Working With Chronological Objects

Saalar Aghili

12/3/2017

Introduction

In my first post, I discussed how versatile data frames are in R and other statistical softwares. We use data frames exstensively in class for very important aspects of the lessons we learn. My first post talked about how data frames are versatile in their applicability and ended on a note regarding timeseries as a package that uses data frames specific to data in terms of time. From the same motivation of my first post, this post is related to my dedication for professional aspirations in economics research. A lot of economics data incorporates chronological objects that are perfect for the R package “timeSeries”. The package was originally created in 2009 under the name “fCalendar” and has changed a bit ever since. In this blog post, I will be looking at the data visualization associated with the package and chronological objects, as well as the important aspects associated with this package.

Package Needed: “timeSeries” & “timeDate”

```
library(timeSeries)
```

```
## Warning: package 'timeSeries' was built under R version 3.4.2
```

```
## Loading required package: timeDate
```

```
## Warning: package 'timeDate' was built under R version 3.4.2
```

```
library(timeDate)
```

*This package allows for dataframes to be converted into a timeseries object in order to easily manipulate data to show growth rates and returns over time. One of the abilities associated with timeseries objects is the use of **date as a class**.*

Date as a Class

In this package, we learn about a new data type we were not taught in class. Before, data classes included “double”, “character”, “integer”, and “logical” among others. This new class, “ts”, only deals with numeric time stamps. Like other data types, time series objects can be either checked to see if they are, in fact, a time series object with the “is.ts” function and “as.ts” function:

```
a <- c(1,2,3,4,5, 25, 12, 14, 0, 6,7,8,9,10)
is.ts(a)
```

```
## [1] FALSE
```

```
as.ts(a)
```

```
## Time Series:
## Start = 1
## End = 14
## Frequency = 1
##  [1]  1  2  3  4  5 25 12 14  0  6  7  8  9 10
```

Or, you can use “timeSeries” to create a timeSeries object from scratch and ensure that it is originally of that data Class.

```
data12 <- c(1,2000,525600,4,-100000000, 999999999, 1400400000)
data1 <- c(20, 21, 29, 20, 6, 123, 108)
timeismoney <- timeSeries(data1, data12, title = "Oil Prices (USD/Barrel) Before the OPEC Oil Crisis")
timeismoney
```

```
## GMT
##                      TS.1
## 1970-01-01 00:00:01   20
## 1970-01-01 00:33:20   21
## 1970-01-07 02:00:00   29
## 1970-01-01 00:00:04   20
## 1966-10-31 14:13:20    6
## 2001-09-09 01:46:39  123
## 2014-05-18 08:00:00  108
```

Above, you can see that I created two vectors: “data12” for the times that these prices were set at and “data1” for the value of the barrel of Gas in USD

As seen above, making a “timeSeries” object from scratch is very complex with many components to it:
- “charvec” is a vector that gets coerced to a “timeDate” object
- “data” is the object that gets coerced to a matrix
- “FinCenter” includes where this time object is taking place
- “zone” the time zone or financial center where the data was recorded

Some of the standard starting points for the TimeSeries object are:

- The time start date is **January 1, 1970**
- The assumed financial center is at **Greenwich Mean Time**
- The data used in time is assumed to be in seconds
- To change the date before 1970, you must use a negative value like above

This example is a brief data set of the (hypothetical) prices of oil in the global market in terms of USD/Gallon prior to the OPEC Oil Crisis. The time starts before the standard start date (1970-01-01) by using a negative value to start in 1969 and then end up in 2014 after the Oil Crisis.

