High-frequency price data analysis in R

R/Finance 2015

Kris Boudt
VU Brussel/Amsterdam

About myself

- Associate Professor of Finance and Econometrics at Free University of Brussels and Amsterdam;
- Research on developing econometric methodology to solve problems in finance.
- R packages to which I contributed: **highfrequency**, PeerPerformance, PerformanceAnalytics, PortfolioAnalytics, CIP, DEoptim;

Roadmap (i)

- "There are old traders, there are bold traders, but there are no old bold traders"
- Focus: How to use high frequency price data to understand better the time-varying risk properties of the investment
- Two types of risk: the normal volatility risk and the jump risk
- Topics:
  - Cleaning and aggregation (univariate and multivariate) of tick prices into log-returns
  - Discrete time model for intraday returns:
    - Spot volatility estimation
    - Price jump detection
  - Continuous time model for log-prices
    - Realized volatility estimation
    - Detection of a jump component in realized volatility
  - Forecasting volatility using realized volatility measures.

Roadmap (ii)

- And how to do these analysis with the functions in the R package **highfrequency**
  - Latest version at: http://r-forge.r-project.org/R/?group_id=1409
  - Main authors are Jonathan Cornelissen (Datacamp), Scott Payseur (UBS) and myself.

Other contributors:
- GSOC:
  - **Giang Nguyen**
  - **Maarten Schermers**
- **Chris Blakely, Brian Peterson, Eric Zivot**
- **You?**

WARNING: The functions in highfrequency were initially designed for the Trades and Quotes database but are generally applicable, as long as:

- They are xts-objects;
- Some functions require tdata/qdata:
  - tdata: Trade data having at least the column name "PRICE"
  - qdata: Quote data having at least the column names "BID" and "OFR"

```
> highfrequency:::tdatacheck
function (tdata)
{
    if (!is.xts(tdata)) {
        stop("The argument tdata should be an xts object")
    }
    if (!any(colnames(tdata) == "PRICE")) {
        stop("The argument tdata should have a PRICE column")
    }
}
<environment: namespace:highfrequency>
> highfrequency:::qdatacheck
function (qdata)
{
    if (!is.xts(qdata)) {
        stop("The argument qdata should be an xts object")
    }
    if (!any(colnames(qdata) == "BID")) {
        stop("The argument qdata should have a column containing the BID. Could not find that column")
    }
    if (!any(colnames(qdata) == "OFR")) {
        stop("The argument qdata should have a column containing the ASK / OFR. Could not find that column")
    }
}
<environment: namespace:highfrequency>
```

CLEANING AND AGGREGATION

- High frequency price data analysis: Making sense of **too** big data

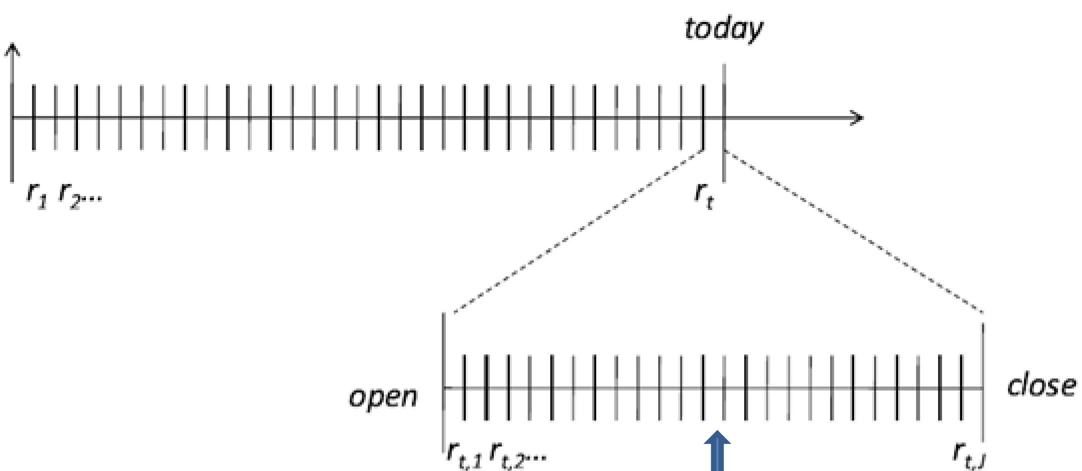
- The tick by tick ‘raw’ price series needs to processed in two ways:
  - **Data cleaning** to remove some obvious “errors” from the data:
    - Trades and quotes with position size of 0;
    - Trades and quotes with time stamp outside the opening hours of the exchange;
    - Trade prices that are below the best bid are above the best ask
    - Bid quotes that are higher than the ask quote
    - Fat finger errors: A human error caused by pressing the wrong key when using a computer to input data.
  - **Aggregation** to the frequency of interest.

Function in highfrequency	Aim	Requirement on input data
exchangeHoursOnly	Restrict data to exchange hours	xts
selectexchange	Restrict data to specific exchange	xts with column "EX"
autoSelectExchangeTrades	Restrict data to exchange with highest trade volume	xts with column "EX“ and “SIZE”
mergeTradesSameTimestamp	Delete entries with same time stamp and use median price	
rmTradeOutliers	Delete entries with prices above/below ask/bid +/- bid/ask spread	xts with column “PRICE” xts with columns “BID” and “OFR” ➔ Transaction and quote data are matched internally with the function matchTradesQuotes
rmOutliers	Remove outliers in quote data based on rolling outlier detection in the spread	xts with columns “BID” and “OFR”

<u>And several others</u>: noZeroPrices, noZeroQuotes, mergeQuotesSameTimestamp, rmNegativeSpread

Aggregation

- Highfrequency price data analysis consists of zooming in on the intraday price data obtained typically as tick data (which occur at irregular times) aggregated at some frequency:



Daily aggregation

Aggregation over higher frequency intervals

Tick by tick price data

Two types of aggregation from tick data:

- **Calendar time** based sampling: Every 10 minutes, Every minute, Every second, Every milisecond ➔ Prices are observed a regularly spaced time intervals
- **Transaction** based sampling (also called business time sampling): Every 10 trades, every trade (tick data).

The choice of the sampling frequency may be a function of, among other things, the liquidity of the stock : Illiquid stocks are infrequently traded implying many zero returns at very high frequencies.

