

# FinalProj\_BStriano

Brendan M. Striano

Before we dive into the analytical question of interest, let's bring in the packages that we will use later on in this code.

## **Project Website** Parking Ticket Analysis

### **Introduction:**

#### *Question of Interest*

In the course of this analysis, we will be exploring idea and motivation of traffic violations and parking tickets. Penalties of this type are one mechanism through which municipalities attempt to manage the behavior of its citizens. The concept behind these fines is that when citizens violate laws, a fine is levied in order to help discourage violating the rules.

In the idealized version of this system of municipalities encouraging legal behavior, traffic violations and parking tickets are given out in every scenario where an individual has violated the law, irrespective of other circumstantial considerations. This idealized conceptualization is ostensibly the most fair scenario.

However, guiding citizen behavior is not the only potential motivation for levying tickets and violations. These fines are also an important income stream that municipalities utilize in order to maintain a balanced budget. This dual functionality of tickets and violations is where some citizens express concerns.

The main form of this concern is the idea that there may be certain scenarios wherein the function of balancing a budget takes precedent over the functionality of guiding citizen behavior. Citizens who express this concern often remark that when it comes time for budgets to be balanced (e.g. -at the end of a month), citizens are more likely to be cited with parking tickets and traffic violations.

This concern can be assessed with data, but unfortunately that data is unrealistically difficult to obtain. To do this, one would need to have data on every time that a citizen is committing an illegal behavior (e.g. - parking illegally), know whether he/she was penalized for that behavior, and whether this act occurred at a time point where a concerned citizen would expect increased rates of fines being levied. In the absence of the ability to generate this ideal data, we will make use of the best available data to perform an analysis to approximate whether there is a difference in the amount of parking tickets given out based on the time of the month.

This analysis will take advantage of publicly available data from the city of Philadelphia, Pennsylvania. The data is made available through [opendataphilly.org](http://opendataphilly.org) (1,2), which is a catalog of open access data of many types from the Philadelphia area and is referenced at the end of this document. One of the major benefits of this data source is that it is quite large and therefore, for ease of conducting this preliminary analysis, we will utilize data just from a single year.

To analyze this data critically, we will begin with simple tests of association to assess the relationship between the number and total fines of tickets and the time of the month. Later, we will perform a regression in order to see if there is a relationship between tickets or fines and time of the month when controlling for additional variables, such as month of the year.

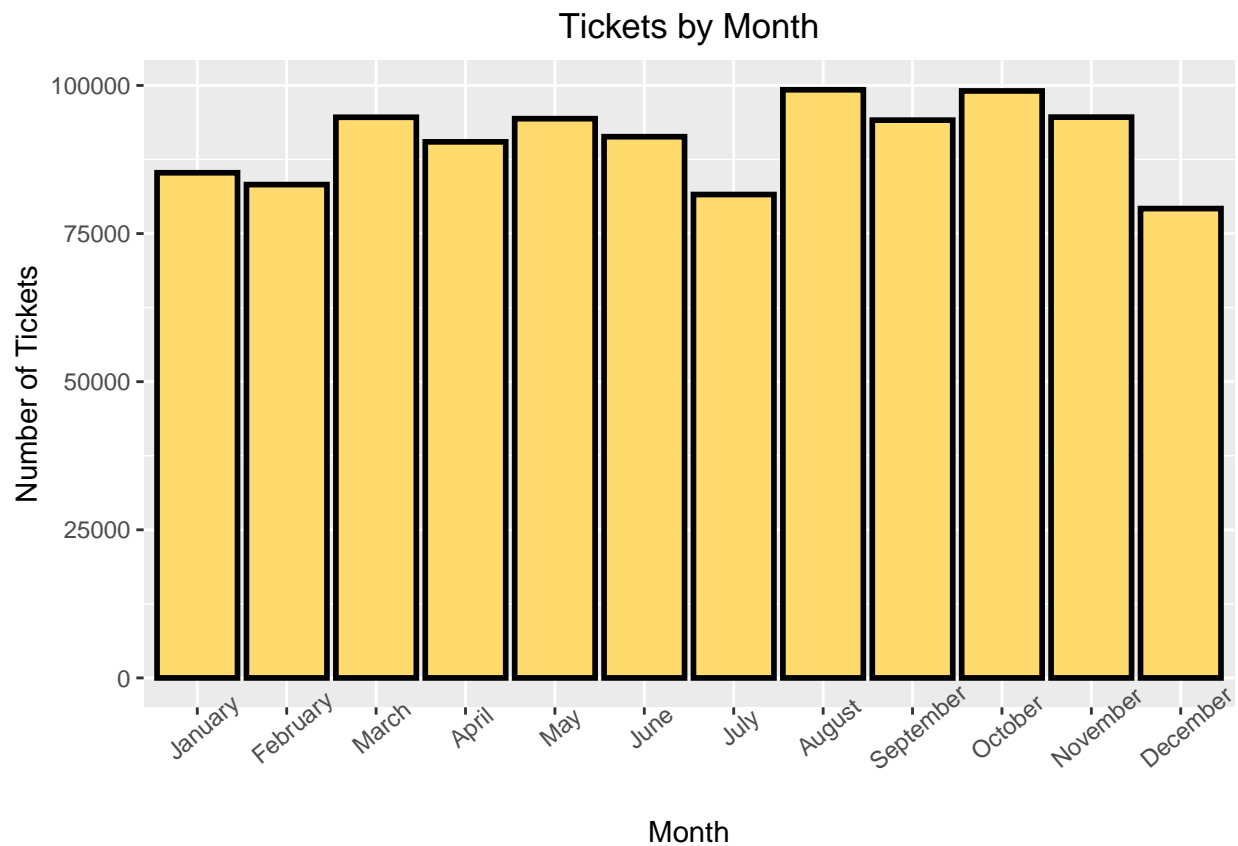
To make this analysis feasible, we will constrain the data to tickets dispensed in the first and last week of each month. This will allow for maximum contrast between the hypothetical financial "motivation" to dispense a ticket. Because we will be looking at the number of tickets and the total dollar amount of tickets

