# Package 'realized'

January 15, 2012

**Version** 1.0.1

**Date** 2010-08-04

**Title** Realized

**Author** Scott Payseur <scott.payseur@gmail.com>

**Maintainer** Scott Payseur <scott.payseur@gmail.com>

**Description** Realized Variance Toolkit

**License** GPL (>= 2)

**Depends** R (>= 2.10)

**Repository** CRAN

**Date/Publication** 2012-01-15 09:51:55

## R topics documented:

---

`realized-package` *Realized Variance, Covariance and Correlation*

---

## Description

Package contains various estimators and tools to research realized estimates of second moments.

## Details

| | |
|---|---|
| Package: | realized |
| Type: | Package |
| Version: | 0.7 |
| Date: | 2007-08-05 |
| License: | |

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

T.G. Andersen, T. Bollerslev, F.X. Diebold, and P. Labys. Great realizations. *Risk*, 13:105 108, 2000.

Y. Fang. Volatility modeling and estimation of high-frequency data with gaussian noise. *unpublished doctoral thesis, MIT, Sloan School of Management*, 1996.

O. E. Barndorff-Nielson and N. Sheppard. Econometric analysis of realised volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society*, Series B 64:253-280, 2002

T.G. Andersen, T. Bollerslev, F.X. Diebold, and P. Labys. The distribution of exchange rate volatility. *Journal of the American Statistical Association*, 96:42-55, 2001.

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

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*, 2005.

Michiel de Pooter, Martin Martens, and Dick van Dijk. Predicting the daily covariance matrix for sp100 stocks using intraday data - but which frequency to use? *Working Paper*, October 2005.

Ole E. Barndorff-Nielsen, Peter Reinhard Hansen, Asger Lunde, and Neil Shephard. Regular and modified kernel-based estimators of integrated variance: The case with independent noise. *Working Paper*, 2004.

J. E. Griffen and R. C. A. Oomen. Covariance measurement in the presence of non-synchronous trading and market microstructure noise. *Working Paper*, June 27th, 2006.

T.G. Andersen, T. Bollerslev, F.X. Diebold, and P. Labys. The distribution of exchange rate volatility. *Journal of the American Statistical Association*, 96:42-55, 2001.

## Examples

```
# see users manual
```

---

`dates.example` *Example dates*

---

## Description

The dates that go along with the other example data sets.

## Usage

```
data(dates.example)
```

## Examples

```
data(dates.example)
print(dates.example)
```

---

`ge.real.cts` *General Electric Realized Objects*

---

## Description

A list of General Electric transaction data aligned to calendar time sampling and already in realizedObjects.

## Usage

```
data(ge.real.cts)
```

## Source

http://faculty.washington.edu/ezivot/ezresearch.htm

## Examples

```
data(ge.real.cts)
plot(ge.real.cts[[1]])
```

---

# ge.real.tts *General Electric Realized Objects*

---

## Description

A list of General Electric transaction data aligned to tick time sampling and already in realizedObjects.

## Usage

```
data(ge.real.tts)
```

## Source

http://faculty.washington.edu/ezivot/ezresearch.htm

---

# merge.realizedObject *Merging realizedObjects*

---

## Description

Merges realizedObjects, see users manual.

## Usage

```
merge.realizedObject(x, y = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | First object |
| y | Second object |
| ... | Third, .... objects |

## Details

see users manual.

### Value

Multiple instrument realized object

### Author(s)

Scott Payseur <spayseur@u.washingon.edu>

---

`msft.real.cts` *Microsoft Realized Objects*

---

### Description

A list of Microsoft transaction data aligned to calendar time sampling and already in realizedObjects.

### Usage

```
data(msft.real.cts)
```

### Source

http://faculty.washington.edu/ezivot/ezresearch.htm

### Examples

```
data(msft.real.cts)
plot(msft.real.cts[[1]])
```

---

`msft.real.tts` *Microsoft Realized Objects*

---

### Description

A list of Microsoft transaction data aligned to tick time sampling and already in realizedObjects.

### Usage