Calendar time based sampling

- Function aggregatets: From transaction time to a **fixed <u>calendar time</u> based frequency**, e.g. every 5 seconds:
  - Default: previous tick: take the last price observed in the interval: [start,end[(i.e. excluding the value at the end time of the interval)
  - Alternative: take the mean value

```
data("sample_tdata");
ts = sample_tdata$PRICE;
# Previous tick aggregation to the 5-seconds
sampling frequency:
tsagg5secs = aggregatets(ts,on="seconds",k=5);
head(tsagg5secs);
```

```
> head(ts,25)
                    PRICE
2008-01-04 09:30:27 "193.71"
2008-01-04 09:30:28 "193.59"
2008-01-04 09:30:29 "193.445"
2008-01-04 09:30:30 "193.38"
2008-01-04 09:30:31 "193.34"
2008-01-04 09:30:33 "193.52"
2008-01-04 09:30:34 "193.58"
2008-01-04 09:30:35 "193.63"
2008-01-04 09:30:36 "193.47"
2008-01-04 09:30:37 "193.26"
2008-01-04 09:30:38 "193.26"
2008-01-04 09:30:39 "193.23"
2008-01-04 09:30:40 "193.15"
2008-01-04 09:30:41 "193.16"
2008-01-04 09:30:46 "192.86"
2008-01-04 09:30:47 "192.85"
2008-01-04 09:30:49 "192.83"
2008-01-04 09:30:50 "193"
2008-01-04 09:30:51 "192.985"
2008-01-04 09:30:54 "193.045"
2008-01-04 09:30:56 "192.735"
2008-01-04 09:30:57 "192.85"
2008-01-04 09:30:59 "192.68"
2008-01-04 09:31:01 "192.99"
2008-01-04 09:31:02 "192.795"
>
```

```
> head(tsagg5secs)
                      PRICE
2008-01-04 09:30:30 193.380
2008-01-04 09:30:35 193.630
2008-01-04 09:30:40 193.150
2008-01-04 09:30:45 193.160
2008-01-04 09:30:50 193.000
2008-01-04 09:30:55 193.045
>
```

Note: Loss of observation, but more tractable , and less market microstructure noise issues

Business time based sampling

- From transaction time to a **fixed <u>business time</u> based frequency**, e.g. every 5 ticks:

```
ts[seq(1,length(ts),5)]
```

```
> head(ts,25)
                    PRICE
2008-01-04 09:30:27 "193.71"
2008-01-04 09:30:28 "193.59"
2008-01-04 09:30:29 "193.445"
2008-01-04 09:30:30 "193.38"
2008-01-04 09:30:31 "193.34"
2008-01-04 09:30:33 "193.52"
2008-01-04 09:30:34 "193.58"
2008-01-04 09:30:35 "193.63"
2008-01-04 09:30:36 "193.47"
2008-01-04 09:30:37 "193.26"
2008-01-04 09:30:38 "193.26"
2008-01-04 09:30:39 "193.23"
2008-01-04 09:30:40 "193.15"
2008-01-04 09:30:41 "193.16"
2008-01-04 09:30:46 "192.86"
2008-01-04 09:30:47 "192.85"
2008-01-04 09:30:49 "192.83"
2008-01-04 09:30:50 "193"
2008-01-04 09:30:51 "192.985"
2008-01-04 09:30:54 "193.045"
2008-01-04 09:30:56 "192.735"
2008-01-04 09:30:57 "192.85"
2008-01-04 09:30:59 "192.68"
2008-01-04 09:31:01 "192.99"
2008-01-04 09:31:02 "192.795"
>
```

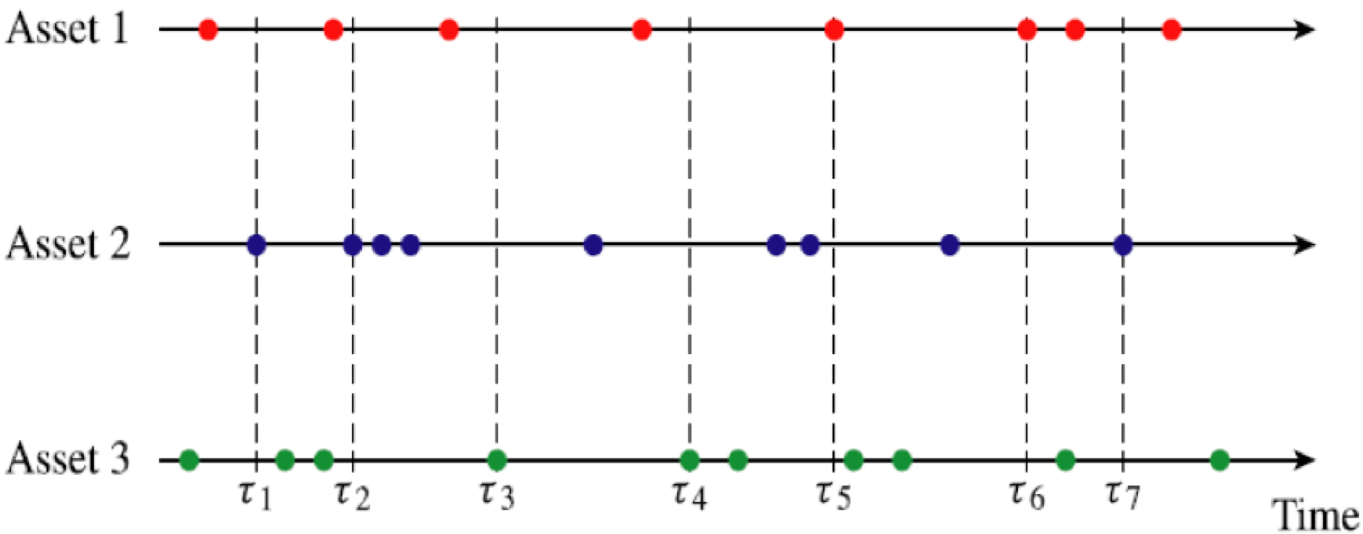
```
> head(ts[seq(1,length(ts),5)])
                    PRICE
2008-01-04 09:30:27 "193.71"
2008-01-04 09:30:33 "193.52"
2008-01-04 09:30:38 "193.26"
2008-01-04 09:30:47 "192.85"
2008-01-04 09:30:56 "192.735"
2008-01-04 09:31:03 "192.69"
>
```

Note: Loss of observation, but more tractable, and less market microstructure noise issues.

