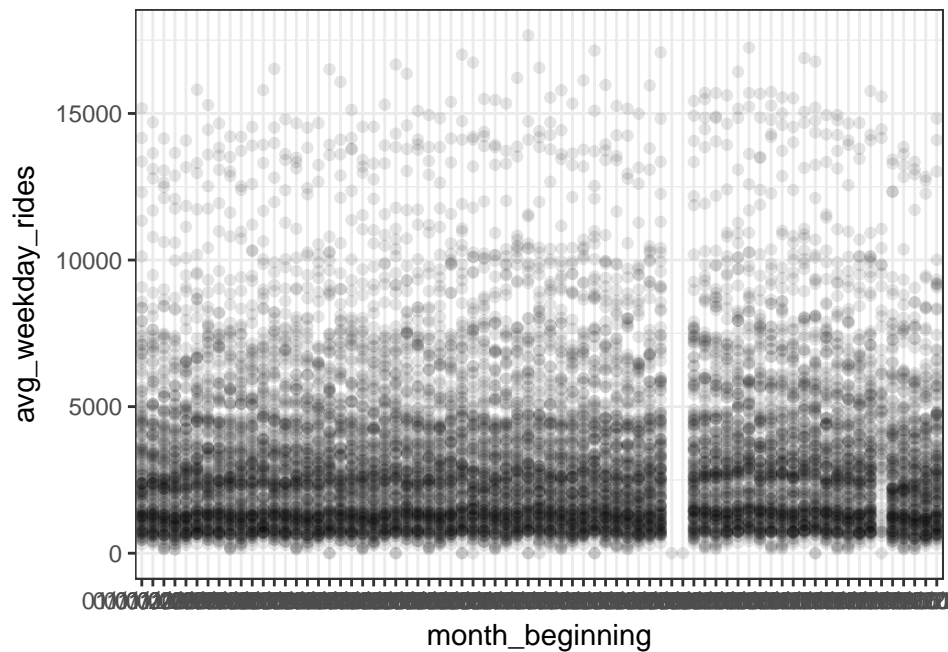
```
sep = '/', remove = F, convert = T)
```

```
head(chicago)
```

```
## # A tibble: 6 x 10
##   station_id stationname month_beginning Month   Day   Year
##   <int>      <chr>      <chr> <int> <int> <int>
## 1     40900    Howard    01/01/2001     1     1  2001
## 2     41190   Jarvis    01/01/2001     1     1  2001
## 3     40100    Morse    01/01/2001     1     1  2001
## 4     41300   Loyola    01/01/2001     1     1  2001
## 5     40760  Granville  01/01/2001     1     1  2001
## 6     40880  Thorndale  01/01/2001     1     1  2001
## # ... with 4 more variables: avg_weekday_rides <dbl>,
## #   avg_saturday_rides <dbl>, `avg_sunday-holiday_rides` <dbl>,
## #   monthtotal <int>
```

I noticed in some initial plots is that a handful of Stations had an average of 0 daily riders for that month. An example is below (note I've only included 10,000 rows in the plot below).

```
# Only show 10,000 rows to prevent overplotting
ggplot(chicago[1:10000, ]) +
  aes(x = month_beginning, y = avg_weekday_rides) +
  geom_point(alpha = 0.1)
```



```
# How many cases are there of 0 monthly weekday rides?
length(chicago[chicago$avg_weekday_rides == 0, ]$avg_weekday_rides)
```

```
## [1] 194
```

```
# length(chicago[chicago$avg_saturday_rides == 0, ]$avg_saturday_rides)
# length(chicago[chicago$`avg_sunday-holiday_rides` == 0, ]$`avg_sunday-holiday_rides` )
```

I'm going to assume that 0 daily riders in a month means that the station was closed or the data were missing,

so I'm going to change these values to NA so that they better reflect a typical day

```
chicago[chicago$avg_weekday_rides == 0, ]$avg_weekday_rides <- NA
# chicago[chicago$avg_saturday_rides == 0, ]$avg_saturday_rides <- NA
# chicago[chicago$`avg_sunday-holiday_rides` == 0, ]$`avg_sunday-holiday_rides` <- NA
```

Unlike the other data sets, these data are supplied for every single station. To make it more similar, we'll have to combine data from all stations for each month.

