

# Package ‘LIHNPSD’

July 2, 2014

**Type** Package

**Title** Poisson Subordinated Distribution

**Version** 0.2.1

**Date** 2012-04-12

**Author** Stephen Horng-Twu Lihn <stevelihn@gmail.com>

**Maintainer** Stephen Horng-Twu Lihn <stevelihn@gmail.com>

**Description** A Poisson Subordinated Distribution to capture major leptokurtic features in log-return time series of financial data.

**License** GPL-2

**Depends** R (>= 2.14.1), sn, moments, BB, Bolstad2, optimx, Rmpfr

**URL** [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2032762](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2032762)

**Repository** CRAN

**Date/Publication** 2012-04-13 05:19:27

**NeedsCompilation** no

## R topics documented:

LIHNPSD-package . . . . .	3
.N . . . . .	4
calcqq . . . . .	4
density . . . . .	5
dji_logr . . . . .	5
generatepdf . . . . .	6
gold_logr . . . . .	6
LihnBetaPoly . . . . .	7
LihnFunctionAnalytic . . . . .	7
LihnFunctionKth . . . . .	8

LihnFunctionSum . . . . .	8
LihnFunctionValidate . . . . .	9
LIHNPSD_plotcdf . . . . .	9
LIHNPSD_plotlogpdf . . . . .	10
LIHNPSD_plotpdf . . . . .	11
LIHNPSD_plotqq . . . . .	11
LIHNPSD_plot_std4gr . . . . .	12
LIHNPSD_prepare_data . . . . .	12
LIHNPSD_standardfit_fn . . . . .	13
LIHNPSD_standardfit_test . . . . .	14
LIHNPSD_theoretical_result . . . . .	15
LIHNPSD_UnitTest . . . . .	16
LihnTildeFunction . . . . .	16
MergeTailHistogram . . . . .	17
MergeTailHistogramOneSide . . . . .	18
mu1_analytic . . . . .	18
mu2_analytic . . . . .	19
mu3_analytic . . . . .	20
mu4_analytic . . . . .	20
mu_n_core . . . . .	21
poisson_sum . . . . .	22
poisson_sum_kth . . . . .	22
prepare . . . . .	23
psdkernel . . . . .	23
psdkurtosis . . . . .	24
psdmagnitude . . . . .	25
psdskewness . . . . .	25
psdunittest . . . . .	26
psdvariance . . . . .	26
r10y_logr . . . . .	27
rawcdf . . . . .	27
rawcdfinv . . . . .	28
rawdensity . . . . .	28
rawdensity0 . . . . .	29
rawdensityslope . . . . .	30
rawdensity_kth . . . . .	30
rawmean . . . . .	31
rawmu1 . . . . .	32
rawmu2 . . . . .	32
rawmu3 . . . . .	33
rawmu4 . . . . .	34
rawsn . . . . .	34
SPSD . . . . .	35
standardfit . . . . .	36
szd_logr . . . . .	37
tailindex . . . . .	37
tailindex_plot . . . . .	38
TimeSeriesLogReturn . . . . .	38

<i>LIHNPSD-package</i>	3
vix_logr . . . . .	39
<b>Index</b>	<b>40</b>

## Description

A new Poisson subordinated distribution is proposed to capture major leptokurtic features in log-return time series of financial data. This distribution is intuitive, easy to calculate, and converge quickly. It fits well to the historical daily log-return distributions of currencies, commodities, Treasury yields, VIX, and, most difficult of all, DJIA. It serves as a viable alternative to the more sophisticated truncated stable distribution.

## Author(s)

Stephen Horng-Twu Lihn <stevelihn@gmail.com>

## References

On a Poisson Subordinated Distribution for Precise Statistical Measurement of Leptokurtic Financial Data, SSRN 2032762, [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2032762](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2032762).

## See Also

[dji\\_logr](#), [rawmean](#), [rawdensity](#), [LIHNPSD\\_prepare\\_data](#), [LIHNPSD\\_theoretical\\_result](#), [LIHNPSD\\_plot\\_std4gr](#)

## Examples

```
# Load the daily log-return data of DJIA
data(dji_logr)

# Construct the S3 object for PSD
dist <- list( sigma= 0.004625, alpha= 0.292645, gamma= 0.482744, beta= -0.154049, location= 0.002968 )
class(dist) <- "LIHNPSD"
dist <- rawmean(dist)

# A simple graph of the distribution's log PDF
x <- seq(-0.1,0.1,by=0.1/1000)
plot( x, log(rawdensity(dist,x)), pch=".")

# The more sophisticated fit and graphs
dt <- LIHNPSD_prepare_data(dji_logr, breaks=160, merge_tails=c(4,2))
th <- LIHNPSD_theoretical_result(dist, dt)
LIHNPSD_plot_std4gr(th, dt)
```

---

`.N` *Simple MPFR constructor*

---

### Description

Construct an MPFR object with provided precision for further use in constructing PSD object in MPFR mode

### Usage

```
.N(x, p=128)
```

### Arguments

<code>x</code>	The number
<code>p</code>	The precision, typically one of 64, 96, 128.