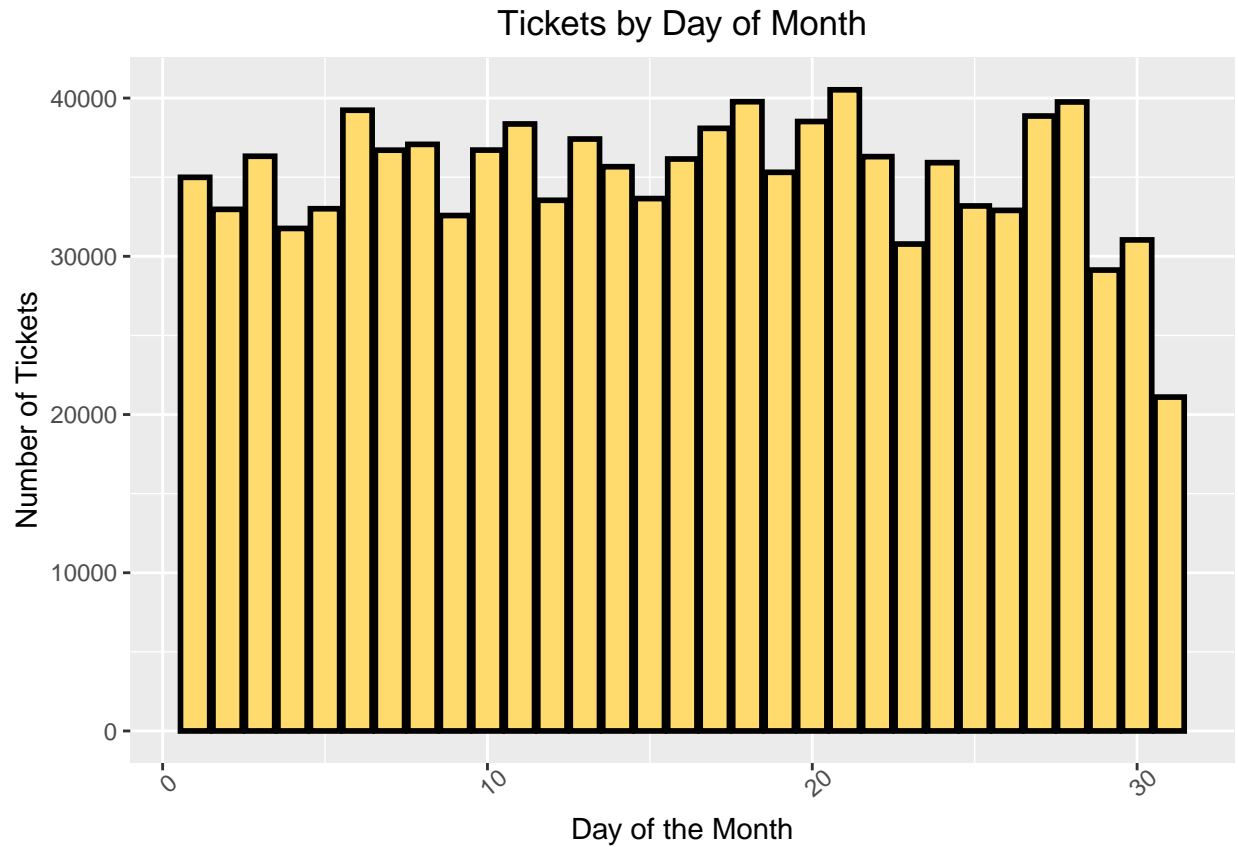
given out, we will utilize t-tests and linear regression. The data set is sufficiently large to utilize parametric statistics (3).

*Exploratory Data Analysis* Initial exploration of the data demonstrates that we have 7 columns the names of which are fairly self-explanatory. The format of the data is such that each row is a ticket and the columns are the information for that individual ticket. The columns of the unmodified data are:

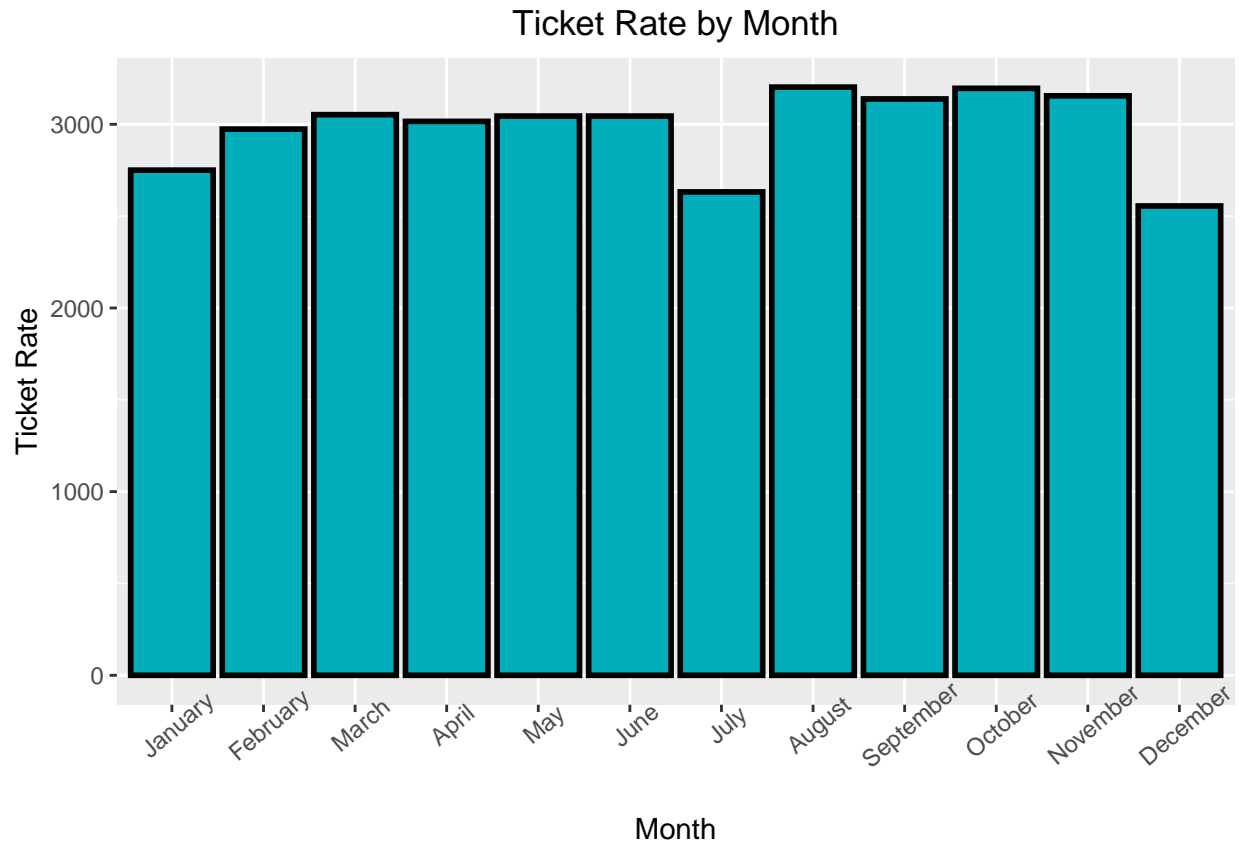
- 1) The type of violation for which the ticket was given
- 2) The date and time at which ticket was given
- 3) The dollar amount of the fine for the ticket
- 4) The agency that gave out the ticket
- 5) The geographic latitude of the violation
- 6) the geographic longitude of the violation
- 7) The zip code in which the infraction occurred

