Realized Volatility

Bernt Arne Ødegaard

24 November 2021

1 Realized volatility

While not necessarily linked to microstructure per see, the concept of realized volatility has gotten its prominence because of the availability of high frequency data at the transaction level.

Realized volatility is the estimation of sample volatility using all transactions using a short period, such as a day.

1.1 Intuition

The intuition behind the use of short periods to estimate volatility is found in Merton (1980). He makes the point that for estimating means (first moment), it does not help to increase the frequency of observations within a given interval, you need to increase the total time to improve the precision of a first moment estimate. For a second moment estimate, that is not the case. For a fixed total time, the precision of the volatility estimate increases if you slice the time period into smaller pieces (higher frequency).

[See lecture notes on Merton's result]

1.2 Estimation of volatility for short periods

So, even if you just use the transactions within a day, you get a consistent estimate of the second moment.

Such short period estimates are for example useful for *volatility forecasting*

Definition (from Andersen, Bollerslev, and Diebold (2010))

> The realized volatility over $[t-h,t]$, for $0 < h \leq t \leq T$, is defined by
>
> $$v^2(t,h;n) = \sum_{i=1}^{n} r(t-h+(i/n)\cdot h, h/n)^2$$
>
> The realized volatility is simply the second (uncentered) sample moment of the return process over a fixed interval of length h, scaled by the number of observations n (corresponding to the sampling frequency $1/n$) so that it provides a volatility measure calibrated to the h-period measurement interval.

2 R tools relevant for analysis of high frequency data

Let us discuss some of the R tools you may need if you run into high frequency data.

2.1 Specifying a day and a time

First issue: Reading in data which is at a higher frequency than daily: need a way of indexing observations.

Two classes:

- `POSIXct`

  Class ‘"POSIXct"’ represents the (signed) number of seconds since the beginning of 1970 (in the UTC time zone) as a numeric vector.

- `POSIXlt`

  Class ‘"POSIXlt"’ is a named list of vectors representing

  - ‘sec’ 0-61: seconds.
  - ‘min’ 0-59: minutes.
  - ‘hour’ 0-23: hours.
  - ‘mday’ 1-31: day of the month
  - ‘mon’ 0-11: months after the first of the year.
  - ‘year’ years since 1900.
  - ‘wday’ 0-6 day of the week, starting on Sunday.
  - ‘yday’ 0-365: day of the year.
  - ‘isdst’ Daylight Saving Time flag. Positive if in force, zero if not, negative if unknown.
  - ‘zone’ (Optional.) The abbreviation for the time zone in force at that time: ‘""’ if unknown (but ‘""’ might also be used for UTC).
  - ‘gmtoff’ (Optional.) The offset in seconds from GMT: positive values are East of the meridian. Usually ‘NA’ if unknown, but ‘0’ could mean unknown.

Questions:
Anybody spot any problems using these classes on current high-frequency datasets?
Which is likely to be most efficient in terms of computer resources?
Illustrate the reading of data. Have a file with the structure

```
date,price,quantity
19970901 09:51:00,395,105
19970901 09:51:00,398,62
19970901 10:05:00,402,260
19970901 10:05:00,402,4
19970901 10:10:00,405,500
....
```

This is read in a follows:

```
data <- read.csv("../data/NHY_trades.csv",header=TRUE)
times <- as.POSIXct(data[,1],format = "%Y%m%d %H:%M:%S")
prices <- xts(data[,2],times)
volumes <- xts(data[,3],times)
```

The special feature here is the conversion to POSIXct, where you specify

```
"%Y%m%d %H:%M:%S"
```

here %H is the hour, %M the minute, and %S the second.

Time series classes like `xts` and `zoo` will work with date-time classes as index. But be careful. *zoo* can not deal with non-unique index values. `xts` can, but makes few guarantees about results with non-unique time indices.

2.2 `highfrequency`

This is the most important library if you are dealing with microstructure data.

It comes with numerous functions to do spesific analyses typically done with high frequency data.

- Read trade/quote type datasets. (But can also deal with sequences of trades, only)
- Numerous time series calculation

  realized volatility, testing for jumps, other high-frequency time series analyses
- Liquidity calculations

We will look at this library in the context of calculation of realized volatility.

