Package 'pinbasic'

November 18, 2018

Title Fast and Stable Estimation of the Probability of Informed Trading (PIN)

Type Package

Version 1.2.2

Description Utilities for fast and stable estimation of the probability of informed trading (PIN) in the model introduced by Easley et al. (2002) <DOI:10.1111/1540-6261.00493> are implemented. Since the basic model developed by Easley et al. (1996) <DOI:10.1111/j.1540-6261.1996.tb04074.x> is nested in the former due to equating the intensity of uninformed buys and sells, functions can also be applied to this simpler model structure, if needed.
State-of-the-art factorization of the model likelihood function as well as most recent algorithms for generating initial values for optimization routines are implemented. In total, two likelihood factorizations and three methodologies for starting values are included.
Furthermore, functions for simulating datasets of daily aggregated buys and sells, calculating confidence intervals for the probability of informed trading and posterior probabilities of trading days' conditions are available.

Depends R (>= 3.1)

Imports stats, fastcluster, lubridate, ggplot2, reshape2, scales, parallel, Rcpp (>= 0.12.9),

LinkingTo Rcpp

NeedsCompilation yes

License GPL-3 | file LICENSE

LazyData true

RoxygenNote 6.1.0

Suggests knitr, rmarkdown, formatR, utils

URL https://github.com/anre005/pinbasic/

BugReports https://github.com/anre005/pinbasic/issues

VignetteBuilder knitr

Author Andreas Recktenwald [aut, cre]

Maintainer Andreas Recktenwald <arecktenwald85@gmail.com>

Repository CRAN

Date/Publication 2018-11-18 10:10:03 UTC

R topics documented:

`BSfrequent` *Daily Buys and Sells*

Description

A matrix containing three months of synthetic daily buys and sells data representing a frequently traded stock.

Usage

```
BSfrequent
```

Format

A matrix with 60 rows and 2 variables:

Buys daily buys

Sells daily sells

Source

own simulation

BSfrequent2015 *Daily Buys and Sells*

Description

A matrix containing one year of synthetic daily buys and sells data representing a frequently traded stock. Rownames equal the business days in 2015.

Usage

```
BSfrequent2015
```

Format

A matrix with 261 rows and 2 variables:

Buys daily buys

Sells daily sells

Source

own simulation

BSheavy *Daily Buys and Sells*

Description

A matrix containing three months of synthetic daily buys and sells data representing a heavily traded stock.

Usage

```
BSheavy
```

Format

A matrix with 60 rows and 2 variables:

Buys daily buys

Sells daily sells

Source

own simulation

BSinfrequent *Daily Buys and Sells*

Description

A matrix containing three months of synthetic daily buys and sells data representing an infrequently traded stock.

Usage

```
BSinfrequent
```

Format

A matrix with 60 rows and 2 variables:

Buys daily buys

Sells daily sells

Source

own simulation

ggplot.posterior *Visualization of Posterior Probabilities*

Description

Visualization of posterior probabilities returned by `post_states` with ggplot2

Usage

```
## S3 method for class 'posterior'
ggplot(x)
```

Arguments

x	*numeric* matrix returned by `posterior`

Value

An object of class `ggplot`.

See Also

`posterior`

Examples

```
## Not run: See Vignette \code{browseVignette(package = 'pinbasic')}
```

`ggplot.qpin` *PIN Visualization*

Description

Visualization of quarterly estimates and probability of informed trading with ggplot2.

Usage

```
## S3 method for class 'qpin'
ggplot(x)
```

Arguments

`x`	List returned by `qpin`

Details

Facets are grouped by probability parameters, intensity parameters and the probability of informed trading.

Value

An object of class `ggplot`.

References

Wickham, Hadley (2009)
ggplot2: Elegant Graphics for Data Analysis
Springer-Verlag New York
doi: 10.1007/9780387981413

Wickham, Hadley (2007)
Reshaping Data with the reshape Package
Journal of Statistical Software, Volume 21, Issue 12, pp. 1 - 20
doi: 10.18637/jss.v021.i12

Wickham, Hadley (2016)
scales: Scale Functions for Visualization
R package version 0.4.0

See Also

`qpin`

Examples

```
# Loading one year of simulated daily buys and sells

data('BSfrequent2015')

# Quarterly estimates for model parameters and the probability of informed trading
# Rownames of 'BSfrequent2015' equal the business days in 2015.

qpin_list <- qpin(numbuys = BSfrequent2015[,"Buys"], numsells = BSfrequent2015[,"Sells"],
                  dates = as.Date(rownames(BSfrequent2015), format = "%Y-%m-%d"))

# Visualization of quarterly estimates
library(ggplot2)
ggplot(qpin_list[["res"]])
```

---

initial_vals *Initial values for PIN optimization*

Description

Generates set(s) of initial values which can be used in PIN optimization routines.

