Package 'PortfolioEffectEstim'

October 12, 2022

Type Package

Title High Frequency Price Estimators by PortfolioEffect

Version 1.4

Date 2016-09-17

Depends methods,PortfolioEffectHFT(>= 1.7)

Imports rJava

LazyData yes

ByteCompile TRUE

Maintainer Andrey Kostin <andrey.kostin@portfolioeffect.com>

Description R interface to PortfolioEffect cloud service for estimating high frequency price variance, quarticity, microstructure noise variance, and other metrics in both aggregate and rolling window flavors. Constructed estimators could use client-side market data or access HF intraday price history for all major US Equities. See <https://www.portfolioeffect.com/> for more information on the PortfolioEffect high frequency portfolio analytics platform.

URL https://www.portfolioeffect.com/

License GPL-3

SystemRequirements Java (>= 1.7)

NeedsCompilation no

Repository CRAN

Author Andrey Kostin [aut, cre],
Aleksey Zemnitskiy [aut],
Oleg Nechaev [aut]

Date/Publication 2016-09-17 19:54:52

R topics documented:

`estimator-class` *Class* `"estimator"`

Description

Class for storing java Estimator object.

Slots

`java`**:** Object of class `"jobjRef"` ~~

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

Examples

```
showClass("estimator")
```

estimator_availableSymbols *Get All Symbol List*

Description

Returns a list of symbols .

Usage

```
estimator_availableSymbols(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Value

List of symbols, exchanges and descriptions

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

Examples

```
## Not run:
estimator=estimator_create(asset='AAPL',fromTime="2014-09-01 09:00:00",
toTime="2014-09-14 16:00:00")
list=estimator_availableSymbols(estimator)

## End(Not run)
```

estimator_create *Creates new estimator*

Description

Creates new empty estimator object.

Usage

```
estimator_create(asset, fromTime, toTime, priceData)
```

Arguments

asset	Unique identifier of the instrument
fromTime	Start of market data interval in "yyyy-MM-dd hh:mm:ss" format when internal market data is used. Offset from last available date/time by N days is denoted as "t-N" (e.g. "t-7" denotes offset by 7 days).
toTime	End of market data interval in "yyyy-MM-dd hh:mm:ss" format when internal market data is used. Offset from last available date/time by N days is denoted as "t-N" (e.g. "t-7" denotes offset by 7 days).
priceData	Vector of (time, price) observations for asset when external market data is used.

Value

estimator object

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

See Also

estimator_settings

Examples

```
## Not run:
data(goog.data)
estimator=estimator_create(priceData=goog.data)
estimator_settings(estimator,resultsSamplingInterval='60s')
util_plot2d(variance_rv(estimator),title="RV")


estimator=estimator_create(asset='AAPL',fromTime="2014-09-01 09:00:00",
toTime="2014-09-14 16:00:00")
estimator_settings(estimator,resultsSamplingInterval='60s')
util_plot2d(variance_tsrv(estimator,K=2),title="TSRV")

estimator=estimator_create(asset='GOOG',fromTime="t-2", toTime="t")
estimator_settings(estimator,resultsSamplingInterval='60s')
util_plot2d(variance_mrv(estimator),title="MRV")

## End(Not run)
```

estimator_defaultSettings *Estimator Default Settings*

Description

Advanced settings that regulate how estimator metrics are computed, returned and stored. Default: jumpsModel = "moments", resultsSamplingInterval = "1s", inputSamplingInterval="none"

Usage

```
estimator_defaultSettings(estimator)
```

Arguments

estimator	Estimator object created using estimator_create( ) function

Value

Void

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

See Also

estimator_create estimator_getSettings

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
estimator_getSettings(estimator)
estimator_defaultSettings(estimator)
estimator_getSettings(estimator)

## End(Not run)
```

estimator_getSettings *Get Estimator Settings*

Description

Method returns active list of settings of a given estimator.