2.3 Example: Norwegian cross section

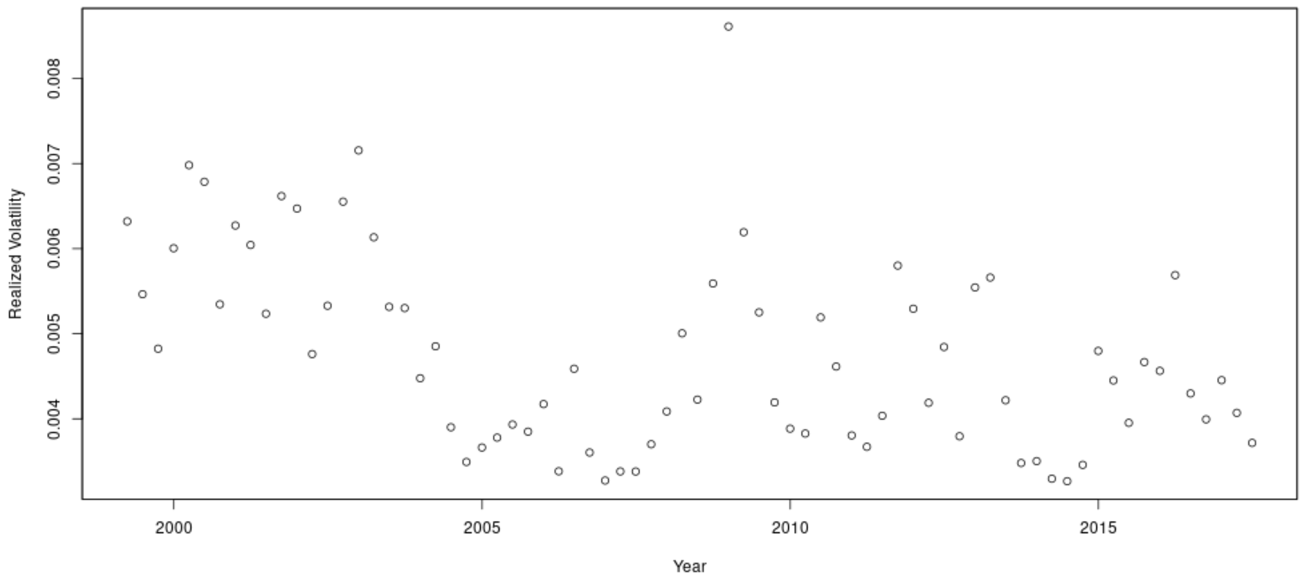
Look at the situation in the Norwegian stock market.

For each stock on the exhange, a simple daily RV is calculated from 15-minute returns.

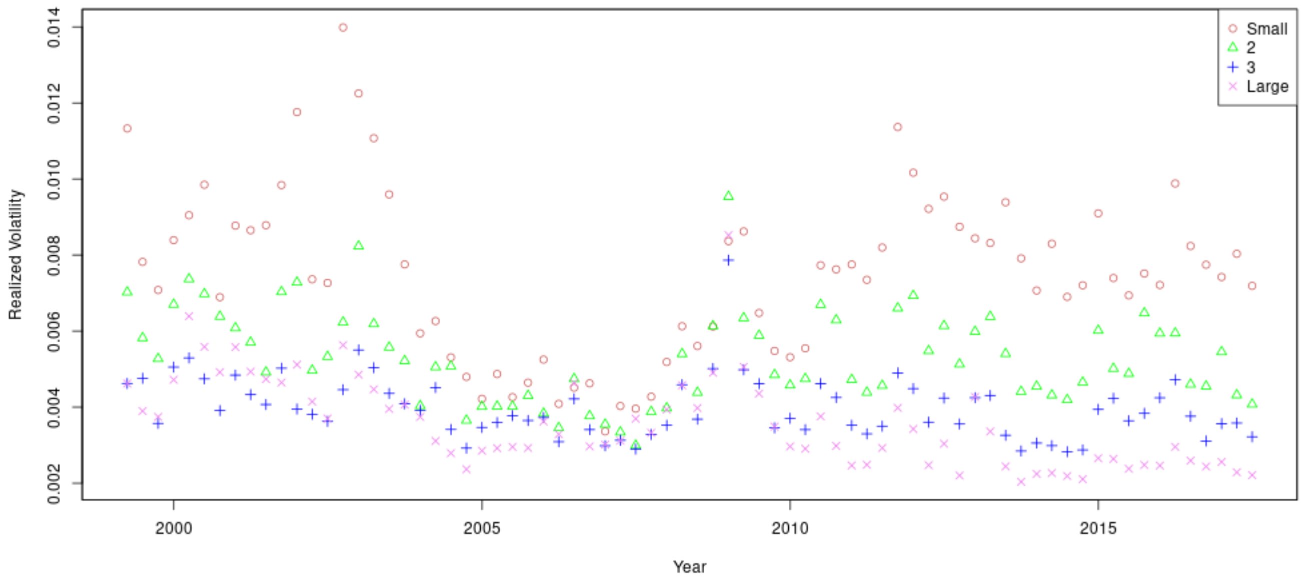
We then aggregate the data into pictures.

