Package 'floodflow'

July 29, 2026

Type Package

Title Map-First Climate-Informed Flood Assessment for Data-Scarce Basins

Version 0.1.1

Description A reproducible, map-oriented workflow for flood hazard assessment that chains rainfall extreme value analysis, rainfall-runoff simulation, terrain-based flow routing and water-depth estimation into a single pipeline. A stationary-versus-nonstationary test for changing rainfall extremes is built in, and any flood scenario can be produced for a present-day or a climate-adjusted design event. Defaults target settings with sparse gauge networks, using satellite or reanalysis rainfall, temperature-based potential evapotranspiration and regional pooling of short records. Heavy modelling engines are wrapped rather than reimplemented so that the core stays lightweight. Methods follow established hydrology, including the generalized extreme value distribution for rainfall maxima (Coles, 2001, <doi:10.1007/978-1-4471-3675-0>) and Manning's equation for open-channel flow.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1)

Imports stats

Suggests terra, sf, extRemes, airGR, whitebox, tmap, leaflet, leafsync, ranger, lmomRFA, shiny, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/gowusu/floodflow, https://gowusu.github.io/floodflow/

BugReports https://github.com/gowusu/floodflow/issues

Config/roxygen2/version 8.0.0

Config/Needs/website pkgdown

NeedsCompilation no

Author George Owusu [aut, cre]

Maintainer George Owusu <owusugeorge@ug.edu.gh>

Repository CRAN

Date/Publication 2026-07-29 16:40:27 UTC

Contents

floodflow_lc_roughness *Default Manning's roughness by land-cover class*

Description

A named numeric vector of representative Manning's n values for common land-cover classes, drawn from standard hydraulic references and distributed hydrological models. Used as the default lookup by `roughness` when `method = "landcover"`. Users may supply their own table.

Usage

```
floodflow_lc_roughness
```

Format

A named numeric vector. Names are land-cover classes; values are Manning's n.

Value

A named numeric vector of length 8. Each element is a representative Manning's roughness coefficient n (dimensionless) and its name is the land-cover class it applies to (`"water"`, `"urban"`, `"bare"`, `"grassland"`, `"cropland"`, `"shrub"`, `"forest"`, `"wetland"`). It is the default land-cover-to-roughness lookup table used by `roughness`.

Examples

```
floodflow_lc_roughness["forest"]
```

flood_extremes *Analyse rainfall extremes and test for a changing climate*

Description

Reduces a daily rainfall record to annual maxima and fits the generalized extreme value (GEV) distribution, both as a stationary model and as a non-stationary model whose location parameter trends linearly with time. A likelihood-ratio test compares the two, providing a formal test of whether extreme rainfall has intensified over the record. Design return levels (for example the 100-year daily rainfall) are computed from the stationary fit.

Usage

```
flood_extremes(
  x,
  periods = c(2, 10, 25, 50, 100),
  engine = c("internal", "extRemes")
)
```

Arguments

x	A flood_project whose rainfall slot holds a data.frame, or a data.frame directly. The data frame must have a date column (of class Date or coercible) and a precip_mm column of daily rainfall in millimetres.
periods	Numeric vector of return periods, in years, at which to report design rainfall. Defaults to c(2, 10, 25, 50, 100).
engine	Which fitting engine to use: "internal" (default, no dependencies) or "extRemes" (uses the **extRemes** package if installed).

Details

By default a small internal maximum-likelihood engine is used, so no extra package is required. If **extRemes** is installed and engine = "extRemes", that package is used for the stationary fit instead.

Value

If x is a flood_project, the same object with its extremes slot populated and the stage recorded in the log. If x is a data.frame, the extremes result list directly. The result is a list of class flood_extremes with elements: annual_max (a data frame of year and maximum), stationary (fitted parameters and negative log-likelihood), trend (the non-stationary fit, including the per-year location trend mu1), lr_test (a list with the likelihood-ratio statistic, df and p_value), trend_detected (logical, TRUE when p_value < 0.05), and return_levels (a data frame of period and level_mm).

References

Coles, S. (2001) An Introduction to Statistical Modeling of Extreme Values. Springer. doi:10.1007/9781447136750

See Also

flood_scenario to turn these design levels into a climate-adjusted event.