Usage

```
estimator_getSettings(estimator)
```

Arguments

estimator	Estimator object created using estimator_create( ) function

Value

List with estimator settings.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

Examples

```
## Not run:
dateStart = "2014-11-17 09:30:00"
dateEnd = "2014-11-17 16:00:00"
estimator=estimator_create('AAPL',dateStart,dateEnd)
estimator_settings(estimator,resultsSamplingInterval='60s')
settings=estimator_getSettings(estimator)

## End(Not run)
```

estimator_settings *Estimator Settings*

Description

Advanced settings that regulate how estimator metrics are computed, returned and stored. Default: jumpsModel = "moments", resultsSamplingInterval = "1s", inputSamplingInterval="none"

Usage

```
estimator_settings(estimator,...)
```

Arguments

estimator	Estimator object created using estimator_create( ) function
...	One of the following estimator settings: "jumpsModel" - Used to select jump filtering mode when computing return statistics. Available modes are: "none" - price jumps are not filtered anywhere, "moments" - price jumps are filtered only when computing moments (variance, skewness, kurtosis) and derived metrics, "all" - price jumps are filtered everywhere. Defaults to "moments", which implies that only return moments and related metrics would be using jump-filtered returns in their calculations. "resultsSamplingInterval" - Interval to be used for sampling computed results before returning them to the caller. Available interval values are: "Xs" - seconds, "Xm" - minutes, "Xh" - hours, "Xd" - trading days (6.5 hours in a trading day), "Xw" - weeks (5 trading days in 1 week), "Xmo" - month (21 trading day in 1 month), "Xy" - years (256 trading days in 1 year), "last" - last result in a series is returned, "none" - no sampling. Large sampling interval would produce smaller vector of results and would require less time spent on data transfer. Default value of "1s" indicates that data is returned for every second during trading hours. "inputSamplingInterval" - Interval to be used as a minimum step for sampling input prices. Available interval values are: "Xs" - seconds, "Xm" - minutes, "Xh" - hours, "Xd" - trading days (6.5 hours in a trading day), "Xw" - weeks (5 trading days in 1 week), "Xmo" - month (21 trading day in 1 month), "Xy" - years (256 trading days in 1 year), "none" - no sampling. Default value is "none", which indicates that no sampling is applied.

Value

Void

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

See Also

estimator_create estimator_getSettings

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_mrv(estimator),title="MRV")

dateStart = "2014-11-17 09:30:00"
dateEnd = "2014-11-17 16:00:00"
```

```
estimator=estimator_create('AAPL',dateStart,dateEnd)
estimator_settings(estimator,
   resultsSamplingInterval = '10s')
util_plot2d(variance_mrv(estimator),title="MRV")

## End(Not run)
```

`noise_acnv` *Autocovariance Noise Variance*

Description

Autocovariance Noise Variance (ACNV) estimates the noise variance based on the autocovariance of returns, rather than the Rescaled Noise Variance (RNV). It is generally preferred to RNV as it leads to a reduction in MSE and is robust to the presence of rare jumps. Also, this approach can be extended straightforwardly to estimate the parameters of higher order noise dependence.

Usage

```
noise_acnv(estimator)
```

Arguments

`estimator`	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/2}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **yes**
- Accounts for time dependence in noise: **yes**
- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

R. C. Oomen, "Comment on realized variance and market microstructure noise by peter r. hansen and asger lunde," pp. 1-15, 23 September, 2005.

See Also

noise_rnv noise_urnv noise_uznv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(noise_acnv(estimator),title="ACNV")

## End(Not run)
```

---

noise_nts *Noise to Signal Ratio*

Description

Noise to Signal Ratio is a measure that compares the level of noise to the level of a desired signal.

Usage

```
noise_nts(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Value

a numeric vector of the same length as data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
```

```
util_plot2d(noise_nts(estimator),title="NTS")

## End(Not run)
```

`noise_rnv` *Rescaled Noise Variance*

Description

Rescaled Noise Variance (RNV) is an asymptotically consistent estimator of noise volatility when daling with additive microstructure noise. It is derived based on Realized Variance property of convergence to noise variance wih the increase of sampling frequency.