Lets look at monthly frequency. First, average RV across all days in a month for each stock.

First, whole exchange, medians of these monthly RV estimates



Then, sorted into size portfolios, medians of monthly RV estimates



While the definition looks simple, estimation has become a very sophisticated exercise.

This is motivated by the fact that the data it is most often applied to, high frequency data, has some issues:

- Possible errors in prices, timing.
- Sudden changes in levels (jumps) (e.g. changes in prices between close and opening)

Want measures of realized volatility robust to such jumps. To illustrate the complexity of the resulting estimators, let us take two examples from Andersen, Dobrev, and Schaumburg (2012)

$$minRV = \frac{\pi}{\pi - 2}\left(\frac{N}{N-1}\right)\sum_{i=1}^{N-1}\min(|\Delta Y_i|, |\Delta Y_{i+1}|)^2$$

$$medRV = \frac{\pi}{6 - 4\sqrt{3} + \pi}\left(\frac{N}{N-2}\right)\sum_{i=1}^{N-1}\text{med}(|\Delta Y_{i+1}|, |\Delta Y_i|, |\Delta Y_{i+1}|)^2$$

Here N is the sample size and Y_i the observations.

The concept of Realized Volatility is most easily introduced by an example

Exercise 1.

We consider the *Realized Volatility* of the stock of Norsk Hydro. You have access to the daily records of trades in NHY for the first half of december 2012.

First plot the time series of tick-by-tick trades.

Then calculate the realized volatility for each of the dates for which you have trades.

Use two measures of realized volatility `medRV` and `minRV`.

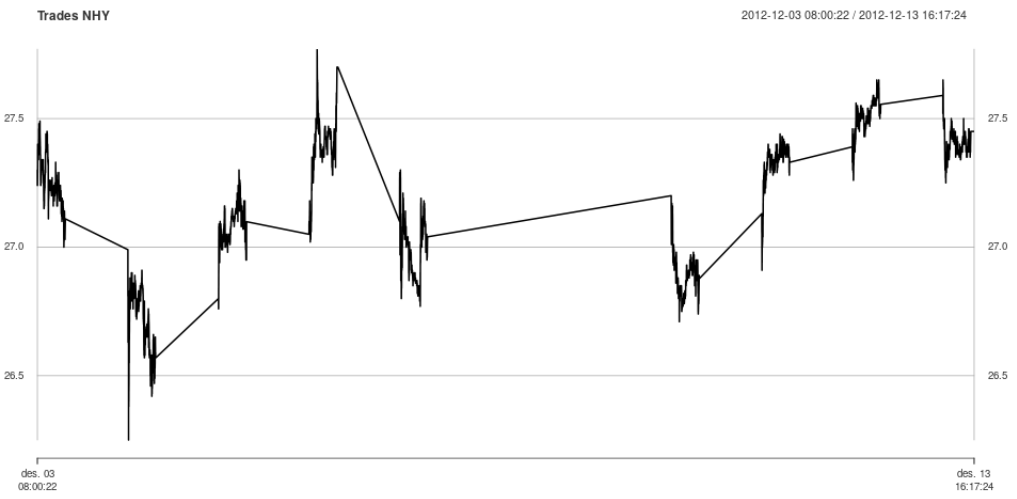
Plot the daily estimates of realized volatility and compare them to the trade records. (You may want to make price plots for each day to make the comparison.

Solution to Exercise 1.

Reading the data is somewhat more involved here. We need to create an `xts` type object indexed by a date and time object.