Examples

```
# Build a synthetic 40-year daily rainfall record with a mild upward trend
set.seed(1)
dates <- seq(as.Date("1985-01-01"), as.Date("2024-12-31"), by = "day")
yr <- as.integer(format(dates, "%Y"))
base <- rgamma(length(dates), shape = 0.7, scale = 6)
trend <- 1 + 0.02 * (yr - 1985)
precip <- round(base * trend * rbinom(length(dates), 1, 0.3), 1)
rain <- data.frame(date = dates, precip_mm = precip)

res <- flood_extremes(rain)
res$return_levels
res$trend_detected
```

`flood_hydraulics` *Derive hydraulic quantities from routed flow*

Description

Computes the family of time-and-motion quantities that a routed flood implies: peak flow velocity, time of concentration (by one or more methods), channel travel time, and the time-to-peak of the routed hydrograph. These are the layers a geographer maps alongside depth.

Usage

```
flood_hydraulics(
  x,
  length_m = 5000,
  overland_m = 100,
  retardance = 0.4,
  dt_hours = 24
)
```

Arguments

`x`	A `flood_project` whose `route` slot has been populated, or a `flood_route` object directly.
`length_m`	Representative flow-path (channel) length in metres, used for time of concentration and travel time. Default `5000`.
`overland_m`	Overland flow length in metres for the Kerby component. Default `100`.
`retardance`	Kerby retardance coefficient. Default `0.4`.
`dt_hours`	Time step of the routed hydrograph in hours, used to convert the time-to-peak index into hours. Default `24` (daily).

Details

Velocity comes from Manning's equation at the routed peak depth. Time of concentration is available by the Kirpich channel method, the Kerby overland method, the combined Kerby-Kirpich sum, and a velocity-based travel time (flow-path length divided by peak velocity). Time-to-peak is read directly from the routed hydrograph.

Value

If `x` is a `flood_project`, the same object with its `hydraulics` slot populated. Otherwise a list of class `flood_hydraulics` with elements `peak_velocity_ms`, `tc` (a named vector of times of concentration in minutes by method: `kirpich`, `kerby`, `kerby_kirpich`, `velocity`), `travel_time_min`, and `time_to_peak_hours`.

See Also

`tc_kirpich`, `tc_kerby`.

Examples

```
disc <- data.frame(
  date = seq(as.Date("2020-06-01"), by = "day", length.out = 15),
  Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1, 0, 0, 0)
)
r <- flood_route(disc, method = "muskingum-cunge")
h <- flood_hydraulics(r, length_m = 4000, overland_m = 120)
h$tc
h$peak_velocity_ms
```

`flood_map` *Map a flood layer*

Description

Renders a chosen pipeline layer for viewing. When **tmap** or **leaflet** is installed and the layer is spatial, an interactive map is produced; otherwise the function returns a tidy data frame of the layer's values (and, for non-spatial results, draws a simple base-R plot) so the pipeline remains usable without the mapping engines. This keeps mapping a first-class output while respecting the package's lightweight core.

Usage

```
flood_map(
  x,
  layer = c("depth", "risk", "velocity", "uncertainty"),
  interactive = TRUE
)
```

Arguments

`x`	A `flood_project` with the requested layer populated.
`layer`	Which layer to map: `"depth"` (routed peak depth), `"risk"` (vulnerability index), `"velocity"`, or `"uncertainty"` (predictive band width). Default `"depth"`.
`interactive`	Logical; if `TRUE` (default) and an engine is available, attempt an interactive map. If no engine is available this is ignored and a data frame is returned.

Value

A list of class `flood_map` with elements `layer`, `rendered` (logical: whether an interactive/graphic map was drawn), `engine` (the engine used, or `"none"`), and `data` (a tidy summary of the mapped values). When an interactive map is produced, the map object is attached as `map`.

See Also

`flood_route`, `flood_vulnerability`.

Examples

```
set.seed(1)
rain <- data.frame(
  date = seq(as.Date("1990-01-01"), as.Date("2020-12-31"), by = "day"),
  precip_mm = round(rgamma(11323, 0.7, scale = 6) *
                    rbinom(11323, 1, 0.3), 1)
)
fp <- flood_project("demo")
fp$rainfall <- rain
fp <- flood_runoff(fp, engine = "simple")
fp <- flood_route(fp, area_km2 = 300)
m <- flood_map(fp, layer = "depth")
m$data
```

`flood_project` *Create a flood project object*

Description

Constructs the container object that carries data and results through every stage of the **floodflow** pipeline. A `flood_project` is a named list with a fixed set of slots; pipeline functions read from and write to these slots, so a single object accumulates the DEM, rainfall record, fitted extreme-value model, routed discharge, water depth and derived maps as it passes through the workflow.

