

# Package ‘CDSim’

May 7, 2026

**Type** Package

**Title** Simulating Climate Data for Research and Modelling

**Version** 0.1.2

**Date** 2026-04-19

**Maintainer** Isaac Osei <ikemillar65@gmail.com>

**Description** Generate synthetic station-based monthly climate time-series including temperature and rainfall, export to Network Common Data Form (NetCDF), and provide visualization helpers for climate workflows. The approach is inspired by statistical weather generator concepts described in Wilks (1999) <[doi:10.1016/S0168-1923\(99\)00037-4](https://doi.org/10.1016/S0168-1923(99)00037-4)> and Richardson (1981) <[doi:10.1029/WR017i001p00182](https://doi.org/10.1029/WR017i001p00182)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** trend, truncnorm, ncd4, lubridate, readr, dplyr, ggplot2, rlang, tidyr, vroom, tibble, stats

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**URL** <https://github.com/ikemillar/CDSim>

**BugReports** <https://github.com/ikemillar/CDSim/issues>

**NeedsCompilation** no

**Author** Isaac Osei [aut, cre],  
Acheampong Baafi-Adomako [aut]

**Repository** CRAN

**Date/Publication** 2026-04-22 14:10:02 UTC

## Contents

|                                     |   |
|-------------------------------------|---|
| CDSim-package . . . . .             | 2 |
| create_stations . . . . .           | 2 |
| plot_station_timeseries . . . . .   | 3 |
| safe_name . . . . .                 | 4 |
| simulate_climate_series . . . . .   | 5 |
| validate_climate . . . . .          | 7 |
| validate_climate_internal . . . . . | 7 |
| visualization . . . . .             | 8 |
| write_station_csv . . . . .         | 8 |
| write_station_netcdf . . . . .      | 9 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>10</b> |
|--------------|-----------|

---

|               |                                               |
|---------------|-----------------------------------------------|
| CDSim-package | <i>CDSim: Climate Data Simulation Toolkit</i> |
|---------------|-----------------------------------------------|

---

### Description

Tools for generating and exporting synthetic climate observation datasets.

### Author(s)

Isaac Osei and Acheampong Baafi-Adomako and Sivaparvathi Dusari

### See Also

Useful links:

- <https://github.com/ikemillar/CDSim>
- Report bugs at <https://github.com/ikemillar/CDSim/issues>

---

|                 |                                        |
|-----------------|----------------------------------------|
| create_stations | <i>Create or load station metadata</i> |
|-----------------|----------------------------------------|

---

### Description

Create a station metadata table (Station, LON, LAT) either by:

- loading from a CSV file,
- accepting an existing data.frame,
- or auto-generating synthetic stations in a bounding box.

**Usage**

```
create_stations(  
  source = NULL,  
  n = 10,  
  bbox = c(-3.5, 1.5, 4.5, 11.5),  
  seed = NULL  
)
```

**Arguments**

|        |                                                                                        |
|--------|----------------------------------------------------------------------------------------|
| source | Path to CSV file OR a data.frame with Station/LON/LAT OR NULL (to generate synthetic). |
| n      | Integer number of stations to generate when source = NULL. Default 10.                 |
| bbox   | numeric vector c(min_lon, max_lon, min_lat, max_lat). Default ~ Ghana bounding box.    |
| seed   | Optional numeric to make generation reproducible.                                      |

**Value**

A data.frame with columns Station, LON, LAT.

**Examples**

```
create_stations(n = 5, seed = 42)  
create_stations(data.frame(Station="A", LON=0, LAT=5))
```

---

plot\_station\_timeseries

*Plot Station Time Series with Seasonal Detection*

---

**Description**

Creates a time-series plot for climate variables with automatic hemisphere-based season detection.

**Usage**

```
plot_station_timeseries(  
  df,  
  station,  
  var = "Avg.Tn",  
  smooth = TRUE,  
  theme_dark = FALSE  
)
```

**Arguments**

|            |                                                                       |
|------------|-----------------------------------------------------------------------|
| df         | A tidy dataset containing columns: Station, Date, LAT, and variables. |
| station    | Station name.                                                         |
| var        | Climate variable to plot.                                             |
| smooth     | Add LOESS smoothing line.                                             |
| theme_dark | Use dark theme.                                                       |

**Value**

A ggplot object.