Usage

```
initial_vals(numbuys = NULL, numsells = NULL, method = "HAC",
  length = 5, num_clust = 5, details = FALSE)
```

Arguments

numbuys	*numeric*: vector of daily buys
numsells	*numeric*: vector of daily sells
method	*character* Switch between algorithms for generating initial values, valid choices are: 'Grid', 'HAC' and 'HAC_Ref'
length	*numeric* length of equidistant sequence from 0.1 to 0.9 for parameters of grid search algorithm, defaults to 5, irrelevant for HAC and refined HAC method
num_clust	*numeric* only relevant for refined HAC method, total number of clusters trading data is grouped into equals num_clust + 1
details	*logical* only relevant for grid search, if TRUE and method = 'Grid' the number of infeasible sets of initial values are returned,

Value

Matrix with set(s) of initial values for PIN model optimization. If `method = 'Grid'` and `details = TRUE` a list with four elements is returned:

inits Matrix of sets of initial values

neg_eps Number of infeasible sets due to negative values for intensity of uninformed sells

irr_mu Number of infeasible sets due to intensity of informed trading larger than any daily buys and sells data

rem Total number of removed sets of initial values

References

Ersan, Oguz and Alici, Asli (2016)
An unbiased computation methodology for estimating the probability of informed trading (PIN)
Journal of International Financial Markets, Institutions and Money, Volume 43, pp. 74 - 94
doi: 10.1016/j.intfin.2016.04.001

Gan, Quan et al. (2015)
A faster estimation method for the probability of informed trading using hierarchical agglomerative clustering
Quantitative Finance, Volume 15, Issue 11, pp. 1805 - 1821
doi: 10.1080/14697688.2015.1023336

Yan, Yuxing and Zhang, Shaojun (2012)
An improved estimation method and empirical properties of the probability of informed trading
Journal of Banking & Finance, Volume 36, Issue 2, pp. 454 - 467
doi: 10.1016/j.jbankfin.2011.08.003

Examples

```
# Loading simulated datasets

data("BSinfrequent")
data("BSfrequent")
data("BSheavy")

# Grid Search

grid <- initial_vals(numbuys = BSinfrequent[,"Buys"],
                     numsells = BSinfrequent[,"Sells"],
                     method = "Grid")

# Grid Search: Detailed Output

grid_detailed <- initial_vals(numbuys = BSinfrequent[,"Buys"],
                              numsells = BSinfrequent[,"Sells"],
                              method = "Grid", details = TRUE)

# HAC

hac <- initial_vals(numbuys = BSfrequent[,"Buys"],
```

```
                    numsells = BSfrequent[,"Sells"],
                    method = "HAC")

# Refined HAC

hac_ref <- initial_vals(numbuys = BSheavy[,"Buys"],
                        numsells = BSheavy[,"Sells"],
                        method = "HAC_Ref")
```

---

pinbasic *A Package for Fast and Stable Estimation of the Probability of Informed Trading (PIN)*

Description

Utilities for fast and stable estimation of the probability of informed trading (PIN) in the model introduced by Easley, Hvidkjaer and O'Hara (EHO, 2002) are implemented. Since the model developed by Easley, Kiefer, O'Hara and Paperman (EKOP, 1996) is nested in the EHO model due to equating the intensity of uninformed buys and sells, functionalities can also be applied to this simpler model structure, if needed. State-of-the-art factorization of the model likelihood function as well as hierarchical agglomerative clustering algorithm for generating initial values for optimization routines are provided. In total, two different likelihood factorizations and three methodologies generating starting values are implemented. The probability of informed trading can be estimated for arbitrary length of daily buys and sells data with `pin_est` function which is a wrapper around the workhorse function `pin_est_core`. No information about the time span of the underlying data is required to perform optimizations. However, recommendation given in the literature is using at least data for 60 trading days to ensure convergence of the likelihood maximization. The `qpin` function delivers quarterly estimates. The number of available quarters in the data are detected utilizing functions from the `lubridate` package. Quarterly estimates can be visualized with the `ggplot` function. Datasets of daily aggregated numbers of buys and sells can be simulated with `simulateBS`. Calculation of confidence intervals for the probability of informed trading can be enabled by `confint` argument in optimization routines (`pin_est_core`, `pin_est` and `qpin`) or by calling `pin_confint` directly. Additionally, posterior probabilities for conditions of trading days can be computed with `posterior` and plotted with `ggplot`.

