Introduction to the **eventstudies** package in R

Ajay Shah, Vimal Balasubramaniam and Vikram Bahure

May 8, 2013

Abstract

The structure of the package and its implementation of event study methodology is explained in this paper. In addition to converting physical dates to event time frame, functions for re-indexing the event time returns and bootstrap inference estimation. The methods and functions are elucidated by employing data-set of SENSEX firms.

1 Introduction

Event study has a long history which dates back to 1938 (Dolley, 1938). It is mostly used to study the response of stock price or value of a firm due to events such as mergers & acquisitions, stock splits, quarterly results and so on. It is one of the most widely used statistical tool.

Event study is used to study the response or the effect on a variable, due to similar events. Efficient and liquid markets are basic assumption in this methodology. It assumes the effect on response variable is without delay. As event study output is further used in econometric analysis, hence significance test such as *t-test*, *J-test*, *Patell-test* which are parametric and *GRANK*, *RANK* which are non-parametric can also be performed.

In this package, there are three major functions *phys2eventtime*, *remap.cumsum* and *inference.Ecar*. *phys2eventtime* changes the physical dates to event time frame on which event study analysis can be done with ease. *remap.cumsum* can be used to convert returns to cumulative sum or product in the event time frame. *inference.Ecar* generates bootstrap inference for the event time response of the variable.

In the section below, we illustrate event study analysis using the package. We measure the impact of stock splits on the stock price of the firm for SENSEX index constituents.

2 Performing Eventstudy analysis

To measure the impact of stock splits on the stock price of the firm, we create a dataset of 30 index companies of Bombay Stock Exchange (BSE). We have a returns of stock price for each firm from 2001 to 2013 and respective stock splits date. Once we have the data then we use following steps to perform event study analysis using the package.

1. Construction of data set

    - A time series object of stock price returns
    - Event dates object with 2 columns, *unit* and *when*.
2. Converting physical dates to event frame
3. Remapping event frame
4. Estimating bootstrap inference

2.1 Construction of data set

We have collected data of index constituents of Bombay stock exchange (BSE) and corresponding stock splits dates. There are 30 firms in SENSEX and we have stock split dates for each firm from 2000 onwards.

A time series *zoo* object is created for stock price returns for 30 firms. For event dates, a data frame with two columns *unit* and *when* is formed. *unit* has name of the response series (firm name as in column name of time series object) along with event date in *when*. *unit* should be in *character* format and *when* in *Date* format.

```
> library(eventstudies)
> data(StockPriceReturns)
> str(StockPriceReturns)
```

```
âĂŸzooâĂŹ series from 2000-04-03 to 2013-03-28
  Data: num [1:3246, 1:30] NA NA NA NA NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:30] "Bajaj.Auto" "BHEL" "Bharti.Airtel" "Cipla" ...
  Index:  Date[1:3246], format: "2000-04-03" "2000-04-04" "2000-04-05" "2000-04-06" ...
```

```
> data(SplitDates)
> head(SplitDates)
```

```
            unit       when
5           BHEL 2011-10-03
6  Bharti.Airtel 2009-07-24
8          Cipla 2004-05-11
9     Coal.India 2010-02-16
10      Dr.Reddy 2001-10-10
11     HDFC.Bank 2011-07-14
```

2.2 Converting physical dates to event frame

After the formation of the dataset, our first step towards event study analysis is to convert the physical dates to event time frame. Using the *phys2eventtime* function we convert the dates in event time frame.

Here, we index the stock split date, stock price returns to day 0 and similarly post event dates are indexed to positive and pre event dates are indexed as negative. As we can see below the stock split dates for BHEL, Bharti Airtel and Cipla are indexed to day 0.

The output for *phys2eventtime* is a list. The first element of a list is a time series object which is converted to event time and the second element is *outcomes* which shows if there was any *NA* in the dataset. If the outcome is *success* then all is well in the given window as specified by the width. It gives *wdatamissing* if there are too many *NAs* within the crucial event window or *wrongspan* if the event date is not placed within the span of data for the unit or *unitmissing* if a unit named in events is not in *z*.

```
> es <- phys2eventtime(z=StockPriceReturns, events=SplitDates, width=10)
> es.w <- window(es$z.e, start=-10, end=10)
> SplitDates[1:3,]
```

```
          unit       when
5         BHEL 2011-10-03
6 Bharti.Airtel 2009-07-24
8        Cipla 2004-05-11
```

```
> StockPriceReturns[SplitDates[1,2],SplitDates[1,1]]
```

```
2011-10-03
 -1.905466
```

```
> StockPriceReturns[SplitDates[2,2],SplitDates[2,1]]
```

```
2009-07-24
  2.079229
```

```
> StockPriceReturns[SplitDates[3,2],SplitDates[3,1]]
```

```
2004-05-11
 -2.681747
```

```
> es.w[,1:3]
```

```
             1           2           3
-10 -1.7405787 -2.21233977  3.22197080
-9  -0.8178890 -2.07950664 -1.98137850
-8  -0.3259862  0.96634168  2.23210993
-7  -2.0657433  0.85718120  1.50960157
-6  -0.3276390  0.66929183  2.17782971
-5  -0.6302048  5.19168746 -1.77666214
-4   3.5561912  2.58180454  0.01819671
-3  -0.6791870 -1.36372757 -1.56790449
-2   1.9429840 -1.76829469 -1.68682067
-1  -1.8788276 -1.09485685 -1.26739614
0   -1.9054656  2.07922865 -2.68174651
1    0.4503463  1.89528291  2.10900726
2   -2.3833050  0.96357972  0.32495489
3    2.7544777 -0.99900931 -5.45126131
```

```
4    2.3946847  0.04723666 -13.67033437
5   -1.5951724 -3.10557899   9.84978345
6    3.0910867  0.13387697   1.86688683
7   -0.8649025 -2.59978912   1.48906372
8   -0.1498801  0.22443900  -0.48711280
9   -2.1375785 -0.53699786   3.71448453
10  -1.4042357 -3.95914305  -1.52101698
```

2.3 Remapping event frame

In event study analysis the variable of interest is cumulative returns. The *remap.cumsum* function is used to convert the returns to cumulative returns.