```
data(msft.real.tts)
```

### Source

http://faculty.washington.edu/ezivot/ezresearch.htm

### Examples

```
data(msft.real.tts)
plot(msft.real.tts[[1]])
```

rAccumulation *Realized Accumulation Plot*

## Description

Plots the realized estimate as it accumulates over a time interval.

## Usage

```
rAccumulation(x, period = 1, y = NULL, align.period = 1, plotit = FALSE, cts = TRUE, makeReturns = FALSE)
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| y | RealizedObject or TimeSeries for S+ |
| period | Sampling period |
| align.period | Align the returns to this period first |
| plotit | T for plot |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |

## Details

Plots the realized estimate as it accumulates over a time interval. This is a good tool to determine what obersations are adding (possibly subtracting for covariance) to the estimate. For version 0.7 this is only implemented for the naive estimators, in 1.0 it will be implemented generically.

## Value

Realized accumulation vector if plotit = F

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

## See Also

rMarginal, rRealizedVariance

## Examples

```
data(msft.real.cts)
data(dates.example)

cumm <- list()
cumm[[1]] <- rCumSum(msft.real.cts[[1]], period=1, align.period=60)
cumm[[2]] <- rCumSum(msft.real.cts[[1]], period=10, align.period=60)
cumm[[3]] <- rCumSum(msft.real.cts[[1]], period=20, align.period=60)
cumm[[4]] <- rCumSum(msft.real.cts[[1]], period=30, align.period=60)
accum <- list()
accum[[1]] <- rAccumulation(msft.real.cts[[1]], period=10, align.period=60)
accum[[2]] <- rAccumulation(msft.real.cts[[1]], period=20, align.period=60)
accum[[3]] <- rAccumulation(msft.real.cts[[1]], period=30, align.period=60)

par(mfrow=c(2,1))
plot(cumm[[1]], xlab="", ylab="Cumulative Ruturns", main="MSFT", sub=dates.example[[1]], type="p", col=16, lwd=2)
lines(cumm[[2]], col=2, lwd=2)
lines(cumm[[3]], col=3, lwd=2)
lines(cumm[[4]], col=4, lwd=2)
plot(accum[[1]], xlab="", ylab="Realized Accumulation", type="l",main="MSFT", sub=dates.example[[1]], col=2, lwd=
lines(accum[[2]], col=3, lwd=2)
lines(accum[[3]], col=4, lwd=2)
```

---

`rc.avg` *Realized Covariance: Average Subsample*

---

## Description

Realized Covariance using average subsample.

## Usage

```
rc.avg(x, y, period, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `y` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `...` | ... |

## Value

Realized covariance using average subsample.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## 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*, 2005.

Michiel de Pooter, Martin Martens, and Dick van Dijk. Predicting the daily covariance matrix for sp100 stocks using intraday data - but which frequency to use? *Working Paper*, October 2005.

## See Also

rv.avg, rRealizedVariance

## Examples

```
 data(msft.real.cts)
data(ge.real.cts)


#
# Average subsampled realized covariance for CTS aligned at one second returns at
# 600 subgrids (10 minutes).
#
rc.avg(x = msft.real.cts[[1]], y = ge.real.cts[[1]], period = 600)

#
# Average subsampled realized covariance for CTS aligned at one minute returns at
# 5 subgrids (5 minutes).
#
rc.avg(x = msft.real.cts[[1]], y = ge.real.cts[[1]], period = 5, align.period=60)
```

---

# rc.hy *Hayashi-Yoshida*

---

## Description

Hayashi-Yoshida covariance estimator

## Usage

```
rc.hy(x, y, period = 1, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| y | RealizedObject or TimeSeries for S+ |
| period | Sampling period |
| align.period | Align the returns to this period first |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |
| ... | ... |

### Author(s)

Scott Payseur

### References

T. Hayashi and N. Yoshida. On covariance estimation of non-synchronously observed diffusion processes. *Bernoulli*, 11:359-379, 2005.