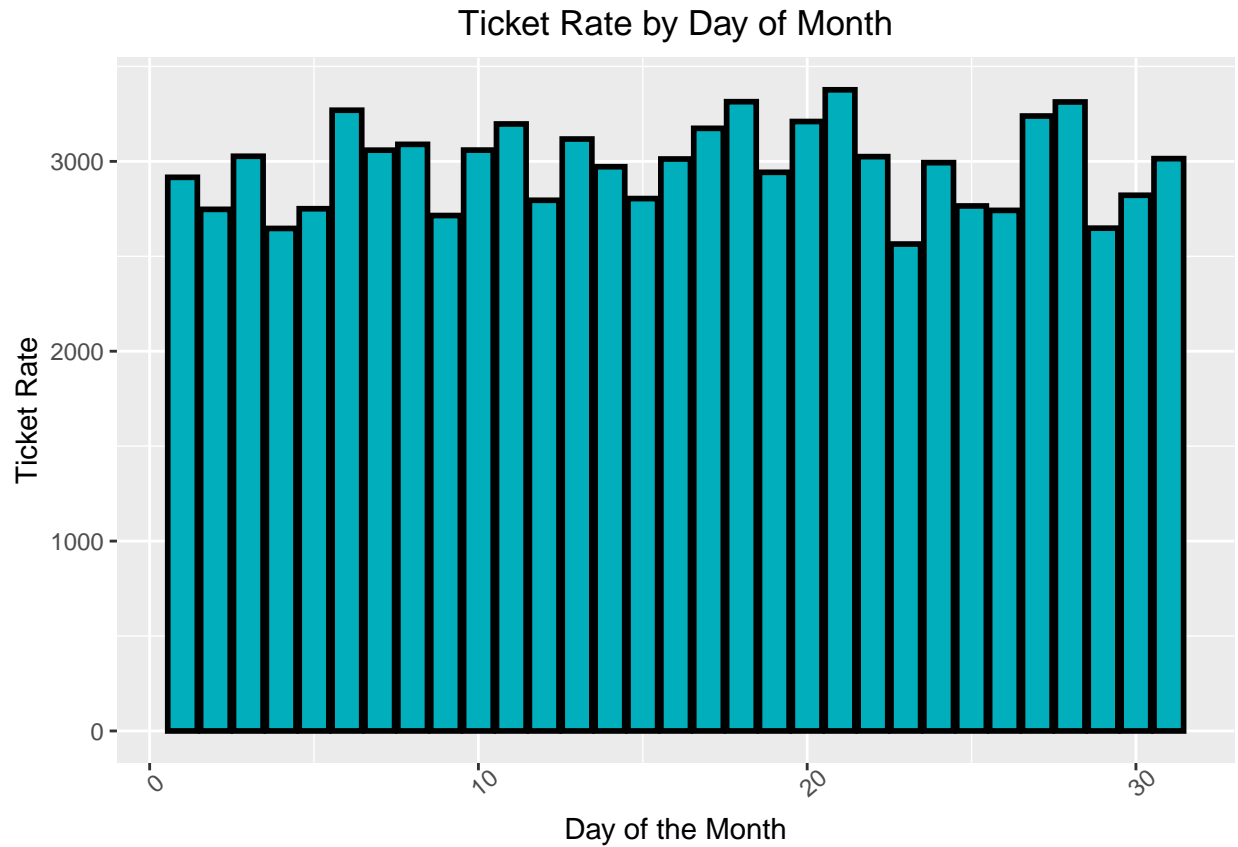
After inspecting the data for missing values/blanks, NAs, outlier data points, reassigning data types, managing dates, and transforming data where necessary, we have an appropriately cleaned data set that will allow us to begin to visualize the data.



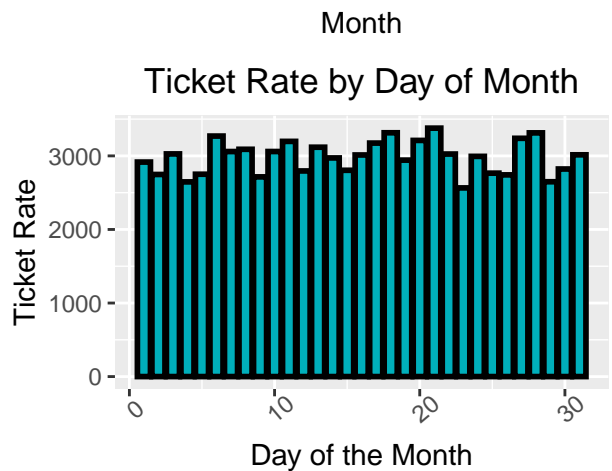
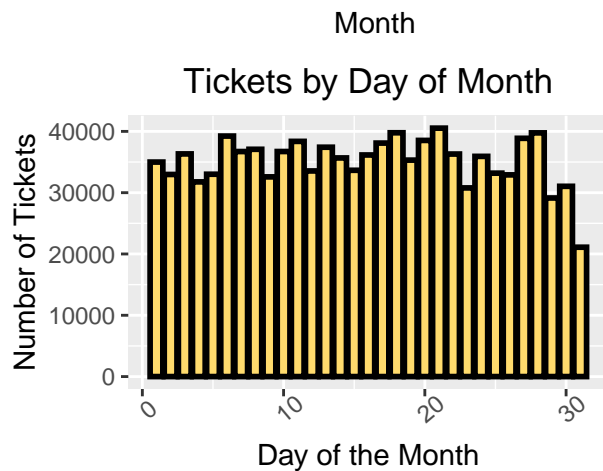
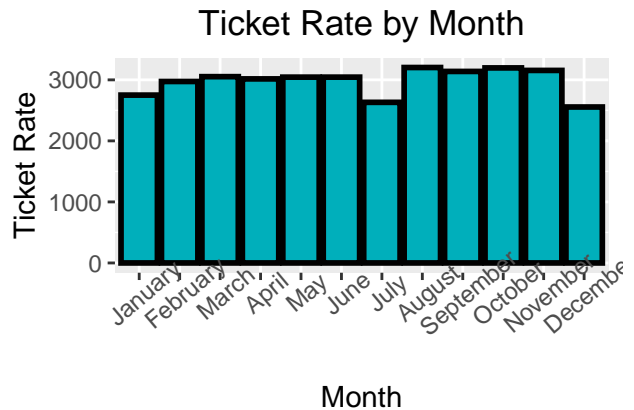
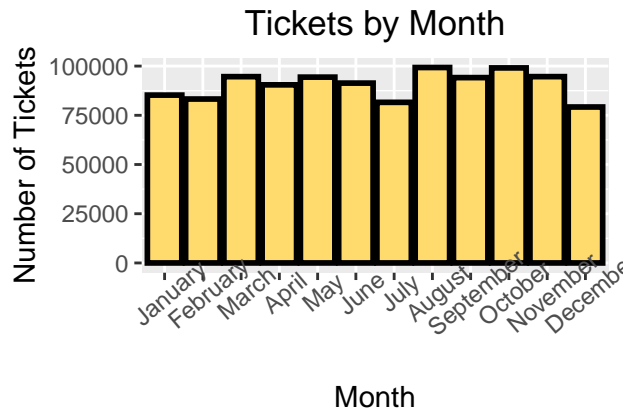