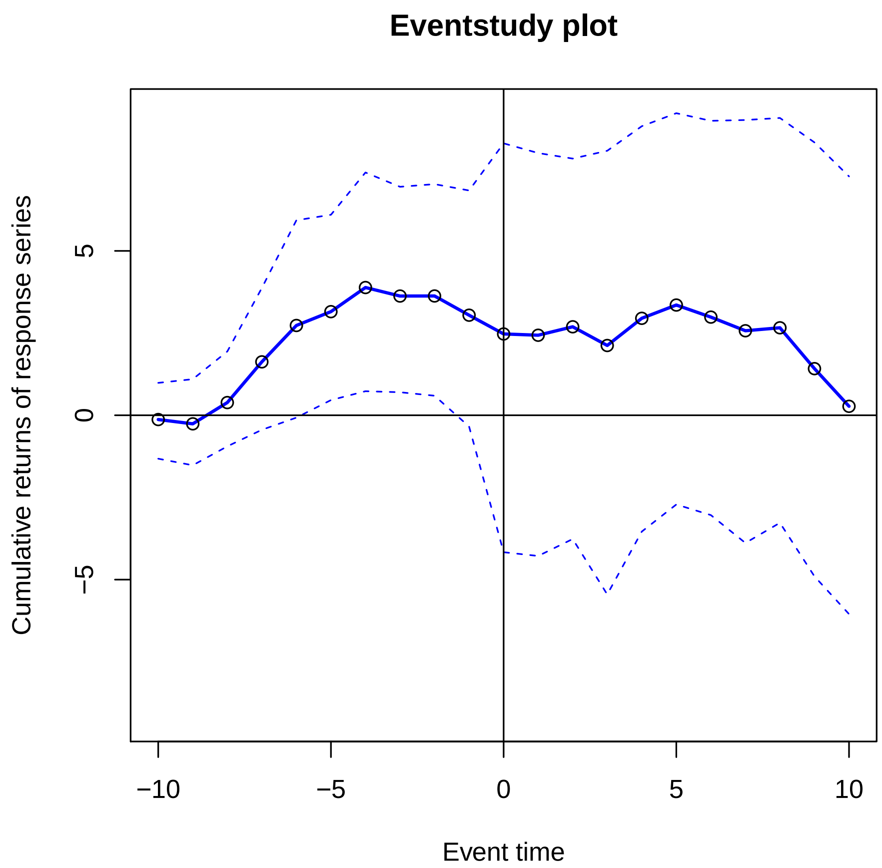
```
> es.cs <- remap.cumsum(es.w,is.pc=FALSE,base=0)
> es.cs[,1:3]
```

```
             1          2           3
-10 -1.7405787 -2.2123398   3.2219708
-9  -2.5584676 -4.2918464   1.2405923
-8  -2.8844539 -3.3255047   3.4727022
-7  -4.9501971 -2.4683235   4.9823038
-6  -5.2778361 -1.7990317   7.1601335
-5  -5.9080409  3.3926558   5.3834714
-4  -2.3518497  5.9744603   5.4016681
-3  -3.0310367  4.6107327   3.8337636
-2  -1.0880527  2.8424380   2.1469429
-1  -2.9668803  1.7475812   0.8795468
0   -4.8723458  3.8268098  -1.8021997
1   -4.4219995  5.7220928   0.3068075
2   -6.8053046  6.6856725   0.6317624
3   -4.0508268  5.6866632  -4.8194989
4   -1.6561421  5.7338998 -18.4898333
5   -3.2513146  2.6283208  -8.6400498
6   -0.1602279  2.7621978  -6.7731630
7   -1.0251304  0.1624087  -5.2840993
8   -1.1750105  0.3868477  -5.7712121
9   -3.3125891 -0.1501502  -2.0567275
10  -4.7168248 -4.1092932  -3.5777445
```

2.4 Bootstrap inference

After converting to event frame and estimating the interest variable, we need to check the stability of the result and derive other estimates like standard errors and confidence intervals. For this, we generate the sampling distribution for the estimate using bootstrap inference. A detailed explanation of the methodology is presented in Patnaik, Shah and Singh (2013).

Figure 1: Stock splits event and response of respective stock returns



This specific approach used here is based on Davinson *et al.* (1986). The *inference.Ecar* function does the bootstrap to generate distribution of $\bar{CR}$. The bootstrap generates confidence interval at 2.5% and 97.5% for the estimate.

```
> result <- inference.Ecar(z.e=es.cs, to.plot=TRUE)
```

3 Computational details

The package code is purely written in R. It has dependencies to zoo (Zeileis 2012) and boot (Ripley 2013). R itself as well as these packages can be obtained from CRAN.

References

Davinson A, Hinkley DV, Schechtman E (1986). "Efficient bootstrap simulation." *Biometrika*, **73**(3), 555–566.

Dolley JC (1938). "The effect of government regulation in the stock-trading volume of the New York Stock Exchange." *The American Economic Review*, **28**(1), 8–26.