### See Also

rRealizedVariance

### Examples

```
data(msft.real.cts)
data(ge.real.cts)

# Hayashi-Yoshida realized covariance for CTS aligned at one second returns at
# 600 subgrids (10 minutes).
#
rc.hy(x = msft.real.cts[[1]], y = ge.real.cts[[1]], period = 600)
```

---

## rc.kernel *Realized Covariance: Kernel*

---

### Description

Realized covariance calculation using a kernel estimator.

### Usage

```
rc.kernel(x, y, q, align.period = 1, adj = TRUE, type = 0, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| y | RealizedObject or TimeSeries for S+ |
| q | Number of lags |
| align.period | Align the returns to this period first |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |
| adj | T to use dof adjustment |
| type | 0-11 or a character string |
| ... | ... |

## Details

The different types of kernels can be found using rKernel.available().

## Value

Kernel estimate of realized covariance.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

Ole E. Barndorff-Nielsen, Peter Reinhard Hansen, Asger Lunde, and Neil Shephard. Regular and modified kernel-based estimators of integrated variance: The case with independent noise. *Working Paper*, 2004.

Michiel de Pooter, Martin Martens, and Dick van Dijk. Predicting the daily covariance matrix for sp100 stocks using intraday data - but which frequency to use? *Working Paper*, October 2005.

J. E. Griffen and R. C. A. Oomen. Covariance measurement in the presence of non-synchronous trading and market microstructure noise. *Working Paper*, June 27th, 2006.

## See Also

rRealizedVariance,rv.kernel, rKernel.available,rKernel

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

# kernel realized covariance for CTS aligned at one minute returns
#
rc.kernel(x = msft.real.cts[[1]], y = ge.real.cts[[1]], q=1, type="bartlett", align.period=60)
```

`rc.naive` *Realized Covariance*

## Description

Traditional realized covariance estimator.

## Usage

```
rc.naive(x, y, period, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `y` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `...` | ... |

## Value

Sum of cross products of high frequency returns.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

T.G. Andersen, T. Bollerslev, F.X. Diebold, and P. Labys. The distribution of exchange rate volatility. *Journal of the American Statistical Association*, 96:42-55, 2001.

## See Also

`rv.naive`, `rSignature`, `rRealizedVariance`

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

# Traditional RC aligned to 60 seconds
#
rc.naive(x = msft.real.cts[[1]], y = ge.real.cts[[1]], period = 60)
```

`rc.timescale` *Realized Covariance: Two Timescales*

## Description

Realized Covariance using a generalization of the popular two timescale variance method.

## Usage

```
rc.timescale(x, y, period, align.period = 1, adj.type = "classic", cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `y` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `adj.type` | "classic", "adj" or "aa" |
| `...` | ... |

## Details

Realized Covariance using two timescale method.

## Value

Realized covariance using two timescale method

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## 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*, 2005.

Michiel de Pooter, Martin Martens, and Dick van Dijk. Predicting the daily covariance matrix for sp100 stocks using intraday data - but which frequency to use? *Working Paper*, October 2005.

## See Also

`rv.timescale`, `rRealizedVariance`

## Examples

```
data(msft.real.cts)
data(ge.real.cts)


rc.timescale(x = msft.real.cts[[1]], y = ge.real.cts[[1]], period = 60, adj.type="aa")
```

---

`rc.zero` *Calculates the percentage of co-zero returns at a specified sampling period*

---

## Description

Calculates the percentage of co-zero returns at a specified sampling period.

## Usage

```
rc.zero(x, y, period, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `y` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `...` | ... |

## Value

Percentage of co-zero returns.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

## See Also

`rSignature`, `rv.zero`

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

#
# Plot the percentage of co-zero returns for sampling periods of 1 second
# to 600 seconds.
#
plot(rSignature(1:600, x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="zero", xscale=1/60), main="Percentage of c
```

---

rCumSum *Plot cummulative returns*

---

## Description

Plots cummulative returns at a certain alignment given a return series.

## Usage

```
rCumSum(x, period = 1, align.period = 1, plotit = FALSE, type = "l", cts = TRUE, makeReturns = FALSE)
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| period | Sampling period |
| align.period | Align the returns to this period first |
| plotit | T for plot |
| type | Line or points |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |

## Value

Cummulative return vector if plotit = F

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## Examples

```
data(msft.real.cts)
data(dates.example)

cumm <- list()
cumm[[1]] <- rCumSum(msft.real.cts[[1]], period=1, align.period=60)
cumm[[2]] <- rCumSum(msft.real.cts[[1]], period=10, align.period=60)
cumm[[3]] <- rCumSum(msft.real.cts[[1]], period=20, align.period=60)
cumm[[4]] <- rCumSum(msft.real.cts[[1]], period=30, align.period=60)
plot(cumm[[1]], xlab="", ylab="Cumulative Ruturns", main="MSFT", sub=dates.example[[1]], type="p", col=16, lwd=2)
lines(cumm[[2]], col=2, lwd=2)
lines(cumm[[3]], col=3, lwd=2)
lines(cumm[[4]], col=4, lwd=2)
```

---

`realizedObject` *realizedObject*

---

## Description

Creates an object of type realizedObject to be passed to most of the functions from the realized library

## Usage

```
realizedObject(x, cts = TRUE, makeReturns = FALSE, millisstart = 34200000, millisend = 57600000)
```

## Arguments

| | |
|---|---|
| `x` | Either an S+ timeSeries or a list with $data that is prices or returns, and $ milliseconds |
| `cts` | Calendar time sampling or tick time sampling |
| `makeReturns` | If you pass prices you should have this set to T |
| `millisstart` | When does your integrated variance start 34200000 is 09:30 |
| `millisend` | When does your integrated variance end 57600000 is 16:00 |

## Details

See user's manual.

## Value

Object of type realizedObject

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## Examples

```
# see users manual
```

---

rKernel *Kernel Function*

---

## Description

The kernel function that is used for the realized variance kernel estimators.

## Usage

```
rKernel(x, type = 0)
```

## Arguments

| | |
|---|---|
| x | Value between [0,1] to calculate kernel weight |
| type | Type of kernel 0-11, or values from rKernel.available() |

## Value

Kernel value

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

Ole E. Barndorff-Nielsen, Peter Reinhard Hansen, Asger Lunde, and Neil Shephard. Regular and modified kernel-based estimators of integrated variance: The case with independent noise. *Working Paper*, 2004.

## See Also

rKernel.available,rv.kernel,rc.kernel

## Examples

```
#
# Plot all available kernels
#
par(mfrow=c(3,4))
x <- (0:100)*.01
for(i in 1:length(rKernel.available()))
   plot(x=x,y=sapply(x, FUN="rKernel", type=rKernel.available()[i]), xlab="", ylab="", main=rKernel.available()[
```

## rKernel.available *Available Kernels*

### Description

Returns a vector of the available kernels.

### Usage

```
rKernel.available()
```

### Author(s)

Scott Payseur <spayseur@u.washington.edu>

### References

Ole E. Barndorff-Nielsen, Peter Reinhard Hansen, Asger Lunde, and Neil Shephard. Regular and modified kernel-based estimators of integrated variance: The case with independent noise. *Working Paper*, 2004.

### See Also

rc.kernel, rv.kernel, rKernel, rRealizedVariance

### Examples

```
rKernel.available()
```

## rMarginal *Maginal Contribution to Realized Estimate*

### Description

Plots the marginal contribution to the realized estimate.

### Usage

```
rMarginal(x, y = NULL, period, align.period = 1, plotit = FALSE, cts = TRUE,makeReturns = TRUE)
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| y | RealizedObject or TimeSeries for S+ |
| period | Sampling period |
| align.period | Align the returns to this period first |
| plotit | T for plot |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |

## Details

Plots the marginal contribution to the realized estimate. This is a good tool to determine what obersations are adding (possibly subtracting for covariance) to the estimate. For version 0.7 this is only implemented for the naive estimators, in 1.0 it will be implemented generically.

## Value

Marginal contribution vector if plotit = F

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

## See Also

rAccumulation, rRealizedVariance

## Examples