### Value

Return an MPFR object

### Examples

```
# Convert the number 0.5 to 128-digit precision in MPFR
.N( 0.5, 128 )
```

---

`calcqq` *Calculate quantile-to-quantile object from PSD and histogram*

---

### Description

Calculate quantile-to-quantile object from PSD and histogram

### Usage

```
calcqq(d, hq, step = 5, debug = 0)
```

### Arguments

<code>d</code>	The PSD S3 object
<code>hq</code>	<code>list( qhx = h\$mids, qhy = h\$counts )</code> where <code>h</code> is the histogram
<code>step</code>	The steps of PSD movement
<code>debug</code>	0/1: Print debug (mostly timing) message or not

**Value**

x	Data's x values
xq	Data's CDF(x)
y	PSD fit's x values
yq	PSD fit's CDF(x)

---

density	<i>The probability density function of PSD with location parameter.</i>
---------	---

---

**Description**

The probability density function of PSD with location parameter. Since location parameter is included, the mean of the PDF is always adjusted to reflect the location parameter.

**Usage**

```
density(d, x)
```

**Arguments**

d	A fully specified PSD S3 object
x	x of PDF

**Value**

Return PDF(x)

**See Also**

[rawdensity](#)

---

dji_logr	<i>Log-return of DJIA</i>
----------	---------------------------

---

**Description**

Log-return data of DJIA from 1930 to 2011

**Usage**

```
data(dji_logr)
```

**References**

See Yahoo Finance ^DJI for more details.

---

generatepdf                      *Internal utility to generate (raw) PDF*

---

**Description**

Internal utility to generate (raw) PDF

**Usage**

```
generatepdf(d, NS, NT, raw=1)
```

**Arguments**

d	A fully specified PSD S3 object
NS	Extend x-axis to number of sigma
NT	Number of tick samples per unit of sigma
raw	1: Use rawdensity; 0: Use density

**Value**

N	Number of data points
x	Array of x
dx	delta x
pb	Array of PDF(x)

**See Also**

[density](#), [rawdensity](#)

---

gold\_logr                      *Log-return of spot gold*

---

**Description**

Log-return data of spot gold (London PM fixing) from 1972 to 2009

**Usage**

```
data(gold_logr)
```

**References**

See LBMA website for more details.

---

LihnBetaPoly	<i>The beta polynomial</i>
--------------	----------------------------

---

**Description**

The beta polynomial in the general form of the N-th moment

**Usage**

```
LihnBetaPoly(N, b)
```

**Arguments**

N	The N-th polynomial
b	Beta

**Value**

The numeric value of the polynomial

**References**

See Section "General Form of the N-th Moment" in the PSD paper

**Examples**

```
# g_1(b) = b  
LihnBetaPoly(1, 0.5)
```

---

LihnFunctionAnalytic	<i>Analytic form of Lihn function</i>
----------------------	---------------------------------------

---

**Description**

Analytic form of Lihn function for integer alpha

**Usage**

```
LihnFunctionAnalytic(alpha, x)
```

**Arguments**

alpha	The order of Lihn function. Must be an integer from -1 to 4
x	The x parameter

**Value**

The numeric value of Lihn function

**References**

Appendix A of the PSD paper.

**See Also**

[LihnFunctionSum](#)

**Examples**

```
# L_1(x) = x+1
LihnFunctionAnalytic(1,1)
```

---

LihnFunctionKth	<i>The k-th term in the sum of Lihn function</i>
-----------------	--

---

**Description**

The k-th term in the sum of Lihn function. This is used internally by LihnFunctionSum.

**See Also**

[LihnFunctionSum](#)

---

LihnFunctionSum	<i>The summation form of Lihn function</i>
-----------------	--

---

**Description**

The summation form of Lihn function

**Usage**

```
LihnFunctionSum(alpha, x, epsilon = 1e-10)
```

**Arguments**

alpha	The order of Lihn function
x	The x parameter
epsilon	tolerance of error



**Value**

The numeric value of Lihn function

**References**

Appendix A of the PSD paper.

**See Also**

[LihnFunctionAnalytic](#)

**Examples**

```
# L_1(x) = x+1
LihnFunctionSum(1,1)
```

---

LihnFunctionValidate    *Validation of Lihn function implementations*

---

**Description**

Validation of Lihn function implementations

**Usage**

```
LihnFunctionValidate()
```

**Examples**

```
LihnFunctionValidate()
```

---

LIHNPSD\_plotcdf    *Internal utility to plot cdf chart*

---

**Description**

Internal utility to plot cdf chart used by LIHNPSD\_plot\_std4gr

**Usage**

```
LIHNPSD_plotcdf(dist, h, data_st, tx, tcdf, xlab = "log(r)", main = "PSD CDF")
```

**Arguments**

dist	the PSD S3 object
h	The histogram
data_st	The descriptive statistics of the data
tx	The x axis for theoretical plot, usually it is tx from LIHNPSD_theoretical_result
tcdf	The CDF for theoretical plot
xlab	The x-axis label
main	The main label

