Estimating spot volatility in R:
The spotvolatility package

Maarten Schermer

July 3, 2014

1 General usage

The `spotvolatility` package offers several methods to estimate spot volatility and its intraday seasonality, using high-frequency data. It is an extension of the function `spotvol` in the `highfrequency` package. That function is included as one of the methods in this package, and multiple other spot volatility estimation methods from recent academic literature have been added.

The spot volatility $\sigma_{t,n}$ can be described as the the volatility of returns on a certain moment n during day t. These returns $r_{t,n}$ are commonly modeled as

$$r_{t,n} = \sigma_{t,n}\varepsilon_{t,n}, \tag{1}$$

where $\varepsilon_{t,n} \sim \mathrm{IID}(0, \frac{1}{N})$ is a white noise process. N denotes the number of intraday periods, and we will denote the number of days with T.

The spot volatility estimates $\hat{sigma}_{t,n}$ can be estimated from these returns using the `spotvol` function. The estimates are stored as `spot` in a `spotvol` object, which is a list containing certain outputs of the function, depending on the method used. These estimates can be called in the following way

```
library(spotvolatility)
out <- spotvol(sample_prices_5min)
class(out)

## [1] "spotvol"

sigma_hat <- out$spot
```

1.1 Input data: prices or returns

The `spotvol` function accepts two sorts of input:

1. An `xts` object containing price data.

2. A **matrix** containing return data.

The first can be used for raw price data, i.e. from the NYSE TAQ database. The **highfrequency** package offers several methods to process data files into **xts** format. The **spotvol** function will then aggregate this price data to the frequency specified by the user and calculate the return series.

```
str(sample_prices_5min)

## An 'xts' object on 2005-03-04 09:30:00/2005-06-01 16:00:00 containing:
##   Data: num [1:4819, 1] 105 105 105 105 104 ...
##   Indexed by objects of class: [POSIXct,POSIXt] TZ:
##   xts Attributes:
##  NULL

out <- spotvol(sample_prices_5min)
```

The second way of inputting data is through a **matrix**, which can be convenient if your data has already been processed. No further permutations will be made to the data, as it is assumed to be in the right format. Therefore, it should satisfy the following restrictions:

- The returns should be equispaced, i.e. the time between each pair of observations should be equal.
- Each row of the matrix should correspond to a day.
- Each column of the matrix should correspond to an intraday time interval.

For example, a matrix containing 5-minute returns of 60 trading days opening at 09:30 and closing at 16:00 should have 100 rows and 288 columns.

```
str(sample_returns_5min)

##  num [1:60, 1:288] 0.00404 -0.02825 0.03223 0.00806 0.04056 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:288] "X2" "X3" "X4" "X5" ...

out <- spotvol(sample_returns_5min)
```