Usage

```
flood_project(name = "flood_project", crs = NULL, meta = list())
```

Arguments

`name`	Character scalar naming the project or study basin, used in printing and map titles. Defaults to `"flood_project"`.
`crs`	Optional character scalar giving the target coordinate reference system as an EPSG string (for example `"EPSG:32630"` for UTM zone 30N, appropriate for Accra). Stored for later stages; not applied here.
`meta`	Optional named list of user metadata to attach to the project (for example data provenance notes). Defaults to an empty list.

Details

The constructor deliberately performs no geospatial work and has no heavy dependencies, so it can be created and inspected without **terra** or any modelling engine installed. Slots that are not yet populated are held as `NULL`.

Value

An object of class `flood_project`: a named list with slots `name`, `crs`, `meta`, `dem`, `rainfall`, `extremes`, `scenario`, `roughness`, `runoff`, `route`, `hydraulics`, `uncertainty`, `vulnerability` and `log`. All data slots are `NULL` until populated by later pipeline functions. The `log` slot is a character vector recording the stages that have been run.

See Also

`is_flood_project` to test the class.

Examples

```
fp <- flood_project(name = "Odaw basin")
fp
is_flood_project(fp)
```

---

flood_route *Route flow and compute water depth*

Description

Converts the discharge series from `flood_runoff` into a routed hydrograph and a water depth, using one of five methods that form a complexity ladder. All methods obtain depth from Manning's equation for a wide channel; they differ in how the flood wave is routed, from a steady baseline to progressively more physics.

Usage

```
flood_route(
  x,
 method = c("muskingum-cunge", "manning-normal", "kinematic", "diffusive", "dynamic"),
  width = 20,
  slope = 0.001,
  n = 0.035,
  celerity = 1.5,
  dx = 1000,
  dt = 86400,
  area_km2 = 100,
  hand = NULL
)
```

Arguments

`x`	A `flood_project` whose `runoff` slot has been populated, or a discharge `data.frame` with columns `date` and `Q_mm`.

method	One of "muskingum-cunge" (default), "manning-normal", "kinematic", "diffusive" or "dynamic".
width	Representative channel width in metres. Default 20.
slope	Representative bed slope (dimensionless). Default 0.001.
n	Manning's roughness. If x is a project with a scalar roughness set, that value is used unless n is given explicitly. Default 0.035.
celerity	Wave celerity in m/s for routing. Default 1.5.
dx	Reach length in metres for routing. Default 1000.
dt	Time step in seconds. Default 86400 (daily).
area_km2	Catchment area in square kilometres, used to convert runoff depth (mm/day) to volumetric discharge (m^3/s). Default 100.
hand	Optional Height Above Nearest Drainage surface as a numeric vector or, with **terra** installed, a SpatRaster. When supplied, the scalar peak depth is turned into a spatial inundation-depth field (flooding cells whose height above drainage is below the peak water level), stored as depth_raster and drawn by flood_map. A raw DEM may be passed for a crude proxy. Default NULL (no spatial depth).

Details

"manning-normal" Steady uniform flow. Depth is the Manning normal depth of the (unrouted) peak discharge. The fast baseline.

"kinematic" Kinematic wave: near-pure translation of the hydrograph with negligible attenuation. Suited to steeper channels.

"diffusive" Diffusive wave: adds hydraulic diffusion so the peak attenuates and backwater effects appear.

"muskingum-cunge" Physically-based storage routing at diffusive-wave accuracy and low cost. The pragmatic default.

"dynamic" Uses the discharge-scaled hydraulic diffusivity as the best stable approximation available in pure R. Full two-dimensional Saint-Venant hydrodynamics are out of scope; for those, couple to a dedicated hydraulic model.

All routing is carried out with the numerically stable Muskingum-Cunge family, varying its diffusion to represent each rung of the ladder.

Value

If x is a flood_project, the same object with its route slot populated. Otherwise a list of class flood_route with elements method, routed (a data frame of date, Q_cms inflow and Q_routed outflow), peak_depth_m, peak_velocity_ms, attenuation (routed peak divided by inflow peak), depth_raster (a spatial inundation-depth field when hand was supplied, otherwise NULL) and the hydraulic settings used.

References

Cunge, J. A. (1969). On the subject of a flood propagation computation method (Muskingum method). *Journal of Hydraulic Research*, 7(2), 205-230.

Manning, R. (1891). On the flow of water in open channels and pipes. *Transactions of the Institution of Civil Engineers of Ireland*, 20, 161-207.

See Also

flood_runoff for the discharge this routes.