```
data(msft.real.cts)
data(dates.example)

par(mfrow=c(2,1))
plot(rCumSum(msft.real.cts[[1]], period=10, align.period=60), xlab="", ylab="Cumulative Ruturns", main="MSFT", su
barplot(rMarginal(msft.real.cts[[1]], period=10, align.period=60)$y, main="Marginal Contribution Plot")
```

`rRealizedVariance` *Calculate realized variance, covariance, or correlation.*

## Description

Calculate realized variance, covariance, correlation, covariance matrix, or correlation matrix.

## Usage

```
rRealizedVariance(x, y = NULL, type = "naive", period = 1, lags = 1, cor = FALSE, rvargs = list(), cts = T
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `y` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `type` | Type of realized estimator to use, a rv. or rc. is appended to this value and that function is called |
| `lags` | Number of lags or subgrids for kernel and subsample based estimators |
| `cor` | T for correlation |
| `rvargs` | List of extra parameters to pass into rv.* or rc.* |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |

## Details

Calculate realized variance, covariance, correlation, covariance matrix, or correlation matrix.

## Value

A single numeric value or a matrix if x is multicolumn matrix.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## See Also

`rc.avg`, `rc.kernel`, `rc.naive`, `rc.timescale`, `rv.avg`, `rv.kernel`, `rv.naive`, `rv.timescale`

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

# Traditional Estimate at highest frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="naive", period=1)

# Traditional Estimate at one minute frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="naive", period=1, rvargs=list(align.period=60)

# Traditional Estimate at 10 minute frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="naive", period=10, rvargs=list(align.period=60

# Bartlett Kernel Estimate with minute aligned data at 20 lags
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="kernel", lags=20, rvargs=list(align.period=60,

# Cubic Kernel Estimate with second aligned data at 400 lags
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="kernel", lags=400, rvargs=list(type="Cubic"))

# Lead-Lag with one lag at one minute frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="kernel", lags=1, rvargs=list(align.period=60))

# Subsample Average Estimate with second aligned data at 600 subgrids
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="avg", period=600)


# Traditional Estimate at highest frequency
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="naive", period=1)

# Traditional Estimate at 10 minute frequency
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="naive", period=10, rvargs=list(align.peri

# Lead-Lag with one lag at one minute frequency>
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="kernel", lags=1, rvargs=list(align.period

# Subsample Average Estimate with second aligned data at 600 subgrids
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="avg", period=600)




# Traditional Estimate at highest frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="naive", period=1, cor=TRUE)

# Traditional Estimate at one minute frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="naive", period=1, rvargs=list(align.period=60)

# Traditional Estimate at 10 minute frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="naive", period=10, rvargs=list(align.period=60

# Bartlett Kernel Estimate with minute aligned data at 20 lags
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="kernel", lags=20, rvargs=list(align.period=60,
```

```
# Cubic Kernel Estimate with second aligned data at 400 lags
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="kernel", lags=400, rvargs=list(type="Cubic"),

# Lead-Lag with one lag at one minute frequency
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="kernel", lags=1, rvargs=list(align.period=60),

# Subsample Average Estimate with second aligned data at 600 subgrids
rRealizedVariance(x=msft.real.cts[[1]], y=ge.real.cts[[1]], type="avg", period=600, cor=TRUE)


# Correlation Matrices
# Traditional Estimate at highest frequency
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="naive", period=1, cor=TRUE)

# Traditional Estimate at 10 minute frequency
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="naive", period=10, rvargs=list(align.peri

# Lead-Lag with one lag at one minute frequency
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="kernel", lags=1, rvargs=list(align.period

# Subsample Average Estimate with second aligned data at 600 subgrids >
rRealizedVariance(x=merge(msft.real.cts[[1]], ge.real.cts[[1]]), type="avg", period=600, cor=TRUE)
```

---

`rScatterReturns` *Scatterplot of aligned returns*

---

## Description

Creates a scatterplot of cross returns.

## Usage

```
rScatterReturns(x,y, period, align.period=1,numbers=FALSE,xlim= NULL, ylim=NULL, plotit=TRUE, pch=NUL
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `y` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `plotit` | T for plot |
| `numbers` | T for count |

| | |
|---|---|
| pch | type of point |
| ylim | ylimit |
| xlim | xlimit |
| scale.size | . |
| col.change | . |
| ... | ... |

