Converting Periodicity

In financial series it is common to find Open-High-Low-Close data (or OHLC) calculated over some repeating and regular interval.Also known as range bars, aggregating a series based on some regular window can make analysis easier amongst series that have varying frequencies. A weekly economic series and a daily stock series can be compared more easily if the daily is converted to weekly. The XTS package has `to.period()` function to do it easily. For example no we will convert a time series object from daily to weekly format with and without OHLC functions.

We need to import the libraries first,

```
library(xts)
library(zoo)
```

Now we will import the dataset and make it a XTS object..

```
> df = read.csv("F:/df.csv",header = TRUE,
stringsAsFactors = FALSE)
> date <- seq(as.Date("2021-03-01"), length = 396, by =
"days")
> data = df$Value
# transform the data frame into time series object
> df = xts(x=data,order.by=date)
```

As it’s a daily time series. Now we will convert it to a weekly format with and without OHLC functions. With OHLC it will look like this,

```
> df_week = to.period(df,period = "weeks")
> head(df_week)
           df.Open df.High df.Low df.Close
2021-03-07 0 0 0 0 2021-03-14 0 0 0 0
```

```
2021-03-21  0  2  0  2  2021-03-28  1  2  0  0
2021-04-04 0 3 0 0 2021-04-11 2 4 0 0
```

And without OHLC it will be like that,

```
> df_week_2 = to.period(df,period = "weeks",OHLC =
FALSE)
> head(df_week_2)
           [,1]
2021-03-07 0
2021-03-14 0
2021-03-21 2
2021-03-28 0
2021-04-04 0
2021-04-11 0
```

Rolling Functions

For doing a calculation within the context of a period R has several rolling functions. Such as cumulative sum, standard deviation etc.

At first we will find a monthly cumulative sum of a time series. For doing this we will use the same data-set that we have made in previous practice.

```
> df_monthly = split(df,f="months")
> df_cumsum = lapply(df_monthly,FUN = cumsum)
> head(df_cumsum)
[[1]]
           [,1]
2021-03-01 0
2021-03-02 0
```

```
2021-03-03 0
2021-03-04 0
2021-03-05 0
```

Now we will find standard deviation of the same time series data after every 3 units of period.

```
> df_sd ← rollapply(df, width=3, FUN = sd) #
width = the number of periods after the sd is
calculated.
> head(df_sd)
                 [,1]
2021-03-01 NA
2021-03-02 NA
2021-03-03 0.0000000
2021-03-04 0.0000000
2021-03-05 0.0000000
2021-03-06 0.0000000
```