- For multivariate analysis, such a univarate aggregation scheme is often not suited due to **non-synchronicity** in the trades of different assets:
  - If one stock has traded, but the other has not, it would seem as there is no relationship, while in fact there is one, but we have not observed it yet.
  - Epss effect: Because trades occur in discrete time, when sampling at ultrahighfrequency observation times, the correlation is biased towards zero.

Multivariate synchronization: Refresh times

- From nonsynchronous transaction times based observations of multiple series to common observations: next observation is when there has been a new observation for all series



Refresh-time sampling. *Source: Barndorff-Nielsen et al., 2011.*

τ_1 is the time it has taken until the three assets have traded, i.e. all the posted prices have been updated.
τ_2 is the first time when all the prices are again refreshed

```
#suppose irregular timepoints:
start = as.POSIXct("2010-01-01 09:30:00")
ta = start + c(1,2,4,5,9,14);
tb = start + c(1,3,6,7,8,9,10,11,15);
tc = start + c(1,2,3,5,7,8,10,13);
#yielding the following timeseries:
a = as.xts(1:length(ta),order.by=ta);
b = as.xts(1:length(tb),order.by=tb);
c = as.xts(1:length(tc),order.by=tc);
#Calculate the synchronized timeseries:
refreshTime(list(a,b,c))
```

```
> a
                    [,1]
2010-01-01 09:30:01    1
2010-01-01 09:30:02    2
2010-01-01 09:30:04    3
2010-01-01 09:30:05    4
2010-01-01 09:30:09    5
2010-01-01 09:30:14    6
> b
                    [,1]
2010-01-01 09:30:01    1
2010-01-01 09:30:03    2
2010-01-01 09:30:06    3
2010-01-01 09:30:07    4
2010-01-01 09:30:08    5
2010-01-01 09:30:09    6
2010-01-01 09:30:10    7
2010-01-01 09:30:11    8
2010-01-01 09:30:15    9
> c
                    [,1]
2010-01-01 09:30:01    1
2010-01-01 09:30:02    2
2010-01-01 09:30:03    3
2010-01-01 09:30:05    4
2010-01-01 09:30:07    5
2010-01-01 09:30:08    6
2010-01-01 09:30:10    7
2010-01-01 09:30:13    8
>
```

```
> refreshTime(list(a,b,c))
                    [,1] [,2] [,3]
2010-01-01 01:30:01    1    1    1
2010-01-01 01:30:03    2    2    3
2010-01-01 01:30:06    4    3    4
2010-01-01 01:30:09    5    6    6
2010-01-01 01:30:14    6    8    8
```

Note:

- The least liquid stock will determine the sampling grid: you risk to lose many observations.
- Even if same liquidity, because of random arrivals, large data losses in high dimensions. Curse of dimensionality!

DISCRETE TIME MODEL FOR THE LOG-RETURNS

- Let us consider first a discrete time location-scale model for the highfrequency returns;
- Notation:
  - P(s) is the price at time s
  - p(s) = log(P(s)) is the natural logarithm of the price
  - We assume for the moment equispaced intraday returns and denote the *i*-th return on day *t* as $r_{t,i}$
  - The length of one day is normalized to [0,1]
  - Assume M observations in a day, then the time between two observations is Δ=1/M.
  - The i-th return on day t is given by:

$$\boxed{r_{t,i} = p(t-1+i\Delta) - p(t-1+(i-1)\Delta)}$$

A discrete time conditional location- scale model for the highfrequency log-return

- Conditional on the information available at the end of the previous intraday time interval $I_{t,i-1}$:

$$\boxed{r_{t,i} = \mu_{t,i}\Delta + \sigma_{t,i}\sqrt{\Delta}z_{t,i}}$$

  - with $\mu_{t,i}$ the conditional mean at the daily level (also called drift)
  - $\sigma_{t,i}$ the conditional volatility at the daily level (also called spot volatility)
  - $z_{t,i}$ standard white noise (i.e. iid with mean 0 and variance 1, typically assumed to be Gaussian).

Estimation?

- At high frequencies, we can assume the drift to be 0:

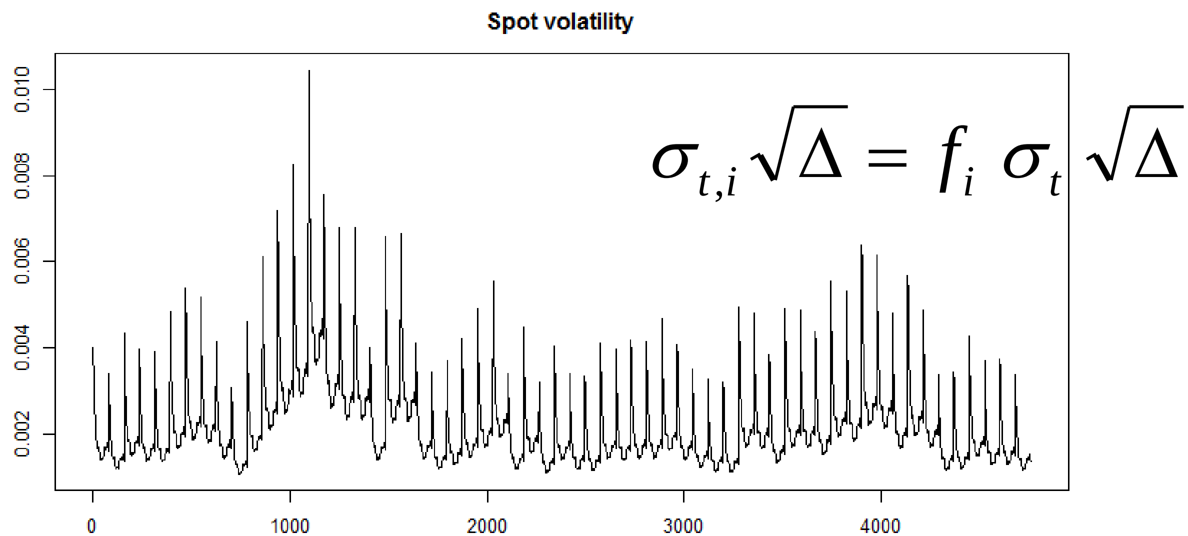
$$\boxed{\mu_{t,i}\Delta = 0}$$

- The function spotvol provides an estimate of $\sigma_{t,i}$:
  - Non-parametric: local kernel estimator;
  - Semi-parametric: volatility is assumed to be given by the product between:
    - A stochastic daily volatility level σ_t
    - A deterministic intraday period process, corresponding to the U-shape f_i:

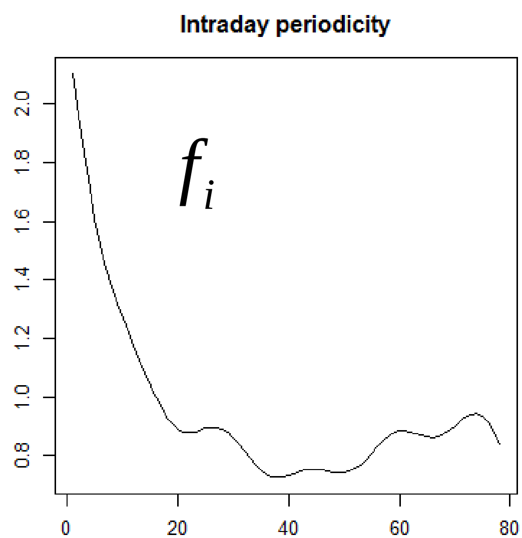
$$\boxed{\sigma_{t,i} = f_i\ \sigma_t}$$

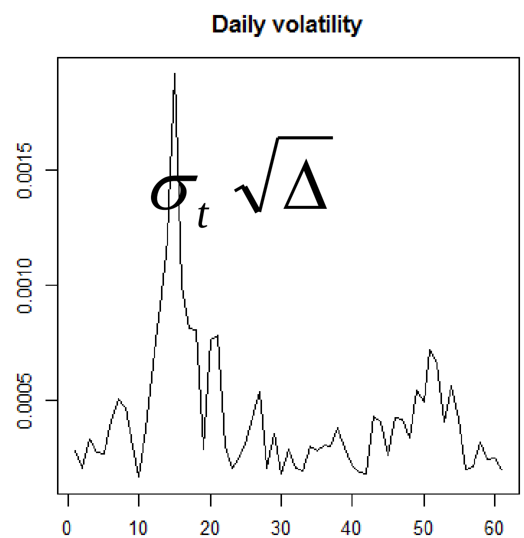
```r
data(sample_real5minprices)
plot(spotvol(sample_real5minprices))
```

Time series of 5-minute spot volatilities for 60 days:



Corresponding components:
* Intraday periodic component that only de





- Extensions in spotvol:
  - Allow for a stochastic component in the periodic pattern;
  - Robust estimators that account for price jumps in the estimation:

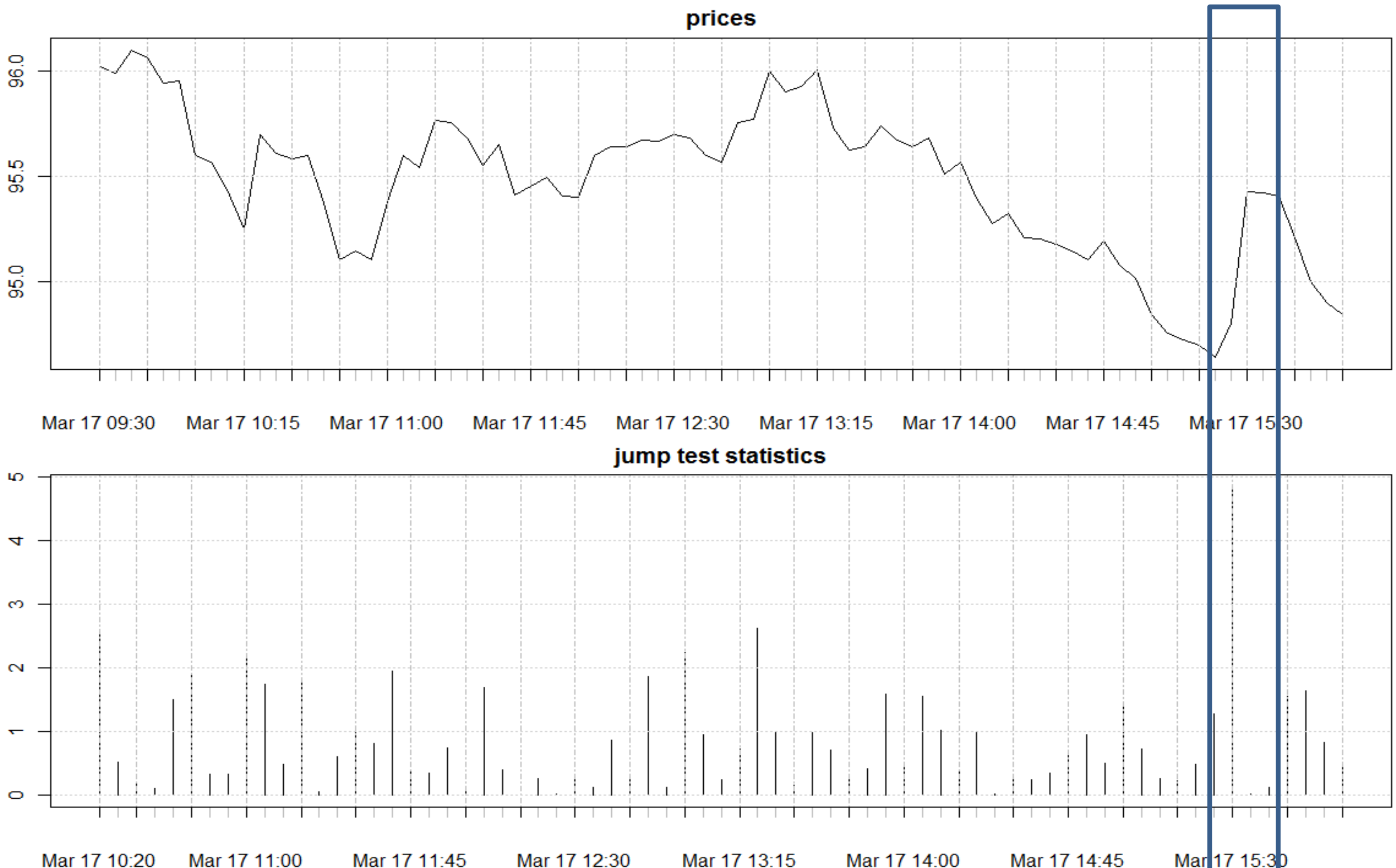
$$\boxed{r_{t,i} = \sigma_{t,i}\sqrt{\Delta} z_{t,i} + j_{t,i}}$$