Functions

`ggplot.posterior` Visualization method for results of `posterior` with ggplot2.

`ggplot.qpin` Visualization method for results of `qpin` with ggplot2.

`initial_vals` Generating initial values by brute force grid search, hierarchical agglomerative clustering algorithm or refined hierarchical agglomerative clustering technique.

`posterior` Calculation of posterior probabilities of trading days' conditions.

`pin_calc` Computing the probability of informed trading (PIN).

`pin_confint` Calculation of confidence intervals for the probability of informed trading.

`pin_est_core` Core function of maximization routines for PIN likelihood function. It grants the most control over optimization procedure. However, the settings chosen in `pin_est` will be sufficient in most applications.

`pin_est` User-friendly wrapper around `pin_est_core`. Default method for creating initial values is set to hierarchical agglomerative clustering, the likelihood formulation defaults to the one proposed by Lin and Ke (2011).

`pin_ll` Evaluating likelihood function values either utilizing the factorization by Easley et. al (2010) or Lin and Ke (2011).

`qpin` Returns quarterly estimates, function is a wrapper around `pin_est` and inherits its optimization settings.

`simulateBS` Simulate datasets of aggregated daily buys and sells.

Datasets

`BSinfrequent` A matrix containing three months of synthetic daily buys and sells data representing an infrequently traded stock.

`BSfrequent` A matrix containing three months of synthetic daily buys and sells data representing a frequently traded stock.

`BSheavy` A matrix containing three months of synthetic daily buys and sells data representing a heavily traded stock.

`BSfrequent2015` A matrix containing one year of synthetic daily buys and sells data representing a frequently traded stock. Rownames equal the business days in 2015.

Source of all included datasets: own simulation

Author

Andreas Recktenwald (Saarland University, Statistics & Econometrics)
Email: <a.recktenwald@mx.uni-saarland.de>

Github

https://github.com/anre005/pinbasic

References

Easley, David et al. (2002)
Is Information Risk a Determinant of Asset Returns?
The Journal of Finance, Volume 57, Number 5, pp. 2185 - 2221
doi: 10.1111/15406261.00493

Easley, David et al. (1996)
Liquidity, Information, and Infrequently Traded Stocks
The Journal of Finance, Volume 51, Number 4, pp. 1405 - 1436
doi: 10.1111/j.15406261.1996.tb04074.x

Easley, David et al. (2010)
Factoring Information into Returns

Journal of Financial and Quantitative Analysis, Volume 45, Issue 2, pp. 293 - 309
doi: 10.1017/S0022109010000074

Ersan, Oguz and Alici, Asli (2016)
An unbiased computation methodology for estimating the probability of informed trading (PIN)
Journal of International Financial Markets, Institutions and Money, Volume 43, pp. 74 - 94
doi: 10.1016/j.intfin.2016.04.001

Gan, Quan et al. (2015)
A faster estimation method for the probability of informed trading using hierarchical agglomerative clustering
Quantitative Finance, Volume 15, Issue 11, pp. 1805 - 1821
doi: 10.1080/14697688.2015.1023336

Grolemund, Garett and Wickham, Hadley (2011)
Dates and Times Made Easy with lubridate
Journal of Statistical Software, Volume 40, Issue 3, pp. 1 - 25
doi: 10.18637/jss.v040.i03

Lin, Hsiou-Wei William and Ke, Wen-Chyan (2011)
A computing bias in estimating the probability of informed trading
Journal of Financial Markets, Volume 14, Issue 4, pp. 625 - 640
doi: 10.1016/j.finmar.2011.03.001