Examples

```
set.seed(1)
dates <- seq(as.Date("2020-06-01"), by = "day", length.out = 30)
Q <- c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1, rep(0, 18))
disc <- data.frame(date = dates, Q_mm = Q)

r_mc  <- flood_route(disc, method = "muskingum-cunge")
r_kin <- flood_route(disc, method = "kinematic")
# Kinematic attenuates less than Muskingum-Cunge
r_kin$attenuation >= r_mc$attenuation

# Spatial inundation depth from a HAND surface (a numeric vector here; a
# terra SpatRaster works the same way and produces a mappable raster)
hand <- c(0, 0.5, 1, 2, 4, 6)
r_spatial <- flood_route(disc, area_km2 = 300, hand = hand)
r_spatial$depth_raster        # deepest in the valley, dry on high ground
```

---

flood_runoff *Simulate discharge from rainfall*

Description

Converts a daily rainfall record into a discharge (streamflow) series, the link between rainfall in the sky and water in the channel. floodflow operates on a daily timestep: each row of the rainfall data frame is one day, and discharge is returned in mm/day; aggregate any sub-daily record to daily totals before use. Potential evapotranspiration is computed by the Oudin formula from temperature and latitude. When **airGR** is installed, the GR4J lumped conceptual model is used; otherwise a simple conceptual fallback runs so the pipeline still produces a discharge series.

Usage

```
flood_runoff(
  x,
  lat_deg = 5.6,
  temp_c = 28,
  params = NULL,
  engine = c("auto", "airGR", "simple")
)
```

Arguments

x	A flood_project whose rainfall slot holds a data frame with date and precip_mm, or such a data frame directly. A temp_c column is used if present; otherwise a constant temperature is assumed.

lat_deg	Latitude in decimal degrees, for PET. Defaults to 5.6 (Accra).
temp_c	Constant air temperature (degrees C) used when the rainfall data has no temp_c column. Default 28.
params	Optional named numeric vector of GR4J parameters X1, X2, X3, X4. Ignored by the fallback.
engine	Which engine to use: "auto" (GR4J if **airGR** is present, else the fallback), "airGR" (require GR4J) or "simple" (force the fallback).

Details

GR4J parameters may be supplied; if not, illustrative defaults are used. In a real study with observed discharge, calibrate the parameters (for example with airGR::Calibration_Michel) before relying on the output.

Infiltration is represented *implicitly*, not as a separate step. Rainfall is partitioned into runoff and soil storage by a production store (a soil-moisture bucket of capacity store_max): when the soil is dry most rain infiltrates and little runs off, and as it saturates the runoff fraction rises toward one. This is a saturation-excess representation. There is no explicit Green-Ampt, Horton, or SCS curve-number infiltration function in this version (curve-number infiltration is planned for a future release).

Value

If x is a flood_project, the same object with its runoff slot populated. Otherwise a list of class flood_runoff with elements discharge (a data frame of date and Q_mm), pet (the PET series), engine used, and peak (the maximum discharge and its date).

See Also

pet_oudin for the evapotranspiration input.

Examples

```
set.seed(1)
dates <- seq(as.Date("2020-01-01"), as.Date("2021-12-31"), by = "day")
rain <- data.frame(
  date = dates,
  precip_mm = round(rgamma(length(dates), 0.7, scale = 8) *
                    rbinom(length(dates), 1, 0.4), 1)
)
rr <- flood_runoff(rain, engine = "simple")
rr$peak
```

`flood_scenario` *Generate a climate-adjusted design flood event*

Description

Turns the design return levels from `flood_extremes` into a scenario for a chosen future, so that a flood can be modelled under present-day or changed-climate conditions. Three methods are offered, from zero-dependency to full-fidelity:

Usage

```
flood_scenario(
  x,
  method = c("delta", "trend", "cmip6"),
  change_factor = 1.15,
  horizon_year = NULL,
  scenario_label = NULL
)
```

Arguments

`x`	A `flood_project` whose `extremes` slot has been populated by `flood_extremes`, or a `flood_extremes` object directly.
`method`	One of `"delta"` (default), `"trend"` or `"cmip6"`.
`change_factor`	Numeric multiplier for `method = "delta"`. A value of 1 leaves rainfall unchanged; 1.15 raises it by 15%. Ignored by other methods.
`horizon_year`	Target year for `method = "trend"`. The location trend is projected from the end of the record to this year. Ignored by other methods.
`scenario_label`	Optional character label for the scenario (for example `"SSP5-8.5 2050"`), stored with the result and used in maps.

Details

`"trend"` Extrapolate the fitted non-stationary location trend forward to a target year. Uses only the record already analysed.

`"delta"` Scale the stationary return levels by a change factor (for example 1.15 for a 15% increase). The default and recommended route for data-scarce settings; change factors can come from published CMIP6 summaries per Shared Socioeconomic Pathway.