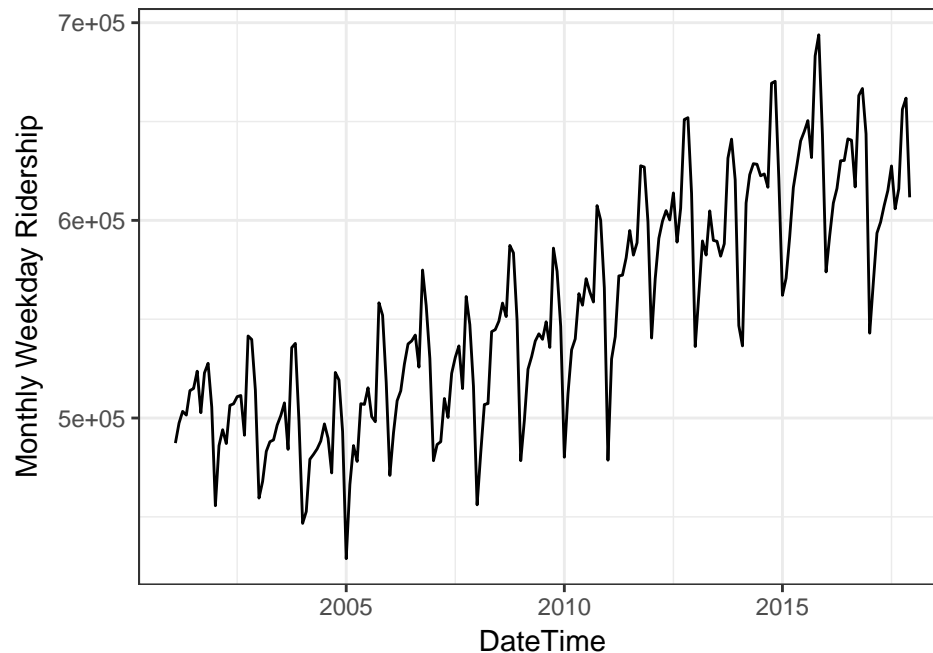
```
chicago_by_month <- chicago %>% group_by(Year, Month) %>%
  summarize(avg_weekday_rides = sum(avg_weekday_rides, na.rm = T),
            avg_saturday_rides = sum(avg_saturday_rides, na.rm = T),
            avg_sunday_holiday_rides = sum(`avg_sunday-holiday_rides`, na.rm = T)) %>%
  mutate(DateTime = Year + Month/12)
```

```
head(chicago_by_month)
```

```
## # A tibble: 6 x 6
## # Groups:   Year [1]
##   Year Month avg_weekday_rides avg_saturday_rides
##   <int> <int>          <dbl>          <dbl>
## 1  2001     1      487319.6      220753.3
## 2  2001     2      497416.6      219440.1
## 3  2001     3      503369.6      245850.6
## 4  2001     4      501485.0      242434.5
## 5  2001     5      513900.5      246570.3
## 6  2001     6      514917.1      279364.0
## # ... with 2 more variables: avg_sunday_holiday_rides <dbl>,
## #   DateTime <dbl>
```

Plots (Monthly ridership)

```
ggplot(chicago_by_month) +
  geom_line(aes(x = DateTime, y = avg_weekday_rides)) +
  ylab('Monthly Weekday Ridership')
```



Chicago's transit looks to be getting busier over time.

Once again, we'll look at the monthly trends

```
p1 <- ggplot(chicago_by_month) +
  geom_line(aes(x = Month, y = avg_weekday_rides / 10000, group = Year, col = Year,
    cumulative = TRUE, frame = Year),
    alpha = 0.6) +
  geom_point(aes(x = Month, y = avg_weekday_rides / 10000, group = Year, col = Year,
    cumulative = TRUE, frame = Year),
    alpha = 0.6) +
  scale_x_continuous(breaks = seq(1, 12)) +
  ylab('Mean Weekday Ridership (10,000s)') +
  ggtitle('Chicago Transit Authority Monthly Ridership')

plot_animation <- gganimate(p1, interval = 1, 'chicago_by_year.gif')
```

Chicago also has some seasonality, but unlike Edmonton, it looks to be less busy in the winter.

Plots (Daily Ridership)

Since we have the luxury of data from every station for every day, let's look at the popularity of each station:

```
chicago_by_station <- chicago %>% group_by(stationname) %>%
  summarize(avg_weekday_rides = mean(avg_weekday_rides),
    avg_saturday_rides = mean(avg_saturday_rides),
    avg_sunday_holiday_rides = mean(`avg_sunday-holiday_rides`))
```

Busiest stations during the week):