Wickham, Hadley (2009)
ggplot2: Elegant Graphics for Data Analysis
Springer-Verlag New York
doi: 10.1007/9780387981413

Wickham, Hadley (2007)
Reshaping Data with the reshape Package
Journal of Statistical Software, Volume 21, Issue 12, pp. 1 - 20
doi: 10.18637/jss.v021.i12

Wickham, Hadley (2016)
scales: Scale Functions for Visualization
R package version 0.4.0

Yan, Yuxing and Zhang, Shaojun (2012)
An improved estimation method and empirical properties of the probability of informed trading
Journal of Banking & Finance, Volume 36, Issue 2, pp. 454 - 467
doi: 10.1016/j.jbankfin.2011.08.003

`pin_calc` *Probability of Informed Trading*

Description

Calculates the probability of informed trading.

Usage

```
pin_calc(param = NULL)
```

Arguments

`param`	*numeric*: (named) vector of model parameters (valid names: `'alpha'`, `'delta'`, `'epsilon_b'`, `'epsilon_s'`, `'mu'`), length must equal 5

Details

If names are not set for `param` or one or more of the vector names do not match the valid choices, they are internally set to `'alpha'`, `'delta'`, `'epsilon_b'`, `'epsilon_s'`, `'mu'` (in this order).

Value

numeric: probability of informed trading

References

Easley, David et al. (2002)
Is Information Risk a Determinant of Asset Returns?
The Journal of Finance, Volume 57, Number 5, pp. 2185 - 2221
doi: 10.1111/15406261.00493

Easley, David et al. (1996)
Liquidity, Information, and Infrequently Traded Stocks
The Journal of Finance, Volume 51, Number 4, pp. 1405 - 1436
doi: 10.1111/j.15406261.1996.tb04074.x

`pin_confint` *PIN confidence intervals*

Description

Computes confidence intervals for the probability of informed trading by simulation

Usage

```
pin_confint(param = NULL, numbuys = NULL, numsells = NULL,
  method = "HAC", lower = rep(0, 5), upper = c(1, 1, rep(Inf, 3)),
  n = 10000, seed = NULL, level = 0.95, ncores = 1)
```

Arguments

`param`	*numeric*: (named) vector of model parameters (valid names: `'alpha'`, `'delta'`, `'epsilon_b'`, `'epsilon_s'`, `'mu'`), length must equal 5
`numbuys`	*numeric*: vector of daily buys
`numsells`	*numeric*: vector of daily sells
`method`	*character* Switch between algorithms for generating initial values, valid choices are: 'Grid', 'HAC' and 'HAC_Ref'
`lower`	*numeric*: lower bounds for optimization, must have length of 5

upper	*numeric*: upper bounds for optimization, must have length of 5
n	*integer*: Number of simulation runs, defaults to 10000
seed	*interpreted as integer or* NULL: defaults to NULL, for more details see set.seed
level	*numeric*: Confidence level, defaults to 0.95
ncores	*integer*: Number of cpu cores utilized in computation, defaults to 1

Details

If names are not set for param or one or more of the vector names do not match the valid choices, they are internally set to 'alpha', 'delta', 'epsilon_b', 'epsilon_s', 'mu' (in this order). By default, only one core is utilized in computations (ncores = 1). Confidence intervals can also be calculated in parallel, however, this only pays off for large values of n.

Value

numeric: confidence interval

---

pin_est *Estimating PIN*

Description

Estimates the probability of informed trading (PIN) for daily buys and sells trading data for arbitrary number of trading days.

Usage

```
pin_est(numbuys = NULL, numsells = NULL, nlminb_control = list(),
  confint = FALSE, ci_control = list(), posterior = TRUE)
```

Arguments

numbuys	*numeric*: vector of daily buys
numsells	*numeric*: vector of daily sells
nlminb_control	*list*: Control list for nlminb
confint	*logical*: Compute confidence intervals for PIN? Defaults to FALSE
ci_control	*list*: see pin_est_core
posterior	*logical*: Should posterior probabilities for conditions of trading days be computed?

Details

User-friendly wrapper around workhorse function `pin_est_core`. `nlminb` function in the **stats** package is used for maximization. In the literature, at least data for 60 trading days is recommended to ensure convergence of optimization. No information about the trading days' dates is needed. Vectors for `numbuys` and `numsells` need to have same length.
Calculation of confidence interval for the probability of informed trading is disabled by default. For more details see `pin_est_core` or `pin_confint`.