`"cmip6"` Placeholder for ingesting downscaled CMIP6 projections directly; not yet implemented, and currently returns an informative error. Use `"delta"` with a CMIP6-derived change factor in the meantime.

Value

If `x` is a `flood_project`, the same object with its `scenario` slot populated. Otherwise a list of class `flood_scenario` with elements `method`, `label`, `baseline` (the present-day return-level data frame), `adjusted` (a data frame of `period` and `level_mm` under the scenario), and `change` (the ratio of adjusted to baseline at each period).

References

IPCC (2021). *Climate Change 2021: The Physical Science Basis*. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press.

See Also

`flood_extremes` for the design levels this adjusts.

Examples

```
set.seed(1)
rain <- data.frame(
  date = seq(as.Date("1985-01-01"), as.Date("2024-12-31"), by = "day"),
  precip_mm = round(rgamma(14610, 0.7, scale = 6) *
                    rbinom(14610, 1, 0.3), 1)
)
ext <- flood_extremes(rain)

# 15% wetter design storm
sc <- flood_scenario(ext, method = "delta", change_factor = 1.15,
                     scenario_label = "SSP2-4.5 2050")
sc$adjusted
```

`flood_surrogate` *Train a fast surrogate emulator of the flood model*

Description

Builds a machine-learning emulator that approximates the routing model's depth response to its parameters, so that many what-if scenarios can be explored far faster than re-running the full pipeline. Training data are generated by evaluating the Manning depth relation across sampled parameters; the surrogate then predicts depth for new parameter sets.

Usage

```
flood_surrogate(
  x,
  n_train = 500,
  n_range = c(0.02, 0.08),
```

```
  width_range = c(10, 40),
  seed = NULL
)
```

Arguments

x	A flood_project whose route slot supplies the fixed slope and discharge context, or a flood_route object directly.
n_train	Number of training samples. Default 500.
n_range, width_range	Length-2 priors for Manning's n and channel width used to generate training data.
seed	Optional integer seed for reproducibility.

Details

When **ranger** is installed, a random forest is used; otherwise a log-linear model provides a dependency-free fallback (exact for the power-law Manning relation, and a reasonable approximation more generally). The surrogate is a convenience for rapid exploration and interpolation, not a replacement for the physical model.

Value

If x is a flood_project, the same object with its surrogate entry stored in meta (the project schema has no dedicated slot) and the stage logged. Otherwise a list of class flood_surrogate with elements engine ("ranger" or "loglinear"), predict (a function taking a data frame with columns Q, n, width and returning predicted depth), performance (in-sample rmse and r2), and settings.

See Also

flood_route for the model being emulated.

Examples

```
disc <- data.frame(
  date = seq(as.Date("2020-06-01"), by = "day", length.out = 12),
  Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1)
)
r <- flood_route(disc, area_km2 = 300)
s <- flood_surrogate(r, n_train = 300, seed = 1)
s$performance
# Predict depth for new parameter sets, fast
s$predict(data.frame(Q = 150, n = 0.04, width = 25))
```

`flood_uncertainty` *Quantify uncertainty and invert for parameters (GLUE)*

Description

Applies Generalized Likelihood Uncertainty Estimation (GLUE) to the routing stage. Uncertain parameters (Manning's roughness and channel width) are sampled from priors, the flood depth is predicted for each sample, and each sample is weighted by an informal likelihood measuring its agreement with an observed depth. This yields a predictive uncertainty band on flood depth and, as the inverse problem, weighted parameter estimates conditioned on the observation.

Usage

```
flood_uncertainty(
  x,
  observed_depth_m,
  n_sim = 5000,
  n_range = c(0.02, 0.08),
  width_range = c(10, 40),
  obs_error = 0.1,
  behavioural_fraction = 0.1,
  seed = NULL
)
```

Arguments

`x`	A `flood_project` whose `route` slot has been populated, or a `flood_route` object directly. The routing settings (slope, area, peak discharge) are taken from it.
`observed_depth_m`	Observed peak flood depth in metres to condition on, for example from a surveyed high-water mark or satellite estimate.
`n_sim`	Number of Monte-Carlo samples. Default `5000`.
`n_range`	Length-2 numeric range for the Manning's n prior. Default `c(0.02, 0.08)`.
`width_range`	Length-2 numeric range for the channel width prior (m). Default `c(10, 40)`.
`obs_error`	Relative observation error (standard deviation as a fraction of the observed depth) used in the Gaussian likelihood. Default `0.1`.
`behavioural_fraction`	Fraction of samples, ranked by likelihood, kept as behavioural. Default `0.1`.
`seed`	Optional integer seed for reproducibility.