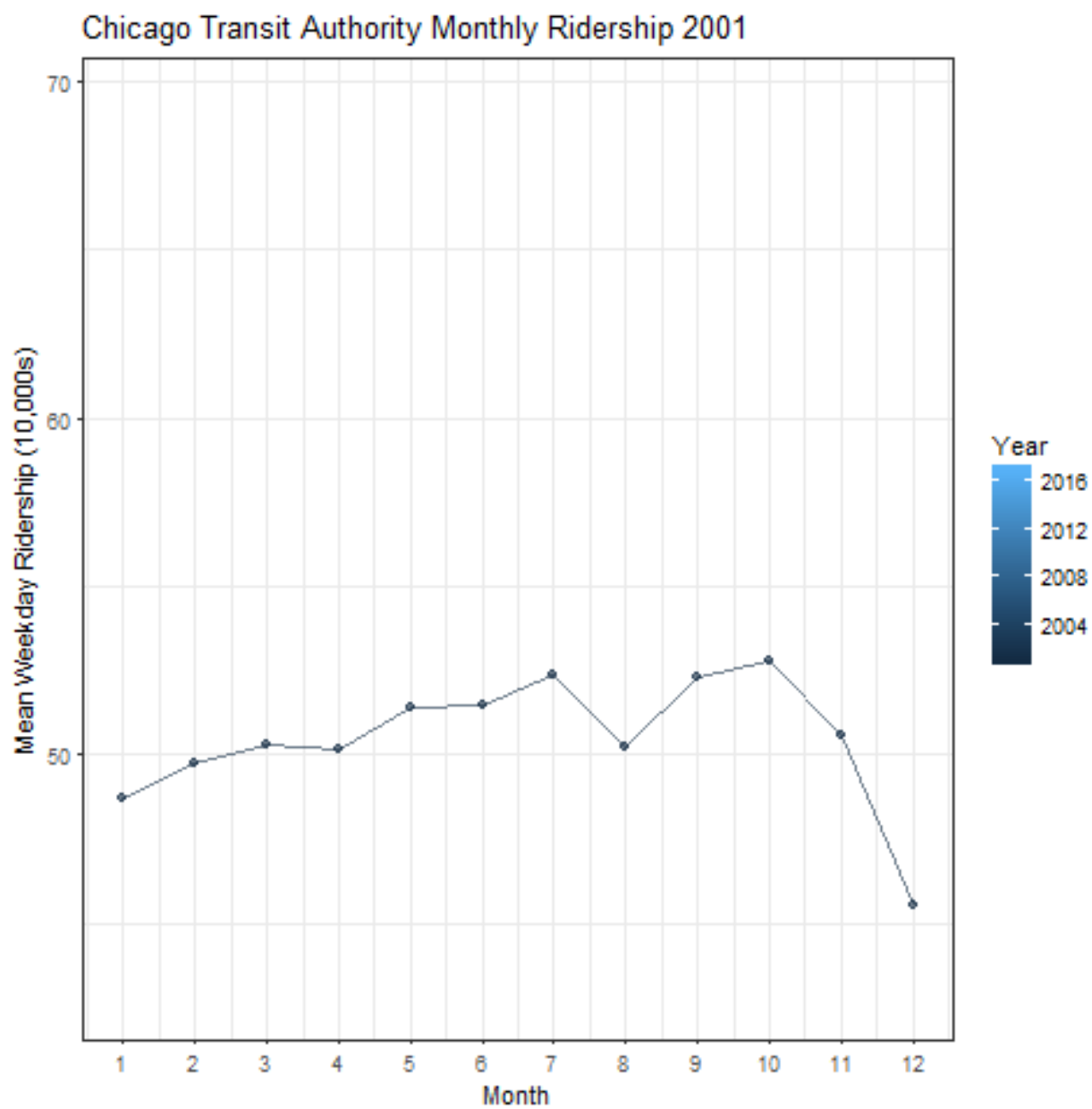


Figure 2:

```
# Ten busiest stations
chicago_by_station <- chicago_by_station %>% arrange(desc(avg_weekday_rides))

head(chicago_by_station, 10)
```