**Value**

N/A

**See Also**

[LIHNPSD\\_plot\\_std4gr](#), [LIHNPSD\\_theoretical\\_result](#)

---

LIHNPSD\_plotlogpdf      *Internal utility to plot log-pdf chart*

---

**Description**

Internal utility to plot cdf chart used by LIHNPSD\_plot\_std4gr

**Usage**

```
LIHNPSD_plotlogpdf(dist, h, tx, tpdf, xlab = "log(r)", main = "PSD Log PDF")
```

**Arguments**

dist	The PSD S3 object
h	The histogram
tx	The x axis for theoretical plot, usually it is tx from LIHNPSD_theoretical_result
tpdf	The PDF for theoretical plot
xlab	The x-axis label
main	The main label

**Value**

N/A

**See Also**

[LIHNPSD\\_plot\\_std4gr](#), [LIHNPSD\\_theoretical\\_result](#)

---

LIHNPSD\_plotpdf      *Internal utility to plot pdf chart*

---

**Description**

Internal utility to plot pdf chart used by LIHNPSD\_plot\_std4gr

**Usage**

```
LIHNPSD_plotpdf(dist, h, tx, tpdf, xlab = "log(r)", main = "PSD PDF")
```

**Arguments**

dist	the PSD S3 object
h	The histogram
tx	The x axis for theoretical plot, usually it is tx from LIHNPSD_theoretical_result
tpdf	The PDF for theoretical plot
xlab	The x-axis label
main	The main label

**Value**

N/A

**See Also**

[LIHNPSD\\_plot\\_std4gr](#), [LIHNPSD\\_theoretical\\_result](#)

---

LIHNPSD\_plotqq      *Internal utility to plot qq chart*

---

**Description**

Internal utility to plot qq chart used by LIHNPSD\_plot\_std4gr

**Usage**

```
LIHNPSD_plotqq(dist, qqp, merge_tails, main = "PSD QQ-Plot")
```

**Arguments**

dist	the PSD S3 object
qqp	The qq-plot data set, usually output of calcqq
merge_tails	Specify the numbers of data points to merge in both tails when processing histogram
main	The main label

**Value**

N/A

**See Also**[LIHNPSD\\_plot\\_std4gr](#), [calcqq](#)


---

LIHNPSD\_plot\_std4gr     *Standard utility to plot a 4-chart graph on a given data set and PSD fit*

---

**Description**

Standard utility to plot a 4-chart graph on a given data set and PSD fit.

**Usage**

```
LIHNPSD_plot_std4gr(th, dt, EPS, file = NA)
```

**Arguments**

th	The theoretical object from LIHNPSD_theoretical_result
dt	The data set object from LIHNPSD_prepare_data
EPS	TRUE: plot for eps file; FALSE: plot to screen
file	File name of the eps output

**Value**

No output value

---

LIHNPSD\_prepare\_data     *Prepare the data set object from log-return series*

---

**Description**

Prepare the data set object from log-return series

**Usage**

```
LIHNPSD_prepare_data(logr, breaks, merge_tails)
```

**Arguments**

logr	The log-return series
breaks	Breaks for histogram
merge_tails	Specify the numbers of data points to merge in both tails when processing histogram

**Value**

logr	The log-return series from the input
N	The length of logr
breaks	Breaks specified from the input
merge_tails	merge_tails from the input
stats	The descriptive statistics of logr: <code>c(mean(logr), sqrt(var(logr)), skewness(logr), kurtosis(logr))</code>
h	The histogram of logr
hq	The tail-merged histogram

**See Also**

[standardfit](#)

---

LIHNPSD\_standardfit\_fn

*Internal utility to perform nonlinear fit using spg*

---

**Description**

Internal utility to perform nonlinear fit using spg

**Usage**

```
LIHNPSD_standardfit_fn(psd, data_stats, hist, plotqq = 1, weights = list(), merge_tails = c(0, 0), debug = FALSE)
```

**Arguments**

psd	An internal array representing a guess of the PSD parameters
data_stats	The descriptive statistics of the input data
hist	Input histogram
plotqq	TRUE/FALSE: Plot intermediate charts or not
weights	Specify the weights of each component in the nonlinear fit, defaults are 1.
merge_tails	Specify the numbers of data points to merge in both tails when processing histogram
debug	TRUE/FALSE: print debug messages or not

**Value**

A numeric value representing the error of the fit

**See Also**

[standardfit](#)

---

LIHNPSD\_standardfit\_test

*Internal utility to test the stability of LIHNPSD\_standardfit\_fn*

---

**Description**

Internal utility to test the stability of LIHNPSD\_standardfit\_fn

**Usage**

```
LIHNPSD_standardfit_test(d, r, hist, plotqq = 1, weights = list(), merge_tails = c(0, 0))
```

**Arguments**

d	A PSD S3 object representing initial guess of the PSD parameters
r	Input log-return series
hist	Input histogram
plotqq	TRUE/FALSE: Plot intermediate charts or not
weights	Specify the weights of each component in the nonlinear fit, defaults are 1.
merge_tails	Specify the numbers of data points to merge in both tails when processing histogram