This initial visualization suggests that there are a fairly consistent number of tickets given out each month and at each day of the month. These are only preliminary representations, so these figures do not address the fact that there are different numbers of days in each month, or different representations of the different days of the month.



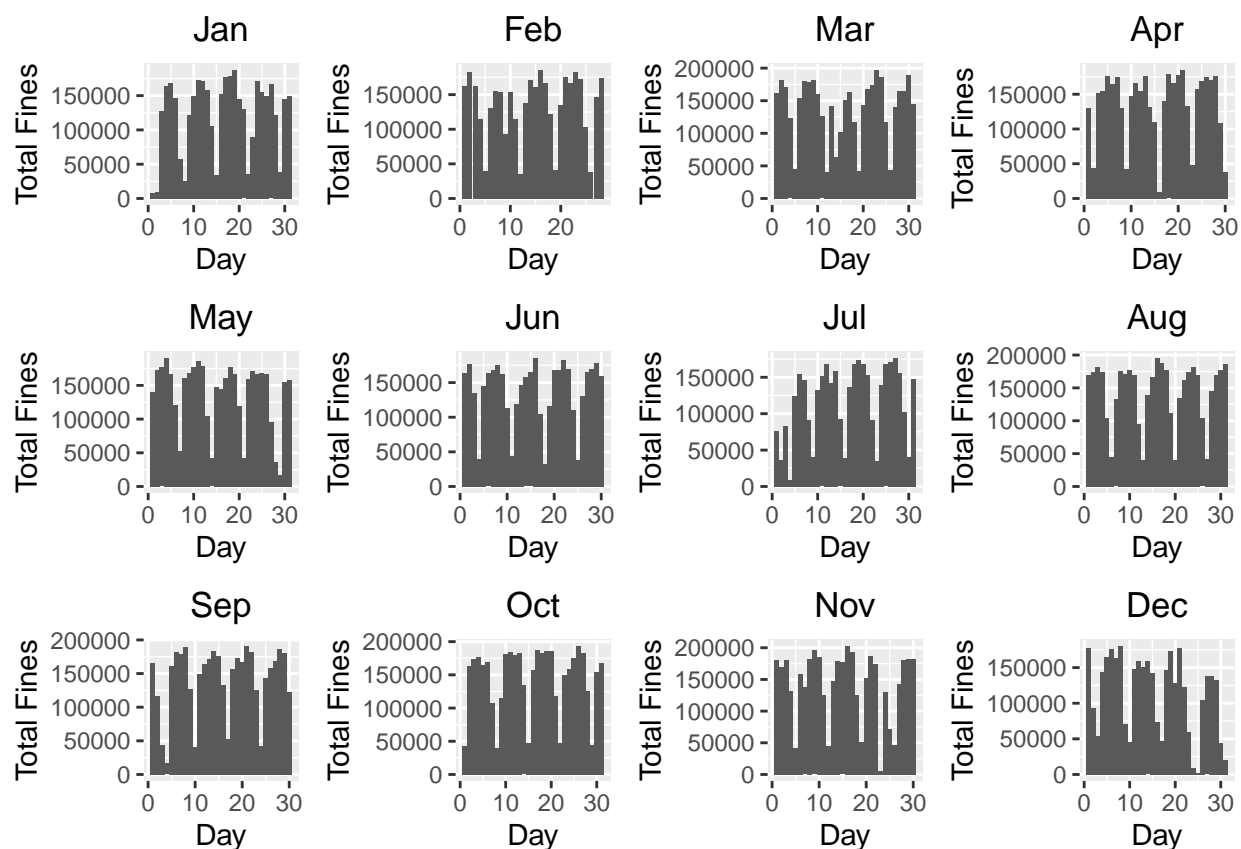


In order to account for the aforementioned differences, here, the data is presented in rates so they columns are therefore all scaled appropriately. With this adjustment, it still seems that there is no difference in how many tickets are being given out across the course of the year or the course of the month.



For visual comparison purposes, this paneled figure places the unadjusted figures alongside the adjusted figures and again we see that there does not appear to be a difference, visually.