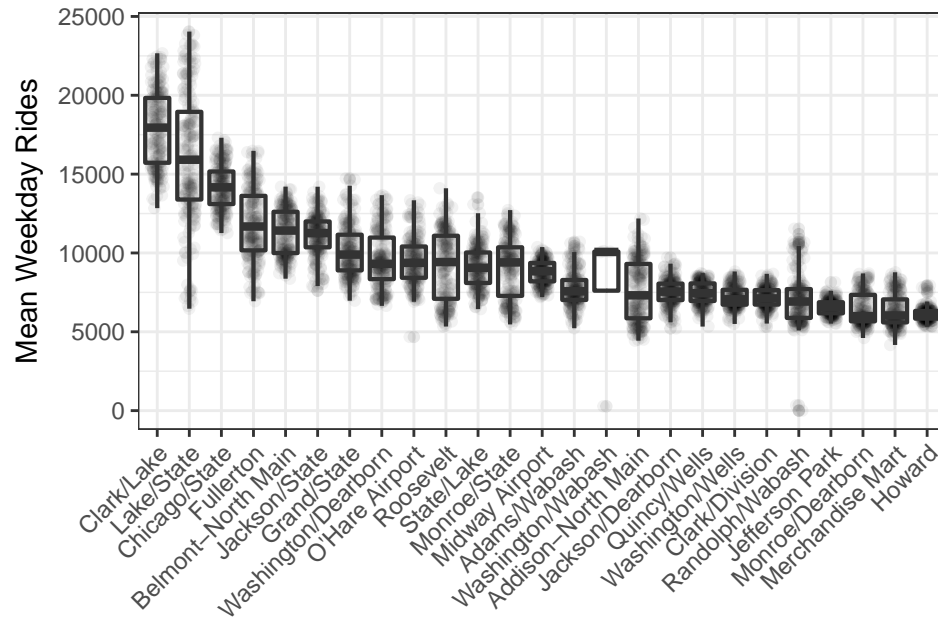
```
## # A tibble: 10 x 4
##       stationname avg_weekday_rides avg_saturday_rides
##       <chr>          <dbl>          <dbl>
## 1      Clark/Lake      17868.422      5072.017
## 2      Lake/State      16062.513      9468.741
## 3    Chicago/State      14194.534     12128.758
## 4      Fullerton      11814.626      7964.955
## 5 Belmont-North Main    11290.372      9927.475
## 6      Jackson/State    11104.076      4661.385
## 7      Grand/State     10084.417      9487.222
## 8 Washington/Dearborn   9573.944      4109.655
## 9      O'Hare Airport   9506.802      7661.039
## 10     Roosevelt       9187.917      7441.314
## # ... with 1 more variables: avg_sunday_holiday_rides <dbl>
```

Plot the 25 busiest stations

```
busiest_stations <- chicago_by_station[1:25, ]$stationname

temp <- filter(chicago, stationname %in% busiest_stations)
temp$stationname <- factor(temp$stationname,
  levels = busiest_stations, ordered = TRUE)

ggplot(temp) +
  aes(x = stationname, y = avg_weekday_rides) +
  geom_point(alpha = 0.05, position = position_jitter(width = 0.2)) +
  geom_boxplot(alpha = 0.1, size = 0.75) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylab('Mean Weekday Rides') + xlab('')
```



Busiest stations on Saturdays:

```
# Ten busiest stations
chicago_by_station <- chicago_by_station %>% arrange(desc(avg_saturday_rides))

head(chicago_by_station, 10)
```

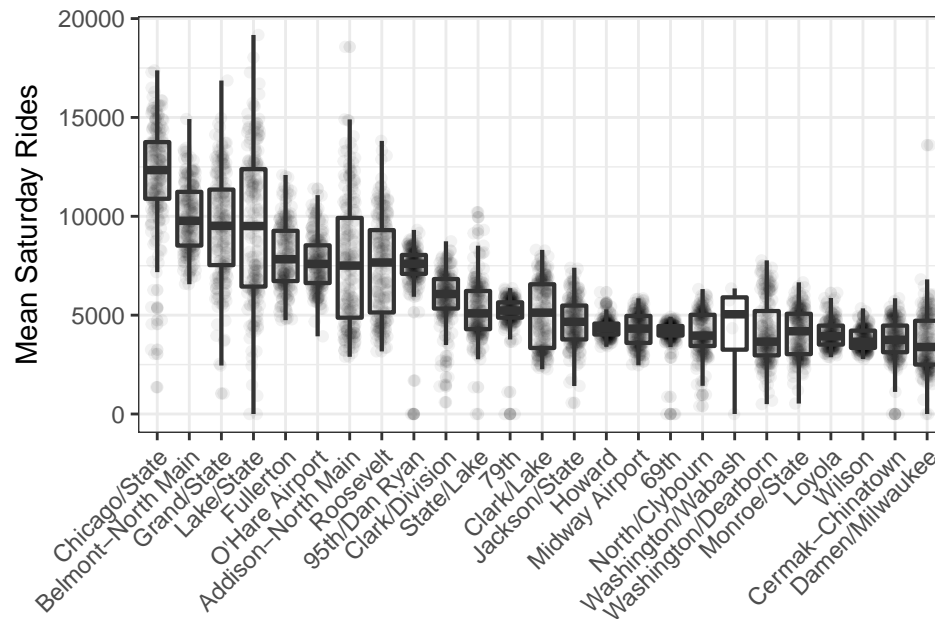
```
## # A tibble: 10 x 4
##       stationname avg_weekday_rides avg_saturday_rides
##       <chr>          <dbl>          <dbl>
## 1 Chicago/State      14194.534      12128.758
## 2 Belmont-North Main 11290.372       9927.475
## 3 Grand/State        10084.417       9487.222
## 4 Lake/State         16062.513       9468.741
## 5 Fullerton          11814.626       7964.955
## 6 O'Hare Airport      9506.802       7661.039
## 7 Addison-North Main 7593.961       7625.417
## 8 Roosevelt           9187.917       7441.314
## 9 95th/Dan Ryan       NA           7418.947
## 10 Clark/Division     7164.378       5964.380
## # ... with 1 more variables: avg_sunday_holiday_rides <dbl>
```

```
busiest_saturday_stations <- chicago_by_station[1:25, ]$stationname

temp <- filter(chicago, stationname %in% busiest_saturday_stations)
temp$stationname <- factor(temp$stationname,
  levels = busiest_saturday_stations, ordered = TRUE)

ggplot(temp) +
  aes(x = stationname, y = avg_saturday_rides) +
  geom_point(alpha = 0.05, position = position_jitter(width = 0.2)) +
  geom_boxplot(alpha = 0.1, size = 0.75) +
```

```
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
ylab('Mean Saturday Rides') + xlab('')
```