Details

GLUE is the workhorse uncertainty method in hydrology. A key feature it reveals is equifinality: many different parameter combinations can reproduce the same observation, so individual parameters may stay uncertain even when the prediction is well constrained. The function reports the parameter spread honestly rather than collapsing it to a single point.

Value

If `x` is a `flood_project`, the same object with its `uncertainty` slot populated. Otherwise a list of class `flood_uncertainty` with elements `observed_depth_m`, `n_behavioural` (count kept), `depth_band` (named vector: lower, median, upper of the weighted predictive band), `obs_in_band` (logical), `estimates` (weighted-mean and range for each parameter), `equifinality` (the n-width correlation among behavioural sets), and `behavioural` (a data frame of kept samples and weights).

References

Beven, K. and Binley, A. (1992) The future of distributed models: model calibration and uncertainty prediction. Hydrological Processes 6, 279–298. doi:10.1002/hyp.3360060305

See Also

`flood_route` for the model being conditioned.

Examples

```
disc <- data.frame(
  date = seq(as.Date("2020-06-01"), by = "day", length.out = 12),
  Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1)
)
r <- flood_route(disc, method = "muskingum-cunge", area_km2 = 300)
u <- flood_uncertainty(r, observed_depth_m = r$peak_depth_m,
                       n_sim = 2000, seed = 1)
u$depth_band
u$obs_in_band
```

---

flood_vulnerability *Combine hazard, exposure and vulnerability into a risk index*

Description

Implements the standard disaster-risk decomposition $Risk = Hazard \times Exposure \times Vulnerability$. Each input layer is normalised to $[0, 1]$, the three are multiplied cell by cell, and the product is re-normalised to give a relative risk index. Because the combination is multiplicative, risk is zero wherever any component is zero: there is no risk without all of hazard, something exposed, and susceptibility.

Usage

```
flood_vulnerability(x, exposure, vulnerability = NULL, hazard = NULL)
```

Arguments

x	A flood_project whose route slot supplies the hazard (peak depth), or a numeric hazard vector / SpatRaster directly.
exposure	Numeric vector or SpatRaster of exposure (for example population or building density).
vulnerability	Optional numeric vector or SpatRaster of social vulnerability (for example a deprivation index). If NULL, treated as uniform.
hazard	Optional explicit hazard layer when x is a flood_project; overrides deriving hazard from the project.

Details

Inputs are numeric vectors (aligned cell values) or, when **terra** is installed, SpatRaster layers. Vulnerability may be omitted, in which case exposure alone weights the hazard.

Value

If x is a flood_project, the same object with its vulnerability slot populated. Otherwise a list of class flood_vulnerability with elements risk (the normalised risk index), components (the normalised hazard, exposure and vulnerability), and summary (min, mean, max of risk).

References

Reimann, L. et al. (2024) An empirical social vulnerability map for flood risk assessment at global scale. Earth's Future 12. doi:10.1029/2023EF003895

See Also

flood_route for the hazard layer.

Examples

```
set.seed(1)
hazard <- runif(100, 0, 5)      # flood depth
pop <- rpois(100, 50)           # exposure
deprivation <- runif(100)       # vulnerability
v <- flood_vulnerability(hazard, exposure = pop,
                         vulnerability = deprivation)
range(v$risk)
```

is_flood_project *Test whether an object is a flood project*

Description

Test whether an object is a flood project

Usage

```
is_flood_project(x)
```

Arguments

x	An object to test.

Value

A single logical value: TRUE if x inherits from class flood_project, otherwise FALSE.

Examples

```
is_flood_project(flood_project())
is_flood_project(list())
```

pet_oudin *Potential evapotranspiration by the Oudin formula*

Description

Computes daily potential evapotranspiration (PET) from air temperature and latitude alone, following Oudin et al. (2005). Because it needs no radiation, humidity or wind data, it suits data-scarce settings where only temperature is available. Extraterrestrial radiation is derived from solar geometry for the given day of year and latitude.

Usage

```
pet_oudin(jday, temp_c, lat_deg)
```

Arguments

jday	Integer vector of Julian day of year (1–366).
temp_c	Numeric vector of mean daily air temperature in degrees Celsius, the same length as jday.
lat_deg	Latitude in decimal degrees (positive north, negative south).

Value

A numeric vector of PET in millimetres per day, never negative.