Value

A list with the following components:

Results Matrix containing the parameter estimates as well as their estimated standard errors, t-values and p-values.

ll Value of likelihood function returned by `nlminb`

pin Estimated probability of informed trading

conv Convergence code for nlminb optimization

message Convergence message returned by the nlminb optimizer

iterations Number of iterations until convergence of nlminb optimizer

init_vals Vector of initial values

confint If `confint = TRUE`; confidence interval for the probability of informed trading

References

Easley, David et al. (2002)
Is Information Risk a Determinant of Asset Returns?
The Journal of Finance, Volume 57, Number 5, pp. 2185 - 2221
doi: 10.1111/15406261.00493

Easley, David et al. (1996)
Liquidity, Information, and Infrequently Traded Stocks
The Journal of Finance, Volume 51, Number 4, pp. 1405 - 1436
doi: 10.1111/j.15406261.1996.tb04074.x

Easley, David et al. (2010)
Factoring Information into Returns
Journal of Financial and Quantitative Analysis, Volume 45, Issue 2, pp. 293 - 309
doi: 10.1017/S0022109010000074

Ersan, Oguz and Alici, Asli (2016)
An unbiased computation methodology for estimating the probability of informed trading (PIN)
Journal of International Financial Markets, Institutions and Money, Volume 43, pp. 74 - 94
doi: 10.1016/j.intfin.2016.04.001

Gan, Quan et al. (2015)
A faster estimation method for the probability of informed trading using hierarchical agglomerative clustering
Quantitative Finance, Volume 15, Issue 11, pp. 1805 - 1821
doi: 10.1080/14697688.2015.1023336

Lin, Hsiou-Wei William and Ke, Wen-Chyan (2011)
A computing bias in estimating the probability of informed trading
Journal of Financial Markets, Volume 14, Issue 4, pp. 625 - 640
doi: 10.1016/j.finmar.2011.03.001

Yan, Yuxing and Zhang, Shaojun (2012)
An improved estimation method and empirical properties of the probability of informed trading
Journal of Banking & Finance, Volume 36, Issue 2, pp. 454 - 467
doi: 10.1016/j.jbankfin.2011.08.003

See Also

`nlminb`, `initial_vals` `pin_est_core` `qpin`

Examples

```
# Loading simulated data for frequently traded stock

data("BSfrequent")

# Optimization with HAC initial values and Lin-Ke likelihood factorization

pin_freq <- pin_est(numbuys = BSfrequent[,"Buys"],
                    numsells = BSfrequent[,"Sells"])
```

`pin_est_core` *Optimization in PIN models*

Description

Core function for optimization routines

Usage

```
pin_est_core(numbuys = NULL, numsells = NULL,
  factorization = "Lin_Ke", init_vals = NULL, lower = rep(0, 5),
  upper = c(1, 1, rep(Inf, 3)), num_best_res = 1,
  only_converged = TRUE, nlminb_control = list(), confint = FALSE,
  ci_control = list(), posterior = TRUE)
```

Arguments

`numbuys`	*numeric*: vector of daily buys
`numsells`	*numeric*: vector of daily sells
`factorization`	*character*: factorization of likelihood function: either 'EHO' or 'Lin_Ke', defaults to: 'Lin_Ke'
`init_vals`	*numeric*: matrix of initial values: either generated by `initial_vals` or user-defined matrix with five columns for which colnames must consist of: `alpha`, `delta`, `epsilon_b`, `epsilon_s`, `mu`

lower	*numeric*: lower bounds for optimization, must have length of 5
upper	*numeric*: upper bounds for optimization, must have length of 5
num_best_res	Number of optimization runs for which results should be returned, either numeric or 'all', only relevant if init_vals = 'Grid', defaults to 1
only_converged	*logical*: Return only results for which the likelihood converged? Defaults to TRUE
nlminb_control	*list*: Control list for nlminb
confint	*logical*: Compute confidence intervals for PIN? Defaults to FALSE
ci_control	*list*: see **Details**
posterior	*logical*: Should posterior probabilities for conditions of trading days be computed?