  - And use this to detect the price jumps by identifying a jump when:

$$\left| \frac{r_{t,i}}{\sigma_{t,i}\sqrt{\Delta}} \right|$$

    is very large

```
# illustration for day 9 in the example data sample_real5minprices
# plot the price series and the corresponding jump test statistics
 d=9 ; par(mfrow=c(2,1),mar=c(3,2,2,1))
 plot(sample_real5minprices[d*79+(1:79)],main="prices")
 plot( abs(diff(log(sample_real5minprices[d*79+(1:79)]))[(-
1)])/spotvol(sample_real5minprices)$spot[d*79+(1:79)],type="h"
,main="jump test statistics")
```



- How large?
  - Take a high critical value to avoid false detections

```
> 2*(1-pnorm(2))
[1] 0.04550026
> 2*(1-pnorm(3))
[1] 0.002699796
> 2*(1-pnorm(3.5))
[1] 0.0004652582
> 2*(1-pnorm(4))
[1] 6.334248e-05
```

  - But not too high to avoid that you lose power to detect the jumps

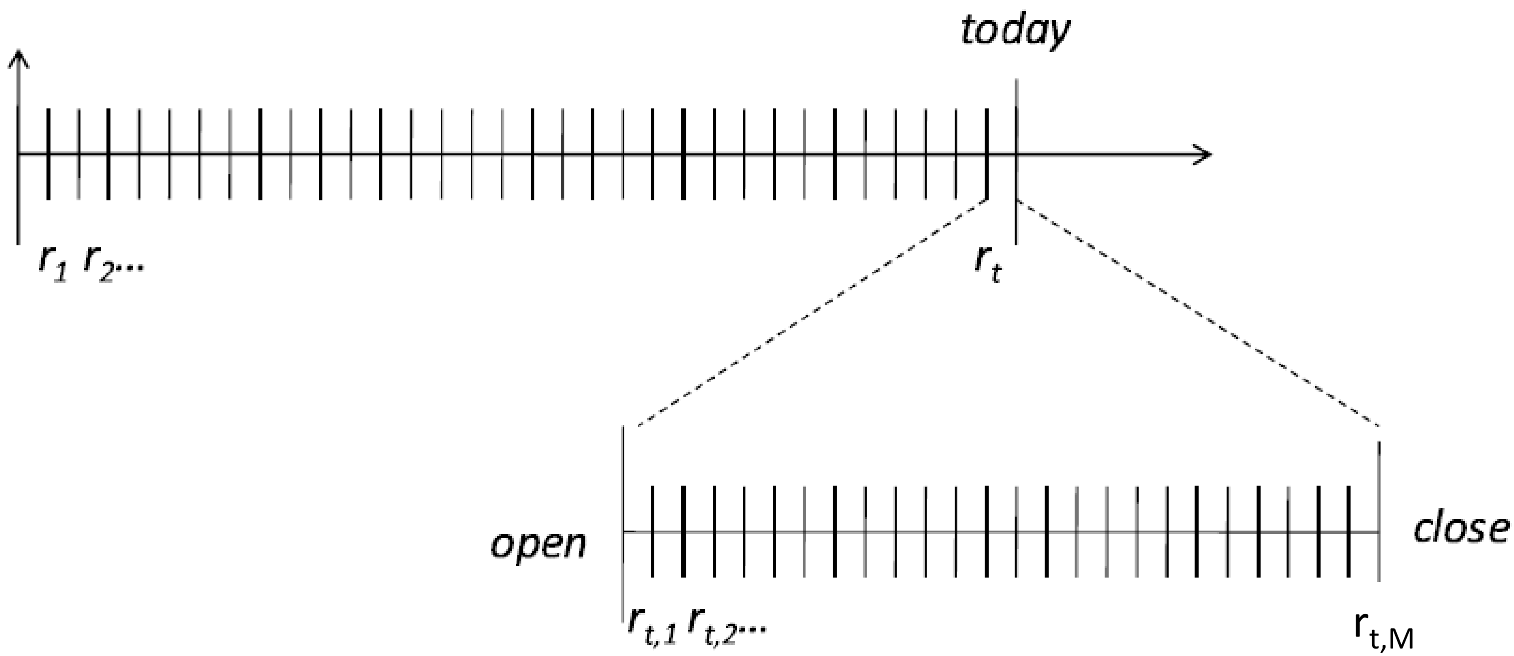
REALIZED VOLATILITY ESTIMATION

- In addition to the estimation of the local spot volatility, it is important to also estimate the variability over a longer time window, such as one day.
- Noisy measure of daily variability: squared daily return
- Potentially more efficient measures use intraday data:
  - daily price range
  - Realized variance: sum of squared intraday returns
  - …

  ➔However, to understand what parameter they actually estimate it is important to have a model for the intraday price evolution: **Continuous-time brownian semimartingale model with jumps**.

  ➔The observed prices are discrete time realizations of that continuous-time process.

From discrete to continuous time model



- Asymptotic analysis: what happens if $M \rightarrow \infty$, that is, when $\Delta \rightarrow 0$.

From discrete to continuous time

- Discrete time conditional location scale model

$$\boxed{r_{t,i} = \mu_{t,i}\Delta + \sigma_{t,i}\sqrt{\Delta} z_{t,i}}$$

- Continuous time brownian semi-martingale diffusion

$$\boxed{dp_s = \mu_s ds + \sigma_s dw_s}$$

μ is the drift parameter

σ is the spot volatility parameter

- A continuous time stochastic proces $\{w_t\}$ is a Brownian motion (Wiener process) is it satisfies that its increments are iid normal with variance equal to the time change.
- More precisely, for any time s, we have that