**Examples**

```
stations <- create_stations(n = 3)
sim <- simulate_climate_series(stations)
plot_station_timeseries(sim, station = "Station_1", var = "Avg.Tn")
```

---

|           |                             |
|-----------|-----------------------------|
| safe_name | <i>Make a safe filename</i> |
|-----------|-----------------------------|

---

**Description**

Ensures file names contain only safe ASCII characters.

**Usage**

```
safe_name(x)
```

```
safe_name(x)
```

**Arguments**

|   |                              |
|---|------------------------------|
| x | A character string to clean. |
|---|------------------------------|

**Value**

A cleaned filename string.

---

```
simulate_climate_series
```

*Simulate monthly climate time series for stations*

---

## Description

Simulate monthly Tmin, Tmax, monthly total rainfall (Sum.Rf) and mean daily rainfall (Avg.Rf) for each station across a year range.

## Usage

```
simulate_climate_series(
  stations,
  start_year = 1996,
  end_year = 2025,
  seed = NULL,
  temp_trend_per_year = 0.02,
  rain_trend_per_year = -0.003,
  phi_temp = 0.85,
  sd = 0.4,
  Tmin_min = 18,
  Tmin_max = 30,
  Tmax_min = 24,
  Tmax_max = 42
)
```

## Arguments

|                     |                                                                                        |
|---------------------|----------------------------------------------------------------------------------------|
| stations            | data.frame from create_stations() (Station, LON, LAT)                                  |
| start_year          | integer (e.g., 1996)                                                                   |
| end_year            | integer (e.g., 2025)                                                                   |
| seed                | optional numeric seed                                                                  |
| temp_trend_per_year | temperature trend per year (°C/year warming)                                           |
| rain_trend_per_year | rain trend per year (slight drying trend)                                              |
| phi_temp            | AR(1) persistence                                                                      |
| sd                  | standard deviation of the AR(1) innovation process controlling temperature variability |
| Tmin_min            | minimum value for minimum temperature                                                  |
| Tmin_max            | maximum value for minimum temperature                                                  |
| Tmax_min            | minimum value for maximum temperature                                                  |
| Tmax_max            | maximum value for maximum temperature                                                  |

## Details

This function generates synthetic monthly climate time series using a stochastic, physically-informed modelling framework. Temperature is modeled as a combination of deterministic seasonality, long-term trend, and stochastic variability. The seasonal component is represented using a sinusoidal function, while temporal persistence is introduced via an autoregressive AR(1) process applied to the innovation term.

Minimum temperature (Avg.Tn) is simulated using a truncated normal distribution to enforce physically realistic lower and upper bounds. Maximum temperature (Avg.Tx) is generated using a gamma-distributed perturbation applied to the mean temperature, producing an asymmetric distribution consistent with observed climatological behavior.

Rainfall occurrence is modeled using a first-order Markov chain, allowing for realistic wet-dry persistence. Conditional on occurrence, rainfall intensity is drawn from a gamma distribution with seasonally varying mean. A temporal trend term can be applied to represent long-term climatic changes such as gradual drying or wetting.

To ensure physical consistency between variables, a coupling mechanism is introduced whereby increased rainfall (proxy for cloud cover) reduces maximum temperature through a linear cooling adjustment. This enforces a negative dependence between precipitation and temperature consistent with atmospheric energy balance principles.

Finally, a minimum diurnal temperature difference constraint is enforced after rounding to guarantee that  $\text{Avg.Tx} > \text{Avg.Tn}$  at all time steps, while preserving the statistical distribution of the simulated variables.

The default parameterization reflects typical tropical conditions for Ghana, but all parameters are user-configurable, allowing adaptation to other climatic regions. The modelling approach follows established stochastic weather generation principles while extending them with distributional asymmetry and cross-variable coupling for improved physical realism.