Details

Grants the most control over optimization procedure. User-friendly wrappers for estimation with trading data of arbitrary length and quarterly data are implemented with pin_est and qpin, respectively. nlminb function in the **stats** package is used for maximization. Vectors for numbuys and numsells need to have same length.

Confidence intervals for the probability of informed trading are calculated via Monte-Carlo-Simulation if confint = TRUE. Settings of the confidence interval simulation can be specified via a named list for ci_control. Valid list names are n, seed, level and ncores which pass number of simulation runs (defaults to 10000), seed for RNG (defaults to seed = NULL), confidence level (defaults to 0.95) and number of cpu cores utilized (defaults to 1).

Value

If num_best_res = 1, a list with following elements is returned:

Results Matrix containing the parameter estimates as well as their estimated standard errors, t-values and p-values.

ll Value of likelihood function returned by nlminb

pin Estimated probability of informed trading

conv Convergence code for nlminb optimization

message Convergence message returned by the nlminb optimizer

iterations Number of iterations until convergence of nlminb optimizer

init_vals Vector of initial values

confint If confint = TRUE; confidence interval for the probability of informed trading

If num_best_res > 1, a named list of lists is returned. Each component of the outer list is again a list structured as shown above. Naming scheme for the outer list is 'Best1',..., 'Bestnum_best_res'.

References

Easley, David et al. (2002)
Is Information Risk a Determinant of Asset Returns?
The Journal of Finance, Volume 57, Number 5, pp. 2185 - 2221
doi: 10.1111/15406261.00493

Easley, David et al. (1996)
Liquidity, Information, and Infrequently Traded Stocks
The Journal of Finance, Volume 51, Number 4, pp. 1405 - 1436
doi: 10.1111/j.15406261.1996.tb04074.x

Easley, David et al. (2010)
Factoring Information into Returns
Journal of Financial and Quantitative Analysis, Volume 45, Issue 2, pp. 293 - 309
doi: 10.1017/S0022109010000074

Ersan, Oguz and Alici, Asli (2016)
An unbiased computation methodology for estimating the probability of informed trading (PIN)
Journal of International Financial Markets, Institutions and Money, Volume 43, pp. 74 - 94
doi: 10.1016/j.intfin.2016.04.001

Gan, Quan et al. (2015)
A faster estimation method for the probability of informed trading using hierarchical agglomerative clustering
Quantitative Finance, Volume 15, Issue 11, pp. 1805 - 1821
doi: 10.1080/14697688.2015.1023336

Lin, Hsiou-Wei William and Ke, Wen-Chyan (2011)
A computing bias in estimating the probability of informed trading
Journal of Financial Markets, Volume 14, Issue 4, pp. 625 - 640
doi: 10.1016/j.finmar.2011.03.001

Revolution Analytics and Steve Weston (2015)
doParallel: Foreach Parallel Adaptor for the 'parallel' Package
R package version 1.0.10

Revolution Analytics and Steve Weston (2015)
foreach: Provides Foreach Looping Construct for R
R package version 1.4.3 Yan, Yuxing and Zhang, Shaojun (2012)
An improved estimation method and empirical properties of the probability of informed trading
Journal of Banking & Finance, Volume 36, Issue 2, pp. 454 - 467
doi: 10.1016/j.jbankfin.2011.08.003