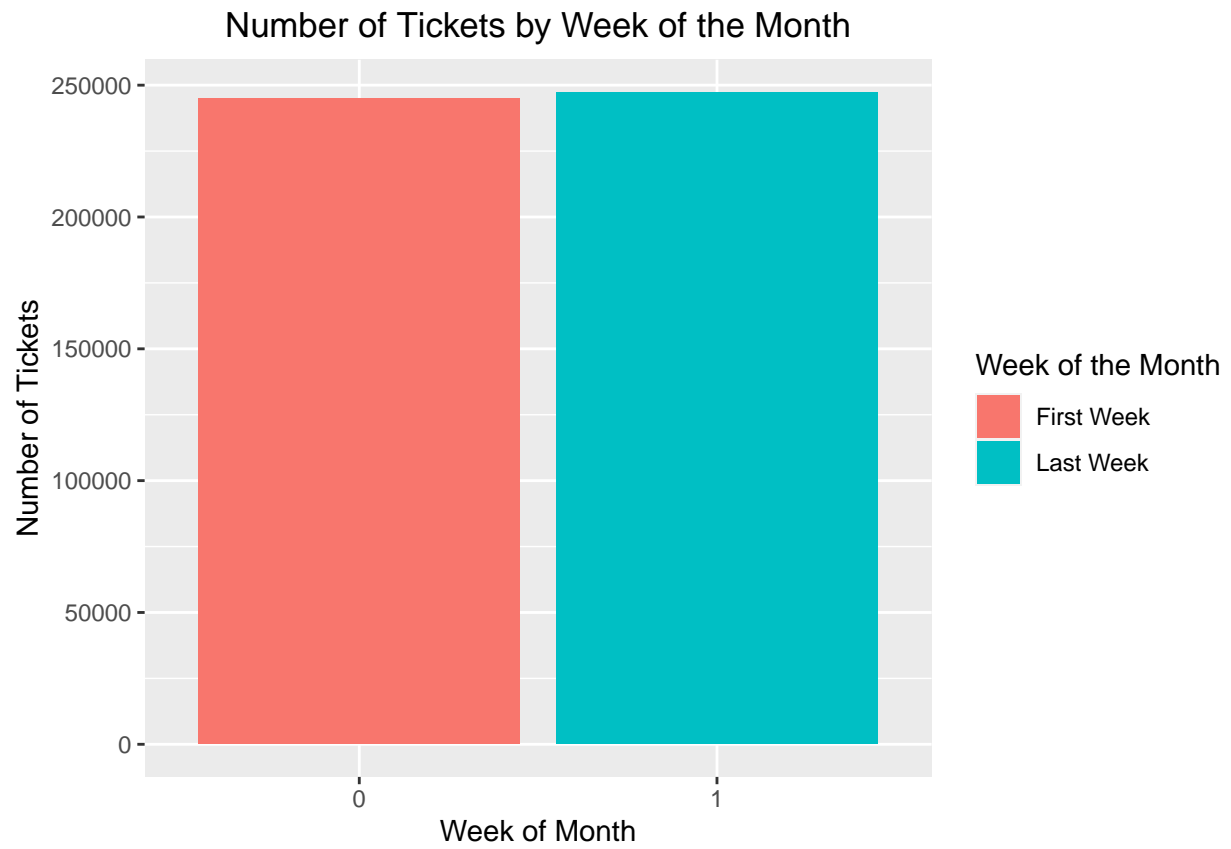
The prior data presented the number of tickets, but it made no exploration of the amount of dollars levied by the tickets. It is possible that in order to help balance the budget, municipalities are not increasing the number or rate of tickets being given out, but rather increasingly giving more expensive tickets. This would be one way to generate increased revenue around times to balance budgets.



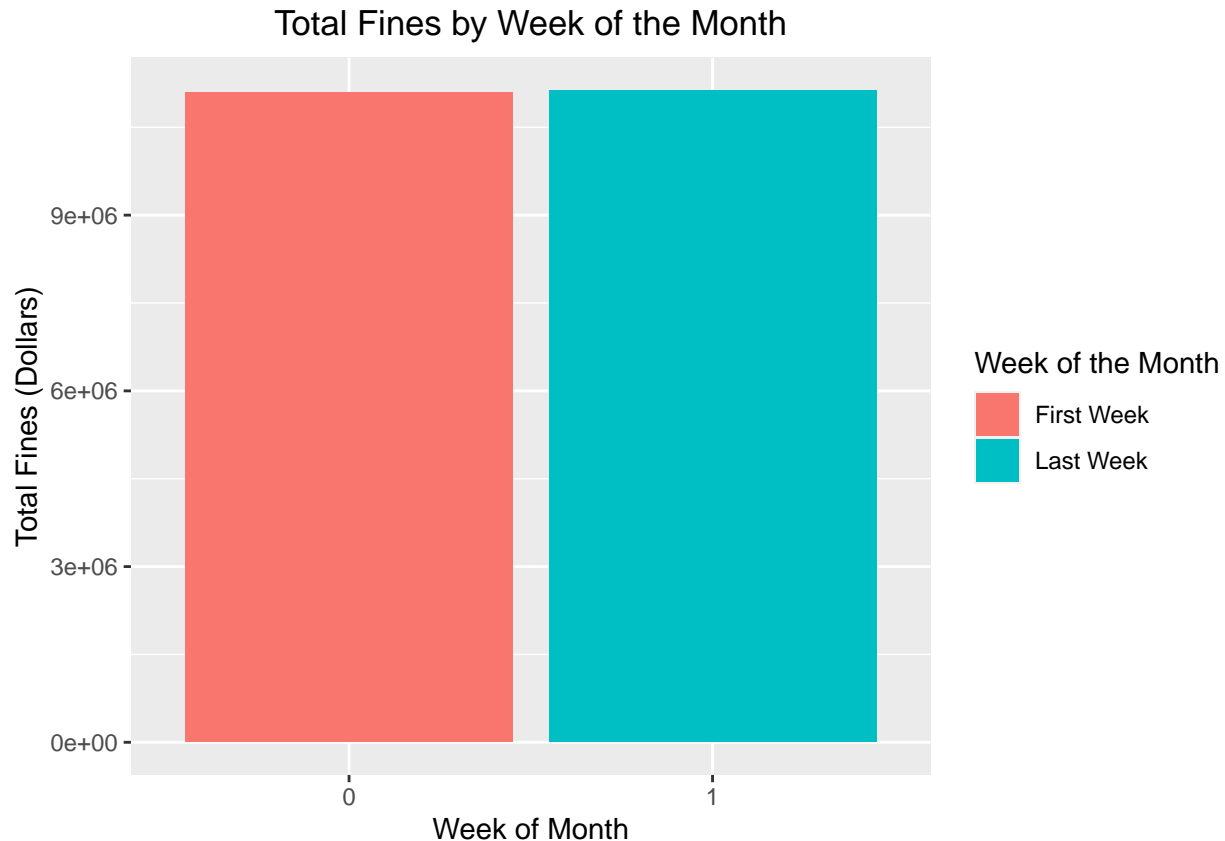
With this plot, we see that there does not seem to be a consistent increasing trend throughout the course of the month as one would suspect if municipalities were attempting to balance a budget with more expensive fines at the end of the month. We do see that there appear to be some trends along the course of an individual week, but we will not be exploring them here.