References

Oudin, L. et al. (2005) Which potential evapotranspiration input for a lumped rainfall-runoff model? Journal of Hydrology 303, 290–306. doi:10.1016/j.jhydrol.2004.08.026

Examples

```
# A year of PET for Accra (latitude ~5.6 N)
jd <- 1:365
temp <- 28 + 3 * sin(2 * pi * (jd - 40) / 365)
pet <- pet_oudin(jd, temp, lat_deg = 5.6)
range(pet)
```

---

print.flood_extremes *Print a flood extremes result*

Description

Print a flood extremes result

Usage

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

Arguments

x	A flood_extremes object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
set.seed(1)
rain <- data.frame(
  date = seq(as.Date("1990-01-01"), as.Date("2020-12-31"), by = "day"),
  precip_mm = round(rgamma(11323, 0.7, scale = 6), 1)
)
print(flood_extremes(rain))
```

print.flood_hydraulics *Print a flood hydraulics result*

Description

Print a flood hydraulics result

Usage

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

Arguments

x	A flood_hydraulics object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
disc <- data.frame(date = seq(as.Date("2020-06-01"), by = "day",
                              length.out = 12),
                   Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1))
print(flood_hydraulics(flood_route(disc)))
```

print.flood_map *Print a flood map*

Description

Print a flood map

Usage

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

Arguments

x	A flood_map object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
set.seed(1)
rain <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2010-12-31"), by = "day"),
  precip_mm = round(rgamma(4018, 0.7, scale = 6) *
                    rbinom(4018, 1, 0.3), 1)
)
fp <- flood_project("demo"); fp$rainfall <- rain
fp <- flood_route(flood_runoff(fp, engine = "simple"), area_km2 = 300)
print(flood_map(fp, layer = "depth"))
```

---

print.flood_project *Print a flood project*

Description

Print a flood project

Usage

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

Arguments

x	A flood_project object.
...	Ignored, present for S3 method consistency.

Value

The flood_project object x, returned invisibly. Called for the side effect of printing a compact summary to the console.

Examples

```
print(flood_project(name = "Odaw basin"))
```

print.flood_roughness *Print a roughness result*

Description

Print a roughness result

Usage

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

Arguments

x	A flood_roughness object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
print(roughness(method = "constant", value = 0.04))
```

print.flood_route *Print a flood route result*

Description

Print a flood route result

Usage

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

Arguments

x	A flood_route object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
disc <- data.frame(date = seq(as.Date("2020-06-01"), by = "day",
                              length.out = 12),
                   Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1))
print(flood_route(disc))
```

`print.flood_runoff` *Print a flood runoff result*

Description

Print a flood runoff result

Usage

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

Arguments

`x`	A `flood_runoff` object.
`...`	Ignored, present for S3 method consistency.

Value

The object `x`, invisibly; prints a compact summary.

Examples

```
set.seed(1)
rain <- data.frame(
  date = seq(as.Date("2020-01-01"), as.Date("2020-12-31"), by = "day"),
  precip_mm = round(rgamma(366, 0.7, scale = 8) *
                    rbinom(366, 1, 0.4), 1)
)
print(flood_runoff(rain, engine = "simple"))
```

print.flood_scenario *Print a flood scenario*

Description

Print a flood scenario

Usage

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

Arguments

x	A flood_scenario object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
set.seed(1)
rain <- data.frame(
  date = seq(as.Date("1990-01-01"), as.Date("2020-12-31"), by = "day"),
  precip_mm = round(rgamma(11323, 0.7, scale = 6), 1)
)
print(flood_scenario(flood_extremes(rain), change_factor = 1.2))
```

print.flood_surrogate *Print a flood surrogate*

Description

Print a flood surrogate

Usage

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

Arguments

x	A flood_surrogate object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
disc <- data.frame(date = seq(as.Date("2020-06-01"), by = "day",
                              length.out = 12),
                   Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1))
print(flood_surrogate(flood_route(disc, area_km2 = 300),
                      n_train = 200, seed = 1))
```

---

print.flood_uncertainty

Print a flood uncertainty result

Description

Print a flood uncertainty result

Usage

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

Arguments

x	A flood_uncertainty object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
disc <- data.frame(date = seq(as.Date("2020-06-01"), by = "day",
                              length.out = 12),
                   Q_mm = c(0, 1, 3, 8, 18, 30, 22, 14, 8, 4, 2, 1))
r <- flood_route(disc, area_km2 = 300)
print(flood_uncertainty(r, observed_depth_m = r$peak_depth_m,
                        n_sim = 1000, seed = 1))
```

print.flood_vulnerability *Print a flood vulnerability result*

Description

Print a flood vulnerability result

Usage

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

Arguments

x	A flood_vulnerability object.
...	Ignored, present for S3 method consistency.

Value

The object x, invisibly; prints a compact summary.