Usage

```
noise_rnv(estimator)
```

Arguments

`estimator`	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/2}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **no**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

Bandi, F. M., and J. R. Russell, 2004, "Microstructure Noise, Realized Variance, and Optimal Sampling" manuscript GSB, The University of Chicago. L. Zhang, P. A. Mykland, and Y. Ait-Sahalia, "A tale of two time scales: Determining integrated volatility with noisy high-frequency data," Journal of the American Statistical Association, vol. 100, No. 472, pp. 1394-1411, December 2005.

See Also

noise_acnv noise_urnv noise_uznv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(noise_rnv(estimator),title="RNV")

## End(Not run)
```

---

noise_urnv *Unbiased Rescaled Noise Variance*

Description

Unbiased Rescaled Noise Variance (URNV) corrects for a bias of Rescaled Noise Variance.

Usage

```
noise_urnv(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/2}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **no**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

L. Zhang, P. A. Mykland, and Y. Ait-Sahalia, "A tale of two time scales: Determining integrated volatility with noisy high-frequency data," Journal of the American Statistical Association, vol. 100, No. 472, pp. 1394-1411, December 2005.

See Also

noise_rnv noise_acnv noise_uznv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(noise_urnv(estimator),title="URNV")

## End(Not run)
```

---

noise_uznv *Uncertainty Zones Noise Variance*

Description

Uncertainty Zones Noise Variance (UZNV) based on the concept of uncertainty zones.

Usage

```
noise_uznv(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/2}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **no**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **yes**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

Robert, C. Y. and Rosenbaum, M. (2012), Volatility and covariation estimation when microstructure noise and trading times are endogenous. Mathematical Finance, 22

See Also

noise_rnv noise_urnv noise_acnv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(noise_uznv(estimator),title="UZNV")

## End(Not run)
```

---

price *Get Asset Price*

Description

Method returns active list of settings of a given estimator.

Usage

```
price(estimator)
```

Arguments

estimator	Estimator object created using estimator_create( ) function

Value

numeric vector of prices.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

Examples

```
## Not run:
dateStart = "2014-11-17 09:30:00"
dateEnd = "2014-11-17 16:00:00"
estimator=estimator_create('AAPL',dateStart,dateEnd)
estimator_settings(estimator,resultsSamplingInterval='60s')
AAPL=price(estimator)
util_plot2d(AAPL,title='AAPL')

## End(Not run)
```

---

quarticity_mrq *Modulated Realized Quarticity*

Description

Modulated Realized Quarticity (MRQ) is an asymptotically unbiased estimator of integrated quarticity in the presence of microstructure noise.

Usage

```
quarticity_mrq(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **yes**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrei.kostin@snowfallsystems.com>

References

M. Podolskij and M. Vetter, "Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps," Bernoulli, vol. 15, No. 3, pp. 634-658, 2009

See Also

quarticity_rq quarticity_rqq quarticity_rtq quarticity_mtq

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(quarticity_mrq(estimator),title="MRQ")

## End(Not run)
```

---

quarticity_mtq *Modulated Tripower Quarticity*

Description

Modulated Tri-power Quarticity (MTQ) is an asymptotically unbiased estimator of integrated quarticity in the presence of microstructure noise. This estimator is also robust to finite activity jumps in price_

Usage

```
quarticity_mtq(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **yes**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrei.kostin@snowfallsystems.com>

References

M. Podolskij and M. Vetter, "Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps," Bernoulli, vol. 15, No. 3, pp. 634-658, 2009

See Also

quarticity_rq quarticity_rqq quarticity_rtq quarticity_mrq

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(quarticity_mtq(estimator),title="MTQ")

## End(Not run)
```

---

quarticity_rq *Realized Quarticity*

Description

Realized Quarticity (RQ) is an asymptotically unbiased estimator of integrated quarticity in the absence of microstructure noise.

Usage

```
quarticity_rq(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)
- Accounts for additive noise: **no**
- Accounts for finite price jumps: **no**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrei.kostin@snowfallsystems.com>

References

Barndorff-Nielsen, O. E. and N. Shephard (2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. Journal of the Royal Statistical Society: Series B 64 (2), 253-280.