$$w_s - w_{s-\Delta} = z_s \sqrt{\Delta}$$

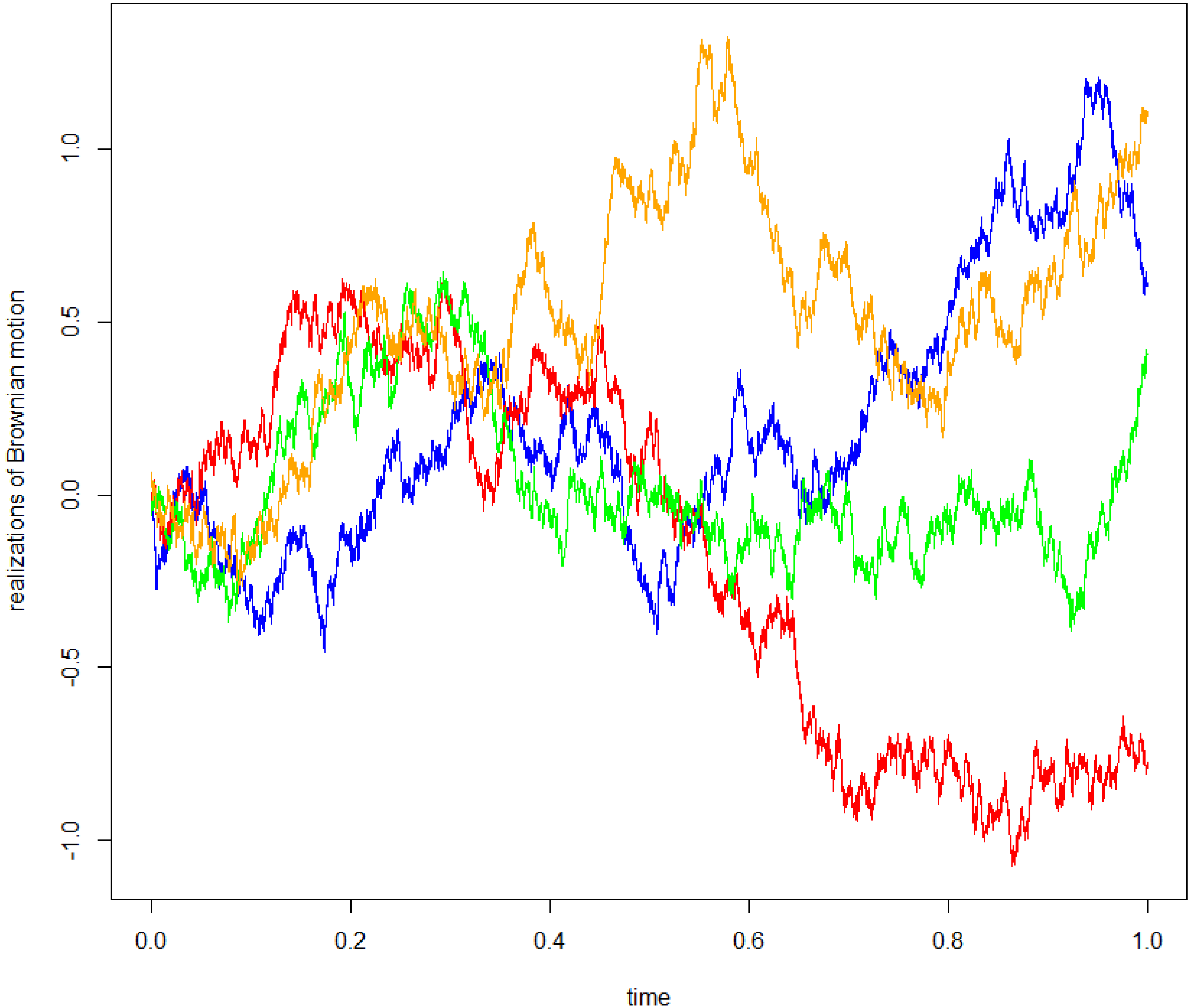
where *z* is a standard normal random variable.

- Martingale property: best possible prediction of future value is most recently observed value

$$E[w_t \mid w_s] = w_s, \forall t \geq s$$

Simulation of Brownian motions

```
set.seed(1234)
# simulate brownian motion
M = 10000
delta = 1/M
z1 = rnorm(M)*sqrt(delta)
z2 = rnorm(M)*sqrt(delta)
z3 = rnorm(M)*sqrt(delta)
z4 = rnorm(M)*sqrt(delta)
w1 = cumsum(z1); w2 = cumsum(z2); w3 = cumsum(z3); w4 = cumsum(z4)
vt = seq(0,1,length.out=M)
plot( vt, w1 , ylim = c( min(c(w1,w2,w3,w4)), max(c(w1,w2,w3,w4)) ) ,
col="blue" , type="l", xlab="time",ylab="realizations of Brownian
motion")
lines( vt , w2 , col="red")
lines( vt , w3 , col="green")
lines( vt , w4 , col="orange")
```


realizations of Brownian motion
1.0
0.5
0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0
time

Including jumps

- For simplicity, assume jumps have finite activity: this means that for each interval the number of jumps that can happen is finite
- The cumulative number of jumps is given by a count process q_s (eg Binomial)
- The magnitude of the jump is given by the process κ_s
- The brownian semimartingale with finite activity jumps is then:

$$\boxed{dp_s = \mu_s ds + \sigma_s dw_s + \kappa_s dq_s}$$

- Different ex post measures of variability can be considered:

  - Quadratic variation: Total variation;
  - Integrated variance: Only the smooth variation;
  - Jump variance: Only the jump variation;
  - Jump tests: Is there a significant jump variability observed on a given day?

Realized variance

- Realized variance is the sum of squared intraday returns

$$RV = \sum_{i=1}^{M} r_{t,i}^2$$

- When $\Delta \rightarrow 0$, the realized variance converges to the quadratic variation, which under the BSMFAJ model equals the integrated variance + sum of squared intraday jumps:

$$QV = \lim_{M \to \infty} \sum_{i=1}^{M} r_{t,i}^2 = \int_0^1 \sigma_s^2 ds + \sum_i \kappa_i^2$$

Realized bipower variation

- Sometimes we only wish to estimate the integrated variance
- Jumps have finite activity: the probability that two contiguous returns have a jump component is 0 almost surely.
- Two continuous returns have almost the same spot variance
- The impact of the product between a “continuous” return and a return with a jump component is neglible
- Hence the realized bipower variation is consistent for the Ivar

$$\boxed{RBV = \frac{\pi}{2}\sum_{i=2}^{M} |r_{t,i}|\,|r_{t,i-1}| \xrightarrow{\Delta \to 0} \int_0^1 \sigma_s^2 ds}$$

(the correction factor π/2 corresponds to the inverse of the square of the expected value of a standard normal random variable)