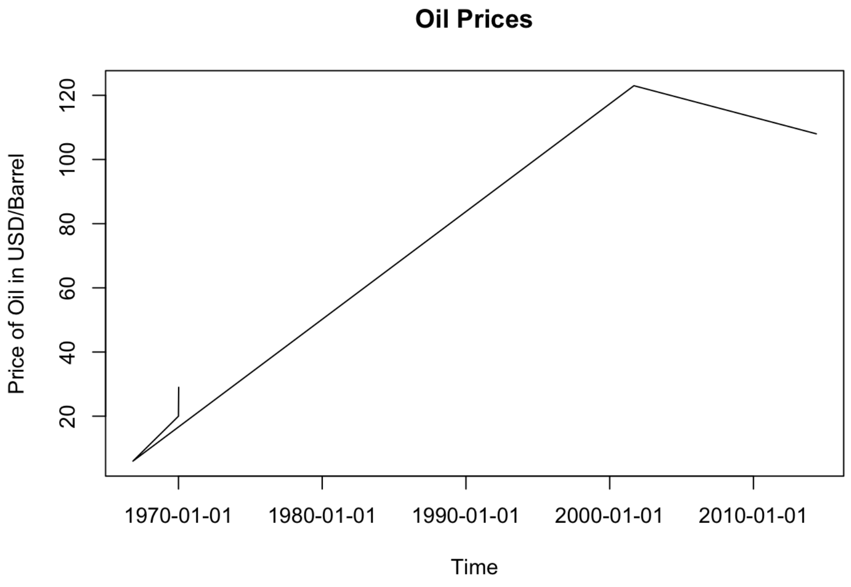
Manipulating Time Data Objects

Effectively, the frequencies of each data type designated as date can be changed based on weeks, months, years, days, etc.
- You can change it from daily to monthly by using the argument “daily2monthly”
- The same can be done from daily to weekly by using the argument “daily2weekly”

Visualizing Time Series Objects

Time Series Objects are used extensively in plotting the chronology of an object through a quantitative unit of measurement. Depending on the data, visualization for time series objects can take many different plots.
The most common plot is the **Univariate Single Plots** that are designed to plot using one single graph frame. One example of this type of data visualization based on the data created previously in this blog post is as follows:

```
plot(timeismoney, xlab = "Time", ylab = "Price of Oil in USD/Barrel", main = "Oil Prices")
```



The graph above depicts the price of oil over the given period of time in USD/Barrel

Other forms of plotting that are more complex can include:
- **Multivariate Single Plots** is to use multivariate objects in one single data frame. This can be good to compare the prices of oil in the US at the time and then comparatively in another country (OPEC average for example).
- **One Column Multiple Plots** separates the separate chronological objects in their own frame.
-**Two Column Multiple Plots** are used when there are 4 or more timeSeries objects in one plot. The number of rows is not restricted in this form of data visualization.

Future Ideas

Some of the related packages associated with Time series and working with chronological objects is **forecasting**. Forecasting is its own package that can be used in econometrics and finance to predict future outcomes based on the retrospective data analysis of the data provided in the time series object. In the future, I would like to explore forecasting and the range of other functions and capabilities that can be produced by this package. I see it as one of the most applicable tools I have found in R that can translate into a potential economics data research project, or even a thesis that includes forecasting. I plan on writing a thesis about the city planning growth of a development project coordinated by India and Iran at Chabahar Port and estimating the economic development growth associated with the growth of a large port city on Iran’s coast. It will be shared by India and Afghanistan, enabling the expansion of industry and trade in South to Western Asia. I can use **forecast** to test these projected growth rates in the future to come.

Works Cited

- Eddelbuettel, Dirk. “CRAN Task View: Empirical Finance.” The Comprehensive R Archive Network, 12 Sept. 2017, cran.r-project.org/web/views/Finance.html.
- “Forecasting Functions for Time Series and Linear Models [R Package Forecast Version 8.2].” The Comprehensive R Archive Network, Comprehensive R Archive Network (CRAN), cran.r-project.org/web/packages/forecast/index.html.
- Wuertz, Diethelm. “Package ‘timeSeries’” The Comprehensive R Archive Network, 17 Nov. 2017.
- Wuertz, Diethelm. “Plotting ’timeSeries’ Objects”. The Comprehensive R Archive Network, 12 May 2014.
- Hewson, Paul. “CRAN Task View: Multivariate Statistics.” The Comprehensive R Archive Network, 1 July 2015, cran.r-

project.org/web/views/Multivariate.html.

- Hyndman, Rob J. “CRAN Task View: Time Series Analysis.” The Comprehensive R Archive Network, CRAN, 19 Oct. 2017, cran.r-project.org/web/views/TimeSeries.html.
- Zeileis, Achim. “CRAN Task View: Econometrics.” The Comprehensive R Archive Network, 5 Oct. 2017, cran.r-project.org/web/views/Econometrics.html.