## RESULTS

The above visualizations were helpful to get a general lay of the data, but they do not specifically address the main analytic question: is there a different in the number of tickets or the dollars of fines dispensed in the first v. last month of the year. Those data can be seen below.







Here, we see that there is almost exact parity in the number of tickets given out in the first week and last week of the month. Similarly, the revenue generated in the first week appears nearly identical to the revenue generated in the last week of the month. This suggests that there is no difference between the two different time points. To go beyond visual inspection, we will quantify the relationship with statistical testing first in the form of a t-test.

Observation of the output from the ttest on the count data demonstrates the mean number of tickets in the first week of the month was 2916.5 compared to 2943.8 in the last week of the month. This indicated that on average, there were about 43 more tickets given in the last week of the month, compared to the first week of the month. However, this difference is far from significantly different with a p value of 0.88 with a range of possible values (i.e. -95% CI) from -385 tickets to 331 tickets. Therefore, in this unadjusted analysis, there is no difference in the number of tickets given out based on the month of the year.

With regard to the financial data, the daily mean dollar amount given out in the last week of the month was 132510.90 dollars compared to 132102.30 dollars in the first week of the month. This difference of 408.60 dollars was not significantly different with a p value of 0.95 and a 95% confidence interval from -16298.92 to 15481.90. Because this difference crosses a difference of 0 dollars, we conclude that there is no consistent difference in the dollar amount of tickets given out on the basis of the week of the month.

These statistical tests are consistent with our graphical representations which suggested that we would likely not find a difference in either the number of tickets or the total dollar amount of fines levied.

Now that we have assessed our outcomes of interest with bivariate statistics, we will move onto the regression analysis. In this linear regression, we will assess the number of tickets as a function of the week of the month (first or last) while controlling for the month of the year.

Running the linear regression for the number of tickets dispensed, we see that when controlling for the month of the year, switching from the first week of the month to the last week of the month only imparts

an expected difference of 27.3 tickets. More importantly, the p-value for this association is not significant at  $p=0.882$ .

It should be noted that when observing the distribution of the ticket data, it was appreciated that the data were left skewed and therefore, a cubic transformation was performed, which did reduce the skewness of the data. However, despite the transformation of the data, the effect of the week of the month was still not significant at  $p=0.879$ .

Similarly, when evaluating fines based on the week of the month, controlling for the month of the year, we see that there is a difference of 408 dollars. Again, this difference is not statistically significant  $p=0.96$ .

As with the ticket count data, a cubic transformation was performed on the fine data to address the leftward skew. Despite improvement in the visual distribution of the data, as assessed via histogram, there was no change in the linear regression with no significant effect of the primary predictor with  $p=0.987$ .

## Conclusion

In conclusion, during the course of this analysis, we explored the assertion that at different times, municipalities selective dispense tickets with financial motivation. Because it is not possible to directly answer this exact question, we analyzed proxy questions: 1) Do the number of tickets vary depending on the week of the month? and 2) Does the amount of money collected in fines vary depending on the week of the month? We first explored this visually with a collection of crude graphs and then with graphs adjusted for differences in the length of months and the occurrences of different days. This required considerable data manipulation and tidying in order to generate the appropriate data format. Visually, it did not appear that there was any trend wither across different months or within months. We explored further by directly graphing the counts of tickets and total fines levied in the first v. last weeks of months. Again, this appeared to not have a difference.

In order to provided statistical comparisons concordant with these graphs, t-tests were performed on both the ticket and fine data demonstrating no significant difference. These bivariate analyses do not adjust for other factors, such as the time of the year. That is to say that it is possible that there was some effect of the week of the month that was being hidden by the effect of different months of the year. In order to evaluate this possibility, linear regression on both the ticket number and fine data were run with the week of the month used as the primary predictor and controlling for differences in the month of the year. Again, these analyses demonstrated that there was no significant effect of week of the month on the number of tickets or total amount of fines. Even when cubic transformations were performed to address the leftward skew of the data, there was no difference in the result.

As best as possible with the available data, these graphical representations, bivariate statistics, and multi-variable statistics demonstrate that there does not appear to be any financial motivation to the way in which parking violations are dispensed in the city of Philadelphia in 2017.

Future analyses could include additional covariates in order to attempt to better control for additional confounders. Further, this analysis could be enhanced by the addition of data from other years to assess beyond just 2017. Finally, this question could be assayed in a different fashion by employing machine learning and attempting to see if with different algorithms, the machine can be trained to identify whether a ticket was given in the first or last week of the month. If these initial analyses are correct, I would expect that the model would not be able to distinguish a first week ticket from a last week ticket with any reliability.

## References

- 1) Data source: <https://www.opendataphilly.org/dataset/parking-violations>
- 2) Additional sources: <https://github.com/rfordatascience/tidytuesday/tree/master/data/2019/2019-12-03>
- 3) Lumley T, Diehr P, Emerson S, Chen L. The importance of the normality assumption in large public health data sets. *Annu Rev Public Health.* 2002;23:151-69. doi: 10.1146/annurev.publhealth.23.100901.140546. Epub 2001 Oct 25. PMID: 11910059.

## Appendix

```

#Install these packages if now already available for your version of Rstudio
##install.packages("tidyverse")
##install.packages("ggplot2")
##install.packages("lubridate")
##install.packages("gridExtra")
##install.packages("data.table")
##install.packages("dplyr")

#Loading relevant libraries
library(tidyverse)
library(ggplot2)
library(lubridate)
library(gridExtra)
library(data.table)
library(dplyr)
tickets <- readr::read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/02/data/tickets.csv")
#view(tickets)

#Understanding data types
data_types <- sapply(tickets, class)
data_types
#We see that the issuing_agency and zip_code variables are stored as character and numeric variables, respectively
tickets$issuing_agency <- as.factor(tickets$issuing_agency)
tickets$zip_code <- as.factor(tickets$zip_code)
#Now that we know the types of the data present and have ensured that they are correct, we will look for missing values
#Looking for NAs in each column
na_count <- colSums(is.na(tickets))
print(na_count)
sum(na_count)

#secondary check for incomplete cases
incomplete_cases <- sum(!complete.cases(tickets))
incomplete_cases

#ensuring consistent between the two methods
sum(na_count)==incomplete_cases

#Doing this, we see that the issuing_agency and zip_code columns contain NAs. In total, there are 173588 rows with missing values
tickets2 <- tickets[complete.cases(tickets),]
nrow(tickets2)
#Now we will quickly look for any blank cells.
#Looking for blanks in columns that are strings
sum(tickets2$violation_desc == "")
sum(tickets2$issuing_agency == "")

#Doing this, we see that there are no blank values in either of the character columns (i.e. - violation_desc and issuing_agency)
#Now that we have checked for blank values in violation_desc and confirmed that there are no blanks, we will convert violation_desc to a factor
tickets2$violation_desc <- as.factor(tickets2$violation_desc)
#Next, we will browse the numerical values and ensure that the numbers seem reasonable and that there are no missing values
date_range <- c(min(tickets2$issue_datetime),max(tickets2$issue_datetime))
date_range