## Details

Scatterplot of returns.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

## Examples

```
data(msft.real.cts)
data(ge.real.cts)
par(mfrow=c(2,1))
rScatterReturns(msft.real.cts[[1]],y=ge.real.cts[[1]], period=1, align.period=20,ylab="GE",xlab="MSFT",numbers=
rScatterReturns(msft.real.cts[[1]],y=ge.real.cts[[1]], period=1, align.period=20,ylab="GE",xlab="MSFT",numbers=
```

---

# rSignature *Signature Plots*

---

## Description

Creates realized variance, covariance, and correlation plots for one or multiple days for each type of estimator.

## Usage

```
rSignature(range, x, y = NULL, type = "naive", cor = FALSE, rvargs = list(), xscale = 1, iteration.funct
```

## Arguments

| | |
|---|---|
| range | x axis of signature plot (inputs for realized estimator) |
| x | RealizedObject or TimeSeries for S+ |
| y | RealizedObject or TimeSeries for S+ |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |
| type | Type of realized estimator to use, a rv. or rc. is appended to this value and that function is called |
| cor | T for correlation |
| rvargs | List of extra parameters to pass into rv.* or rc.* |
| xscale | value to multiply range by for x axis (see below) |
| iteration.funct | Name of function to be used for average signature plot |
| iterations | Numeric vector to pass into iteration.funct |
| plotit | T for plot |

## Details

Creates a list that contains an x and y value that is used for plotting a signature plot.

## Value

List containing:

| | |
|---|---|
| x | x axis of signature plot (range * xscale) |
| y | Realized variance or covariance |
| type | type of estimator used for calculation |
| cor | is this a correlation signture |
| cov | it this realized covariance |

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

T.G. Andersen, T. Bollerslev, F.X. Diebold, and P. Labys. Great realizations. *Risk*, 13:105 108, 2000.

Y. Fang. Volatility modeling and estimation of high-frequency data with gaussian noise. *unpublished doctoral thesis, MIT, Sloan School of Management*, 1996.

J. E. Griffen and R. C. A. Oomen. Covariance measurement in the presence of non-synchronous trading and market microstructure noise. *Working Paper*, June 27th, 2006.

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

## See Also

rRealizedVariance

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

# see users manual for more examples

test.cov <- rSignature(1:1200,x=msft.real.cts[[1]], y=ge.real.cts[[1]], xscale=1/60)
test.rect <- rSignature(1:600,msft.real.cts[[1]], ge.real.cts[[1]],type="kernel",rvargs=list(type="rectangular"
test.mth <- rSignature(1:600,msft.real.cts[[1]], ge.real.cts[[1]],type="kernel",rvargs=list(type="mth"), xscale=
plot(test.cov, ylab="Realized Covariance", xlab="Minutes", main="GE | MSFT")
lines(test.rect, col=3, lwd=1)
lines(test.mth, col=4, lwd=2)
axis(3, c(0,(1:5)*4), c("Lags:",as.character((1:5)*120)))
legend(13,.00015,c("Rectangular", "Mod TH"), lwd=c(1,2), col=c(3,4))
```

---

# rv.avg *Realized Variance: Average Subsample*

---

## Description

Realized Variance using average subsample.

## Usage

```
rv.avg(x,period, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| period | Sampling period |
| align.period | Align the returns to this period first |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |
| ... | ... |

## Value

Realized variance using average subsample.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## 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*, 2005.

## See Also

rv.avg, rRealizedVariance

## Examples

```
data(msft.real.cts)


#
# Average subsampled realized variance for CTS aligned at one second returns at
# 600 subgrids (10 minutes).
#
rv.avg(x = msft.real.cts[[1]], period = 600)

#
# Average subsampled realized variance for CTS aligned at one minute returns at
# 5 subgrids (5 minutes).
#
rv.avg(x = msft.real.cts[[1]], period = 5, align.period=60)
```

---

# rv.kernel *Realized Variance: Kernel*

---

## Description

Realized variance calculation using a kernel estimator.

