Package ‘avar’

January 15, 2020

Type Package

Title Allan Variance

Version 0.1.1

Date 2020-01-15

LazyData true

Maintainer Stéphane Guerrier <stef.guerrier@gmail.com>

Description Implements the allan variance and allan variance linear regression estimator for latent time series models. More details about the method can be found, for example, in Guerrier, S., Molinari, R., & Stebler, Y. (2016) <doi:10.1109/LSP.2016.2541867>.

Depends R (>= 3.5.0)

License AGPL-3

Imports Rcpp, stats, simts

Suggests knitr, rmarkdown

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.0.2

VignetteBuilder knitr

URL https://github.com/SMAC-Group/avar

BugReports https://github.com/SMAC-Group/avar/issues

NeedsCompilation yes

Author Stéphane Guerrier [aut, cre],
James Balamuta [aut],
Gaetan Bakalli [aut],
Roberto Molinari [aut],
Justin Lee [aut],
Ahmed Radi [aut],
Haotian Xu [aut],
Yuming Zhang [aut],
Nathanael Claussen [aut]

Repository CRAN

Date/Publication 2020-01-15 19:10:03 UTC

R topics documented:

`adis_av` *Allan variance of IMU Data from an ADIS 16405 sensor*

Description

This data set contains Allan variance of gyroscope and accelerometer data from an ADIS 16405 sensor.

Usage

```
adis_av
```

Format

A list of the following elements:

- "sensor": Name of the sensor.
- "freq": The frequency at which the error signal is measured.
- "n": Sample size of the data.
- "type": The types of sensors considered in the data.
- "axis": The axes of sensors considered in the data.
- "avar": A list containing the computed Allan variance based on the data.

Source

The IMU data comes from Department of Geomatics Engineering, University of Calgary.

---

avlr *Computes the Allan Variance Linear Regression estimator*

Description

Estimate the parameters of time series models based on the Allan Variance Linear Regression (AVLR) approach

Usage

```
avlr(x, ...)

## Default S3 method:
avlr(
  x,
  qn = NULL,
  wn = NULL,
  rw = NULL,
  dr = NULL,
  ci = FALSE,
  B = 100,
  alpha = 0.05,
  ...
)

## S3 method for class 'imu_avar'
avlr(
  x,
  qn_gyro = NULL,
  wn_gyro = NULL,
  rw_gyro = NULL,
  dr_gyro = NULL,
  qn_acc = NULL,
  wn_acc = NULL,
  rw_acc = NULL,
  dr_acc = NULL,
  B = 100,
  alpha = 0.05,
  ...
)
```

Arguments

x	A vec of time series observations or an imu object.
...	Further arguments passed to other methods.
qn	A vec specifying on which scales the parameters of a Quantization Noise (QN) should be computed.
wn	A vec specifying on which scales the parameters of a White Noise (WN) should be computed.
rw	A vec specifying on which scales the parameters of a Random Wakk (RW) should be computed.
dr	A vec specifying on which scales the parameters of a Drift (DR) should be computed.
ci	A boolean to compute parameter confidence intervals.
B	A double for the number of bootstrap replicates to compute the parameter confidence intervals.
alpha	A double defining the level of the confidence interval (1 - ‘alpha‘).
qn_gyro	A vec specifying on which scales the parameters of a Quantization Noise (QN) should be computed for the gyroscope component.
wn_gyro	A vec specifying on which scales the parameters of a White Noise (WN) should be computed for the gyroscope component.
rw_gyro	A vec specifying on which scales the parameters of a Random Wakk (RW) should be computed for the gyroscope component.
dr_gyro	A vec specifying on which scales the parameters of a Drift (DR) should be computed for the gyroscope component.
qn_acc	A vec specifying on which scales the parameters of a Quantization Noise (QN) should be computed for the accelerometer component.
wn_acc	A vec specifying on which scales the parameters of a White Noise (WN) should be computed for the accelerometer component.
rw_acc	A vec specifying on which scales the parameters of a Random Wakk (RW) should be computed for the accelerometer component.
dr_acc	A vec specifying on which scales the parameters of a Drift (DR) should be computed for the accelerometer component.

Value

If the input x is a vec, then the function returns a list that contains:

- "estimates": The estimated value of the parameters.
- "implied_ad": The Allan deviation implied by the estimated parameters.
- "implied_ad_decomp": The Allan deviation implied by the estimated parameters for each individual model (if more than one is specified).
- "av": The avar object computed from the provided data.

If the input x is of the class imu_avar, then the function returns a list that contains:

- "gyro": The estimation results correseponding to the gyroscope component.
- "acc": The estimation results correseponding to the accelerometer component.
- "imu_av": The `imu_avar` object computed based on the IMU data.

Examples

```
set.seed(999)

N = 100000
Xt = rnorm(N) + cumsum(rnorm(N, 0, 3e-3))

av = avar(Xt)
plot(av)

# Input time series
fit = avlr(Xt, wn = 1:8, rw = 11:15)
fit

# Input directly Allan variance
fit = avlr(av, wn = 1:8, rw = 11:15)
fit

# Plot functions
plot(fit)
plot(fit, decomp = TRUE)
plot(fit, decomp = TRUE, show_scales = TRUE)
```

`av_ar1` *Calculate Theoretical Allan Variance for Stationary First-Order Autoregressive (AR1) Process*

Description

This function allows us to calculate the theoretical allan variance for stationary first-order autoregressive (AR1) process.

Usage

```
av_ar1(n, phi, sigma2)
```

Arguments

`n`	An `integer` value for the size of the cluster.
`phi`	A `double` value for the autocorrection parameter ϕ.
`sigma2`	A `double` value for the variance parameter σ^2.

Value

A `double` indicating the theoretical allan variance for AR1 process.

Note

This function is based on the calculation of the theoretical allan variance for stationary AR1 process raised in "Allan Variance of Time Series Models for Measurement Data" by Nien Fan Zhang, 2008, Metrologia, 45(5): 549. This calculation is fundamental and necessary for the study in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Author(s)

Yuming Zhang

Examples

```
av1 = av_ar1(n = 5, phi = 0.9, sigma2 = 1)
av2 = av_ar1(n = 8, phi = 0.5, sigma2 = 2)
```

---

av_dr — *Calculate Theoretical Allan Variance for Drift Process*

Description

This function allows us to calculate the theoretical allan variance for drift process.

Usage

```
av_dr(delta, n)
```

Arguments

`delta`	A `double` value for the noise parameter δ.
`n`	An `integer` value for the size of the cluster.

Value

A `double` indicating the theoretical allan variance for the drift process.

Note

This function is based on the calculation of the theoretical allan variance for drift process raised in "Allan Variance of Time Series Models for Measurement Data" by Nien Fan Zhang, 2008, Metrologia, 45(5): 549. This calculation is fundamental and necessary for the study in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Examples

```
av1 = av_dr(delta = 1, n = 5)
av2 = av_dr(delta = 2, n = 8)
```

`av_qn` *Calculate Theoretical Allan Variance for Stationary Quantization Noise Process*

Description

This function allows us to calculate the theoretical allan variance for stationary quantization noise process.

Usage

```
av_qn(Q2, n)
```

Arguments

Q2	A `double` value for the noise parameter Q^2.
n	An `integer` value for the size of the cluster.

Value

A `double` indicating the theoretical allan variance for the quantization noise process.

Note

This function is based on the calculation of the theoretical allan variance for stationary quantization noise process raised in "Allan Variance of Time Series Models for Measurement Data" by Nien Fan Zhang, 2008, Metrologia, 45(5): 549. This calculation is fundamental and necessary for the study in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Examples

```
av1 = av_qn(Q2 = 1, n = 5)
av2 = av_qn(Q2 = 2, n = 8)
```

av_rw — *Calculate Theoretical Allan Variance for Random Walk Process*

Description

This function allows us to calculate the theoretical allan variance for random walk process.

Usage

```
av_rw(omega2, n)
```

Arguments

omega2	A `double` value for the noise parameter ω^2.
n	An `integer` value for the size of the cluster.

Value

A `double` indicating the theoretical allan variance for the random walk process.

Note

This function is based on the calculation of the theoretical allan variance for random walk process raised in "Allan Variance of Time Series Models for Measurement Data" by Nien Fan Zhang, 2008, Metrologia, 45(5): 549. This calculation is fundamental and necessary for the study in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Examples

```
av1 = av_rw(omega2 = 1, n = 5)
av2 = av_rw(omega2 = 2, n = 8)
```

av_wn — *Calculate Theoretical Allan Variance for Stationary White Noise Process*

Description

This function allows us to calculate the theoretical allan variance for stationary white noise process.

Usage

```
av_wn(sigma2, n)
```

Arguments

sigma2	A double value for the variance parameter σ^2.
n	An integer value for the size of the cluster.

Value

A double indicating the theoretical allan variance for the white noise process.

Note

This function is based on the calculation of the theoretical allan variance for stationary white noise process raised in "Allan Variance of Time Series Models for Measurement Data" by Nien Fan Zhang, 2008, Metrologia, 45(5): 549. This calculation is fundamental and necessary for the study in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Examples

```
av1 = av_wn(sigma2 = 1, n = 5)
av2 = av_wn(sigma2 = 2, n = 8)
```

---

covmat_ar1blocks *Calculate Theoretical Covariance Matrix of AR(1) Blocks Process*

Description

This function allows us to calculate the theoretical covariance matrix of a non-stationary AR(1) blocks process.

Usage

```
covmat_ar1blocks(n_total, n_block, phi, sigma2)
```

Arguments

n_total	An integer indicating the length of the whole AR(1) blocks process.
n_block	An integer indicating the length of each block of the AR(1) blocks process.
phi	A double value for the autocorrection parameter ϕ.
sigma2	A double value for the variance parameter σ^2.

Value

The theoretical covariance matrix of the AR(1) blocks process.

Note

This function helps calculate the theoretical covariance matrix of a non-stationary process, AR(1) blocks. It is helpful to calculate the theoretical allan variance of non-stationary processes, which can be used to compare with the theoretical allan variance of stationary processes as shown in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Author(s)

Yuming Zhang

Examples

```
covmat1 = covmat_ar1blocks(n_total = 1000, n_block = 10,
phi = 0.9, sigma2 = 1)
covmat2 = covmat_ar1blocks(n_total = 800, n_block = 20,
phi = 0.5, sigma2 = 2)
```

---

covmat_bi *Calculate Theoretical Covariance Matrix of Bias-Instability Process*

Description

This function allows us to calculate the theoretical covariance matrix of a bias-instability process.

Usage

```
covmat_bi(sigma2, n_total, n_block)
```

Arguments

sigma2	A `double` value for the variance parameter σ^2.
n_total	An `integer` indicating the length of the whole bias-instability process.
n_block	An `integer` indicating the length of each block of the bias-instability process.

Value

The theoretical covariance `matrix` of the bias-instability process.

Note

This function helps calculate the theoretical covariance matrix of a non-stationary process, bias-instability. It is helpful to calculate the theoretical allan variance of non-stationary processes, which can be used to compare with the theoretical allan variance of stationary processes as shown in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Author(s)

Yuming Zhang

Examples

```
covmat1 = covmat_bi(sigma2 = 1, n_total = 1000, n_block = 10)
covmat2 = covmat_bi(sigma2 = 2, n_total = 800, n_block = 20)
```

---

covmat_nswn — *Calculate Theoretical Covariance Matrix of Non-Stationary White Noise Process*

Description

This function allows us to calculate the theoretical covariance matrix of a non-stationary white noise process.

Usage

```
covmat_nswn(sigma2, n_total)
```

Arguments

`sigma2`	A `double` value for the variance parameter σ^2.
`n_total`	An `integer` indicating the length of the whole non-stationary white noise process.

Value

The theoretical covariance `matrix` of the non-stationary white noise process.

Note

This function helps calculate the theoretical covariance matrix of a non-stationary process, non-stationary white noise. It is helpful to calculate the theoretical allan variance of non-stationary processes, which can be used to compare with the theoretical allan variance of stationary processes as shown in "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260.

Author(s)

Yuming Zhang

Examples

```
covmat1 = covmat_nswn(sigma2 = 1, n_total = 1000)
covmat2 = covmat_nswn(sigma2 = 2, n_total = 800)
```

fit_avlr *Internal function to the Allan Variance Linear Regression estimator*

Description

Estimate the parameters of time series models based on the Allan Variance Linear Regression (AVLR) approach

Usage

```
fit_avlr(qn, wn, rw, dr, ad, scales)
```

Arguments

qn	A vec specifying on which scales the parameters of a Quantization Noise (QN) should be computed.
wn	A vec specifying on which scales the parameters of a White Noise (WN) should be computed.
rw	A vec specifying on which scales the parameters of a Random Wakk (RW) should be computed.
dr	A vec specifying on which scales the parameters of a Drift (DR) should be computed.
ad	A vec of the Allan variance.
scales	A vec of the scales.

Value

A list with the estimated parameters.

imar_av *Allan variance of IMU Data from IMAR Gyroscopes*

Description

This data set contains Allan variance of IMAR gyroscopes data.

Usage

```
imar_av
```

Format

A list of the following elements:

- "sensor": Name of the sensor.
- "freq": The frequency at which the error signal is measured.
- "n": Sample size of the data.
- "type": The types of sensors considered in the data.
- "axis": The axes of sensors considered in the data.
- "avar": A list containing the computed Allan variance based on the data.

Source

The IMU data comes from Geodetic Engineering Laboratory (TOPO) and Swiss Federal Institute of Technology Lausanne (EPFL).

`kvh1750_av` *Allan variance of IMU Data from a KVH1750 IMU sensor*

Description

This data set contains Allan variance of gyroscope and accelerometer data from an KVH1750 sensor.

Usage

```
kvh1750_av
```

Format

A list of the following elements:

- "sensor": Name of the sensor.
- "freq": The frequency at which the error signal is measured.
- "n": Sample size of the data.
- "type": The types of sensors considered in the data.
- "axis": The axes of sensors considered in the data.
- "avar": A list containing the computed Allan variance based on the data.

Source

The IMU data comes from Department of Geomatics Engineering, University of Calgary.

MOAV *Non-stationary Maximal-overlapping Allan Variance*

Description

Calculation of the theoretical Maximal-overlapping Allan variance for constant-mean non-stationary time series data.

Usage

```
MOAV(n, covmat)
```

Arguments

`n`	An `integer` indicating the length of each vector of consecutive observations considered for the average.
`covmat`	A `matrix` indicating the T-by-T covariance matrix of the time series with length T.

Details

This calculation of Maximal-overlapping Allan variance is based on the definition on "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260. Here n is an integer larger than 1 and smaller than $floor\left(log_2\left(dim\left(covmat\right)[1]\right)\right) - 1$.

Value

A `field` <`numeric`> that is the theoretical Maximal-overlapping Allan variance for constant-mean non-stationary time series data.

Author(s)

Haotian Xu

Examples

```
set.seed(999)
Xt = arima.sim(n = 100, list(ar = 0.3))
avar(Xt, type = "to")

a = matrix(rep(0, 1000^2), nrow = 1000)
for (i in 1:1000){
  a[,i] = seq(from = 1 - i, length.out = 1000)
}
a.diag = diag(a)
a[upper.tri(a,diag=TRUE)] = 0
```

```
a = a + t(a) + diag(a.diag)
covmat = 0.3^a
sapply(1:8, function(y){MOAV(2^y, covmat)})
```

`navchip_av` *Allan variance of IMU Data from a navchip sensor*

Description

This data set contains Allan variance of gyroscope and accelerometer data from a navchip sensor.

Usage

```
navchip_av
```

Format

A list of the following elements:

- "sensor": Name of the sensor.
- "freq": The frequency at which the error signal is measured.
- "n": Sample size of the data.
- "type": The types of sensors considered in the data.
- "axis": The axes of sensors considered in the data.
- "avar": A list containing the computed Allan variance based on the data.

Source

The IMU data of the navchip sensor comes from Geodetic Engineering Laboratory (TOPO) and Swiss Federal Institute of Technology Lausanne (EPFL).

`NOAV` *Non-stationary Non-overlapping Allan Variance*

Description

Calculation of the theoretical Non-overlapping Allan variance for constant-mean non-stationary time series data.

Usage

```
NOAV(n, covmat)
```

Arguments

- `n` — An `integer` indicating the length of each vector of consecutive observations considered for the average.
- `covmat` — A `matrix` indicating the T-by-T covariance matrix of the time series with length T.

Details

This calculation of Non-overlapping Allan variance is based on the definition on "A Study of the Allan Variance for Constant-Mean Non-Stationary Processes" by Xu et al., 2017, IEEE Signal Processing Letters, 24(8): 1257–1260. Here n is an integer larger than 1 and smaller than $floor\left(log_2\left(dim\left(covmat\right)[1]\right)\right) - 1$.

Value

A `field` <`numeric`> that is the theoretical Non-overlapping Allan variance for constant-mean non-stationary time series data.

Author(s)

Haotian Xu

Examples

```
set.seed(999)
Xt = arima.sim(n = 100, list(ar = 0.3))
avar(Xt, type = "to")

a = matrix(rep(0, 1000^2), nrow = 1000)
for (i in 1:1000){
  a[,i] = seq(from = 1 - i, length.out = 1000)
}
a.diag = diag(a)
a[upper.tri(a,diag=TRUE)] = 0
a = a + t(a) + diag(a.diag)
covmat = 0.3^a
sapply(1:8, function(y){NOAV(2^y, covmat)})
```

`plot.avar` *Plot Allan Deviation*

Description

Displays a plot of Allan deviation with its corresponding pointwise confidence intervals.

Usage

```
## S3 method for class 'avar'
plot(
  x,
  units = NULL,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  col_ad = NULL,
  col_ci = NULL,
  nb_ticks_x = NULL,
  nb_ticks_y = NULL,
  legend_position = NULL,
  ci_ad = NULL,
  point_cex = NULL,
  point_pch = NULL,
  ...
)
```

Arguments

x	An avar object.
units	A string that specifies the units of time plotted on the x axis.
xlab	A string that gives a title for the x axis.
ylab	A string that gives a title for the y axis.
main	A string that gives an overall title for the plot.
col_ad	A string that specifies the color of the line allan deviation line.
col_ci	A string that specifies the color of the shaded area covered by the confidence intervals.
nb_ticks_x	An integer that specifies the maximum number of ticks for the x-axis.
nb_ticks_y	An integer that specifies the maximum number of ticks for the y-axis.
legend_position	A string that specifies the position of the legend (use legend_position = NA to remove legend).
ci_ad	A boolean that determines whether to plot the confidence interval shaded area.
point_cex	A double that specifies the size of each symbol to be plotted.
point_pch	A double that specifies the symbol type to be plotted.
...	Additional arguments affecting the plot.

Value

A plot of the Allan deviation and relative confidence interval for each scale.

Author(s)

Stephane Guerrier, Nathanael Claussen and Justin Lee

Examples

```
set.seed(999)
Xt = rnorm(10000)
av = avar(Xt)

plot(av)
plot(av, main = "Simulated white noise", xlab = "Scales")
plot(av, units = "sec", legend_position = "topright")
plot(av, col_ad = "darkred", col_ci = "pink")
```

`plot.avlr` *Plot the AVLR with the Allan Deviation*

Description

Displays a plot of the Allan deviation (AD) with the CI values and the AD implied by the estimated parameters.

Usage

```
## S3 method for class 'avlr'
plot(
  x,
  decomp = FALSE,
  units = NULL,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  col_ad = NULL,
  col_ci = NULL,
  nb_ticks_x = NULL,
  nb_ticks_y = NULL,
  legend_position = NULL,
  ci_ad = NULL,
  point_cex = NULL,
  point_pch = NULL,
  show_scales = FALSE,
  ...
)
```

Arguments

`x`	An `avlr` object.
`decomp`	A `boolean` that determines whether the contributions of each individual model are plotted.

units	A string that specifies the units of time plotted on the x axis.
xlab	A string that gives a title for the x axis.
ylab	A string that gives a title for the y axis.
main	A string that gives an overall title for the plot.
col_ad	A string that specifies the color of the line allan deviation line.
col_ci	A string that specifies the color of the shaded area covered by the confidence intervals.
nb_ticks_x	An integer that specifies the maximum number of ticks for the x-axis.
nb_ticks_y	An integer that specifies the maximum number of ticks for the y-axis.
legend_position	A string that specifies the position of the legend (use legend_position = NA to remove legend).
ci_ad	A boolean that determines whether to plot the confidence interval shaded area.
point_cex	A double that specifies the size of each symbol to be plotted.
point_pch	A double that specifies the symbol type to be plotted.
show_scales	A boolean that specifies if the scales used for each process should be plotted.
...	Additional arguments affecting the plot.

Value

Plot of Allan deviation and relative confidence intervals for each scale.

Author(s)

Stephane Guerrier and Justin Lee

Examples

```
set.seed(999)

N = 100000
Xt = rnorm(N) + cumsum(rnorm(N, 0, 3e-3))
av = avlr(Xt, wn = 1:7, rw = 12:15)

plot.avlr(av)
plot.avlr(av, decomp = TRUE, main = "Simulated white noise", xlab = "Scales")
plot.avlr(av, units = "sec", legend_position = "topright")
plot.avlr(av, col_ad = "darkred", col_ci = "pink")
plot(fit, decomp = TRUE, show_scales = TRUE)
```

`plot.imu_avar` *Plot Allan Deviation based on IMU Data*

Description

Displays a plot of Allan deviation based on IMU data with its corresponding pointwise confidence intervals.

Usage

```
## S3 method for class 'imu_avar'
plot(
  x,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  col_ad = NULL,
  col_ci = NULL,
  nb_ticks_x = NULL,
  nb_ticks_y = NULL,
  ci_ad = NULL,
  point_pch = NULL,
  point_cex = NULL,
  ...
)
```

Arguments

`x`	An `avar` object.
`xlab`	A `string` that gives a title for the x axis.
`ylab`	A `string` that gives a title for the y axis.
`main`	A `string` that gives an overall title for the plot.
`col_ad`	A `string` that specifies the color of the line allan deviation line.
`col_ci`	A `string` that specifies the color of the shaded area covered by the confidence intervals.
`nb_ticks_x`	An `integer` that specifies the maximum number of ticks for the x-axis.
`nb_ticks_y`	An `integer` that specifies the maximum number of ticks for the y-axis.
`ci_ad`	A `boolean` that determines whether to plot the confidence interval shaded area.
`point_pch`	A `double` that specifies the symbol type to be plotted.
`point_cex`	A `double` that specifies the size of each symbol to be plotted.
`...`	Additional arguments affecting the plot.

Value

A plot of the Allan deviation and relative confidence interval for each scale.

Author(s)

Stephane Guerrier and Yuming Zhang

Examples

```
data("navchip_av")
plot(navchip_av)
```

`plot.imu_avlr` *Plot the AVLR with the Allan Deviation for IMU*

Description

Displays a plot of the Allan deviation (AD) with the CI values and the AD implied by the estimated parameters for the IMU.

Usage

```
## S3 method for class 'imu_avlr'
plot(
  x,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  col_ad = NULL,
  col_ci = NULL,
  nb_ticks_x = NULL,
  nb_ticks_y = NULL,
  ci_ad = NULL,
  point_pch = NULL,
  point_cex = NULL,
  ...
)
```

Arguments

`x`	An `avlr` object.
`xlab`	A `string` that gives a title for the x axis.
`ylab`	A `string` that gives a title for the y axis.
`main`	A `string` that gives an overall title for the plot.
`col_ad`	A `string` that specifies the color of the line allan deviation line.
`col_ci`	A `string` that specifies the color of the shaded area covered by the confidence intervals.
`nb_ticks_x`	An `integer` that specifies the maximum number of ticks for the x-axis.
`nb_ticks_y`	An `integer` that specifies the maximum number of ticks for the y-axis.

ci_ad	A boolean that determines whether to plot the confidence interval shaded area.
point_pch	A double that specifies the symbol type to be plotted.
point_cex	A double that specifies the size of each symbol to be plotted.
...	Additional arguments affecting the plot.

Value

Plot of Allan deviation and relative confidence intervals for each scale.

Author(s)

Stephane Guerrier and Justin Lee

Examples

```
data(navchip_av)
navchip_avlr = avlr(navchip_av, wn_gyro = 1:20, rw_gyro = 1:20, wn_acc = 1:20, rw_acc = 1:20)
plot(navchip_avlr)
```

---

print.avar *Prints Allan Variance*

Description

Displays the information on the output of the ‘avar()‘ function

Usage

```
## S3 method for class 'avar'
print(x, ...)
```

Arguments

x	A avar object.
...	Arguments to be passed to methods

Value

console output

Examples

```
set.seed(999)
Xt = rnorm(10000)
out = avar(Xt)
print(out)
```

summary.avar *Summary Allan Variance*

Description

Displays the summary table of the output of the ‘avar()‘ function

Usage

```
## S3 method for class 'avar'
summary(object, ...)
```

Arguments

`object`	A `avar` object.
`...`	Additional arguments affecting the summary produced. A `table` that contains: • "Time": The averaging time at each level. • "AVar": The estimated Allan variance. • "ADev": The estimated Allan deviation. • "Lower CI": The lower bound of the confidence interval for the Allan deviation (ADev). • "Upper CI": The upper bound of the confidence interval for the Allan deviation (ADev).

Examples

```
set.seed(999)
Xt = rnorm(10000)
out = avar(Xt)
summary(out)
```

Index