Back to top

Maryland

From: <https://catalog.data.gov/dataset/mta-average-weekday-ridership-by-month>

These are data for the entire state, rather than for a single city.

The data are given as the average weekday ridership across each medium of transit:

- Intra-city transit: bus, metro, light rail
- Accessibility services: MobilityLink and Taxi Access
- MARC commuter train system, which connects the Baltimore-Washington DC area.
- Commuter Bus system, which serves to connect suburban areas to urban Baltimore/Washington DC.

Also note that these data are daily ridership numbers, whereas Edmonton's data were supplied as Monthly numbers. We can easily calculate that by multiplying by days in the month, but I won't be doing anything that requires that.

Data Manipulation

```
json_maryland <- fromJSON('Data/maryland_ridership.json', flatten = TRUE)

maryland <- data.frame(json_maryland$data)
names(maryland) <- json_maryland$meta$view$columns$fieldname
```

```
names(maryland)
```

```
## [1] ":sid"                ":id"
## [3] ":position"           ":created_at"
## [5] ":created_meta"       ":updated_at"
## [7] ":updated_meta"       ":meta"
## [9] "average_weekday_ridership" "bus"
## [11] "metro"               "light_rail"
## [13] "mobility"            "taxi_access"
## [15] "marc_average"        "marc_brunswick"
## [17] "marc_camden"         "marc_penn"
## [19] "commuter_bus_total"  "baltimore"
## [21] "washington"         "icc"
## [23] "total_average_weekday_ridership"
```

total_average_weekday_ridership is the total average ridership across all transit media, calculated by the folks at the MTA. In general, we'll prefer to use our own calculations rather than ones provided to us.

```
maryland[1:3, 9:23] # Display first 3 rows/The first 8 columns are meta data
```

```
##   average_weekday_ridership   bus metro light_rail mobility taxi_access
## 1                Jul-06 205015 43358      22997      2838      1072
## 2                Aug-06 215455 44427      22708      2860      1162
## 3                Sep-06 251719 44062      24085      2985      1211
##   marc_average marc_brunswick marc_camden marc_penn commuter_bus_total
## 1          <NA>          <NA>      <NA>      <NA>          <NA>
## 2          <NA>          <NA>      <NA>      <NA>          <NA>
## 3          <NA>          <NA>      <NA>      <NA>          <NA>
##   baltimore washington   icc total_average_weekday_ridership
## 1          <NA>          <NA> <NA>                275280
## 2          <NA>          <NA> <NA>                286612
## 3          <NA>          <NA> <NA>                324062
```

Some problems with the data:

- Everything has been read in as a string/factor
- The MARC and Commuter Bus data are missing (at least for these rows)

Fix the variable typing:

```
# Convert columns to correct type
# Also add usable date columns
maryland[] <- lapply(maryland[], function(x) type.convert(as.character(x)))

maryland <- maryland %>%
  separate(average_weekday_ridership, into = c('Month', 'Year'), sep = '-') %>%
  mutate(Month = match(Month, month.abb),
         Year = as.numeric(paste('20', Year, sep = '')),
         DateTime = Year + Month / 12)
```

