

Data Download Tutorial

MBA 592 Team

9/27/2019

Use the code below to get a list of the stocks in the S&P500 along with their respective ticker symbols. If the stock is not in the index, you can get the ticker symbol directly from yahoo finance.

```
sp500 <- GetSP500Stocks()
SnP500 <- sp500[,c(1,2,4)]
rm(sp500)
```

Once you know the stock ticker, you can use the code below to obtain the stock data from the date you choose. We will be extracting the 6th column only, because it contains the adjusted closing prices of the stock. When working with multiple stocks, you can download the data individually and then club them together into a dataframe as shown below. In this case we will use Micron Technologies(MU) and Tesla(TSLA).

```
Micron <- getSymbols("MU", source="yahoo", auto.assign=FALSE,
return.class="xts", from = "2015-07-07")[,6]

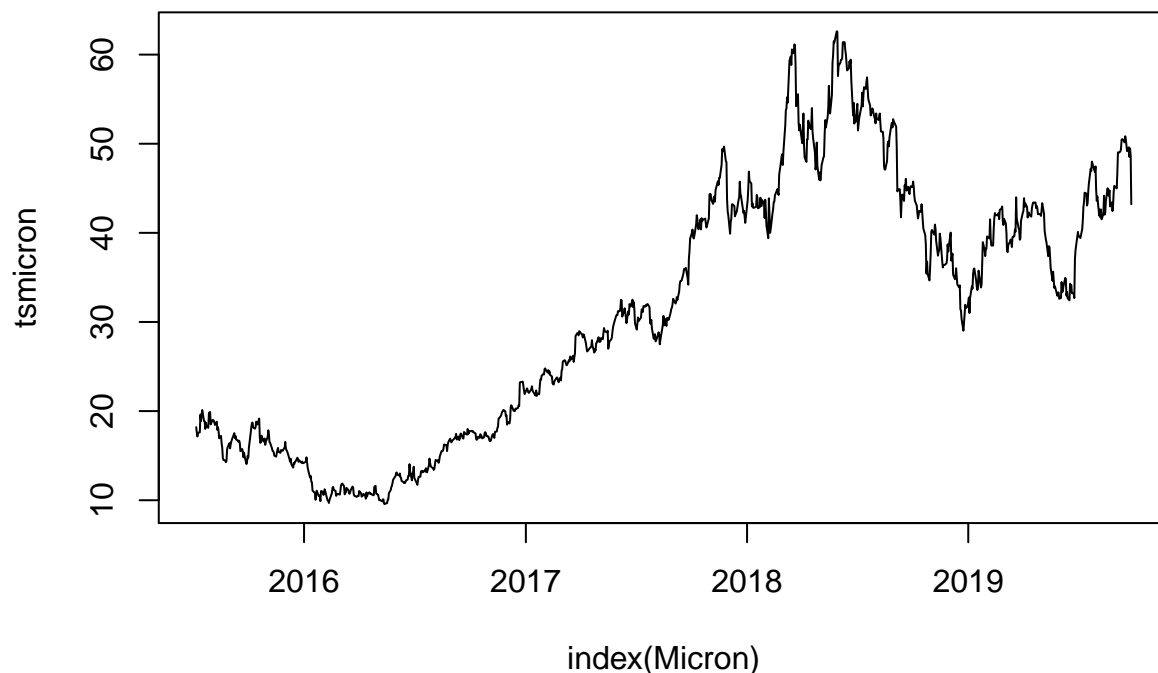
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

Tesla <- getSymbols("TSLA", source="yahoo", auto.assign=FALSE,
return.class="xts", from = "2015-07-07")[,6]

prices <- data.frame(Micron, Tesla)
```

We will use the `ts()` function to convert the data into a time series data. To get the dates on the x axis when plotting a time series data, we need to use the index of the original stock data. For this, we use the `index()` function. With the dates on the x axis, and the time series data on the y axis, we will get a presentable graph. x and y labels can be added using `xlabel` and `ylabel` parameters within `plot()`.

```
tsmicron <- ts(Micron)
plot(index(Micron), tsmicron, type = "l")
```

