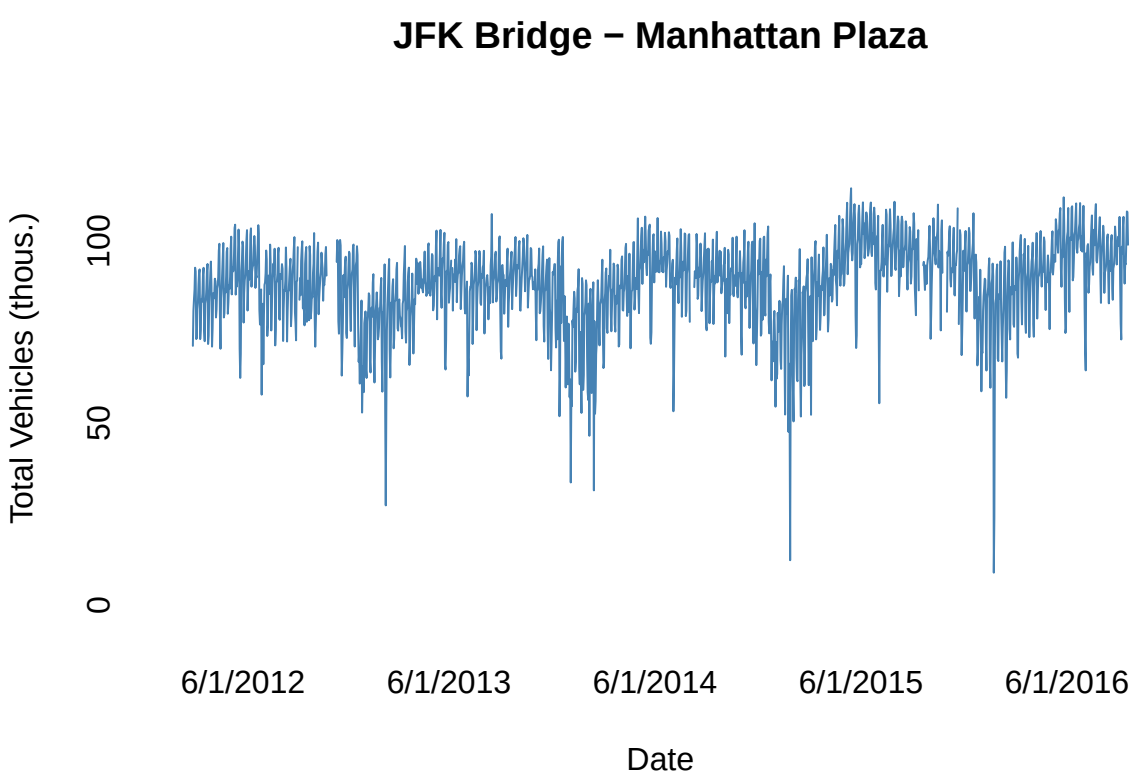
Seasonal Trends in New York City Traffic: STL Part I

Dillon R. Gardner

September 29, 2016



Everyone knows that traffic is much worse during the week than Sundays. But how much so? And how much does traffic change from month to month? Does traffic decrease in the summer because people take vacation... or do those vacationers clog the roadways? Given all of these fluctuations in traffic, how do we quantify long term trends?

In data science, these questions fall are referred to as *seasonality*. A wonderful technique to address these questions is Seasonal and Trend decomposition using LOESS (STL). This the first of a three part series on STL. In this post, we'll look at a test case of traffic from the John F. Kennedy Bridge in Manhattan. Part II delves into the weeds of how this works, and Part III discusses how STL can be used for imputing data over missing values, a key advantage over other means of decomposition.

The dataset we have is of total daily tolls across the JFK bridge. The data were scrapped from the MTA. All of the data aggregation and analysis including this post in .Rmd form is available on GitHub. The cleaned data is a time series of daily data from March 2012 through September 2016.

```
head(jfkManhattan)
```

```
## Source: local data frame [6 x 2]
##
##         Date  Total
##       (date)  (dbl)
## 1 2012-03-04 71.236
## 2 2012-03-05 80.855
```

```
## 3 2012-03-06 84.055
## 4 2012-03-07 86.691
## 5 2012-03-08 92.716
## 6 2012-03-09 91.656
```

From the plot at the top of the post, there is a clear yearly pattern. Traffic peaks in the summer and drops down, bottoming out around the January or February. On top of this pattern, there is a general increase in traffic over time.

To quantify this behavior the monthly trend we aggregate the data from daily totals to the daily average for each month.

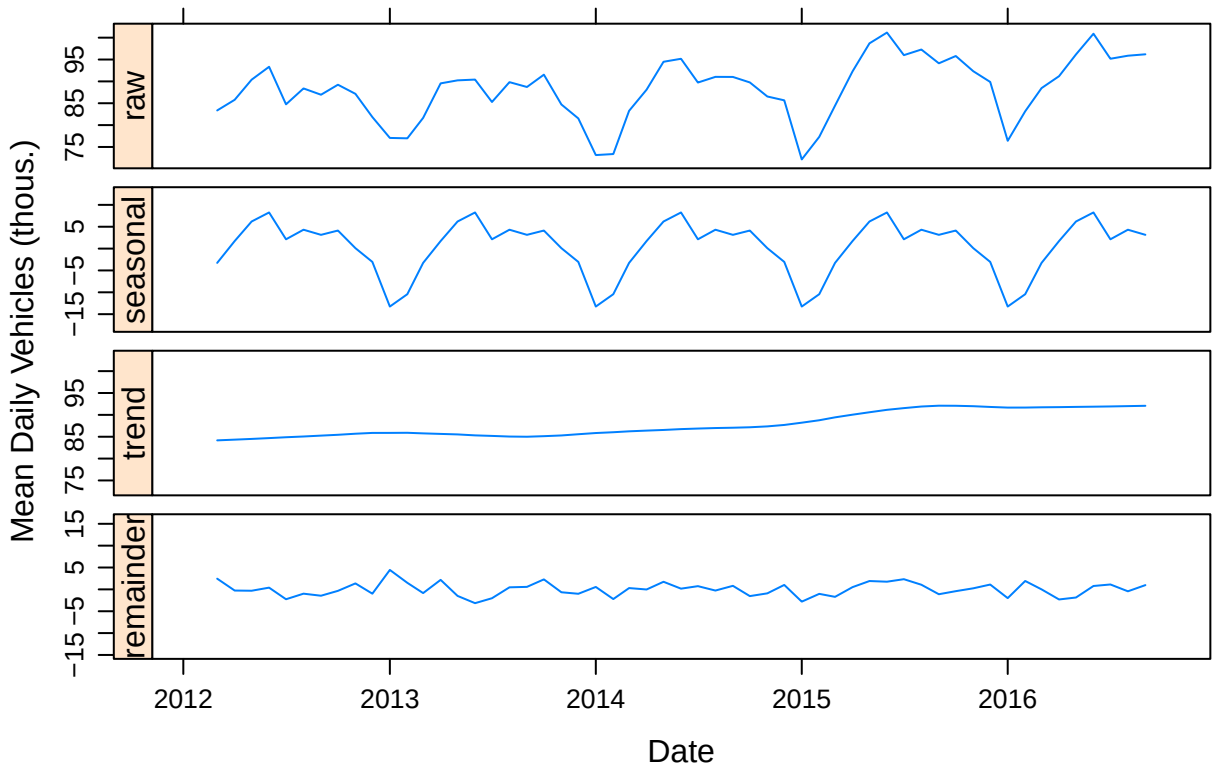
```
library(dplyr)
monthlyData <- jfkManhattan
day(monthlyData$Date) <- 1
monthlyData <- monthlyData %>%
  group_by(Date) %>%
  summarise_each(funs(mean(., na.rm=T))) %>%
  select(Date, DailyAverage=Total)
```

The decomposition takes the average daily traffic for each month as the sum of three components: a seasonal component, a trend component, and the remainder.

$$MonthlyTraffic_t = Seasonal_t + Trend_t + Remainder_t$$

STL is a particular algorithm to make this separation. The `stl` function exists in base R, but the `stlplus` package implements the same algorithm while allowing for missing data. It also has some nicer plotting features. The details of the algorithm are address in PartII. But the important parameter is `n.p`, which is the number of measurements in a full period of seasonal behavior. Since our data is now monthly and we anticipate yearly seasonality, `n.p = 12`

```
library(stlplus)
monthNames <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")

stlMonthly <- stlplus(monthlyData$DailyAverage, t=monthlyData$Date,
                      n.p=12, s.window="periodic",
                      sub.start=3, sub.labels = monthNames)

plot(stlMonthly, xlab="Date", ylab="Mean Daily Vehicles (thous.)")
```

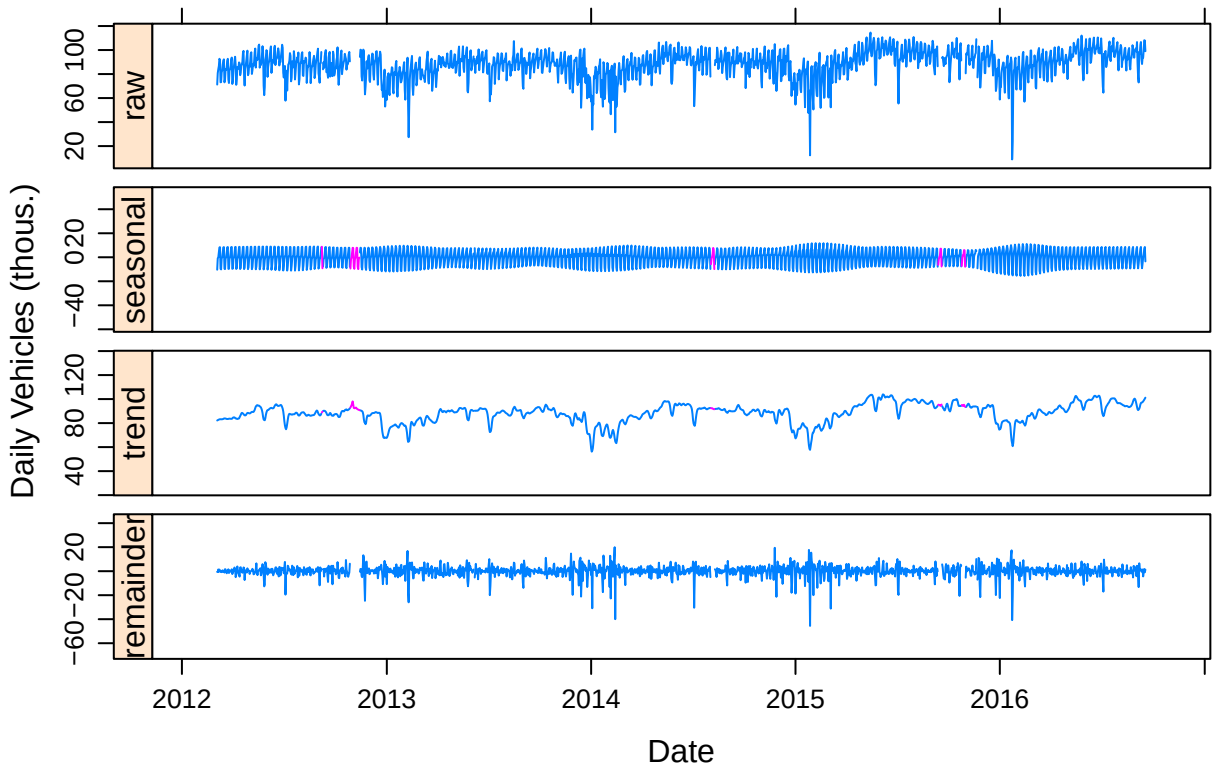


The top plot is the raw monthly aggregated data. Once the seasonal and trend components are fit, they are subtracted from the raw data to give the remainder. The result quantifies exactly what our eye told us. The average daily usage for each month follows a yearly pattern visible in the "seasonal" component. The decomposition is done such that this component averages out to nearly zero over each period. This makes the trend interpretable as the average value over time, excluding the periodic fluctuations. Overall, the trend line shows that traffic has increased by almost 10 percent over the past four years. The remainder is whatever short fluctuations are not captured in the other two components. At a monthly level, this terms fairly flat with no clear anomalous months.

Of course, there is a potential problem with the analysis so far. By aggregating data to monthly, we ignored the fact that there are also variations in traffic over the course of the week. Since weeks do not evenly fit into months, some months will have 5 Mondays and 4 Sundays. Other months, will have the opposite. This weekly variation will therefore affect the measured monthly average. This is exacerbated by the fact that the dataset is not complete and is missing some days. This sort of problem is endemic in trying to extract seasonal behavior from time series data. The lack of consistency of days and weeks in months, days in years, indivisibility of weeks into years, changing dates of holidays

We can address this problem by first performing STL on the daily data to extract the weekly seasonality. Now that it is daily data and the seasonality is seven days, we set `n.p=7`.

```
weekDays <- c("Sun", "Mon", "Tues","Wed", "Thur", "Fri", "Sat")

stlDaily <- stlplus(jfkManhattan$Total,t=jfkManhattan$Date,
                    n.p=7, s.window=25,
                    sub.labels=weekDays, sub.start=1)

plot(stlDaily, xlab="Date", ylab="Daily Vehicles (thous.)")
```



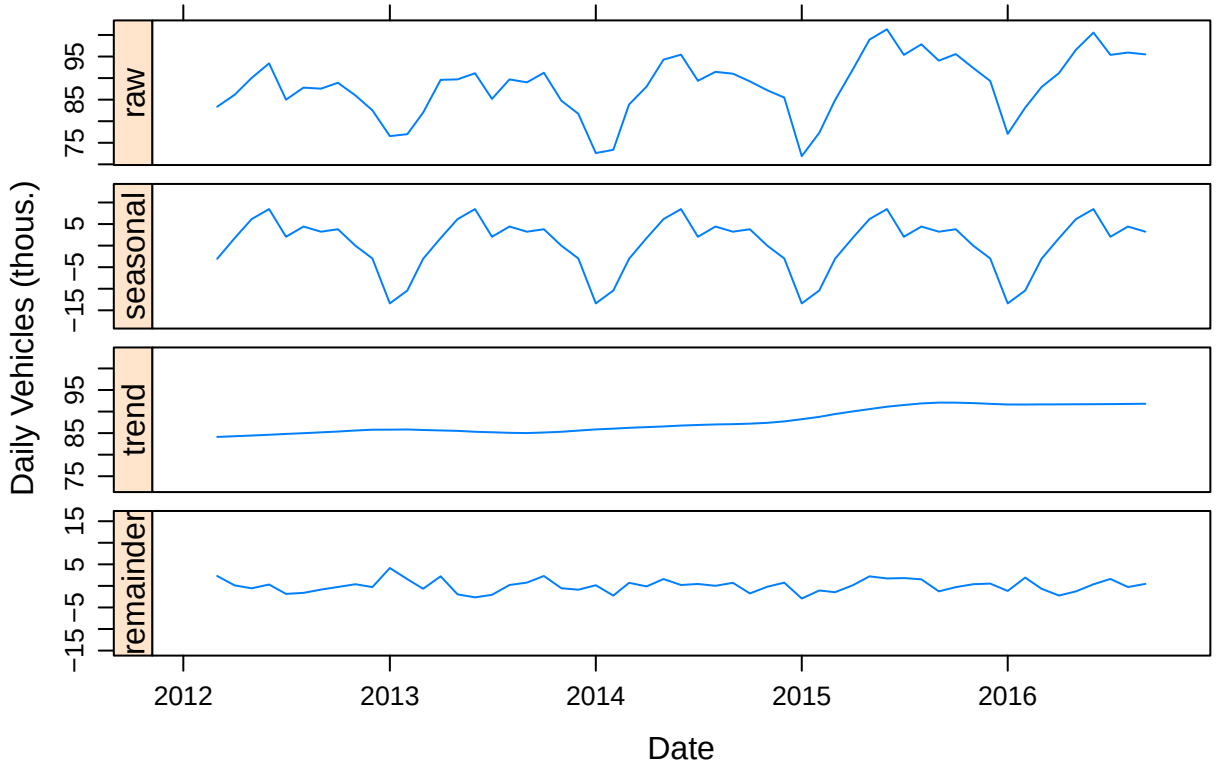
Unsurprisingly, the daily variation is quite strong. Saturdays and Sundays have the least traffic (This decomposition shows a couple new aspects of STL.

1. Even though the raw data has some missing data, the seasonal and trend components are complete. The pink data are the imputations. How this is accomplished will be discussed in the later posts in this series.
2. The seasonal component is not completely periodic every seven days. There is additional structure to the seasonal trend. The difference between weekday and weekend traffic is smaller in the summer than in the winter. The ability to capture slow changes in the seasonal terms is a key advatage of STL over other decomposition methods such as `decompose` in R and `seasonal_decompose` in Python.
3. The remainder term now has a lot more strucutre. Every year around the winter holidays, there are a series of big negative spikes.
4. The trend component now exhibits the year long oscillation we had previously described as seasonal. This is a key point about any seasonal decomposition: the division of the data into these three components is, to a degree, a matter of opinion. Changing model parameters and time scales shifts activity from one component to another. There is no "true" decomposition.

Next step, we subtract the seasonal component from the raw data. Because the seasonal component averages to (approximately) zero, this does not change the total traffic over the data set. Now, the value for each day is interpreted as how many vehicles passed adjusted for the day of the week. This should allow a Sunday to be compared to Monday. The higher value means that there was more traffic compared to a typical day. This also solves the problem of weekly variation affecting monthly aggregation.

```
normalizedData <- jfkManhattan
normalizedData$Total <- normalizedData$Total - stlDaily$data$seasonal
day(normalizedData$Date) <- 1
normalizedData <- normalizedData %>%
  group_by(Date) %>%
  summarise_each(funs(mean(., na.rm=T)))
```

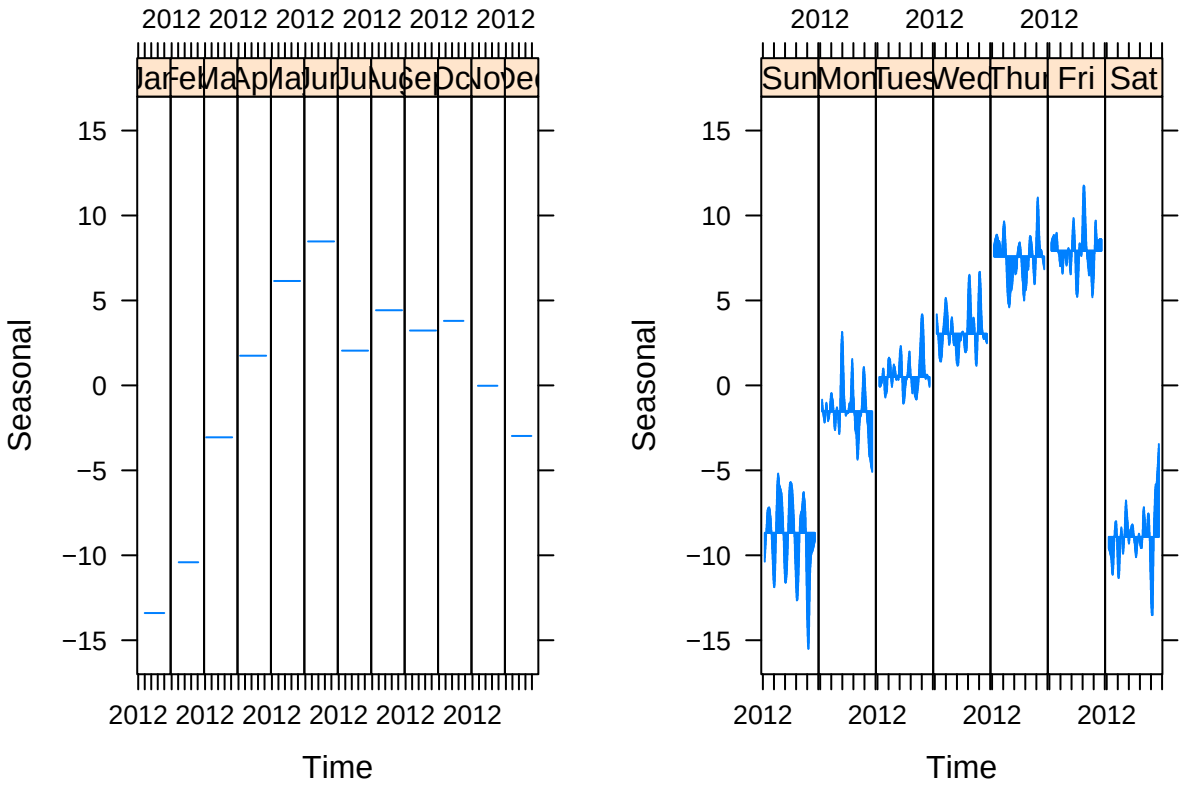
```
stlNormalizedMonthly <- stlplus(normalizedData$Total, t=normalizedData$Date,
                                n.p=12, s.window="periodic",
                                sub.start=3, sub.labels = monthNames)

plot(stlNormalizedMonthly, xlab="Date", ylab="Daily Vehicles (thous.)")
```



The result hasn't changed much. The sum of the absolute value of the remainder is slightly smaller, which means more of the data is captured in the seasonal and trend terms. But more importantly, this excludes a possible error in our analysis making the conclusions more robust.

A insightful way of looking at the decomposition, is plotting the seasonal component by the subseries. A subseries is the set of observations of one season type. For example, all of the Mondays, or all of the Januaries. The plot for Mondays is a time series of the seasonal component across the whole time range of the data set. These plots provide a clear picture of how the seasonality changes over time.

```
library(gridExtra)
p1 <- plot_cycle(stlNormalizedMonthly, ylim=c(-17, 17))
p2 <- plot_cycle(stlDaily, ylim=c(-17, 17))
grid.arrange(p1,p2, ncol=2)
```



Conclusions

This has shown how STL can be used to assess the seasonal components of a time series. Using this technique on the traffic on the JFK bridges shows that, nsurprisingly, traffic has a strong weekly seasonality. This approach allows us to quantify that the difference between a typical Sunday and a typical Thursday or Friday is 15-18 thousand vehicles, which about 15-20% of the total traffic. More surprising, the difference between a typical day in January and a day in June is about 23 thousand vehicles. The STL trend component shows that traffic has increased from an average of 84 thousand vehicles to to 92 thousand vehicles (about 10%) since 2012.