Missingness Plot

```
# Plot each NA as a bar, sorted by date.

# Sort by date
maryland <- maryland %>% arrange(DateTime)
```

```

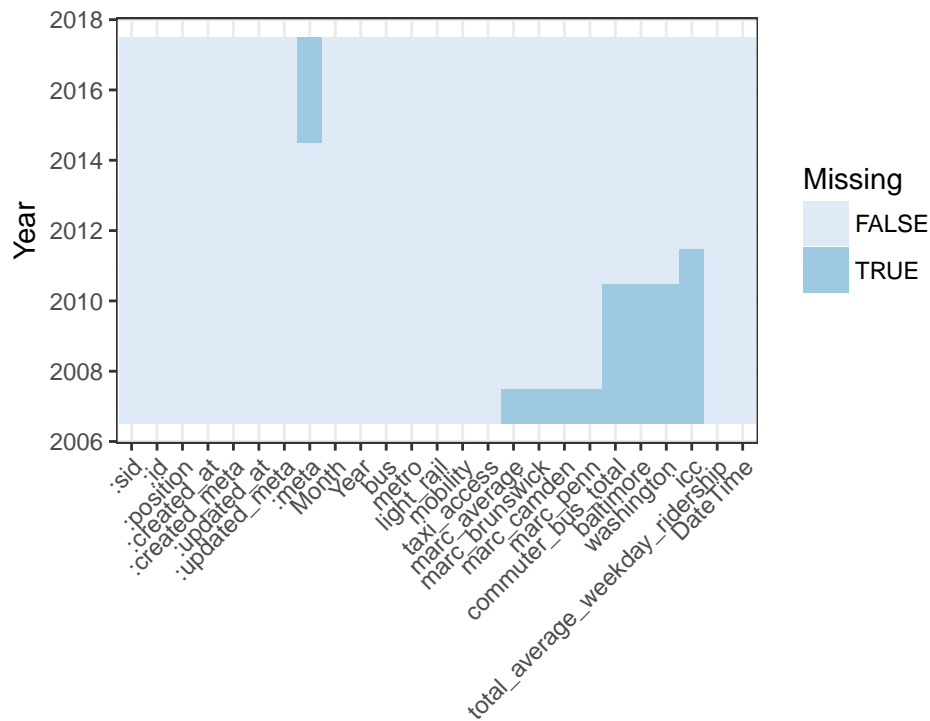
# Find if entry is missing
x <- as.matrix(is.na(maryland))

# Name columns to match year. To prevent congestion of the axis, only name rows
# that correspond to January (i.e. Month == 1)
dimnames(x)[[1]] <- ifelse(maryland$Month == 1, maryland$Year, NA)

# Reshape matrix x into a single column of data
x <- reshape2::melt(x)

ggplot(x, aes(x = Var2, y = Var1, fill = value)) +
  geom_tile() +
  scale_fill_brewer('Missing') +
  ylab('Year') + xlab('') +
  theme(axis.text.x = element_text(angle=45, hjust = 1))

```



We see that we don't have data on the MARC until ~2007, and we don't have data on the Commuter Bus or Intracounty Commuter Bus (ICC) until 2011. These will bias the `total_average_weekday_ridership` variable, so we should probably calculate totals ourselves when making comparisons across years.

We can also see that our data do not extend to the end of 2017.

```
max(maryland$DateTime, na.rm = T)
```

```
## [1] 2017.5
```

They only extend until June.

Compare total ridership including MARC, commuter bus, and ICC to intracity ridership

```

# New calculation of total ridership without MARC, Commuter bus, and ICC

maryland <- maryland %>%
  mutate(ridership_intracity = bus + metro + light_rail + mobility + taxi_access)

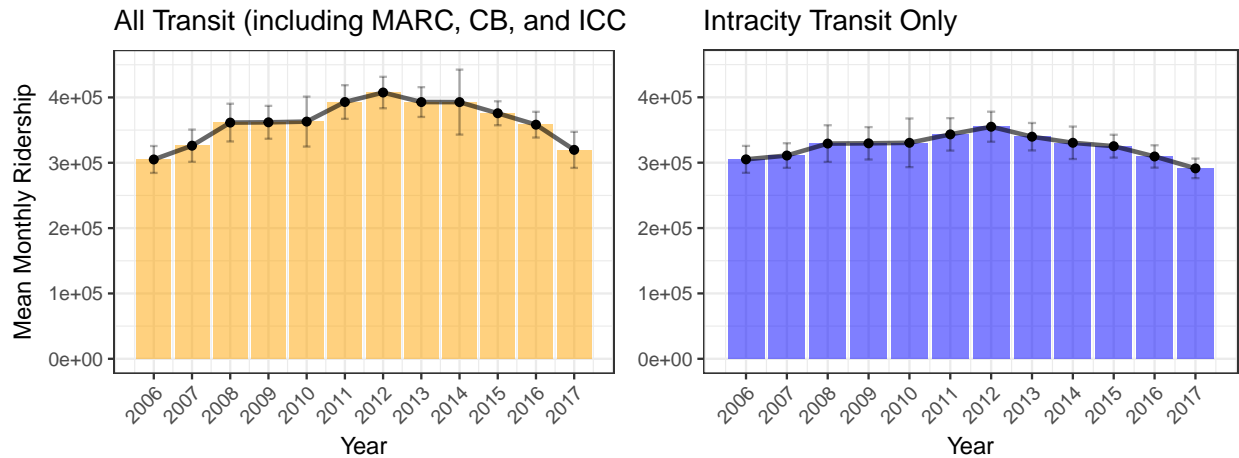
maryland_by_year <- maryland %>% group_by(Year) %>%
  summarize(mean = mean(total_average_weekday_ridership),
            sd = sd(total_average_weekday_ridership),
            mean_intracity = mean(ridership_intracity),
            sd_intracity = sd(ridership_intracity))

# Compare all trips
p1 <- ggplot(maryland_by_year) +
  aes(x = Year, y = mean) +
  geom_col(fill = 'orange', alpha = 0.5) +
  geom_point() +
  geom_line(alpha = 0.6, size = 1) +
  geom_errorbar(aes(ymin = mean - sd,
                    ymax = mean + sd),
                width = 0.2, alpha = 0.3) +
  ggtitle('All Transit (including MARC, CB, and ICC)') +
  ylab('Mean Monthly Ridership') +
  scale_x_continuous(breaks = seq(2006, 2017)) +
  ylim(c(0, 450000)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p2 <- ggplot(maryland_by_year) +
  aes(x = Year, y = mean_intracity) +
  geom_col(fill = 'blue', alpha = 0.5) +
  geom_point() +
  geom_line(alpha = 0.6, size = 1) +
  geom_errorbar(aes(ymin = mean_intracity - sd_intracity,
                    ymax = mean_intracity + sd_intracity),
                width = 0.2, alpha = 0.3) +
  ggtitle('Intracity Transit Only') +
  scale_x_continuous(breaks = seq(2006, 2017)) +
  ylim(c(0, 450000)) +
  theme(axis.title.y = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1))

gridExtra::grid.arrange(p1, p2, ncol = 2)