See Also

`nlminb`, `initial_vals` `pin_est` `qpin` `pin_confint`

Examples

```
# Loading simulated data for frequently traded stock

data("BSfrequent")

# Generate several matrices of initial values utilizing all methods implemented
```

```
inits_grid <- initial_vals(numbuys = BSfrequent[,"Buys"],
                           numsells = BSfrequent[,"Sells"],
                           method = "Grid")

inits_hac <- initial_vals(numbuys = BSfrequent[,"Buys"],
                          numsells = BSfrequent[,"Sells"],
                          method = "HAC")

inits_hac_ref <- initial_vals(numbuys = BSfrequent[,"Buys"],
                              numsells = BSfrequent[,"Sells"],
                              method = "HAC_Ref")

# Optimization with different matrices of initial values

pin_core_grid <- pin_est_core(numbuys = BSfrequent[,"Buys"],
                              numsells = BSfrequent[,"Sells"],
                              factorization = "Lin_Ke", init_vals = inits_grid,
                              lower = rep(0,5), upper = c(1,1, rep(Inf,3)),
                              num_best_res = 5)

pin_core_hac <- pin_est_core(numbuys = BSfrequent[,"Buys"],
                             numsells = BSfrequent[,"Sells"],
                             factorization = "Lin_Ke", init_vals = inits_hac,
                             lower = rep(0,5), upper = c(1,1, rep(Inf,3)))

pin_core_hac_ref <- pin_est_core(numbuys = BSfrequent[,"Buys"],
                                 numsells = BSfrequent[,"Sells"],
                                 factorization = "Lin_Ke", init_vals = inits_hac_ref,
                                 lower = rep(0,5), upper = c(1,1, rep(Inf,3)))

## Not run:
pin_core_hac <- pin_est_core(numbuys = BSfrequent[,"Buys"],
                             numsells = BSfrequent[,"Sells"],
                             factorization = "Lin_Ke", init_vals = inits_hac,
                             lower = rep(0,5), upper = c(1,1, rep(Inf,3)),
                             confint = TRUE)

## End(Not run)
```

`pin_ll` *Likelihood factorizations*

Description

Evaluates likelihood function either utilizing factorization by Easley et. al (2010) or Lin and Ke (2011).

Usage

```
pin_ll(param = NULL, numbuys = NULL, numsells = NULL,
  factorization = "Lin_Ke")
```

Arguments

`param`	*numeric*: (named) vector of model parameters (valid names: `'alpha'`, `'delta'`, `'epsilon_b'`, `'epsilon_s'`, `'mu'`), length must equal 5
`numbuys`	*numeric*: vector of daily buys
`numsells`	*numeric*: vector of daily sells
`factorization`	*character*: switch between EHO ('EHO') and Lin-Ke ('Lin_Ke') factorization

Details

If names are not set for `param` or one or more of the vector names do not match the valid choices, they are internally set to `'alpha'`, `'delta'`, `'epsilon_b'`, `'epsilon_s'`, `'mu'` (in this order). Vectors for `numbuys` and `numsells` need to have same length.

Value

numeric: likelihood function value

References

Easley, David et al. (2010)
Factoring Information into Returns
Journal of Financial and Quantitative Analysis, Volume 45, Issue 2, pp. 293 - 309
doi: 10.1017/S0022109010000074

Lin, Hsiou-Wei William and Ke, Wen-Chyan (2011)
A computing bias in estimating the probability of informed trading
Journal of Financial Markets, Volume 14, Issue 4, pp. 625 - 640
doi: 10.1016/j.finmar.2011.03.001

`posterior` *Posterior Probabilities*

Description

Calculates posterior probabilities of conditions of trading days

Usage

```
posterior(param = NULL, numbuys = NULL, numsells = NULL)
```

Arguments

param	*numeric*: (named) vector of model parameters (valid names: 'alpha', 'delta', 'epsilon_b', 'epsilon_s', 'mu'), length must equal 5
numbuys	*numeric*: vector of daily buys
numsells	*numeric*: vector of daily sells

Details

For more details see corresponding section in vignette

Value

numeric matrix with columns 'no', 'good' and 'bad' representing posterior probabilities for the corresponding trading day conditions

See Also

ggplot.posterior

Examples

```
## Not run: See Vignette \code{browseVignette(package = 'pinbasic')}
```

---

qpin *Quarterly PIN estimates*

Description

Estimation of model parameters and probability of informed trading for quarterly data.

Usage

```
qpin(numbuys = NULL, numsells = NULL, dates = NULL,
  nlminb_control = list(), confint = FALSE, ci_control = list(),
  posterior = TRUE)
```

Arguments

numbuys	*numeric*: vector of daily buys
numsells	*numeric*: vector of daily sells
dates	see **Details**
nlminb_control	*list*: Control list for nlminb
confint	*logical*: Compute confidence intervals for PIN? Defaults to FALSE
ci_control	*list*: see pin_est_core
posterior	*logical*: Should posterior probabilities for conditions of trading days be computed?