See Also

quarticity_mrq quarticity_rqq quarticity_rtq quarticity_mtq

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(quarticity_rq(estimator),title="RQ")

## End(Not run)
```

---

quarticity_rqq *Realized Quadpower Quarticity*

Description

Realized Quadpower Quarticity (RQQ) is an asymptotically unbiased estimator of integrated quarticity in the absence of microstructure noise.

Usage

```
quarticity_rqq(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)

- Accounts for additive noise: **no**

- Accounts for finite price jumps: **yes**

- Accounts for time dependence in noise: **no**

- Accounts for endogenous effects in noise: **no**

Author(s)

Kostin Andrey <andrei.kostin@snowfallsystems.com>

References

O. E. Barndorff-Nielsen and N. Shephard. Power and bipower variation with stochastic volatility and jumps. Journal of Financial Econometrics, Vol.2(No.1):1-37,2004

See Also

quarticity_rq quarticity_mrq quarticity_rtq quarticity_mtq

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(quarticity_rqq(estimator),title="RQQ")

## End(Not run)
```

quarticity_rtq *Realized Tripower Quarticity*

Description

Realized Tri-power Quarticity (RTQ) is an asymptotically unbiased estimator of integrated quarticity in the absence of microstructure noise.

Usage

```
quarticity_rtq(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)

- Accounts for additive noise: **no**

- Accounts for finite price jumps: **yes**

- Accounts for time dependence in noise: **no**

- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrei.kostin@snowfallsystems.com>

References

Andersen, T. G., Bollerslev, T., and Diebold, F. X. (2005),"Roughing it Up: Including Jump Components in the Measurement, Modeling and Forecasting of Return Volatility" Tech. rep., NBER

See Also

quarticity_rq quarticity_rqq quarticity_mrq quarticity_mtq

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(quarticity_rtq(estimator),title="RTQ")

## End(Not run)
```

`variance_jrmrv` *Jump Robust Modulated Realized Variance*

Description

Jump Robust Modulated Realized Variance (JRMRV) is an integrated variance estimator introduced by Podolskij and Vetter. It is based on the concept of multipower variation, is robust to finite activity jumps and assumes additive noise structure.

Usage

```
variance_jrmrv(estimator)
variance_jrmrvRolling(estimator,wLength=23400)
```

Arguments

`estimator`	Vector of (time, price) observations for market asset when external market data is used.
`wLength`	Length of a rolling window for rolling estimators. Default window length is 23400 (number of seconds in a trading day)

Details

Converges to integrated variance

- Convergence speed: $m^{1/6}$ (m - number of observation)

- Accounts for additive noise: **yes**

- Accounts for finite price jumps: **yes**

- Accounts for time dependence in noise: **no**

- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

M. Podolskij and M. Vetter, "Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps," Bernoulli, vol. 15, No. 3, pp. 634-658, 2009.

See Also

variance_rv variance_tsrv variance_msrv variance_mrv variance_uzrv variance_krv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_jrmrv(estimator),title='JRMRV',legend='Simple')+
util_line2d(variance_jrmrvRolling(estimator,wLength=3600),legend='Rolling Window')

## End(Not run)
```

---

variance_krv *Kernel Realized Variance*

Description

Kernel Realized Variance (KRV) is an asymptotically consistent estimator of integrated volatility based on the concept of realized kernels for dealing with additive microstructure noise.