- Since:

$$RV = \sum_{i=1}^{M} r_{t,i}^2 \xrightarrow{\Delta \to 0} \int_0^1 \sigma_s^2 ds + \sum_i \kappa_i^2$$

$$RBV = \frac{\pi}{2} \sum_{i=2}^{M} | r_{t,i} || r_{t,i-1} | \xrightarrow{\Delta \to 0} \int_0^1 \sigma_s^2 ds$$

- We have that:

$$RV - RBV \xrightarrow{\Delta \to 0} \sum_i \kappa_i^2$$

Other robust estimators exist

- MedRV

$$medRV = c\sum_{i=2}^{M-1} median(|r_{t,i-1}|,|r_{t,i-1}|,|r_{t,i}|)^2 \xrightarrow{\Delta\to 0} \int_0^1 \sigma_s^2 ds$$

(c is a correction factor to ensure consistency)

- ROWVar

$$ROWVar = c_k \sum_{i=1}^{M} r_{t,i}^2 I[\frac{|r_{t,i}|}{\sigma_{t,i}\sqrt{\Delta}} \leq k] \xrightarrow{\Delta\to 0} \int_0^1 \sigma_s^2 ds$$

- Because of estimation error, the jump robust estimator and the total variation estimator will always give a different number in finite samples;
- How to decide whether that difference is large enough to say that there has been a price jump?

Jump test

- H_0: no jump on day t
- H_A: at least one jump on day t
- Under H_0 the RV and robust alternatives estimate the same quantity (IV): the difference is estimation error that is normally distributed around 0 and variance proportional to the integrated quarticity

$$\boxed{\sqrt{M}\left(RV - RBV\right) \xrightarrow{d} N(0, \theta \int_0^1 \sigma_s^4 ds)}$$

Estimation of integrated quarticity

- MedRQ

$$medRQ = c\sum_{i=2}^{M-1} median(|r_{t,i-1}|,|r_{t,i-1}|,|r_{t,i}|)^4 \xrightarrow{\Delta\to 0} \int_0^1 \sigma_s^4 ds$$

(c is a correction factor to ensure consistency)

- Then the jump test statistic is:

$$\sqrt{M}\frac{\left(RV-\hat{I}V\right)}{\sqrt{\theta\hat{I}Q}} \xrightarrow{d} N(0,1)$$

Example

```
data(sample_tdata)
BNSjumptest(sample_tdata$PRICE[1:79], QVestimator= "RV",
IVestimator= "medRV", IQestimator = "medRQ", type=
"linear", makeReturns = TRUE)
```

```
> data(sample_tdata)
> BNSjumptest(sample_tdata$PRICE[1:79], QVestimator= "RV", IVestimator= "medRV",
+             IQestimator = "medRQ", type= "linear", makeReturns = TRUE)
$ztest
[1] -0.6748374

$critical.value
[1] -1.959964  1.959964

$pvalue
[1] 0.4997791
```

- Other jump tests implemented in highfrequency:
  - Ait-Shalia and Jacod: AJjumptest,
  - Jian and Oomen: JOjumptest

Extensions: Microstructure noise

- In practice, we don’t observe the efficient price, but the price with some microstructure noise (rounding, bid-ask bounce)

$$\boxed{p_{t_i}^{observed} = p_{t_i}^{efficient} + \varepsilon_{t_i}}$$

- Then there are three sources of variability to distinguish: IV, jump variance and the noise variance:
  - Two time scale estimator: function (R)TSCov in highfrequency
  - Preaveraging: function MRCov in highfrequency

Extensions: Multivariate analysis

- Asset pricing models, portfolio selection, hedging, arbitrage strategies, value-at-Risk forecasts: they typically need a multivariate approach and require a covariance estimate
- Same challenges as in the univariate case because of the three sources of variability, in addition to estimation troubles coming from non-synchronous trading
  - If ignored: leads to underestimation of dependence.
- In highfrequency:
  - Multivariate refresh time sampling
  - Several covariance estimators: rCov, rHYCov, rTSCov, rThresholdCov, rOWCov, MRC,…

FORECASTING THE REALIZED VOLATILITY (OPEN TO CLOSE)

HAR model

- Realized volatilities model the open to close variabilities
- It’s of interest to forecast future open to close variability
- This is done through a **Heterogeneous AutoRegressive model** in which the RV is predicted based on averages of k past RV
  - Lagged RV (k=1)
  - Average RV of the past week (k=5)
  - Average RV of the past month (k=22)

$$\boxed{RV_t = \beta_0 + \beta_1 RV_{t-1} + \beta_2 \sum_{i=1}^{5} RV_{t-i} + \beta_3 \sum_{i=1}^{22} RV_{t-i} + \varepsilon_t}$$

- Note: parsiomonious, linear in parameters, so OLS estimation.

```
# Forecasting daily Realized volatility for DJI 2008
# using the basic harModel: HARRV and give RVs as
# input
data(realized_library);
#Get sample daily Realized Volatility data
DJI_RV =
realized_library$Dow.Jones.Industrials.Realized.Variance; #Select DJI
DJI_RV = DJI_RV[!is.na(DJI_RV)]; #Remove NA's
DJI_RV = DJI_RV['2008'];
x = harModel(data=DJI_RV , periods = c(1,5,22),
RVest = c("rCov"), type="HARRV",h=1,transform=NULL);
summary(x);
plot(x);
```