**Value**

The debug output of LIHNPSD\_standardfit\_fn

**See Also**

[LIHNPSD\\_standardfit\\_fn](#)

---

 LIHNPSD\_theoretical\_result

*Prepare the theoretical result on a given data set and PSD fit*

---

### Description

Prepare the theoretical result on a given data set and PSD fit

### Usage

```
LIHNPSD_theoretical_result(dist, dt, N=5000)
```

### Arguments

dist	A PSD S3 object to evaluate theoretical result. Location parameter is included.
dt	The data set output from LIHNPSD_prepare_data
N	The number of data points when calculating PDF / CDF

### Value

dist	the PSD S3 object from the input
N	N from the input
qqp	Output from calcqq
merge_tails	Copied from dt
tx	x by seq(min(dt\$logr), max(dt\$logr), length=N+1)
dx	dx on tx
tpdf	Theoretical PDF calculated on tx
tcdf	Theoretical CDF calculated on tx

### See Also

[calcqq](#), [density](#), [rawcdf](#)

---

LIHNPSD\_UnitTest      *Perform major unit tests on the package*

---

**Description**

Perform major unit tests on the package

**Usage**

LIHNPSD\_UnitTest(mpfr = 0)

**Arguments**

mpfr                      If non-zero value is specified, MPFR will be used.

**Value**

Error if not passed

**See Also**

[psdunittest](#)

---

LihnTildeFunction      *The tilde Lihn function*

---

**Description**

The tilde Lihn function

**Usage**

LihnTildeFunction(x, alpha, p, epsilon = 1e-10)

**Arguments**

x                          The x parameter  
alpha                      The alpha parameter  
p                            The p parameter  
epsilon                    the tolerance of error

**Value**

The numeric value of the tilde Lihn function



## References

Section "Pareto Tail" of the PSD paper.

## Examples

```
LihnTildeFunction(0,0,0)
# should equal to 1/sqrt(2*pi)
```

---

MergeTailHistogram      *Utility function to merge tails in the histogram*

---

## Description

Utility function to merge data points in the tails of the histogram.

## Usage

```
MergeTailHistogram(q, merge_tails)
```

## Arguments

**q**                      In the form of `list( qhx = h$mids, qhy = h$counts )` where `h` is a histogram

**merge\_tails**          Number of data points to merge on each side, in the form of `c(left, right)`

## Value

Modified `q` after the merge

## Examples

```
# Load the daily log-return data of DJIA
data(dji_logr)
h <- hist(dji_logr, breaks = 60, plot = FALSE)
# Merge two data points from the left tail, and one data points from the right tail
q2 <- MergeTailHistogram( list(qhx=h$mids, qhy=h$counts), c(2,1) )
```

---

MergeTailHistogramOneSide

*Internal utility function to merge the left tail in the histogram*


---

**Description**

Internal utility function to merge the left tail in the histogram

**Usage**

```
MergeTailHistogramOneSide(q, allowed_merge)
```

**Arguments**

q                    In the form of `list( qhx = h$mids, qhy = h$counts )` where h is a histogram

allowed\_merge      Number of data points that should be merged in the left tail

**Value**

Modified q after the merge

**See Also**

[MergeTailHistogram](#)

---

mu1\_analytic

*Calculate the analytic form of the first moment (mean) of PSD without location parameter*


---

**Description**

Calculate the analytic form of the first moment (mean) of PSD without location parameter. The analytic form uses Lihn function.

**Usage**

```
mu1_analytic(d)
```

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the mean

**References**

See Section "The Mean" in the PSD paper

**See Also**

[rawmean](#), [rawmu1](#)

---

mu2_analytic	<i>Calculate the analytic form of the second moment of PSD without location parameter</i>
--------------	---

---

**Description**

Calculate the analytic form of the second moment of PSD without location parameter. The analytic form uses Lihn function.

**Usage**

```
mu2_analytic(d)
```

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the second moment

**References**

See Section "The Variance" in the PSD paper

**See Also**

[rawmu2](#)

---

mu3_analytic	<i>Calculate the analytic form of the third moment of PSD without location parameter</i>
--------------	--

---

**Description**

Calculate the analytic form of the third moment of PSD without location parameter. The analytic form uses Lihn function.

**Usage**

```
mu3_analytic(d)
```

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the third moment

**References**

See Section "The Skewness" in the PSD paper

**See Also**

[rawmu3](#)

---

mu4_analytic	<i>Calculate the analytic form of the fourth moment of PSD without location parameter</i>
--------------	---

---

**Description**

Calculate the analytic form of the fourth moment of PSD without location parameter. The analytic form uses Lihn function.

**Usage**

```
mu4_analytic(d)
```

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the fourth moment

**References**

See Section "The Kurtosis" in the PSD paper

**See Also**

[rawmu4](#)

---

mu_n_core	<i>Calculate the analytic core part of N-th moment of PSD without location parameter</i>
-----------	--

---

**Description**

Calculate the analytic core part of N-th moment of PSD without location parameter. The analytic form uses Lihn function. The core part does not include the the beta polynomial ([LihnBetaPoly](#)) since it does not have a general analytic form.

**Usage**

```
mu_n_core(d, n)
```

**Arguments**

d	A fully specified PSD S3 object
n	The N-th moment

**Value**

Numeric value of the core part of the N-th moment

**References**

See Section "General Form of the N-th Moment" in the PSD paper

**See Also**

[LihnBetaPoly](#)

---

poisson_sum	<i>Utility to calculate the Poisson sum</i>
-------------	---

---

**Description**

Utility to calculate the Poisson sum

**Usage**

```
poisson_sum(d, fn)
```

**Arguments**

d	A fully specified PSD S3 object
fn	The input function

**Value**

The numeric value of the sum on fn

---

poisson_sum_kth	<i>Internal utility to calculate the k-th item of the Poisson sum</i>
-----------------	---

---

**Description**

Internal utility to calculate the k-th item of the Poisson sum

**Usage**

```
poisson_sum_kth(d, fn, k)
```

**Arguments**

d	A fully specified PSD S3 object
fn	The input function
k	Specify the k-th item

**Value**

The numeric value of the k-th item on fn

**See Also**

[poisson\\_sum](#)

---

prepare	<i>Internal utility to construct a PSD S3 object</i>
---------	--

---

**Description**

Internal utility to construct a PSD S3 object. This utility makes sure unspecified parameters are set to zero. If MPFR precision is set, all the parameters will be converted to MPFR. Several standard numbers are also stored within the object for easy access, such as pi, e. This utility is for internal use most of time, which is wrapped by [rawmean](#).

**Usage**

```
prepare(d)
```

**Arguments**

d                    A primitive PSD S3 object

**Value**

A fully specified PSD S3 object

**See Also**

[rawmean](#)

**Examples**

```
# Construct the S3 object for PSD
dist <- list( sigma= 0.004625, alpha= 0.292645, gamma= 0.482744, beta= -0.154049, location= 0.002968 )
class(dist) <- "LIHNPSD"
dist <- prepare(dist)
```

---

psdkernel	<i>Calculate the ratio of the actual volatility vs the unit volatility</i>
-----------	--

---

**Description**

Calculate the ratio of the actual volatility vs the unit volatility

**Usage**

```
psdkernel(d, k)
```

**Arguments**

d	A fully specified PSD S3 object
k	The k in the Poisson distribution

**Value**

The numeric value of the kernel

---

psdkurtosis	<i>Calculate the kurtosis based on Poisson sum of moments</i>
-------------	---

---

**Description**

Calculate the kurtosis based on Poisson sum of moments.

**Usage**

```
psdkurtosis(d)
```

**Arguments**

d	A fully specified PSD S3 object
---	---------------------------------

**Value**

Numeric value of the kurtosis

**References**

See Section "The Kurtosis" in the PSD paper

**See Also**

[rawmu4](#)



---

psdmagnitude                      *A simplistic algorithm of estimating magnitude of shock*

---

**Description**

A simplistic algorithm of estimating magnitude of shock

**Usage**

```
psdmagnitude(d, r, step=0.1)
```

**Arguments**

d	A fully specified PSD S3 object
r	Log-return of the day
step	Fractional k step for the precision of the estimate, default is 0.1.

**Value**

Estimated k magnitude

---

psdskewness                      *Calculate the skewness based on Poisson sum of moments*

---

**Description**

Calculate the skewness based on Poisson sum of moments.

**Usage**

```
psdskewness(d)
```

**Arguments**

d	A fully specified PSD S3 object
---	---------------------------------

**Value**

Numeric value of the skewness

**References**

See Section "The Skewness" in the PSD paper

**See Also**

[rawmu3](#)

---

psdunittest	<i>Perform unit tests on the specified PSD S3 object</i>
-------------	--

---

**Description**

Perform unit tests on the specified PSD S3 object

**Usage**

```
psdunittest(d)
```

**Arguments**

d	A PSD S3 object
---	-----------------

**Value**

Error if not passed

**See Also**

[LIHNPSD\\_UnitTest](#)

---

psdvariance	<i>Calculate the variance based on Poisson sum of moments</i>
-------------	---

---

**Description**

Calculate the variance based on Poisson sum of moments.

**Usage**

```
psdvariance(d)
```

**Arguments**

d	A fully specified PSD S3 object
---	---------------------------------

**Value**

Numeric value of the variance

**References**

See Section "The Variance" in the PSD paper

**See Also**

[rawmu2](#)

---

r10y_logr	<i>Log-return of R10Y</i>
-----------	---------------------------

---

**Description**

Log-return data of R10Y (10-Year Treasury yield) from 1962 to 2011

**Usage**

```
data(r10y_logr)
```

**References**

See Federal Reserve Board website for more details.

---

rawcdf	<i>Calculate the raw CDF</i>
--------	------------------------------

---

**Description**

Calculate the raw CDF

**Usage**

```
rawcdf(d, x)
```

**Arguments**

d	A fully specified PSD S3 object
x	x of CDF

**Value**

Return CDF(x)

**References**

See Section "Development of PSD" in the PSD paper

**See Also**

[rawcdfinv](#)

rawcdfinv

*Calculate Inverse of CDF using Newton's method*

---

**Description**

Calculate Inverse of CDF using Newton's method

**Usage**

```
rawcdfinv(d, c, xinit)
```

**Arguments**

d	A fully specified PSD S3 object
c	CDF
xinit	Initial guess of x. Use 0 if not sure.

**Value**

Return x where  $CDF(x)=c$

**References**

See Section "Development of PSD" in the PSD paper

**See Also**

[rawcdf](#)

---

rawdensity*The probability density function of PSD without location parameter*

---

**Description**

The probability density function of PSD without location parameter. Since there is no additional location parameter, the PDF is exactly the same as what is described in the PSD paper. We use "raw" to differentiate such PDF from the more complicated (yet more complete) PDF with location parameter.

**Usage**

```
rawdensity(d, x)
```

**Arguments**

d	A fully specified PSD S3 object
x	x of PDF

**Value**

Return PDF(x)

**References**

See Section "Development of PSD" in the PSD paper

**See Also**

[SPSD, density](#)

---

rawdensity0	<i>The raw PDF at x=0</i>
-------------	---------------------------

---

**Description**

The raw PDF at  $x=0$ . It is implemented from the analytic result primarily for internal validation purpose.

**Usage**

rawdensity0(d)

**Arguments**

d	A fully specified PSD S3 object
---	---------------------------------

**Value**

Return PDF(x)

**References**

See Section "Development of PSD" in the PSD paper

**See Also**

[rawdensity](#)

---

rawdensityslope	<i>Calculate the slope of the PDF (dP/dx)</i>
-----------------	---

---

**Description**

Calculate the slope of the PDF (dP/dx)

**Usage**

```
rawdensityslope(d, x)
```

**Arguments**

d	A fully specified PSD S3 object
x	x of dP/dx

**Value**

Return dP/dx

**References**

See Section "Tail Index" in the PSD paper

**See Also**

[rawdensity](#)

---

rawdensity_kth	<i>The k-th item of the raw PDF</i>
----------------	-------------------------------------

---

**Description**

The k-th item of the raw PDF. This is used primarily to understand the internal structure of the subordination.

**Usage**

```
rawdensity_kth(d, x, k)
```

**Arguments**

d	A fully specified PSD S3 object
x	x of PDF
k	The k-th item in the Poisson sum

**Value**

Return PDF(x)

**References**

See Section "Development of PSD" in the PSD paper

**See Also**

[rawdensity](#)

---

rawmean

*Utility to construct a PSD S3 object with calculated mean*

---

**Description**

Utility to construct a PSD S3 object with calculated mean. Mean of a PSD distribution is required in many calculations. So it makes sense to calculate it and store it in the S3 object once for all. The word "raw" means the calculation is performed without considering the location parameter.

**Usage**

```
rawmean(d)
```

**Arguments**

d                    A PSD S3 object

**Value**

A fully specified PSD S3 object with calculated mean

**References**

See Section "The Mean" in the PSD paper

**Examples**

```
# Construct the S3 object for PSD
dist <- list( sigma= 0.004625, alpha= 0.292645, gamma= 0.482744, beta= -0.154049, location= 0.002968 )
class(dist) <- "LIHNPSD"
dist <- rawmean(dist)
```

---

rawmu1	<i>Calculate the Poisson sum of the first moment (mean) of PSD without location parameter</i>
--------	---

---

**Description**

Calculate the Poisson sum of the first moment (mean) of PSD without location parameter.

**Usage**

rawmu1(d)

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the mean

**References**

See Section "The Mean" in the PSD paper

**See Also**

[rawmean](#), [mu1\\_analytic](#)

---

rawmu2	<i>Calculate the Poisson sum of the second moment of PSD without location parameter</i>
--------	---

---

**Description**

Calculate the Poisson sum of the second moment of PSD without location parameter.

**Usage**

rawmu2(d)

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the second moment



**References**

See Section "The Variance" in the PSD paper

**See Also**

[psdvvariance](#), [mu2\\_analytic](#)

---

rawmu3	<i>Calculate the Poisson sum of the third moment of PSD without location parameter</i>
--------	--

---

**Description**

Calculate the Poisson sum of the third moment of PSD without location parameter.

**Usage**

```
rawmu3(d)
```

**Arguments**

d                    A fully specified PSD S3 object

**Value**

Numeric value of the third moment

**References**

See Section "The Skewness" in the PSD paper

**See Also**

[psdskewness](#), [mu3\\_analytic](#)

---

rawmu4	<i>Calculate the Poisson sum of the fourth moment of PSD without location parameter</i>
--------	---

---

**Description**

Calculate the Poisson sum of the fourth moment of PSD without location parameter.

**Usage**

```
rawmu4(d)
```

**Arguments**

d	A fully specified PSD S3 object
---	---------------------------------

**Value**

Numeric value of the fourth moment

**References**

See Section "The Kurtosis" in the PSD paper

**See Also**

[psdkurtosis](#), [mu4\\_analytic](#)

---

rawsn	<i>Internal utility for SN related function</i>
-------	---

---

**Description**

Internal utility for SN related function

**Usage**

```
rawsn(d, type, x, k)
```

**Arguments**

d	A fully specified PSD S3 object
type	1: for PDF, 2: for 2nd term of dP/dx
x	x of SN
k	The k-th item in the Poisson sum

**Value**

A numeric value

**See Also**

[rawdensity](#), [rawdensityslope](#)

---

 SSPSD

*Simple PSD constructor*

---

**Description**

Construct an S3 object for PSD in double-precision or MPFR

**Usage**

```
SPSD(sigma, alpha, gamma, beta=0, mpfr=0)
```

**Arguments**

sigma	sigma value of PSD
alpha	alpha value of PSD
gamma	gamma value of PSD
beta	optional beta value of PSD for skewness
mpfr	optional mpfr precision. Default is 0, which sets all calculations in double precision. For MPFR, set it to an integer, typically one of 64, 96, 128.

**Value**

Return an S3 object of LIHNPSD class that can be used for subsequent calculation.

**Note**

This constructor doesn't include the location parameter.

**See Also**

See also package's example for the DJIA parameters.

**Examples**

```
# Normal distribution
SPSD( 1,0,0 )
# PSD that approximate DJIA
SPSD(0.004625, 0.292645, 0.482744, -0.154049)
```

---

 standardfit

*Standard utility to perform nonlinear PSD fit*


---

**Description**

Standard utility to perform nonlinear PSD fit

**Usage**

```
standardfit(d, r, hist, trace, iter, plotqq, weights, merge_tails)
```

**Arguments**

d	A PSD S3 object representing initial guess of the PSD parameters
r	Input log-return series
hist	Input histogram
trace	TRUE/FALSE: turn trace on/off
iter	Maximum number of iterations
plotqq	TRUE/FALSE: Plot intermediate charts or not
weights	Specify the weights of each component in the nonlinear fit, defaults are 1.
merge_tails	Specify the numbers of data points to merge in both tails when processing histogram

**Value**

dist	A PSD S3 object representing best nonlinear fit
psdout	The output of optimix/psg function. This is for debugging purpose only.

**See Also**

[LIHNPSD\\_standardfit\\_fn](#)

**Examples**

```
# Load the daily log-return data of DJIA
data(szd_logr)

# Prepare the input data set
merge_tails <- c(1,3)
dt <- LIHNPSD_prepare_data(szd_logr, breaks=68, merge_tails=merge_tails)

# Prepare the input PSD
dist <- list( sigma= 0.0036, alpha= 0.9, gamma= 0.0, beta= -0.014 )
class(dist) <- "LIHNPSD"
dist <- rawmean(dist)
dist$location <- 0.00014
```

```

# Invoke the nonlinear fit (This will take some time!)
#fit <- standardfit(dist, dt$logr, dt$h, trace=1, iter=10,
#   plotqq=1, weights=list(m3=5,m4=1,qq_df=4), merge_tails=merge_tails )

# The final PSD
#dist <- fit$dist

```

---

szd\_logr

*Log-return of SZD/USD*


---

### Description

Log-return data of SZD/USD exchange rate from 1975 to 2008

### Usage

```
data(szd_logr)
```

### References

See Federal Reserve Board website for more details.

---

tailindex

*Calculate the tail index*


---

### Description

Calculate the tail index

### Usage

```
tailindex(d, x)
```

### Arguments

d	A fully specified PSD S3 object
x	x where the tail index is evaluated

### Value

Return the tail index

### References

See Section "Tail Index" in the PSD paper

---

tailindex_plot	<i>Generate the plot of tail index for the specified PSD</i>
----------------	--

---

**Description**

Generate the plot of tail index for the specified PSD

**Usage**

```
tailindex_plot(d, xmin, xmax, ymax=0.4)
```

**Arguments**

d	A fully specified PSD S3 object
xmin	Minimum of x-axis
xmax	Maximum of x-axis
ymax	Maximum of y-axis, default is 0.4

**Value**

N/A

**References**

See Section "Tail Index" in the PSD paper

**See Also**

[tailindex](#)

---

TimeSeriesLogReturn	<i>Convert price series to log-return series</i>
---------------------	--

---

**Description**

Convert daily price series to log-return series by a specified time interval

**Usage**

```
TimeSeriesLogReturn(pr, days)
```

**Arguments**

pr	Array of daily prices
days	Time interval, typically 1 for one day

**Value**

Array of log-return series

**Examples**

```
pr <- c( 100.0, 102.0, 106.0, 105.0 )  
logr <- TimeSeriesLogReturn(pr,1)
```

---

vix_logr	<i>Log-return of VIX</i>
----------	--------------------------

---

**Description**

Log-return data of VIX from 1990 to 2011

**Usage**

```
data(vix_logr)
```

**References**

See Yahoo Finance ^VIX for more details.