Usage

```
variance_krv(estimator,kernelName="ParzenKernel",bandwidth=1)
variance_krvRolling(estimator,kernelName="ParzenKernel",bandwidth=1,wLength=23400)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.
wLength	Length of a rolling window for rolling estimators. Default window length is 23400 (number of seconds in a trading day)
kernelName	Kernel name is one of the following (default:"ParzenKernel") • "BartlettKernel"

- "EpanichnikovKernel"
- "SecondOrderKernel"
- "CubicKernel"
- "ParzenKernel"
- "TukeyHanningKernel"
- "TukeyHanningModifiedKernel"
- "FifthOrderKernel"
- "SixthOrderKernel"
- "SeventhOrderKernel"
- "EighthOrderKernel"

`bandwidth` "optimal" to compute optimal bandwidth from the data, or the value of bandwidth (default:1)

Details

Flat Top kernel types:

(Bartlett, Epanichnikov and Second order kernel)

- Convergence speed: $m^{1/6}$ (m - number of observation)

- Accounts for additive noise: **yes**

- Accounts for finite price jumps: **no**

- Accounts for time dependence in noise: **no**

- Accounts for endogenous effects in noise: **no**

Non Flat Top kernel types:

(Cubic,Parzen,Tukey Hanning,Tukey Hanning modified and 5,6,7,8 order kernel)

- Convergence speed: $m^{1/4}$ (m - number of observation)

- Accounts for additive noise: **yes**

- Accounts for finite price jumps: **no**

- Accounts for time dependence in noise: **yes**

- Accounts for endogenous effects in noise: **yes**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

O.E.Barndorff-Nielsen, P.Reinhard Hansen, A.Lunde, and N.Shephard, "Designing realised kernels to measure the ex-post variation of equity prices in the presence of noise", Economics Series Working Papers 264, University of Oxford, Department of Economics, 2006.

See Also

variance_rv variance_tsrv variance_msrv variance_mrv variance_uzrv variance_jrmrv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_krv(estimator,kernelName="EpanichnikovKernel"),
title='KRV',legend='Simple')+
util_line2d(variance_krvRolling(estimator,kernelName="ParzenKernel",
wLength=3600),legend='Rolling Window')

## End(Not run)
```

---

variance_mrv *Modulated Realized Variance*

Description

Modulated Realized Variance (MRV) is an integrated variance estimator introduced by Podolskij and Vetter. It is based on the concept of multipower variation and assumes additive noise structure.

Usage

```
variance_mrv(estimator)
variance_mrvRolling(estimator,wLength=23400)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.
wLength	Length of a rolling window for rolling estimators. Default window length is 23400 (number of seconds in a trading day)

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)

- Accounts for additive noise: **yes**

- Accounts for finite price jumps: **no**

- Accounts for time dependence in noise: **no**

- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

M. Podolskij and M. Vetter, "Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps," Bernoulli, vol. 15, No. 3, pp. 634-658, 2009.

See Also

variance_rv variance_tsrv variance_msrv variance_jrmrv variance_uzrv variance_krv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_mrv(estimator),title='MRV',legend='Simple')+
util_line2d(variance_mrvRolling(estimator,wLength=3600),legend='Rolling Window')

## End(Not run)
```

---

variance_msrv *Multiple Scales Realized Variance*

Description

Multiple Series Realized Variance (MSRV) is a generalization of the TSRV estimator of integrated volatility. It uses multiple time scales to account for the effects of additive market microstructure noise.

Usage

```
variance_msrv(estimator,K=2,J=1)
variance_msrvRolling(estimator,K=2,J=1,wLength=23400)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.
K	number of subsamples in the slow time series (default: 2)
J	number of subsamples in the fast time series (default: 1)
wLength	Length of a rolling window for rolling estimators. Default window length is 23400 (number of seconds in a trading day)

Details

- Convergence speed: $m^{1/4}$ (m - number of observation)

- Accounts for additive noise: **yes**

- Accounts for finite price jumps: **no**

- Accounts for time dependence in noise: **yes**

- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

Zhang, L. (2006). Efficient estimation of stochastic volatility using noisy observations: A multiscale approach.

See Also

variance_rv variance_tsrv variance_jrmrv variance_mrv variance_uzrv variance_krv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_msrv(estimator),title='MSRV',legend='Simple')+
util_line2d(variance_msrvRolling(estimator,wLength=3600),legend='Rolling Window')

## End(Not run)
```

variance_rv *Realized Variance*

Description

Realized Variance (RV) is the sum of squared returns. For instance the RV can be the sum of squared daily returns for a particular month, which would yield a measure of price variation over this month. This variance estimator does not account for market microstructure effects.