## Usage

```
rv.kernel(x, q, align.period = 1, adj = TRUE, type = 0, cts = TRUE, makeReturns = FALSE,rvargs=list(),...
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| q | Number of lags |
| align.period | Align the returns to this period first |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |
| adj | T to use dof adjustment |

| | |
|---|---|
| type | 0-11 or a character string |
| rvargs | . |
| ... | ... |

## Details

The different types of kernels can be found using rKernel.available().

## Value

Kernel estimate of realized variance.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

Ole E. Barndorff-Nielsen, Peter Reinhard Hansen, Asger Lunde, and Neil Shephard. Regular and modified kernel-based estimators of integrated variance: The case with independent noise. *Working Paper*, 2004.

B. Zhou. High-frequency data and volatility in foreign-exchange rates. *Journal of Buiness & Economic Statistics*, 14:45-52, 1996.

P. Hansen and A. Lunde. Realized variance and market microstructure noise. *Journal of Business and Economic Statistics*, 24:127-218, 2006.

## See Also

rRealizedVariance,rc.kernel, rKernel.available,rKernel

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

# kernel realized variance for CTS aligned at one minute returns
#
rv.kernel(x = msft.real.cts[[1]], q=1, type="bartlett", align.period=60)
```

`rv.naive` *Realized Variance*

## Description

Traditional realized variance estimator.

## Usage

```
rv.naive(x, period, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `...` | ... |

## Value

Sum of products of high frequency returns.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

O. E. Barndorff-Nielson and N. Sheppard. Econometric analysis of realised volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society*, Series B 64:253-280, 2002

T.G. Andersen, T. Bollerslev, F.X. Diebold, and P. Labys. The distribution of exchange rate volatility. *Journal of the American Statistical Association*, 96:42-55, 2001.

## See Also

`rc.naive`, `rSignature`, `rRealizedVariance`

## Examples

```
data(msft.real.cts)
data(ge.real.cts)

# Traditional RV aligned to 60 seconds
#
rv.naive(x = msft.real.cts[[1]], period = 60)
```

---

`rv.timescale` *Realized Variance: Two Timescales*

---

## Description

Realized variance using the two timescale variance method.

## Usage

```
rv.timescale(x, period, align.period = 1, adj.type = "classic", cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | RealizedObject or TimeSeries for S+ |
| `period` | Sampling period |
| `align.period` | Align the returns to this period first |
| `cts` | Create calendar time sampling if a non realizedObject is passed |
| `makeReturns` | Prices are passed make them into log returns |
| `adj.type` | "classic", "adj" or "aa" |
| `...` | ... |

## Details

Realized variance using two timescale method.

## Value

Realized variance using two timescale method

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## 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*, 2005.

## See Also

rc.timescale, rRealizedVariance

## Examples

```
data(msft.real.cts)

rv.timescale(x = msft.real.cts[[1]], period = 60, adj.type="aa")
```

---

# rv.zero *Calculates the percentage of zero returns at a specified sampling period*

---

## Description

Calculates the percentage of zero returns at a specified sampling period.

## Usage

```
rv.zero(x, period, align.period = 1, cts = TRUE, makeReturns = FALSE, ...)
```

## Arguments

| | |
|---|---|
| x | RealizedObject or TimeSeries for S+ |
| period | Sampling period |
| align.period | Align the returns to this period first |
| cts | Create calendar time sampling if a non realizedObject is passed |
| makeReturns | Prices are passed make them into log returns |
| ... | ... |

## Value

Percentage of zero returns.

## Author(s)

Scott Payseur <spayseur@u.washington.edu>

## References

S. W. Payseur. A One Day Comparison of Realized Variance and Covariance Estimators. *Working Paper: University of Washington*, 2007

## See Also

rSignature, rc.zero

## Examples

```
data(msft.real.cts)

#
# Plot the percentage of zero returns for sampling periods of 1 second
# to 600 seconds.
#
plot(rSignature(1:600, x=msft.real.cts[[1]], type="zero", xscale=1/60), main="Percentage of co-zeros", ylab="Perc
```

# Index