# Index

- \*Topic **CDF**
  - LIHNPSD\_plotcdf, 9
  - rawcdf, 27
  - rawcdfinv, 28
- \*Topic **Constructor**
  - SPSD, 35
- \*Topic **Lihn function**
  - LihnFunctionAnalytic, 7
  - LihnFunctionKth, 8
  - LihnFunctionSum, 8
  - LihnFunctionValidate, 9
- \*Topic **MPFR**
  - .N, 4
- \*Topic **PDF**
  - density, 5
  - generatepdf, 6
  - LIHNPSD\_plotlogpdf, 10
  - LIHNPSD\_plotpdf, 11
  - rawdensity, 28
  - rawdensity0, 29
  - rawdensity\_kth, 30
  - rawdensityslope, 30
  - rawsn, 34
- \*Topic **Poisson sum**
  - LihnFunctionKth, 8
  - LihnFunctionSum, 8
  - poisson\_sum, 22
  - poisson\_sum\_kth, 22
- \*Topic **analytic**
  - LihnFunctionAnalytic, 7
- \*Topic **constructor**
  - .N, 4
  - prepare, 23
  - rawmean, 31
- \*Topic **data**
  - dji\_logr, 5
  - gold\_logr, 6
  - r10y\_logr, 27
  - szd\_logr, 37
  - vix\_logr, 39
- \*Topic **histogram**
  - MergeTailHistogram, 17
  - MergeTailHistogramOneSide, 18
- \*Topic **package**
  - LIHNPSD-package, 3
- \*Topic **plot**
  - LIHNPSD\_plot\_std4gr, 12
  - LIHNPSD\_plotcdf, 9
  - LIHNPSD\_plotlogpdf, 10
  - LIHNPSD\_plotpdf, 11
  - LIHNPSD\_plotqq, 11
- \*Topic **statistics**
  - mu1\_analytic, 18
  - mu2\_analytic, 19
  - mu3\_analytic, 20
  - mu4\_analytic, 20
  - mu\_n\_core, 21
  - psdkurtosis, 24
  - psdskewness, 25
  - psdvariance, 26
  - rawmu1, 32
  - rawmu2, 32
  - rawmu3, 33
  - rawmu4, 34
- \*Topic **tail**
  - MergeTailHistogram, 17
  - MergeTailHistogramOneSide, 18
  - tailindex, 37
  - tailindex\_plot, 38
- \*Topic **testing**
  - LIHNPSD\_UnitTest, 16
  - psdunittest, 26
- .N, 4
- calcqq, 4, 12, 15
- density, 5, 6, 15, 29
- dji\_logr, 3, 5



generatepdf, 6  
gold\_logr, 6

LihnBetaPoly, 7, 21  
LihnFunctionAnalytic, 7, 9  
LihnFunctionKth, 8  
LihnFunctionSum, 8, 8  
LihnFunctionValidate, 9  
LIHNPSD (LIHNPSD-package), 3  
LIHNPSD-package, 3  
LIHNPSD\_plot\_std4gr, 3, 10–12, 12  
LIHNPSD\_plotcdf, 9  
LIHNPSD\_plotlogpdf, 10  
LIHNPSD\_plotpdf, 11  
LIHNPSD\_plotqq, 11  
LIHNPSD\_prepare\_data, 3, 12  
LIHNPSD\_standardfit\_fn, 13, 14, 36  
LIHNPSD\_standardfit\_test, 14  
LIHNPSD\_theoretical\_result, 3, 10, 11, 15  
LIHNPSD\_UnitTest, 16, 26  
LihnTildeFunction, 16

MergeTailHistogram, 17, 18  
MergeTailHistogramOneSide, 18  
mu1\_analytic, 18, 32  
mu2\_analytic, 19, 33  
mu3\_analytic, 20, 33  
mu4\_analytic, 20, 34  
mu\_n\_core, 21

poisson\_sum, 22, 22  
poisson\_sum\_kth, 22  
prepare, 23  
psdkernel, 23  
psdkurtosis, 24, 34  
psdmagnitude, 25  
psdskewness, 25, 33  
psdunittest, 16, 26  
psdvariance, 26, 33

r10y\_logr, 27  
rawcdf, 15, 27, 28  
rawcdfinv, 27, 28  
rawdensity, 3, 5, 6, 28, 29–31, 35  
rawdensity0, 29  
rawdensity\_kth, 30  
rawdensityslope, 30, 35  
rawmean, 3, 19, 23, 31, 32  
rawmu1, 19, 32  
rawmu2, 19, 26, 32  
rawmu3, 20, 25, 33  
rawmu4, 21, 24, 34  
rawsn, 34

SPSD, 29, 35  
standardfit, 13, 14, 36  
szd\_logr, 37

tailindex, 37, 38  
tailindex\_plot, 38  
TimeSeriesLogReturn, 38

vix\_logr, 39