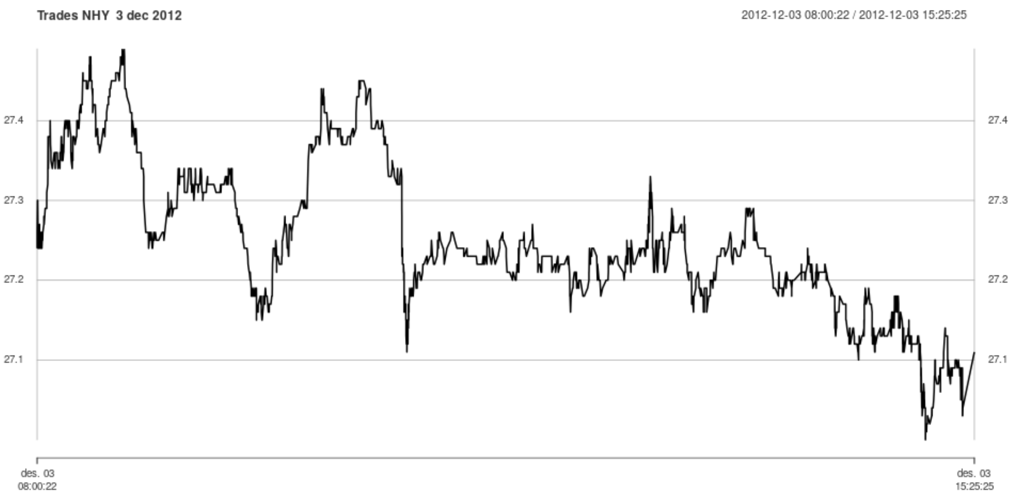
```
library(xts)
options(digits.sec=4)
tradedata12  <-  read.table("../data/nhy_dec_2012_trades.csv",sep=",",header=TRUE)
times12 <- as.POSIXct(tradedata12[,1],format = "%Y%m%d %H:%M:%S")
prices12 <- xts(tradedata12[,2],times12)
names(prices12)[1]="Price"
volumes12 <- xts(tradedata12[,3],times12)
names(volumes12)[1]="Volume"
```

Let us look at the trade prices



See that there is a lot of movement within the trading day, and some large movements between days. (The opening is done by an auction).

Let us look at the first day (3 dec)



To work with this kind of high frequency data we use the R library `highfrequency`.

```
> library("highfrequency")
```

The realized volatility uses the trading during the day as a basis for estimating volatility. The first step is to create equidistant price observations as a basis for calculating returns. We need to choose the interval between observations we use for sampling.

```
> TradesOneDay12 <- prices12['2012-12-03']
> tr <- aggregatets(TradesOneDay12,on="minutes",k=1)
```

Here we observe prices every minute, which is then the basis for calculating returns.

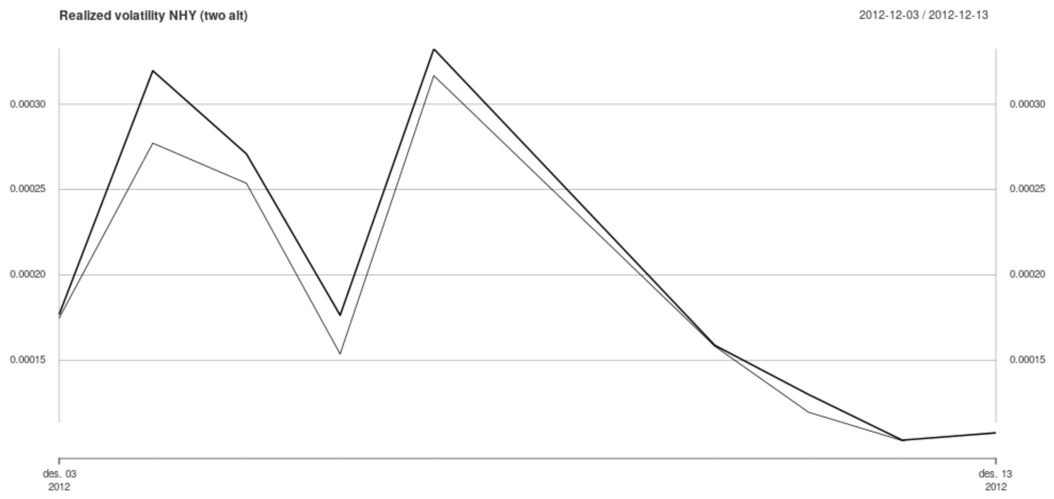
```
> rets <- makeReturns(tr)
> rets <- na.omit(rets)
```

Here are two etimates of realized volatility

```
> mr1 <- medRV(rets,makeReturns=FALSE)
> print(mr1)
[1] 0.0001611687
> mr2 <- minRV(rets,makeReturns=FALSE)
> print(mr2)
[1] 0.0001604513
```

Not a major difference between the two estimation methods.