```



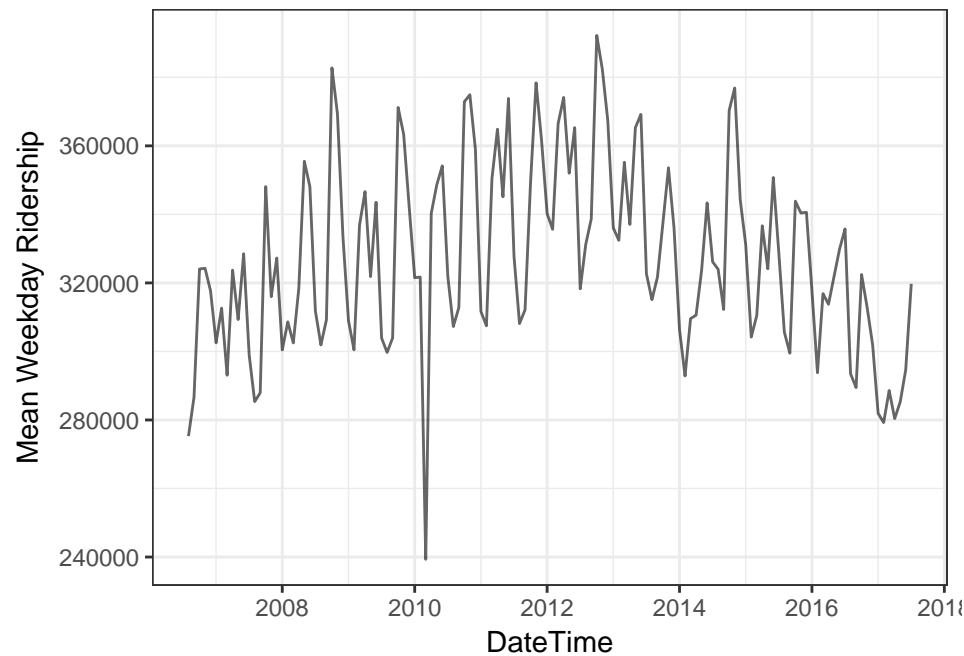
We can see that the ‘jumps’ in ridership in 2007/2008 and 2011/2012 mostly disappear when we only look at intracity transit.

Also recall that the data for 2017 only go up to June.

Exploratory Plots

Let’s just look at ridership over time for the urban transit system

```
ggplot(maryland) +
  aes(x = DateTime, y = maryland$ridership_intracity) +
  geom_line(alpha = 0.6) +
  ylab('Mean Weekday Ridership')
```



Unlike the Edmonton transit system, the obvious trend is that the transit system is being used less over time. We also don’t see any sort of obvious monthly trends.

The dip in February 2010 is interesting. A quick internet search reveals that there was a bad blizzard in February 2010 that covered the roads with 2 feet of snow.

