

Package ‘Rdrw’

July 2, 2026

Version 1.0.3

Date 2026-07-02

Title Univariate and Multivariate Damped Random Walk Processes

Author Zhirui Hu [aut],
Hyungsuk Tak [aut, cre]

Maintainer Hyungsuk Tak <hyungsuk.tak@gmail.com>

Depends R (>= 3.5.0)

Imports MASS

Description

Provides tools for fitting and simulating univariate and multivariate damped random walk processes, also known as Ornstein-Uhlenbeck processes or first-order continuous-time autoregressive models, CAR(1) or CARMA(1, 0). The package supports irregularly spaced observation times, heteroscedastic measurement errors, missing measurements across multivariate time series, and polynomial mean trends in normalized time. The current implementation models up to ten time series jointly. Kalman filtering is used to evaluate the likelihood efficiently for maximum likelihood estimation and Bayesian posterior sampling. Users should preserve sufficient numerical precision when loading astronomical observation times; see the manual for details. Also see Hu and Tak (2020) <[doi:10.48550/arXiv.2005.08049](https://doi.org/10.48550/arXiv.2005.08049)>.

License GPL-2

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2026-07-02 05:40:02 UTC

Contents

drw	2
drw.sim	4
Rdrw	6

Index	8
--------------	----------

Description

The function `drw` fits univariate and multivariate damped random walk processes to one or more irregularly spaced time series with possibly heteroscedastic measurement errors. The current implementation allows each time series to have a polynomial mean trend in normalized time. This feature is useful in applications such as stock price prediction, where a constant mean may be too restrictive. The likelihood is evaluated by Kalman filtering, and the function returns maximum likelihood estimates or posterior samples of the model parameters.

Usage

```
drw(data1, data2 = NULL, data3 = NULL, data4 = NULL, data5 = NULL,
    data6 = NULL, data7 = NULL, data8 = NULL, data9 = NULL,
    data10 = NULL, n.datasets, method = "mle", poly.order = 1,
    bayes.n.burn = 3000, bayes.n.sample = 1000)
```

Arguments

<code>data1</code>	An <code>n_1</code> by 3 matrix or data frame for the first time series. The first column contains observation times, the second column contains measurements, and the third column contains measurement error standard deviations. If measurement error standard deviations are unavailable, use zeros in the third column.
<code>data2</code>	Optional second data set in the same format as <code>data1</code> .
<code>data3</code>	Optional third data set in the same format as <code>data1</code> .
<code>data4</code>	Optional fourth data set in the same format as <code>data1</code> .
<code>data5</code>	Optional fifth data set in the same format as <code>data1</code> .
<code>data6</code>	Optional sixth data set in the same format as <code>data1</code> .
<code>data7</code>	Optional seventh data set in the same format as <code>data1</code> .
<code>data8</code>	Optional eighth data set in the same format as <code>data1</code> .
<code>data9</code>	Optional ninth data set in the same format as <code>data1</code> .
<code>data10</code>	Optional tenth data set in the same format as <code>data1</code> . The current version allows up to ten time series data sets.
<code>n.datasets</code>	The number of time series data sets to be modeled jointly. This must be an integer from 1 to 10.
<code>method</code>	Either "mle" for maximum likelihood estimation or "bayes" for posterior sampling.
<code>poly.order</code>	The order of the polynomial trend used for the mean function of each time series. The default is 1, corresponding to an intercept and a linear trend in normalized time. Use 0 for the constant-mean damped random walk model.
<code>bayes.n.burn</code>	The number of warm-up iterations when <code>method = "bayes"</code> .
<code>bayes.n.sample</code>	The number of posterior samples saved after warm-up when <code>method = "bayes"</code> .

Details

The multivariate damped random walk process is also known as the Ornstein–Uhlenbeck process or a CARMA(1, 0) process. In the updated model, the long-term mean of the j -th time series is represented as

$$m_j(u) = \text{beta_j0} + \text{beta_j1} * u + \dots + \text{beta_jp} * u^p,$$

where `p` is `poly.order`. The time variable u is the internally normalized observation time,

$$u = (t - t_{\min}) / (t_{\max} - t_{\min}),$$

where $t_{\min}$ and $t_{\max}$ are the minimum and maximum combined observation times, respectively. Setting `poly.order = 0` recovers the constant-mean specification in the original package.

Because the likelihood is evaluated on normalized time, the estimated `tau` values are also on the normalized-time scale. To convert them back to the original time scale, multiply by `time.range = max(time) - min(time)`. For example, if the combined observation times span 2,000 days and `tau = 0.05`, then the corresponding timescale is $0.05 * 2000 = 100$ days. The returned object includes this converted value as `tau.days`.

The polynomial coefficients in `beta` should be interpreted on the same normalized-time scale. For example, with `poly.order = 1`, the fitted mean is $\text{beta_j0} + \text{beta_j1} * u$, so `beta_j1` is the total linear change over the normalized interval $[0, 1]$, not the daily slope.

For multivariate fitting, the individual data sets are merged by observation time. Missing measurements at a given time are allowed and are handled by the Kalman filter. The input measurement-error column should contain standard deviations, not variances.

When astronomical time series data are loaded into R by `read.table`, `read.csv`, or related functions, decimal places of observation times may be rounded because of R's display and input defaults. If high time precision is needed, consider setting `options(digits = 11)` before loading the data.

Value

The returned object is a list. For `method = "mle"`, it contains:

beta Maximum likelihood estimates of the polynomial trend coefficients. The coefficients are returned as a vector ordered by time series. For each time series, the coefficients correspond to powers 0 through `poly.order` of normalized time, $u = (t - t_{\min}) / (t_{\max} - t_{\min})$. Thus, `beta_j0` is the fitted mean at the first normalized time point, and higher-order coefficients describe changes over the unit interval of normalized time, not per day.

sigma2 Maximum likelihood estimates of the short-term variance parameters.

sigma Maximum likelihood estimates of the short-term standard deviation parameters.

tau Maximum likelihood estimates of the timescale parameters on the normalized-time scale.

tau.days Maximum likelihood estimates of the timescale parameters on the original time scale. If the input observation times are measured in days, then `tau.days` is the estimated timescale in days. It is computed as `tau * time.range`.

rho Maximum likelihood estimates of the cross-correlation parameters when `n.datasets > 1`; otherwise `numeric(0)`.

time.range The range of the combined observation times, `max(time) - min(time)`.

time.min The minimum combined observation time.

data.comb The merged data matrix used for fitting.

For method = "bayes", beta, sigma, tau, tau.days, and rho contain posterior samples, with one row per saved MCMC iteration. The output also includes sigma.accept.rate, tau.accept.rate, and rho.accept.rate.

Author(s)

Zhirui Hu and Hyungsuk Tak

References

Zhirui Hu and Hyungsuk Tak (2020), "Modeling Stochastic Variability in Multi-Band Time Series Data," doi:10.48550/arXiv.2005.08049.

Examples

```
set.seed(1)
n1 <- 20
obs.time1 <- cumsum(rgamma(n1, shape = 3, rate = 1))
y1 <- rnorm(n1)
measure.error.SD1 <- rgamma(n1, shape = 0.01)
data1 <- cbind(obs.time1, y1, measure.error.SD1)

res1.mle <- drw(data1 = data1, n.datasets = 1, method = "mle", poly.order = 1)
names(res1.mle)
res1.mle$beta
res1.mle$sigma
res1.mle$tau
res1.mle$tau.days

res1.bayes <- drw(data1 = data1, n.datasets = 1, method = "bayes",
  poly.order = 1, bayes.n.burn = 10, bayes.n.sample = 10)

n2 <- 15
obs.time2 <- cumsum(rgamma(n2, shape = 3, rate = 1))
y2 <- rnorm(n2)
measure.error.SD2 <- rgamma(n2, shape = 0.01)
data2 <- cbind(obs.time2, y2, measure.error.SD2)

res2.mle <- drw(data1 = data1, data2 = data2, n.datasets = 2,
  method = "mle", poly.order = 1)
```

Description

The function `drw.sim` simulates one or more time series from univariate or multivariate damped random walk processes.

Usage

```
drw.sim(time, n.datasets, measure.error.SD, mu, sigma, tau, rho)
```

Arguments

<code>time</code>	A vector of observation times. Let n denote the length of this vector.
<code>n.datasets</code>	A positive integer giving the number of time series to be simulated. In simulation, there is no upper limit on the number of time series. Let k denote this number.
<code>measure.error.SD</code>	Optional measurement error standard deviations. If one time series is simulated, this is a vector of length n . If more than one time series is simulated, this is an n by k matrix of measurement error standard deviations. If this information is unavailable, it is automatically set to zero.
<code>mu</code>	A vector of length k containing the long-term mean parameter or parameters.
<code>sigma</code>	A vector of length k containing the short-term variability parameter or parameters, expressed as standard deviations.
<code>tau</code>	A vector of length k containing the timescale parameter or parameters.
<code>rho</code>	Required when more than one time series is simulated, $k > 1$. This is a vector of length $k(k - 1) / 2$ containing the cross-correlation parameters. For example, if $k = 3$, this vector contains <code>rho_12</code> , <code>rho_13</code> , and <code>rho_23</code> . If $k = 5$, this vector contains <code>rho_12</code> , <code>rho_13</code> , <code>rho_14</code> , <code>rho_15</code> , <code>rho_23</code> , <code>rho_24</code> , <code>rho_25</code> , <code>rho_34</code> , <code>rho_35</code> , and <code>rho_45</code> .

Details

Given observation times and model parameter values `mu`, `sigma`, `tau`, and optionally `rho`, this function simulates k time series, optionally including known measurement error standard deviations.

Value

The output of `drw.sim` is a list containing:

x An n by k matrix of simulated time series. Each column corresponds to one simulated time series.

Author(s)

Zhirui Hu and Hyungsuk Tak

References

Zhirui Hu and Hyungsuk Tak (2020), "Modeling Stochastic Variability in Multi-Band Time Series Data," doi:10.48550/arXiv.2005.08049.

Examples

```
##### Simulating a multivariate damped random walk process

n <- 100
k <- 5
obs.time <- cumsum(rgamma(n, shape = 3, rate = 1))

tau <- 100 + 20 * (1:5)
sigma <- 0.01 * (1:5)
mu <- 17 + 0.5 * (1:5)

rho.m <- matrix(0, k, k)
for (i in 1:k) {
  for (j in 1:k) {
    rho.m[i, j] <- 1.1^(-abs(i - j))
  }
}

rho <- rho.m[upper.tri(rho.m)]

measure.error.band <- c(0.010, 0.014, 0.018, 0.022, 0.026)
measure.error <- NULL
for (i in 1:k) {
  measure.error <- cbind(measure.error, rnorm(n, measure.error.band[i], 0.002))
}

x <- drw.sim(time = obs.time, n.datasets = 5, measure.error.SD = measure.error,
             mu = mu, sigma = sigma, tau = tau, rho = rho)

plot(obs.time, x[, 1], xlim = c(min(obs.time), max(obs.time)), ylim = c(17, 20),
     xlab = "time", ylab = "observation")
points(obs.time, x[, 2], col = 2, pch = 2)
points(obs.time, x[, 3], col = 3, pch = 3)
points(obs.time, x[, 4], col = 4, pch = 4)
points(obs.time, x[, 5], col = 5, pch = 5)

##### Simulating a univariate damped random walk process

x <- drw.sim(time = obs.time, n.datasets = 1, measure.error.SD = measure.error[, 1],
             mu = mu[1], sigma = sigma[1], tau = tau[1])
plot(obs.time, x)
```

Description

The **Rdrw** package provides tools for fitting and simulating univariate and multivariate damped random walk processes with optional measurement error standard deviations in a state-space framework. The damped random walk process is also known as an Ornstein–Uhlenbeck process or a first-order continuous-time autoregressive model, CAR(1) or CARMA(1, 0). The package uses Kalman

filtering to evaluate the likelihood efficiently, with linear complexity in the number of unique observation times. The main function `drw` fits the model and returns maximum likelihood estimates or posterior samples of the model parameters. The function `drw.sim` simulates time series from univariate or multivariate damped random walk processes.

Details

Package:	Rdrw
Type:	Package
Version:	1.0.6
Date:	2026-07-01
License:	GPL-2
Main functions:	<code>drw</code> , <code>drw.sim</code>

Author(s)

Zhirui Hu and Hyungsuk Tak

References

Zhirui Hu and Hyungsuk Tak (2020), "Modeling Stochastic Variability in Multi-Band Time Series Data," doi:10.48550/arXiv.2005.08049.

Index

`drw`, [2](#), [7](#)

`drw.sim`, [4](#), [7](#)

`Rdrw`, [6](#)

`Rdrw-package (Rdrw)`, [6](#)