Package 'weatherMRJD'

August 20, 2026

Title Weather Analysis and Markov Regime Switching Jump Diffusion Models

Version 0.1.1

Description Provides statistical tools for analyzing weather patterns, temperature anomalies, and climate risk. Implements Markov regime-switching jump diffusion (MRJD) models to capture abrupt shifts, extreme weather events, and structural breaks in environmental time series data. Estimates model parameters using maximum likelihood estimation and offers utility functions for simulating regime-dependent stochastic processes. The regime-switching methodology is based on Hamilton (1989) ``Analysis of Time Series Subject to Changes in Regime" <doi:10.2307/1912559>.

License MIT + file LICENSE

Encoding UTF-8

Imports stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Okechukwu J. Obulezi [aut, cre] (ORCID: <https://orcid.org/0000-0002-7753-1868>)

Maintainer Okechukwu J. Obulezi <oj.obulezi@unizik.edu.ng>

Repository CRAN

Date/Publication 2026-08-20 15:20:02 UTC

Contents

calibrate_mrjd *Calibrate Dynamic SDE and Jump Parameters*

Description

Estimates mean-reversion speed, unconditional volatility, and jump characteristics.

Usage

```
calibrate_mrjd(residuals, n_years, threshold_sigma = 2.5)
```

Arguments

residuals	Numeric vector of detrended temperature residuals X(t).
n_years	Numeric scalar representing total observation period in years.
threshold_sigma	Threshold multiplier for jump identification (default: 2.5).

Value

A named list of calibrated stochastic differential parameters.

Examples

```
set.seed(123)
res <- stats::rnorm(730, sd = 2)
calib <- calibrate_mrjd(res, n_years = 2)
print(calib$kappa)
```

fit_seasonal_trend *Fit Fourier Seasonal Baseline Trend*

Description

Fits a truncated Fourier series to daily atmospheric temperature observations.

Usage

```
fit_seasonal_trend(temp, day_of_year)
```

Arguments

temp	Numeric vector of observed daily average temperatures.
day_of_year	Numeric vector of day-of-year indices (1 to 365/366).

Value

A list containing the fitted OLS model, fitted values, and isolated residuals.

Examples

```
doy <- rep(1:365, times = 2)
temp <- 15 + 10 * sin(2 * pi * doy / 365) + stats::rnorm(730, sd = 2)
fit <- fit_seasonal_trend(temp, doy)
print(fit$rmse)
```

---

price_weather_option *Price HDD and CDD Weather Derivatives*

Description

Evaluates fair values and Monte Carlo standard errors for HDD and CDD Call Options.

Usage

```
price_weather_option(
  temp_paths,
  strike,
  type = c("HDD", "CDD"),
  r = 0.04,
  base_temp = 18
)
```

Arguments

temp_paths	Matrix of simulated temperature paths from simulate_weather_paths.
strike	Strike index level K.
type	Option contract type: "HDD" or "CDD".
r	Risk-free discount rate (default 0.04).
base_temp	Threshold index baseline (default 18.0 deg C).

Value

A list containing estimated price, standard error, and index distribution metrics.

Examples

```
paths <- simulate_weather_paths(n_paths = 100, days = 90)
hdd_opt <- price_weather_option(paths, strike = 500, type = "HDD")
print(hdd_opt$price)
```

`simulate_weather_paths` *Simulate Temperature Paths via Non-Homogeneous Jump-Diffusion*

Description

Generates discretized Monte Carlo sample trajectories of daily average temperatures.

Usage

```
simulate_weather_paths(
  n_paths = 5000,
  days = 365,
  kappa = 0.08,
  sigma_0 = 2.5,
  lambda_0 = 0.05,
  lambda_1 = 0.03,
  mu_jump = -3,
  sigma_jump = 1.5,
  baseline_temp = 14.5,
  amplitude_temp = 11.2
)
```

Arguments

`n_paths`	Integer count of simulation paths.
`days`	Horizon length in days (default 365).
`kappa`	Mean-reversion speed.
`sigma_0`	Baseline volatility.
`lambda_0`	Baseline jump intensity.
`lambda_1`	Amplitude of dynamic seasonal jump intensity.
`mu_jump`	Mean jump magnitude.
`sigma_jump`	Volatility of jump magnitude.
`baseline_temp`	Baseline temperature parameter A.
`amplitude_temp`	Seasonal temperature amplitude.

Value

An (n_paths x days) matrix of simulated daily temperatures.

Examples

```
paths <- simulate_weather_paths(n_paths = 50, days = 30)
dim(paths)
```

Index