Looking at Monthly trends

```
p1 <- ggplot(maryland) +  
  geom_line(aes(x = Month, y = ridership_intracity / 10000, group = Year, col = Year,  
               cumulative = TRUE, frame = Year),  
            alpha = 0.6) +  
  geom_point(aes(x = Month, y = ridership_intracity / 10000, group = Year, col = Year,  
                cumulative = TRUE, frame = Year),  
            alpha = 0.6) +  
  scale_x_continuous(breaks = seq(1, 12)) +  
  ylab('Mean Weekday Ridership (10,000s)') +  
  ggtitle('Maryland Transit Administration Monthly Ridership')  
  
plot_animation <- gganimate(p1, interval = 1, 'maryland_by_year.gif')
```

There's a dip in the summer, but you really can't tell what's going on.

[Back to top](#)

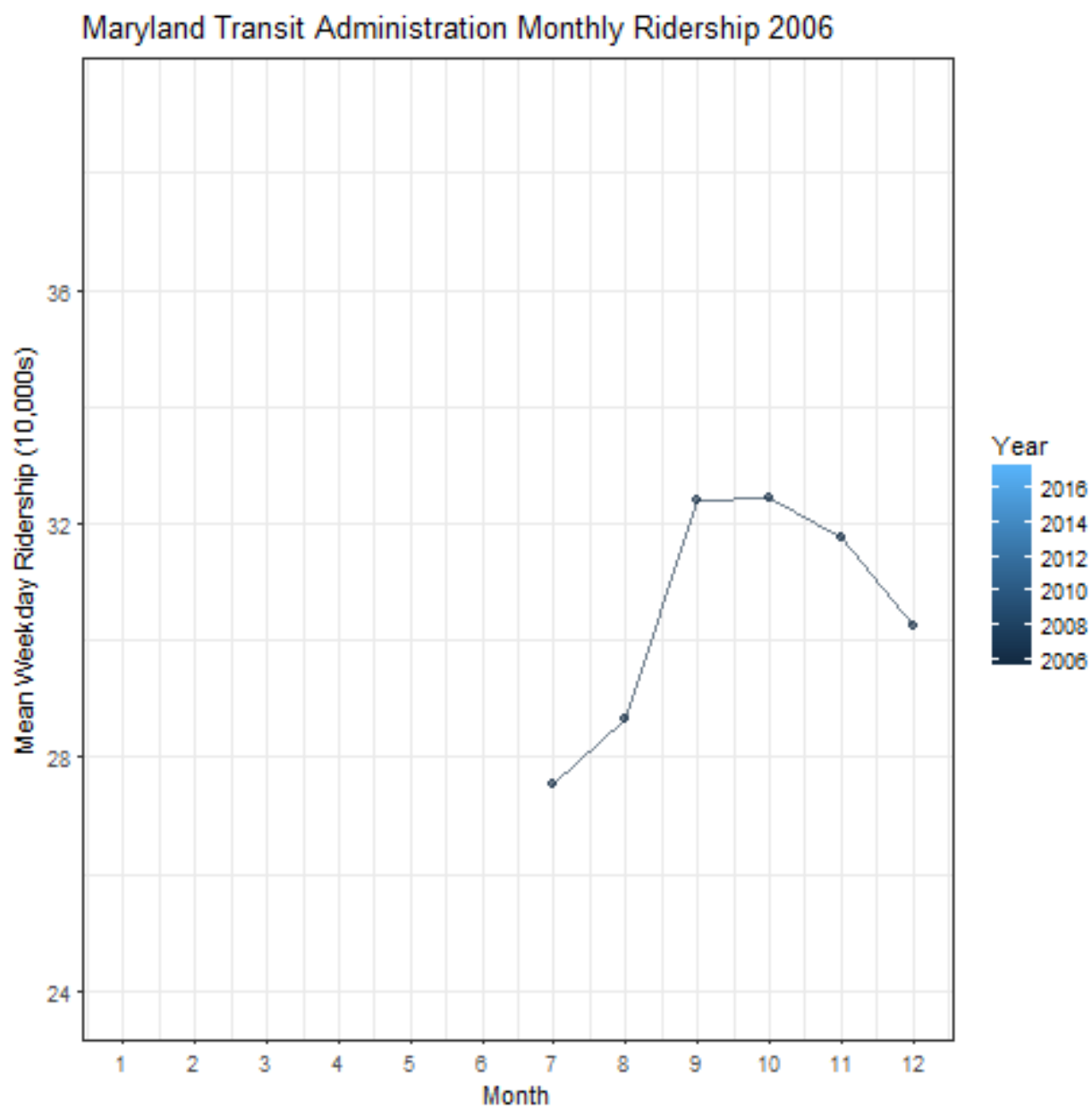


Figure 3: