Data Science for Supply Chain

Moving Average

The average demand during the last n periods.

$$f_n = \frac{1}{n}\sum_{i=1}^{n} d_{t-i}$$

Where:

- n is the number of periods we take the average of
- d_t the demand we observe during period t
- f_t is the forecast we made for period t

The first forecast will be done for $t = n + 1$

In scientific literature, you will find often see the output you want to predict noted y

A prediction would be noted as $\hat{y}$

When we want to point to a specific occurrence of the forcast at time t, we will not it f_t or d_t

Demand observation: we will call the demand of each period.

Noise: an unexplained variation in the data. It is often due to the randomness of the different processes at hand.

References

- Moving Average:

```
dmd <- tibble::tribble(
              ~Date, ~Demand,
         "2010-1-1",      37,
         "2011-1-1",      60,
         "2012-1-1",      85,
         "2013-1-1",     112,
         "2014-1-1",     132,
         "2015-1-1",     145,
         "2016-1-1",     179,
         "2017-1-1",     198,
         "2018-1-1",     212,
         "2019-1-1",     232,
         "2020-1-1",      NA,
         "2021-1-1",      NA,
         "2022-1-1",      NA
         )



dmd$Date <- as.Date(dmd$Date)
```

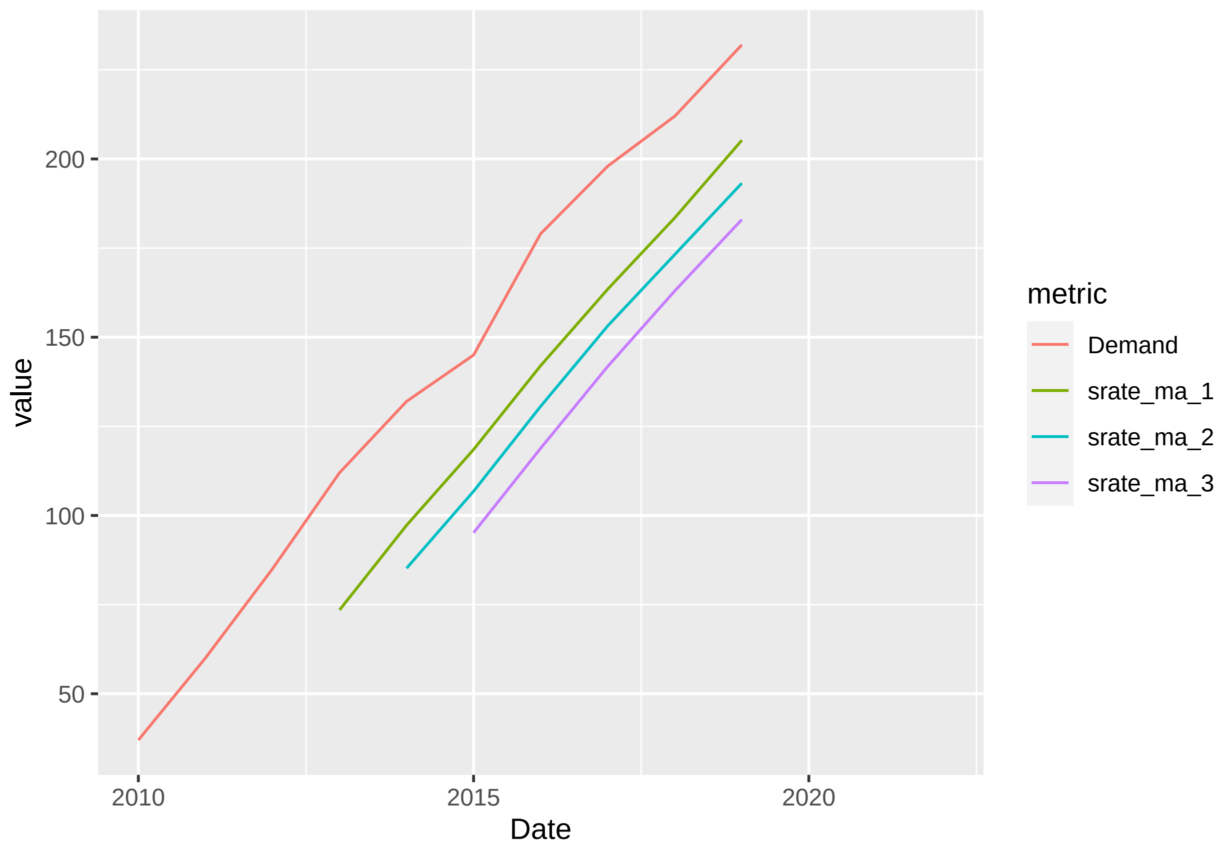
```r
demand <- dmd %>% mutate(srate_ma_1 = rollmean(Demand, k = 4, fill = NA, align = "right"),
                         srate_ma_2 = rollmean(Demand, k = 5, fill = NA, align = "right"),
                          srate_ma_3 = rollmean(Demand, k = 6, fill = NA, align = "right"))

demand
```

```
## # A tibble: 13 x 5
##    Date       Demand srate_ma_1 srate_ma_2 srate_ma_3
##    <date>      <dbl>      <dbl>      <dbl>      <dbl>
##  1 2010-01-01     37       NA         NA         NA
##  2 2011-01-01     60       NA         NA         NA
##  3 2012-01-01     85       NA         NA         NA
##  4 2013-01-01    112       73.5       NA         NA
##  5 2014-01-01    132       97.2       85.2       NA
##  6 2015-01-01    145      118.       107.        95.2
##  7 2016-01-01    179      142        131.       119.
##  8 2017-01-01    198      164.       153.       142.
##  9 2018-01-01    212      184.       173.       163
## 10 2019-01-01    232      205.       193.       183
## 11 2020-01-01     NA       NA         NA         NA
## 12 2021-01-01     NA       NA         NA         NA
## 13 2022-01-01     NA       NA         NA         NA
```

```r
demand %>% gather(metric, value, Demand:srate_ma_3) %>%
  ggplot(aes(Date, value, color = metric)) +
  geom_line()
```

```
## Warning: Removed 24 row(s) containing missing values (geom_path).
```



This is by far the simplest way to generate a trending forecast based on the information. This really hends up giving common trend but does not take seasonality into consideration and will thus understate/overstate when there are significant changes over a period of time.

One example of this is seasonal sales. Generally after Black Friday (Holiday in the United States similar to singles day in Asia), sales will dramatically increase due to the Christmas season.