harModel giving the RV as input

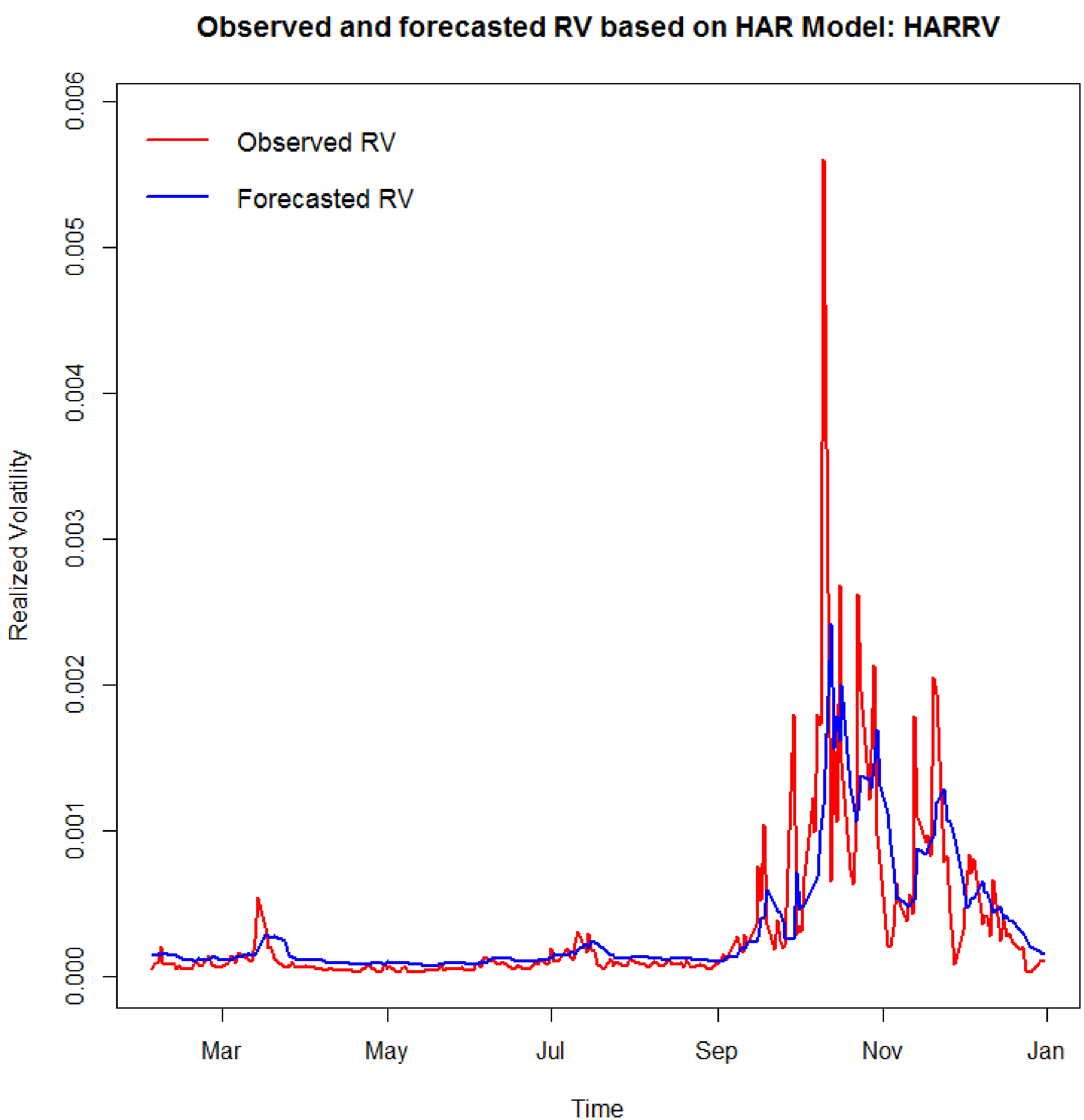
```
> summary(x);

Call:
"RV1 = beta0  +  beta1 * RV1 +  beta2 * RV5 +  beta3 * RV22"

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0017683 -0.0000626 -0.0000427 -0.0000087  0.0044331 

Coefficients:
       Estimate Std. Error t value Pr(>|t|)    
beta0 4.432e-05  3.695e-05   1.200   0.2315    
beta1 1.586e-01  8.089e-02   1.960   0.0512 .  
beta2 6.213e-01  1.362e-01   4.560 8.36e-06 ***
beta3 8.721e-02  1.217e-01   0.716   0.4745    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0004344 on 227 degrees of freedom
Multiple R-squared:  0.4679, Adjusted R-squared:  0.4608 
F-statistic: 66.53 on 3 and 227 DF,  p-value: < 2.2e-16
```


Observed and forecasted RV based on HAR Model: HARRV
Observed RV
Forecasted RV
Realized Volatility
0.000
0.001
0.002
0.003
0.004
0.005
0.006
Mar
May
Jul
Sep
Nov
Jan
Time

- Several extensions of the HAR model: including jumps (type == "HARRVJ“), leverage effects (not implemented yet).

- Limitation: open to close variability

- For forecasting close to close variance using realized measures: see the heavyModel in highfrequency (non-linear: QML estimation).


THE END
IS
NEAR!
LOLME.ORG

Roadmap (i)

- "There are old traders, there are bold traders, but there are no old bold traders"
- Focus: How to use high frequency price data to understand better the time-varying risk properties of the investment
- Two types of risk: the normal volatility risk and the jump risk
- Topics:
  - Cleaning and aggregation (univariate and multivariate) of tick prices into log-returns
  - Discrete time model for intraday returns:
    - Spot volatility estimation
    - Price jump detection
  - Continuous time model for log-prices
    - Realized volatility estimation
    - Detection of a jump component in realized volatility
  - Forecasting volatility using realized volatility measures.

Roadmap (ii)

- And how to do these analysis with the functions in the R package **highfrequency**
  - Latest version at: http://r-forge.r-project.org/R/?group_id=1409
- Other functionality: Calculation of liquidity measures (effective spreads, depth imbalance, etc.)
- Convert large multiday files from WRDS, TAQ, Tickdata into xts objects organized by day
- Realized higher order moments: rSkew, rKurt
- **Your contribution?**

References

- Plenty!
- Some of my own:

  - Boudt, K, Laurent, S., Lunde, A. and Quaedvlieg, R. 201x. Positive Semidefinite Integrated Covariance Estimation, Factorizations and Asynchronicity. Wp.
  - Boudt K. and Zhang, J. (2015). Jump robust two time scale covariance estimation and realized volatility budgets. Quantitative Finance, 15, 1041-1054.
  - Boudt K. and Petitjean M. (2014). Intraday liquidity dynamics and news releases around price jumps: Evidence from the DJIA stocks. Journal of Financial Markets 17, 121-149.
  - Boudt, K., Cornelissen, J. and Croux, C. (2012) Jump robust daily covariance estimation by disentangling variance and correlation components. Computational Statistics & Data Analysis 56, 2993-3005.
  - Boudt, K., Croux, C. and Laurent, S. (2011) Outlyingness weighted covariation. Journal of Financial Econometrics 9, 657-684.
  - Boudt, K., Croux, C. and Laurent, S. (2011) Robust estimation of intraweek periodicity in volatility and jump detection. Journal of Empirical Finance 18, 353-369.