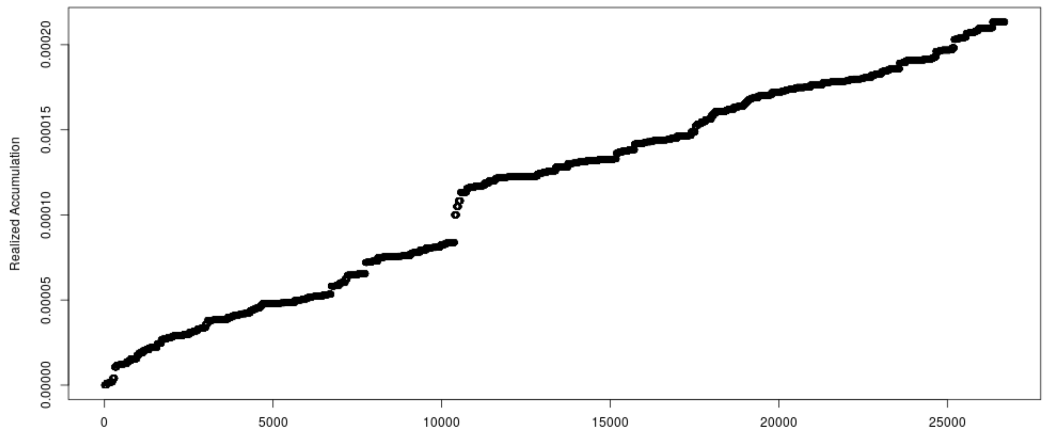
Doing this calculation for each of the trading days, we find the following time series of realized volatility.



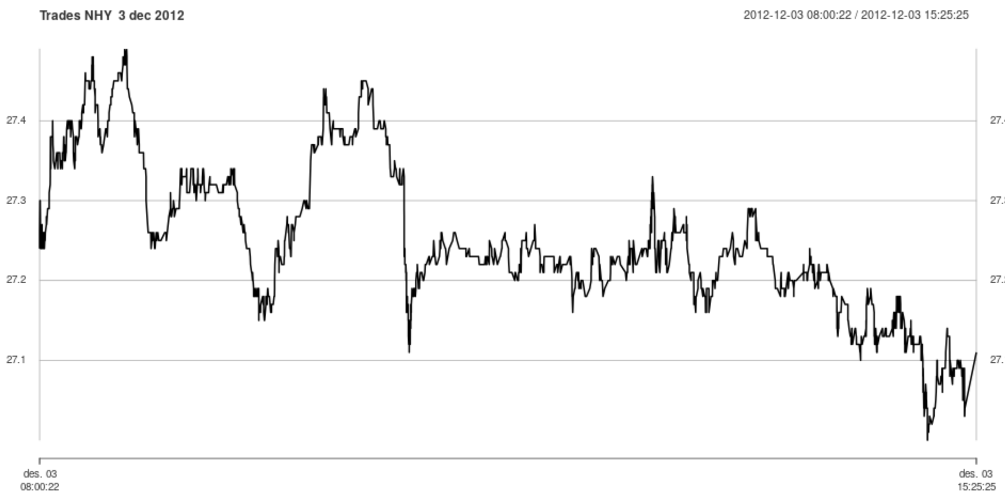
Let us first look at the first date, 3 dec.

An interesting time series is created by `rAccumulation`. This accumulates the squared volatilities, and shows where during the day the volatility contribution happens