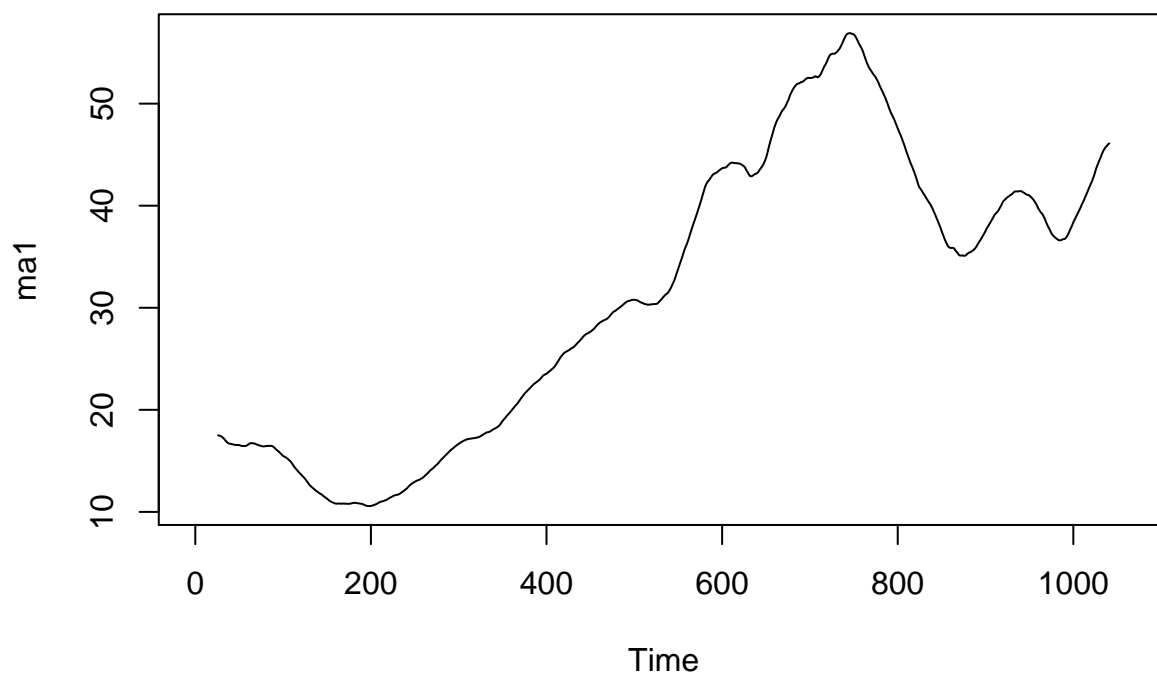


Taking out the differences in prices using `diff()`. When doing the Stationarity Test/ARIMA modelling, this might come in handy.

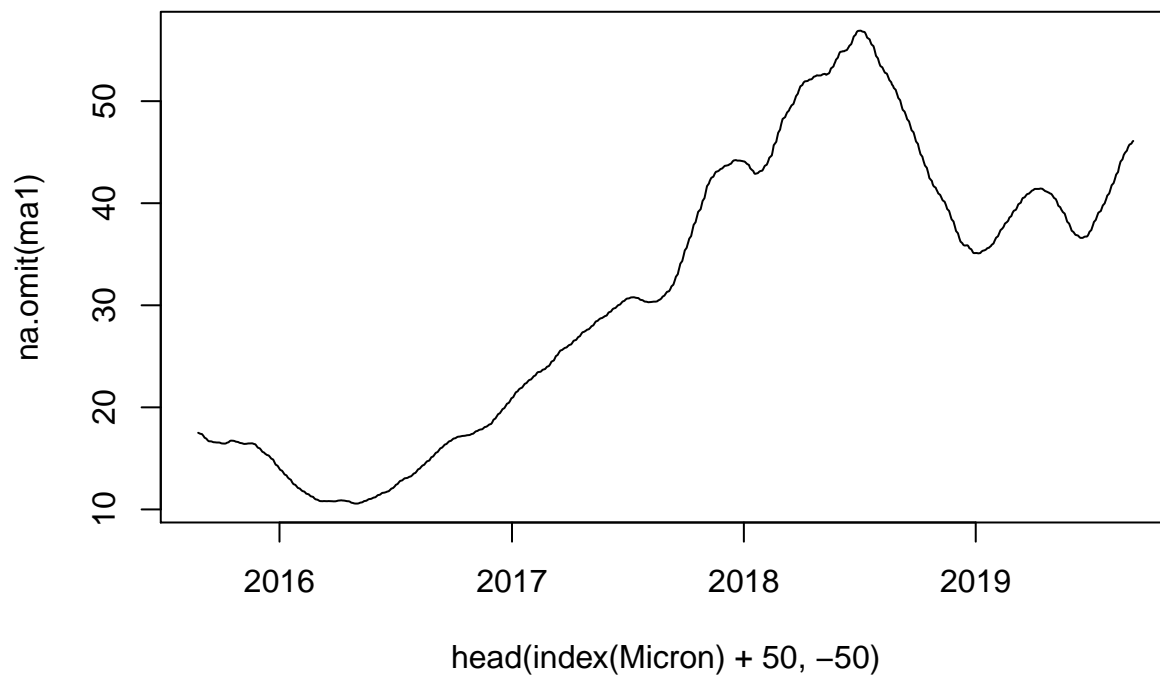
```
microndiff <- diff(tsmicron)
```

When plotting the moving average functions, we have to be careful to adjust for the missing information. If we chose a 50 MA, we will have no data for the first 50 data points. To adjust for this, we add 50 to the date index to shift the x axis by 50 days as shown below. We also remove the last 50 days as we won't have the moving average data for the same.

```
ma1 <- ma(tsmicron, 50)
plot(ma1)
```



```
plot(head(index(Micron)+50,-50), na.omit(ma1), type = "l")
```



The timeseries plot for arima forecasts are a bit more complicated and will not be covered in this tutorial. We can use axis function to replace the x axis with our own tickers. It is tricky to do when using include() within the plot function.