Usage

```
variance_rv(estimator)
variance_rvRolling(estimator,wLength=23400)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.
wLength	Length of a rolling window for rolling estimators. Default window length is 23400 (number of seconds in a trading day)

Details

- Convergence speed: $m^{1/2}$ (m - number of observation)
- Accounts for additive noise: **no**
- Accounts for finite price jumps: **no**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **no**

Value

A vector of integrated variance estimates

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

T. G. Andersen, T. Bollerslev, F. X. Diebold, and P. Labys. "The distribution of realized exchange rate volatility". Journal of American Statistical Association, 96(453):4255, March 2001. Barndorff-Nielsen, O. E. and N. Shephard (2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. Journal of the Royal Statistical Society: Series B 64 (2), 253-280.

See Also

variance_jrmrv variance_tsrv variance_msrv variance_mrv variance_uzrv variance_krv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_rv(estimator),title='RV',legend='Simple')+
util_line2d(variance_rvRolling(estimator,wLength=3600),legend='Rolling Window')

## End(Not run)
```

---

variance_tsrv *Two Scales Realized Variance*

Description

Two Scale Realized Variance (TSRV) estimates integrated volatility consistently. The idea is to use realized variance type estimators over two time scales to correct the effect of additive market microstructure noise.

Usage

```
variance_tsrv(estimator,K=2)
variance_tsrvRolling(estimator,K=2,wLength=23400)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.
K	number of subsamples in the slow time series (default: 2)
wLength	Length of a rolling window for rolling estimators. Default window length is 23400 (number of seconds in a trading day)

Details

- Convergence speed: $m^{1/6}$ (m - number of observation)

- Accounts for additive noise: **yes**

- Accounts for finite price jumps: **no**

- Accounts for time dependence in noise: **no**

- Accounts for endogenous effects in noise: **no**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

L. Zhang, P. A. Mykland, and Y. Ait-Sahalia, "A tale of two time scales: Determining integrated volatility with noisy high-frequency data," Journal of the American Statistical Association, vol. 100, No. 472, pp. 1394-1411, December 2005.

See Also

variance_rv variance_jrmrv variance_msrv variance_mrv variance_uzrv variance_krv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_tsrv(estimator),title='TSRV',legend='Simple')+
util_line2d(variance_tsrvRolling(estimator,wLength=3600),legend='Rolling Window')

## End(Not run)
```

---

variance_uzrv *Uncertainty Zones Realized Variance*

Description

Uncertainty Zones Realized Variance (UZRV) is an integrated variance estimator that accounts for stochastic rounding noise like bid-ask bounce effects.

Usage

```
variance_uzrv(estimator)
```

Arguments

estimator	Vector of (time, price) observations for market asset when external market data is used.

Details

- Convergence speed: $m^{1/2}$ (m - number of observation)
- Accounts for additive noise: **yes**
- Accounts for finite price jumps: **no**
- Accounts for time dependence in noise: **no**
- Accounts for endogenous effects in noise: **yes**

Value

a numeric vector of the same length as input data.

Author(s)

Kostin Andrey <andrey.kostin@portfolioeffect.com>

References

Robert, C. Y. and Rosenbaum, M. (2012), Volatility and covariation estimation when microstructure noise and trading times are endogenous. Mathematical Finance, 22

See Also

variance_rv variance_tsrv variance_msrv variance_mrv variance_jrmrv variance_krv

Examples

```
## Not run:
data(spy.data)
estimator=estimator_create(priceData=spy.data)
estimator_settings(estimator,
   inputSamplingInterval = '10s',
   resultsSamplingInterval = '10s')
util_plot2d(variance_uzrv(estimator),title="UZRV")

## End(Not run)
```

Index