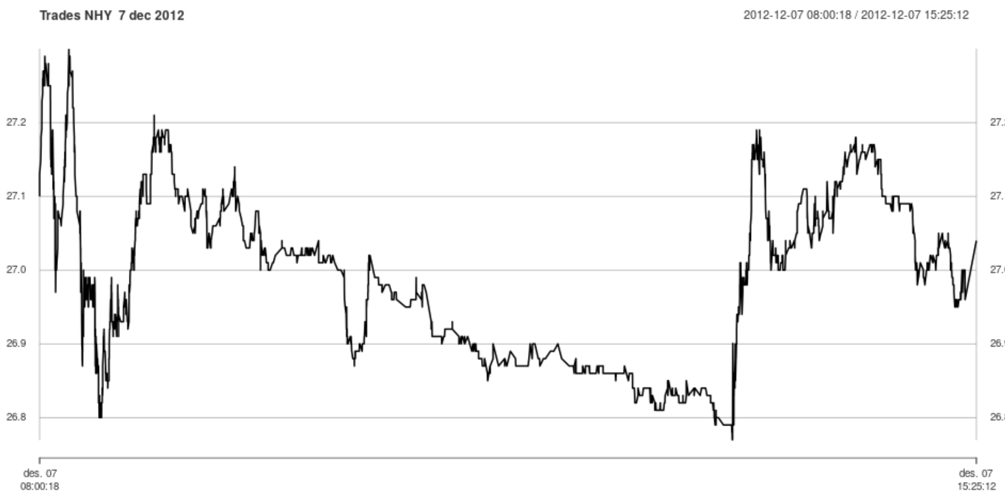
```
a <- rAccumulation(rets,plotit=TRUE)
```



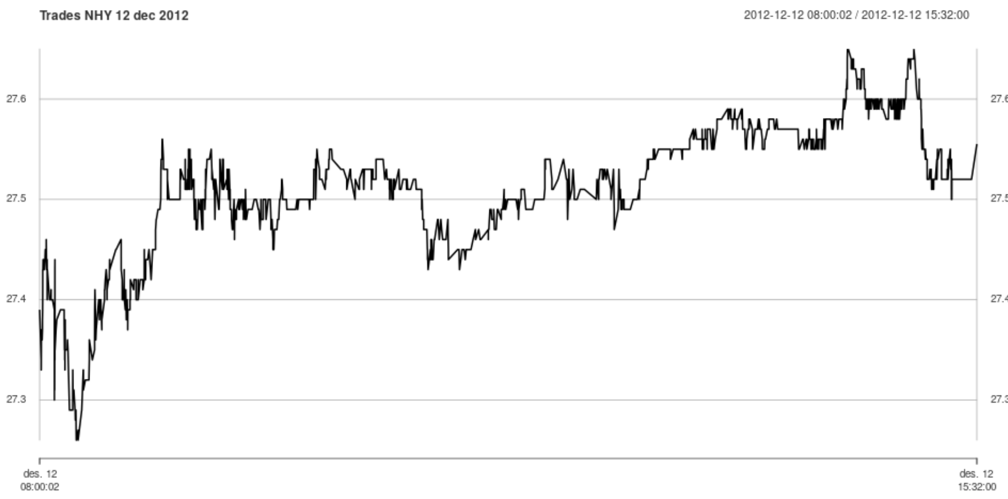
Comparing this picture with the daily trade record:



The huge increase in the accumulated RV is related to the huge drop at one point during the day.
Now, there is clear differences in RV estimates for the different days.
Let us compare the daily trade records for the day with the highest RV (7 dec)



with the one with the lowest RV (12 dec)



Observe that the day with the highest RV also has some large up *and* down movements.

3 Literature

Early paper:

Andersen and Benzoni (2008) is a quick introduction to RV.
For a recent survey of volatility modelling, including RV, see Andersen et al. (2010)
For computer modelling of RV in R, see Boudth, Cornelissen, and Payseur (2013)
For a statistical perspective, see Mykland and Zhang (2012)

References

Torben G Andersen and Luca Benzoni. Realized volatility. Working Paper Federal Reserve Bank of Chicago, to be published in, *Handbook of Financial Time Series*, 2008.

Torben G Andersen, Tim Bollerslev, and Francis X Diebold. Parametric and nonparametric volatility estimation. In Yacine Ait-Sahalia and Lars Peter Hansen, editors, *Handbook of Financial Econometrics*. North-Holland, 2010.

Torben G. Andersen, Dobrislav Dobrev, and Ernst Schaumburg. Jump-robust volatility estimation using nearest neighbor truncation. *Journal of Econometrics*, 169(1):75 – 93, 2012.

Kris Boudth, Jonathan Cornelissen, and Scott Payseur. Highfrequency: Toolkit for the analysis of highfrequency financial data in r. Working paper, CRAN, April 2013.

Robert C Merton. On estimating the expected return on the market. *Journal of Financial Economics*, pages 323–362, 1980.

P A Mykland and L Zhang. The econometrics of high frequency data. In *Statistical Methods for Stochastic Differential Equations*, pages 109–190. 2012.