Details

Wrapper around `pin_est` function and therefore inherits its settings for optimization. Data is split into quarters with the `quarter` function from **lubridate** package. According to the help page of this function `dates` argument must be
a date-time object of class POSIXct, POSIXlt, Date, chron, yearmon, yearqtr, zoo, zooreg, timeDate, xts, its, ti, jul, timeSeries, fts or anything else that can be converted with as.POSIXlt.
`nlminb` function in the **stats** package is used for maximization. Vectors for `numbuys` and `numsells` need to have same length.
Calculation of confidence interval for the probability of informed trading is disabled by default. For more details see `pin_est_core` or `pin_confint`.

Value

A list of lists with class 'qpin'. The length of the outer list equals the number of available quarters in the data. Naming scheme for the outer list is 'Year.QuarterNumber', where QuarterNumber equals an integer from 1 to 4. The inner list is structured as follows:

Results Matrix containing the parameter estimates as well as their estimated standard errors, t-values and p-values.

ll Value of likelihood function returned by `nlminb`

pin Value(s) of the estimated probability of informed trading

conv Convergence code for nlminb optimization

message Convergence message returned by the nlminb optimizer

iterations Number of iterations until convergence of nlminb optimizer

init_vals Vector of initial values

confint If `confint = TRUE`; confidence interval for the probability of informed trading

References

Easley, David et al. (2002)
Is Information Risk a Determinant of Asset Returns?
The Journal of Finance, Volume 57, Number 5, pp. 2185 - 2221
doi: 10.1111/15406261.00493

Easley, David et al. (1996)
Liquidity, Information, and Infrequently Traded Stocks
The Journal of Finance, Volume 51, Number 4, pp. 1405 - 1436
doi: 10.1111/j.15406261.1996.tb04074.x

Easley, David et al. (2010)
Factoring Information into Returns
Journal of Financial and Quantitative Analysis, Volume 45, Issue 2, pp. 293 - 309
doi: 10.1017/S0022109010000074

Ersan, Oguz and Alici, Asli (2016)
An unbiased computation methodology for estimating the probability of informed trading (PIN)
Journal of International Financial Markets, Institutions and Money, Volume 43, pp. 74 - 94
doi: 10.1016/j.intfin.2016.04.001

Gan, Quan et al. (2015)
A faster estimation method for the probability of informed trading using hierarchical agglomerative clustering
Quantitative Finance, Volume 15, Issue 11, pp. 1805 - 1821
doi: 10.1080/14697688.2015.1023336

Grolemund, Garett and Wickham, Hadley (2011)
Dates and Times Made Easy with lubridate
Journal of Statistical Software, Volume 40, Issue 3, pp. 1 - 25
doi: 10.18637/jss.v040.i03

Lin, Hsiou-Wei William and Ke, Wen-Chyan (2011)
A computing bias in estimating the probability of informed trading
Journal of Financial Markets, Volume 14, Issue 4, pp. 625 - 640
doi: 10.1016/j.finmar.2011.03.001

Yan, Yuxing and Zhang, Shaojun (2012)
An improved estimation method and empirical properties of the probability of informed trading
Journal of Banking & Finance, Volume 36, Issue 2, pp. 454 - 467
doi: 10.1016/j.jbankfin.2011.08.003

See Also

`nlminb`, `initial_vals` `pin_est` `pin_est_core` `pin_confint` `quarter` `year`

Examples

```
# Loading one year of simulated daily buys and sells

data('BSfrequent2015')

# Quarterly estimates for model parameters and the probability of informed trading
# Rownames of 'BSfrequent2015' equal the business days in 2015.

qpin2015 <- qpin(numbuys = BSfrequent2015[,"Buys"],
                 numsells = BSfrequent2015[,"Sells"],
                 dates = as.Date(rownames(BSfrequent2015), format = "%Y-%m-%d"))
```

---

simulateBS *Simulate trading data*

Description

Simulates a matrix consisting of synthetic data for daily buys and sells

Usage

```
simulateBS(param, ndays)
```

Arguments

param	*numeric*: (named) vector of model parameters (valid names: 'alpha', 'delta', 'epsilon_b', 'epsilon_s', 'mu'), length must equal 5
ndays	*integer*: Number of trading days for which aggregated buys and sells are simulated, defaults to 60

Details

If names are not set for param or one or more of the vector names do not match the valid choices, they are internally set to 'alpha', 'delta', 'epsilon_b', 'epsilon_s', 'mu' (in this order).

Value

numeric: Matrix with ndays rows and two columns which are named 'Buys' and 'Sells'.

Index