```

```

fine_range <- c(min(tickets2$fine),max(tickets2$fine))
fine_range

lat_range <- c(min(tickets2$lat),max(tickets2$lat))
lat_range

lon_range <- c(min(tickets2$lon),max(tickets2$lon))
lon_range

zip_codes <- levels(tickets2$zip_code)
zip_codes

agencies <- levels(tickets2$issuing_agency)
agencies

#Observation of these data reveal that there does not appear to be any outlier or inappropriate values

#quickly clearing house of no longer necessary objects
rm(data_types,date_range, fine_range, lat_range, lon_range, zip_codes, agencies, incomplete_cases, na_c

#First, we will break down the current issuing date into component parts and assign a label for the day

#Extracting the components of the issue date including day of the week
tickets2 <- tickets2 %>% mutate(year = year(issue_datetime),
                                month = month(issue_datetime),
                                day = day(issue_datetime),
                                weekday = lubridate::wday(issue_datetime, label = TRUE),
                                months = months(issue_datetime))

#With the date information parsed into its component elements, we can now generate novel variables as t

#generating new variable for the total number of days in each month
tickets2 <- tickets2 %>% mutate(daysthismonth = case_when(month==1 ~ 31,
                                                           month==2 ~ 28,
                                                           month==3 ~ 31,
                                                           month==4 ~ 30,
                                                           month==5 ~ 31,
                                                           month==6 ~ 30,
                                                           month==7 ~ 31,
                                                           month==8 ~ 31,
                                                           month==9 ~ 30,
                                                           month==10 ~ 31,
                                                           month==11 ~ 30,
                                                           month==12 ~ 31))

#flagging tickets that occurred in the first and last week of the month
tickets2 <- tickets2 %>% mutate(last_week = ifelse(daysthismonth - day <=6, 1,0),
                                first_week = ifelse(day <=7,1,0))

#creating a last or first variable where last=1, first=0, and everything else =2
tickets2 <- tickets2 %>% mutate(week_of_month_tri = case_when(last_week==1 ~ 1,
                                                              first_week==1 ~ 0,
                                                              TRUE ~ 2))

#Now, the tickets that occurred in the first and last week of the months are flagged. This will be usef

```

*#Next, we will begin to visualize the data:*

```
tickets2$months <- factor(tickets2$months, levels=month.name)
```

```
month_plot <- tickets2 %>%  
  ggplot(aes(x=months)) +  
    geom_bar(fill="#FFDB6D", color="black", size=1) +  
    xlab("Month") +  
    ylab("Number of Tickets") +  
    ggtitle("Tickets by Month") +  
    theme(plot.title = element_text(hjust = 0.5),  
          axis.text.x = element_text(angle = 40))
```

```
daily_plot <- tickets2 %>%  
  ggplot(aes(x=day)) +  
    geom_bar(fill="#FFDB6D", color="black", size=1) +  
    xlab("Day of the Month") +  
    ylab("Number of Tickets") +  
    ggtitle("Tickets by Day of Month") +  
    theme(plot.title = element_text(hjust = 0.5),  
          axis.text.x = element_text(angle = 40))
```

```
#print(month_plot)
```

```
#print(daily_plot)
```

*#Reviewing these charts suggests that there does not appear to be an difference in the number of ticket*

*#calculating the adjusted monthly rate*

```
monthly <- tickets2 %>% group_by(months) %>%  
  summarize(count = n())
```

```
long_months <- c("January", "March", "May", "July", "August", "October", "December")  
short_months <- c("April", "June", "September", "November")
```

```
monthly <- monthly %>% mutate(rate = case_when(months %in% long_months ~ count/31,  
                                                months == "February" ~ count/28,  
                                                months %in% short_months ~ count/30))
```

```
adj_month_plot <- monthly %>%  
  ggplot(aes(x = months, y = rate)) +  
    geom_bar(fill="#00AFBB", color="black", size=1, stat="identity") +  
    xlab("Month") +  
    ylab("Ticket Rate") +  
    ggtitle("Ticket Rate by Month") +  
    theme(plot.title = element_text(hjust = 0.5),  
          axis.text.x = element_text(angle = 40))
```

*#calculating the daily rate*

```
daily <- tickets2 %>% group_by(day) %>%  
  summarize(count = n())
```

```

daily <- daily %>% mutate(rate = case_when(day <=28 ~ count/12,
      day %in% c(28:30) ~ count/11,
      day == 31 ~ count /7))

adj_daily_plot <- daily %>%
  ggplot(aes(x = day, y = rate)) +
  geom_bar(fill="#00AFBB", color="black", size=1, stat="identity") +
  xlab("Day of the Month") +
  ylab("Ticket Rate") +
  ggtitle("Ticket Rate by Day of Month") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 40))

#plotting
#print(adj_month_plot)
#print(adj_daily_plot)

#making paneled figure for side-by-side comparison
#grid.arrange(month_plot, adj_month_plot, daily_plot, adj_daily_plot, nrow=2)
#When looking at these plots side by side, there is no major change, but we do see that the month and d

#Just quickly cleaning house again for the sake of speed and clutter prevention
#rm(adj_daily_plot, adj_month_plot, daily, daily_plot, month_plot, monthly)
#It is possible that in order to help balance the budget, municipalities are not increasing the number

tickets2 <- tickets2 %>% arrange(month, day) %>%
  group_by(months) %>%
  mutate(fine_aux = ifelse(is.na(fine),0,fine),
        monthly_cumul_fine = cumsum(fine_aux))

tickets2 <- tickets2 %>% arrange(months, day) %>%
  group_by(months, day) %>%
  mutate(daily_cumul_fine = cumsum(fine_aux))

jan_plot <- tickets2 %>% filter(months == "January") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y= day_total)) +
  geom_col() +
  ggtitle("Jan") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

feb_plot <- tickets2 %>% filter(months == "February") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y= day_total)) +
  geom_col() +
  ggtitle("Feb") +
  xlab("Day") +
  ylab("Total Fines") +