Examples

```
set.seed(1)
print(flood_vulnerability(runif(50, 0, 5), exposure = rpois(50, 40),
                          vulnerability = runif(50)))
```

roughness *Assign Manning's roughness*

Description

Produces a Manning's roughness coefficient (n) for use by the routing stage, by one of three methods. Roughness is the single most sensitive hydraulic parameter, so the function makes the choice explicit and always lets the user override it.

Usage

```
roughness(
  x = NULL,
  method = c("constant", "landcover", "ndvi"),
  value = 0.035,
  table = floodflow_lc_roughness,
  n_min = 0.02,
  n_max = 0.12,
  data = NULL
)
```

Arguments

x	A flood_project, or a data input directly. For method = "constant" the data input is ignored. For method = "landcover" it is a character/factor vector of classes or a land-cover SpatRaster. For method = "ndvi" it is a numeric vector or an NDVI SpatRaster.
method	One of "constant", "landcover" or "ndvi".
value	Manning's n for method = "constant". Default 0.035 (natural channel).
table	Named numeric vector mapping land-cover classes to n for method = "landcover". Defaults to floodflow_lc_roughness.
n_min, n_max	Roughness bounds for method = "ndvi".
data	Optional explicit data input when x is a flood_project; overrides looking in the project. Land-cover classes or NDVI values, as a vector or SpatRaster.

Details

"constant" A single n applied everywhere. Simplest and fully reproducible.

"landcover" Look up n from land-cover classes using a table (the built-in floodflow_lc_roughness by default, or a user-supplied named vector).

"ndvi" Derive n from NDVI with a monotonic empirical function, so remotely-sensed vegetation density sets roughness.

The function operates on plain vectors and, when **terra** is installed, on raster inputs (SpatRaster). Raster handling is optional: if the input is a raster and **terra** is not available, the function stops with an informative message.

Value

If x is a flood_project, the same object with its roughness slot populated by a list of class flood_roughness. Otherwise the flood_roughness list directly, with elements method, n (the resulting roughness: a scalar, numeric vector, or SpatRaster), and summary (min, mean and max of n).

References

Manning, R. (1891). On the flow of water in open channels and pipes. *Transactions of the Institution of Civil Engineers of Ireland*, 20, 161-207.

Chow, V. T. (1959). *Open-Channel Hydraulics*. McGraw-Hill, New York.

See Also

floodflow_lc_roughness for the default lookup table.

Examples

```
# Constant roughness
roughness(method = "constant", value = 0.03)

# From land-cover classes
```

```
cls <- c("urban", "cropland", "forest", "water")
roughness(cls, method = "landcover")$n

# From NDVI values
roughness(c(0, 0.3, 0.6, 0.9), method = "ndvi")$n
```

---

summary.flood_project *Summarise a flood project*

Description

Summarise a flood project

Usage

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

Arguments

object	A flood_project object.
...	Ignored, present for S3 method consistency.

Value

A data.frame with one row per pipeline slot and columns slot and status (either "populated" or "empty"), returned invisibly after printing.

Examples

```
summary(flood_project(name = "Odaw basin"))
```

---

tc_kerby *Time of concentration by the Kerby method (overland flow)*

Description

The Kerby formula for overland (sheet) flow time of concentration, suited to the upstream portion of a catchment before channel flow begins. Often combined with Kirpich channel time in the Kerby-Kirpich approach.

Usage

```
tc_kerby(length_m, slope, retardance = 0.4)
```

Arguments

length_m	Overland flow length in metres.
slope	Overland slope (dimensionless).
retardance	Kerby retardance roughness coefficient (dimensionless); higher for rougher surfaces. Default 0.4 (average grass).

Value

Overland time of concentration in minutes.

References

Kerby, W. S. (1959) Time of concentration for overland flow. Civil Engineering, 29(3), 174.

Examples

```
tc_kerby(100, 0.01, retardance = 0.4)
```

---

tc_kirpich *Time of concentration by the Kirpich method*

Description

The Kirpich (1940) empirical formula for the time of concentration of a channelised catchment, in the SI form with length in metres. Time of concentration is the time for water to travel from the hydraulically most distant point to the outlet, a key control on peak discharge timing.

Usage

```
tc_kirpich(length_m, slope)
```

Arguments

length_m	Flow-path length in metres.
slope	Channel slope (dimensionless, m/m).

Value

Time of concentration in minutes.

References

Kirpich, Z. P. (1940) Time of concentration of small agricultural watersheds. Civil Engineering 10(6), 362.

Examples

```
tc_kirpich(1500, 0.05)
```

Index