Package: gmwmx2 (via r-universe)

June 10, 2026

Title Estimate Functional and Stochastic Parameters of Linear Models with Correlated Residuals and Missing Data

Version 0.0.5

Description Implements the Generalized Method of Wavelet Moments with Exogenous Inputs estimator (GMWMX) presented in Voirol, L., Xu, H., Zhang, Y., Insolia, L., Molinari, R. and Guerrier, S. (2024) <doi:10.48550/arXiv.2409.05160>. The GMWMX estimator allows to estimate functional and stochastic parameters of linear models with correlated residuals in presence of missing data. The 'gmwmx2' package provides functions to load and plot Global Navigation Satellite System (GNSS) data from the Nevada Geodetic Laboratory and functions to estimate linear model model with correlated residuals in presence of missing data.

License AGPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Language en-US

Depends R (>= 4.0.0)

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, data.table, dplyr, magrittr, wv, Matrix, httr2, rlang, longmemo

Suggests knitr, rmarkdown, knitcitations, raster, rnaturalearth, shape, tibble, tidygeocoder, sf, geodata

URL https://smac-group.github.io/gmwmx2/, https://github.com/SMAC-Group/gmwmx2

BugReports https://github.com/SMAC-Group/gmwmx2/issues

LazyData true

NeedsCompilation yes

Author Lionel Voirol [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1696-1407>), Haotian Xu [aut]
(ORCID: <https://orcid.org/0000-0001-8336-4536>), Yuming Zhang
[aut] (ORCID: <https://orcid.org/0000-0003-2293-9601>), Luca
Insolia [aut] (ORCID: <https://orcid.org/0000-0003-4169-5446>),
Roberto Molinari [aut] (ORCID:
<https://orcid.org/0000-0001-8766-6902>), Stéphane Guerrier
[aut] (ORCID: <https://orcid.org/0000-0003-0002-8794>)

Maintainer Lionel Voirol <lionelvoirol@hotmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-06-10 11:06:28 UTC

RemoteUrl https://github.com/cran/gmwmx2

RemoteRef HEAD

RemoteSha 3ffd11a242ed7e398f7c832ed006c78049c05cc8

Contents

+.sum_model *Add to a* sum_model *object*

Description

Add to a sum_model object

Usage

```
## S3 method for class 'sum_model'
e1 + e2
```

Arguments

e1	Left operand.
e2	Right operand.

Value

A sum_model.

Examples

```
m1 <- wn(sigma2 = 1)
m2 <- ar1(phi = 0.8, sigma2 = 0.5)
m3 <- pl(kappa = 0.3, sigma2 = 2)
model <- (m1 + m2) + m3
```

+.time_series_model *Add to a* time_series_model *object*

Description

Combines time_series_model and/or sum_model into a sum_model.

Usage

```
## S3 method for class 'time_series_model'
e1 + e2
```

Arguments

e1	Left operand.
e2	Right operand.

Value

A sum_model.

Examples

```
m1 <- wn(sigma2 = 1)
m2 <- ar1(phi = 0.8, sigma2 = 0.5)
model <- m1 + m2
model
```

---

ar1 *AR(1) process (*time_series_model*)*

Description

Constructs a time_series_model for a stationary AR(1) process with parameter phi and innovation variance sigma2. The model is $X_t = \phi X_{t-1} + \varepsilon_t,\ \varepsilon_t \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. The autocovariance is $\gamma(h) = \mathrm{cov}(X_t, X_{t+h}) = \frac{\sigma^2}{1-\phi^2}\,\phi^{|h|}$.

Usage

```
ar1(phi = NULL, sigma2 = NULL)
```

Arguments

phi	AR(1) coefficient in (-1, 1).
sigma2	Innovation variance (> 0).

Value

A time_series_model object.

Examples

```
mod <- ar1(phi = 0.8, sigma2 = 1)
mod
```

`df_estimated_velocities_gmwmx` *Estimated northward and eastward velocity and their standard deviation using the GMWMX estimator*

Description

Estimated northward and eastward velocity and standard deviation for a subset of 1202 GNSS station with more than 10 years of daily data.

Usage

```
df_estimated_velocities_gmwmx
```

Format

A data frame with 1202 rows and 12 variables:

station_name Name of the GNSS station.

estimated_trend_N Estimated northward velocity trend (in meters per day).

std_estimated_trend_N Standard deviation of the estimated northward velocity trend.

estimated_trend_E Estimated eastward velocity trend (in meters per day).

std_estimated_trend_E Standard deviation of the estimated eastward velocity trend.

length_signal Length of the signal (in days).

estimated_trend_N_scaled Scaled estimated northward velocity trend (multiplying by 365.25 for yearly values).

std_estimated_trend_N_scaled Scaled standard deviation of the estimated northward velocity trend.

estimated_trend_E_scaled Scaled estimated eastward velocity trend (multiplying by 365.25 for yearly values).

std_estimated_trend_E_scaled Scaled standard deviation of the estimated eastward velocity trend.

latitude Latitude of the GNSS station.

longitude Longitude of the GNSS station.

`download_all_stations_ngl`

Download all stations name and location from the Nevada Geodetic Laboratory

---

Description

Download all stations name and location from the Nevada Geodetic Laboratory

Usage

```
download_all_stations_ngl(verbose = FALSE)
```

Arguments

`verbose`	A `boolean` that controls the level of detail in the output of the `wget` command used to load data. Default is `FALSE`.

Value

Return a `data.frame` with all stations name, latitude, longitude and heights.

Examples

```
df_all_stations <- download_all_stations_ngl()
head(df_all_stations)
```

`download_estimated_velocities_ngl`

Download estimated velocities using the MIDAS estimator provided by the Nevada Geodetic Laboratory for all stations.

Description

Download estimated velocities using the MIDAS estimator provided by the Nevada Geodetic Laboratory for all stations.

Usage

```
download_estimated_velocities_ngl(verbose = FALSE)
```

Arguments

`verbose`	A `boolean` that controls the level of detail in the output of the `wget` command used to load data. Default is `FALSE`.

Value

Return a data.frame with all stations name, information about the time series for each station, estimated velocities and estimated standard deviation of the estimated velocities.

Examples

```
df_estimated_velocities <- download_estimated_velocities_ngl()
head(df_estimated_velocities)
```

---

download_station_ngl *Download GNSS position time series and steps reference from the Nevada Geodetic Laboratory with IGS14 or IGS20 reference frame.*

Description

Download GNSS position time series and steps reference from the Nevada Geodetic Laboratory with IGS14 or IGS20 reference frame.

Usage

```
download_station_ngl(station_name, verbose = FALSE, reference_frame = "IGS20")
```

Arguments

station_name	A string specifying the station name.
verbose	A boolean that controls the level of detail in the output of the wget command used to load data. Default is FALSE.
reference_frame	A string with value either "IGS14" or "IGS20" that specify which reference frame to use. Default is "IGS20".

Value

A list of class gnss_ts_ngl that contains three data.frame: The data.frame df_position which contains the position time series extracted from the .tenv3 file available from the Nevada Geodetic Laboratory, the data.frame df_equipment_software_changes which specify the equipment or software changes for that stations and the data.frame df_earthquakes that specify the earthquakes associated with that station.

Examples

```
station_1LSU <- download_station_ngl("1LSU")
attributes(station_1LSU)
```

`flicker` *Flicker noise process (*`time_series_model`*)*

Description

Constructs a `time_series_model` for flicker noise with variance `sigma2`. The process has spectral density $S(f) \propto \frac{1}{|f|}$. Hence, $\kappa = -1$ (Bos et al., 2008). The process is non-stationary and its covariance matrix is assumed to be given by

$$\mathbf{C} = \sigma^2 \mathbf{U}^\top \mathbf{U},$$

where $\mathbf{U} \in \mathbb{R}^{N \times N}$ is an upper-triangular Toeplitz matrix with entries

$$U_{i,j} = \begin{cases} h_{j-i}, & j \geq i, \\ 0, & j < i, \end{cases} \qquad i, j = 1, \ldots, N.$$

The coefficients $\{h_i\}_{i \geq 0}$ define a causal linear filter and are given recursively by

$$h_0 = 1, \qquad h_i = \left(i - \frac{\kappa}{2} - 1\right) \frac{h_{i-1}}{i}, \quad i > 0.$$

Usage

```
flicker(sigma2 = NULL)
```

Arguments

`sigma2`	Innovation variance (> 0).

Value

A `time_series_model` object.

References

Bos MS, Fernandes RMS, Williams SDP, Bastos L (2008). "Fast error analysis of continuous GPS observations." *Journal of Geodesy*, 82, 157-166.

Examples

```
mod <- flicker(sigma2 = 1)
mod
```

generate — *Generate a time series from a* time_series_model *or* sum_model *object*

Description

Generate a time series from a time_series_model or sum_model object

Usage

```
generate(object, n, seed = NULL, ...)
```

Arguments

object	A time_series_model or sum_model.
n	Length of series to generate.
seed	Optional integer seed for reproducibility.
...	Passed to method.

Value

A generated_time_series (single model) or generated_composite_model_time_series (sum model).

Examples

```
# Single model
m1 <- ar1(phi = 0.8, sigma2 = 1)
y1 <- generate(m1, n = 200, seed = 123)
plot(y1)

# Composite model
m2 <- wn(sigma2 = 1) + pl(kappa = 0.3, sigma2 = 2)
y2 <- generate(m2, n = 200, seed = 123)
plot(y2)
```

gmwm2 — *GMWM estimator*

Description

Implements the Generalized Method of Wavelet Moments (GMWM) estimator to fit a time_series_model, a sum_model or a numeric vector.

Usage

```
gmwm2(x, model, omega = NULL, method = "L-BFGS-B", control = list(), ...)
```

Arguments

x	Numeric vector, or a generated_time_series / generated_composite_model_time_series object (its series is used).
model	A time_series_model or sum_model.
omega	Optional weighting matrix. If NULL, a default based on the empirical WV confidence intervals is used.
method	Optimization method passed to stats::optim.
control	Optional list of control parameters for stats::optim.
...	Additional arguments passed to stats::optim.

Details

The GMWM estimator solves a weighted least-squares criterion of the form

$$\{\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}(\boldsymbol{\theta})\}^{\top} \boldsymbol{\Omega} \{\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}(\boldsymbol{\theta})\}$$

where $\hat{\boldsymbol{\nu}}$ denotes the empirical wavelet variance and $\boldsymbol{\nu}(\boldsymbol{\theta})$ the corresponding theoretical wavelet variance implied by the model parameters $\boldsymbol{\theta}$. The weighting matrix $\boldsymbol{\Omega}$ defaults to a diagonal matrix with entries proportional to the inverse squared width of the empirical WV asymptotic confidence intervals. Provide omega to use a custom weighting (e.g., from a theoretical covariance).

Value

An object of class gmwm2_fit with elements: theta_hat (real space), theta_domain (constrained space), model, empirical_wvar, theoretical_wvar, optim, and n.

References

Guerrier, S., Skaloud, J., Stebler, Y., and Victoria-Feser, M.-P. (2013). Wavelet-variance-based estimation for composite stochastic processes. *Journal of the American Statistical Association*, 108(503), 1021-1030. doi:10.1080/01621459.2013.799920.

Examples

```
n = 10000
mod = wn(20) + ar1(phi = .995, sigma2 = .2)
y = generate(mod, n = n, seed = 123)
plot(y)
fit = gmwm2(y, model = wn() + ar1())
fit
plot(fit)
```

gmwmx2 *GMWMX estimator*

Description

Dispatches either to the generic regression interface (design matrix + response) or to a gnss_ts_ngl workflow.

Convenience wrapper that selects the missing or non-missing implementation based on the presence of NA values in y.

Usage

```
gmwmx2(X, ...)

## Default S3 method:
gmwmx2(X, y, model, omega = NULL, method = "L-BFGS-B", control = list(), ...)

## S3 method for class 'gnss_ts_ngl'
gmwmx2(
  X,
  n_seasonal = 2,
  vec_earthquakes_relaxation_time = NULL,
  component = NULL,
  model = NULL,
  omega = NULL,
  method = "L-BFGS-B",
  control = list(),
  ...
)
```

Arguments

X	A gnss_ts_ngl object (GNSS time-series interface).
...	Reserved for future extensions.
y	Response vector for a generic regression interface.
model	Stochastic model specification.
omega	Optional weighting matrix. If NULL, uses inverse CI width.
method	Optimization method passed to stats::optim.
control	Control list passed to stats::optim.
n_seasonal	Number of seasonal signals.
vec_earthquakes_relaxation_time	Relaxation time for each earthquake.
component	Component to estimate ("N", "E", or "V").

Value

A fitted model object.

A fitted model object.

A fitted model object.

---

markov_two_states *Markov two-state missingness model (*missingness_model*)*

Description

Constructs a missingness_model representing a two-state Markov process for missing/observed indicators. The process takes values in $\{0, 1\}$, where 1 indicates observed and 0 indicates missing.

Usage

```
markov_two_states(p1 = NULL, p2 = NULL)
```

Arguments

p1	Transition probability from observed (1) to missing (0).
p2	Transition probability from missing (0) to observed (1).

Value

A missingness_model object.

Examples

```
mod <- markov_two_states(p1 = 0.05, p2 = 0.95)
mod
z <- generate(mod, n = 200, seed = 123)
plot(z)
```

---

matern *Matern process (*time_series_model*)*

Description

Constructs a time_series_model for a Matern covariance process with variance sigma2, range lambda, and smoothness alpha. The autocovariance is $\gamma(h) = \mathrm{cov}(X_t, X_{t+h}) = \frac{2\sigma^2}{\Gamma(\alpha-1/2)2^{\alpha-1/2}}|\lambda h|^{\alpha-1/2}\mathcal{K}_{|\alpha-1/2|}(|\lambda h|)$ where $\mathcal{K}_\omega(x)$ is the modified Bessel function of the second kind of order ω.

Usage

```
matern(sigma2 = NULL, lambda = NULL, alpha = NULL)
```

Arguments

sigma2	Marginal variance (> 0).
lambda	Range/scale parameter (> 0).
alpha	Smoothness parameter in (1/2, 10).

Value

A time_series_model object.

References

Lilly JM, Sykulski AM, Early JJ, Olhede SC (2017). "Fractional Brownian motion, the Matérn process, and stochastic modeling of turbulent dispersion." *Nonlinear Processes in Geophysics*, 24(3), 481-514.

Examples

```
mod <- matern(sigma2 = 1, lambda = 0.2, alpha = 1.0)
mod
```

---

pl *Stationary Power-Law process (*time_series_model*)*

Description

Constructs a time_series_model representing a stationary power-law process with parameters kappa and sigma2. In the frequency domain, a power-law process is often described by a spectrum $P(f) = P_0 f^{\kappa}$ (Bos et al., 2008), where f is the frequency, P_0 is a constant and κ is the spectral index. Note that we use the convention that the power spectral density satisfies $P(f) \propto |f|^{\kappa}$, where $\kappa > -1$ ensures second-order stationarity. This corresponds to the alternative notation $P(f) \propto |f|^{-\alpha}$ with $\alpha = -\kappa$. The autocovariance $\gamma(h) = \mathrm{cov}(X_t, X_{t+h})$ used here (Hosking, 1981) is $\gamma(0) = \sigma^2 \frac{\Gamma(1+\kappa)}{\Gamma(1+\kappa/2)^2}$, and for $h > 0$ $\gamma(h) = \frac{-\kappa/2+h-1}{\kappa/2+h}\,\gamma(h-1)$.

Usage

```
pl(kappa = NULL, sigma2 = NULL)
```

Arguments

kappa	Power-law parameter in (-1, 1).
sigma2	Process variance (> 0).

Value

A time_series_model object.

References

Bos MS, Fernandes RMS, Williams SDP, Bastos L (2008). "Fast error analysis of continuous GPS observations." *Journal of Geodesy*, 82, 157-166.

Hosking JRM (1981). "Fractional differencing." *Biometrika*, 68(1), 165-176.

Examples

```
mod <- pl(kappa = -0.5, sigma2 = 2)
mod
```

---

plot.generated_composite_model_time_series

Plot a `generated_composite_model_time_series` *object*

Description

Produces stacked line plots for each component and the sum for a `generated_composite_model_time_series` object.

Usage

```
## S3 method for class 'generated_composite_model_time_series'
plot(x, ...)
```

Arguments

`x`	A `generated_composite_model_time_series`.
`...`	Additional arguments passed to `plot()`.

Value

Invisibly returns `x`.

Examples

```
m2 <- wn(sigma2 = 1) + ar1(phi = 0.8, sigma2 = 0.5)
y2 <- generate(m2, n = 200, seed = 123)
plot(y2)
```

plot.generated_missingness *Plot a* generated_missingness *object*

Description

Produces a step plot for a generated_missingness object.

Usage

```
## S3 method for class 'generated_missingness'
plot(x, ...)
```

Arguments

x	A generated_missingness.
...	Additional arguments passed to plot().

Value

Invisibly returns x.

Examples

```
m0 <- markov_two_states(p1 = 0.05, p2 = 0.9)
z0 <- generate(m0, n = 200, seed = 123)
plot(z0)
```

plot.generated_time_series *Plot a* generated_time_series *object*

Description

Produces a single line plot for a generated_time_series object.

Usage

```
## S3 method for class 'generated_time_series'
plot(x, ...)
```

Arguments

x	A generated_time_series.
...	Additional arguments passed to plot().

Value

Invisibly returns `x`.

Examples

```
m1 <- wn(sigma2 = 1)
y1 <- generate(m1, n = 200, seed = 123)
plot(y1)
```

---

plot.gmwm2_fit *Plot method for a* gmwm2_fit *object*

Description

Plots empirical wavelet variance with the fitted theoretical curve and, for sum models, component-implied theoretical curves.

Usage

```
## S3 method for class 'gmwm2_fit'
plot(
  x,
  show_ci = TRUE,
  col_emp = "black",
  col_theo = "darkorange",
  col_ci = "#e6f7fb",
  lwd = 2,
  pch_emp = 16,
  pch_theo = 21,
  cex_theo = 1.4,
  legend_pos = "auto",
  ...
)
```

Arguments

`x`	A `gmwm2_fit` object.
`show_ci`	Logical; if TRUE and available, show empirical CI bars.
`col_emp`	Color for empirical WV points/line.
`col_theo`	Color for theoretical WV line.
`col_ci`	Color for empirical WV CI band.
`lwd`	Line width for theoretical curve.
`pch_emp`	Plotting character for empirical points.
`pch_theo`	Plotting character for theoretical points.
`cex_theo`	Size for theoretical points.
`legend_pos`	Legend position (e.g., "topleft") or "auto".
`...`	Additional arguments passed to `plot()`.

Value

The input object, invisibly.

Examples

```
n = 10000
mod = wn(20) + ar1(phi = .995, sigma2 = .2)
y = generate(mod, n = n, seed = 123)
plot(y)
fit = gmwm2(y, model = wn() + ar1() )
fit
plot(fit)
```

---

plot.gmwmx2_fit_gnss_ts_ngl *Plot a* gmwmx2_fit_gnss_ts_ngl *object*

Description

Plot a gmwmx2_fit_gnss_ts_ngl object

Usage

```
## S3 method for class 'gmwmx2_fit_gnss_ts_ngl'
plot(x, ...)
```

Arguments

x	A gmwmx2_fit_gnss_ts_ngl object.
...	Additional graphical parameters.

Value

No return value. Plot a gmwmx2_fit_gnss_ts_ngl object.

Examples

```
# station_data = gmwmx2::download_station_ngl("1LSU")
# fit station with WN and PL
# fit1 <- gmwmx2(
#   station_data,
#   n_seasonal = 2,
#   model = wn() + pl(), component = "N"
# )
# fit1
# plot(fit1)
```

plot.gnss_ts_ngl *Plot a* gnss_ts_ngl *object*

Description

Plot a gnss_ts_ngl object

Usage

```
## S3 method for class 'gnss_ts_ngl'
plot(x, component = NULL, ...)
```

Arguments

x	A gnss_ts_ngl object.
component	A string with value either "N", "E" or "V" that specify which component to plot (Northing, Easting or Vertical).
...	Additional graphical parameters.

Value

No return value. Plot a gnss_ts_ngl object.

Examples

```
station_1LSU <- download_station_ngl("1LSU")
plot(station_1LSU)
plot(station_1LSU, component = "N")
plot(station_1LSU, component = "E")
plot(station_1LSU, component = "V")
```

print.gmwm2_fit *Print method for a* gmwm2_fit *object*

Description

Print method for a gmwm2_fit object

Usage

```
## S3 method for class 'gmwm2_fit'
print(x, digits = 4, show_initial_parameters = FALSE, ...)
```

Arguments

x	A gmwm2_fit object.
digits	Significant digits for printing.
show_initial_parameters	Logical; if TRUE, also show the initial parameters used for optimization.
...	Unused.

Value

The input object, invisibly.

Examples

```
n = 10000
mod = wn(20) + ar1(phi = .995, sigma2 = .2)
y = generate(mod, n = n, seed = 123)
plot(y)
fit = gmwm2(y, model = wn() + ar1() )
fit
```

---

print.gmwmx2_fit *Print method for a* gmwmx2_fit *object*

Description

Displays a table of regression coefficients with standard errors and summarizes the fitted stochastic model with estimated parameters.

Usage

```
## S3 method for class 'gmwmx2_fit'
print(x, digits = 4, ...)
```

Arguments

x	A gmwmx2_fit object.
digits	Significant digits to display.
...	Passed to print methods.

Value

The input object, invisibly.

print.gmwmx2_fit_gnss_ts_ngl *Print method for a* gmwmx2_fit_gnss_ts_ngl *object*

Description

Displays regression coefficients with standard errors and confidence intervals, along with the fitted stochastic and missingness models.

Usage

```
## S3 method for class 'gmwmx2_fit_gnss_ts_ngl'
print(x, digits = 4, ...)
```

Arguments

x	A gmwmx2_fit_gnss_ts_ngl object.
digits	Significant digits to display.
...	Passed to print methods.

Value

The input object, invisibly.

rw *Random walk process (*time_series_model*)*

Description

Constructs a time_series_model for a random walk with innovation variance sigma2. The auto-covariance returned is the mean of the diagonal and super-diagonals of the covariance matrix. The model is $X_t = X_{t-1} + \varepsilon_t,\ \varepsilon_t \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$.

Usage

```
rw(sigma2 = NULL)
```

Arguments

sigma2	Innovation variance (> 0).

Value

A time_series_model object.

Examples

```
mod <- rw(sigma2 = 1)
mod
```

---

wn *White noise process (*time_series_model*)*

Description

Constructs a time_series_model for white noise with variance sigma2. The process is defined as $X_t \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$ with autocovariance $\gamma(h) = \text{cov}(X_t, X_{t+h}) = \sigma^2 \mathbf{1}\{h = 0\}$

Usage

```
wn(sigma2 = NULL)
```

Arguments

sigma2	Innovation variance (> 0).

Value

A time_series_model object.

Examples

```
mod <- wn(sigma2 = 1)
mod
```

Index