```

```

  theme(plot.title = element_text(hjust = 0.5))

mar_plot <- tickets2 %>% filter(months == "March") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Mar") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

apr_plot <- tickets2 %>% filter(months == "April") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Apr") +
  xlab("Day") +
  ylab("Total Fines ") +
  theme(plot.title = element_text(hjust = 0.5))

may_plot <- tickets2 %>% filter(months == "May") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("May") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

june_plot <- tickets2 %>% filter(months == "June") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Jun") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

july_plot <- tickets2 %>% filter(months == "July") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Jul") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

aug_plot <- tickets2 %>% filter(months == "August") %>%

```

```

group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Aug") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

sep_plot <- tickets2 %>% filter(months == "September") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Sep") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

oct_plot <- tickets2 %>% filter(months == "October") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Oct") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

nov_plot <- tickets2 %>% filter(months == "November") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Nov") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

dec_plot <- tickets2 %>% filter(months == "December") %>%
  group_by(day) %>%
  summarize(day_total = max(daily_cumul_fine)) %>%
  ggplot(aes(x = day, y = day_total)) +
  geom_col() +
  ggtitle("Dec") +
  xlab("Day") +
  ylab("Total Fines") +
  theme(plot.title = element_text(hjust = 0.5))

#grid.arrange(jan_plot, feb_plot, mar_plot, apr_plot, may_plot, june_plot, july_plot, aug_plot, sep_plot,
#leaving code for individual options because the paneled figure can be hard to see
#print(jan_plot)

```



```

#print(feb_plot)
#print(mar_plot)
#print(apr_plot)
#print(may_plot)
#print(june_plot)
#print(july_plot)
#print(aug_plot)
#print(sep_plot)
#print(oct_plot)
#print(nov_plot)
#print(dec_plot)
#These data can be a little difficult to visualize in the paneled figure (if you are having alot of trouble)

#removing the plots from above for tidyness
rm(jan_plot,feb_plot,mar_plot,apr_plot, may_plot, june_plot, july_plot, aug_plot, sep_plot, oct_plot,nov_plot,dec_plot)
#To finalize our visualization of the data, we will represent the number and value of tickets given in

by_week <- tickets2 %>%
  filter(week_of_month_tri != 2) %>% #filtering out the tickets that don't occur in the first or last week of the month
  group_by(week_of_month_tri) %>%
  summarize(count = n(),
            total_fine = sum(fine))

tickets_by_week_plot <-
  by_week %>% ggplot(aes(x= as.factor(week_of_month_tri), y = count, fill=as.factor(week_of_month_tri))) +
  geom_col() +
  xlab("Week of Month") +
  ylab("Number of Tickets") +
  ggtitle("Number of Tickets by Week of the Month") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_discrete(labels=c("First Week", "Last Week")) +
  labs(fill="Week of the Month")

fines_by_week_plot <-
  by_week %>% ggplot(aes(x= as.factor(week_of_month_tri), y = total_fine, fill=as.factor(week_of_month_tri))) +
  xlab("Week of Month") +
  ylab("Total Fines (Dollars)") +
  ggtitle("Total Fines by Week of the Month") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_discrete(labels=c("First Week", "Last Week")) +
  labs(fill="Week of the Month")

#print(tickets_by_week_plot)
#print(fines_by_week_plot)

#Consistent with our prior visualizations, there does not appear to be a difference in the number of tickets given in each week of the month
#bringing in all of the plots that have been silenced above
print(month_plot)
print(daily_plot)
print(adj_month_plot)
print(adj_daily_plot)
grid.arrange(month_plot, adj_month_plot, daily_plot, adj_daily_plot, nrow=2)

```

```

grid.arrange(jan_plot,feb_plot,mar_plot,apr_plot, may_plot, june_plot, july_plot, aug_plot, sep_plot, oct_plot)
print(tickets_by_week_plot)
print(fines_by_week_plot)

#isolating down to cases in the first or last week
#and generating a label variable while informs how many days from the beginning of the month, or end of the month
days <- tickets2 %>% mutate(days_from_end = day - daysthismonth) %>%
  filter(days_from_end %in% c(-6:0) | day %in% c(1:7)) %>% select(-fine_aux,
                                                                -monthly_cumul_fine,
                                                                -daily_cumul_fine) %>%

  mutate(first_last = case_when(first_week ==1 ~ 0,
                                last_week==1 ~ 1))

days <- days %>% mutate(day_label = days_from_end)
days <- days %>% mutate(day_label = case_when(day_label %in% c(-6:0)~ day_label,
                                              day_label %in% c(-30:-21) ~ daysthismonth + days_from_end,
                                              TRUE ~ 0))

#For the current analysis, we have organized the data by month and then by the number of days from the beginning of the month

wip <- days %>% group_by(months, day_label) %>%
  summarize(count = n(), total_fine = sum(fine), avg_fine = total_fine/count)

wip <- wip %>% mutate(first_last = case_when(day_label %in% c(-6:0) ~ 1,
                                              day_label %in% c(1:7) ~0))

#ttest for the count data
ttest_count <- t.test(wip$count ~ wip$first_last, var.equal=TRUE)

ttest_count$p.value
ttest_count$conf.int
ttest_count$estimate

#ttest for the fine data
ttest_fines <- t.test(wip$total_fine ~ wip$first_last, var.equal=TRUE)

ttest_fines$p.value
ttest_fines$conf.int
ttest_fines$estimate

count_lm <- lm(count ~ as.factor(first_last) + months, data = wip)
summary(count_lm)

#cubing the count of tickets as an attempt to address the left-skewed nature of the date)
wip <- wip %>% mutate(count_cube = count^3)
hist(wip$count_cube)
count3_lm <- lm(count_cube ~ as.factor(first_last) + months, data = wip)
summary(count3_lm)

#Linear reg with fine data
fines_lm <- lm(total_fine ~ as.factor(first_last) + months, data = wip)
summary(fines_lm)

```

```
#cubing fine data and doing lm on the cubed data
wip <- wip %>% mutate(fine_cube = total_fine^3)
hist(wip$fine_cube)
fines3_lm <- lm(fine_cube ~ as.factor(first_last) + months, data = wip)
summary(fines3_lm)
hist(wip$count)
hist(wip$count_cube)
```