## Value

A tidy data.frame with one row per station  $\times$  month containing: Station, LON, LAT, Year, Month, Date, Avg.Tn, Avg.Tx, Sum.Rf, Avg.Rf

## See Also

[write\\_station\\_csv\(\)](#), [write\\_station\\_netcdf\(\)](#)

## Examples

```
st <- create_stations(n = 3, seed = 1)
sim <- simulate_climate_series(st, 1996, 2025, seed = 42)
head(sim)
```

---

|                  |                                                             |
|------------------|-------------------------------------------------------------|
| validate_climate | <i>Validate simulated climate data against observations</i> |
|------------------|-------------------------------------------------------------|

---

**Description**

Performs statistical and physical validation of simulated climate data against observed datasets, including distributional tests, mean comparison, dependence structure, and temporal persistence.

**Usage**

```
validate_climate(sim, obs)
```

**Arguments**

|     |                              |
|-----|------------------------------|
| sim | Simulated climate data.frame |
| obs | Observed climate data.frame  |

**Value**

A list containing validation metrics and test results

---

|                           |                                                      |
|---------------------------|------------------------------------------------------|
| validate_climate_internal | <i>Internal validation of simulated climate data</i> |
|---------------------------|------------------------------------------------------|

---

**Description**

Evaluates physical plausibility and statistical properties of simulated climate data in the absence of observational datasets. The function assesses distributional characteristics, temporal persistence, inter-variable relationships, and physical constraints.

**Usage**

```
validate_climate_internal(sim)
```

**Arguments**

|     |                              |
|-----|------------------------------|
| sim | Simulated climate data.frame |
|-----|------------------------------|

**Value**

A list of validation diagnostics

---

|               |                                                 |
|---------------|-------------------------------------------------|
| visualization | <i>Visualization Functions for Climate Data</i> |
|---------------|-------------------------------------------------|

---

**Description**

Visualization Functions for Climate Data

---

|                   |                                                                                     |
|-------------------|-------------------------------------------------------------------------------------|
| write_station_csv | <i>Write station CSV Exports a simulated climate station dataset to a CSV file.</i> |
|-------------------|-------------------------------------------------------------------------------------|

---

**Description**

Write station CSV Exports a simulated climate station dataset to a CSV file.

**Usage**

```
write_station_csv(df, file = "simulated_station_climate.csv")
```

**Arguments**

|      |                                                                  |
|------|------------------------------------------------------------------|
| df   | A dataframe returned by <code>simulate_climate_series()</code> . |
| file | The output CSV filename.                                         |

**Value**

Returns the file path invisibly.

**Examples**

```
stations <- create_stations(n = 3)
sim <- simulate_climate_series(stations)
tmp <- tempfile(fileext = ".csv")
write_station_csv(sim, tmp)
```

---

write\_station\_netcdf *Write station NetCDF (station x time) Exports a simulated climate station dataset to a NetCDF file.*

---

### Description

Write station NetCDF (station x time) Exports a simulated climate station dataset to a NetCDF file.

### Usage

```
write_station_netcdf(  
  df,  
  out_nc = "simulated_station_climate.nc",  
  fillvalue = -9999  
)
```

### Arguments

|           |                                                                     |
|-----------|---------------------------------------------------------------------|
| df        | station x time long dataframe returned by simulate_climate_series() |
| out_nc    | Output NetCDF filename                                              |
| fillvalue | Value used for missing entries                                      |

### Value

Returns the file path invisibly.

### Examples

```
stations <- create_stations(n = 3)  
sim <- simulate_climate_series(stations)  
tmp <- tempfile(fileext = ".nc")  
write_station_netcdf(sim, tmp)
```

# Index

## \* IO Functions

simulate\_climate\_series, 5

## \* climate

CDSim-package, 2

## \* netcdf

CDSim-package, 2

## \* rainfall

CDSim-package, 2

## \* simulation

CDSim-package, 2

## \* temperature

CDSim-package, 2

CDSim (CDSim-package), 2

CDSim-package, 2

create\_stations, 2

plot\_station\_timeseries, 3

safe\_name, 4

simulate\_climate\_series, 5

validate\_climate, 7

validate\_climate\_internal, 7

visualization, 8

write\_station\_csv, 8

write\_station\_csv(), 6

write\_station\_netcdf, 9

write\_station\_netcdf(), 6