Package ‘xaiHydro’

September 24, 2026

Type Package

Title Explainable AI Tools for Hydro-Climate Modelling

Version 0.1.0

Date 2026-09-09

Description Provides a unified workflow for applying Explainable Artificial Intelligence (XAI) methods to hydro-climate predictive models. Functions implement a permutation-based Monte Carlo SHAP estimator (Strumbelj and Kononenko (2014) <doi:10.1007/s10115-013-0679-x>; Lundberg and Lee (2017) <doi:10.48550/arXiv.1705.07874>), a self-contained locally weighted linear surrogate LIME (Ribeiro et al. (2016) <doi:10.1145/2939672.2939778>), and Partial Dependence Plots with Accumulated Local Effects (Friedman (2001) <doi:10.1214/aos/1013203451>; Apley and Zhu (2020) <doi:10.1111/rssb.12377>) with hydrology-specific visualisations and interpretation utilities. Supports any model object compatible with the 'DALEX' explainer interface (Biecek (2018) <doi:10.18637/jss.v097.i01>), including random forests, gradient boosting, and neural networks trained on streamflow, drought indices, flood risk, or evapotranspiration data. Hydrology-standard performance metrics Nash-Sutcliffe Efficiency (NSE, Nash and Sutcliffe (1970) <doi:10.1016/0022-1694(70)90255-6>) and Kling-Gupta Efficiency (KGE, Gupta et al. (2009) <doi:10.1016/j.jhydrol.2009.08.003>) are computed alongside standard regression metrics. Designed to accompany the book chapter: Islam, S., Dheeraj, A., Ali, S., Kaushal, R. and Venkatesh, G. (2026). Explainable Artificial Intelligence for Hydro-Climatic Modelling: Methods, Applications, and Implementation Using the xaiHydro R Package. In Chandniha, S. K. et al. (Eds.), Hydro-Climate Analytics: Remote Sensing, AI and Geospatial Modelling. Springer.

License GPL-3

Encoding UTF-8

NeedsCompilation no

Depends R (>= 4.1.0)

Imports DALEX (>= 2.4.0), ggplot2 (>= 3.4.0), dplyr (>= 1.1.0), tidyr

(>= 1.3.0), patchwork (>= 1.2.0), cli (>= 3.6.0), rlang (>= 1.1.0)

Suggests randomForest (>= 4.7.0), xgboost (>= 1.7.0), ranger (>= 0.16.0), testthat (>= 3.0.0), knitr (>= 1.45), rmarkdown (>= 2.25)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Author Sadikul Islam [aut, cre] (ORCID: <https://orcid.org/0000-0003-2924-7122>),
Shakir Ali [aut],
Rajesh Kaushal [aut]

Maintainer Sadikul Islam <sadikul.islamiasri@gmail.com>

Repository CRAN

Date/Publication 2026-09-24 13:30:02 UTC

Contents

`hydro_breakdown` *Prediction breakdown profile for a single observation*

Description

Decomposes a single model prediction into additive feature contributions using `predict_parts` with the SHAP or Break Down algorithm. Shows how each feature pushes the prediction above or below the baseline (mean prediction).

Usage

```
hydro_breakdown(explainer, new_obs, type = c("shap", "break_down"), B = 25L)
```

Arguments

explainer	A hydro_explainer object created by hydro_explainer.
new_obs	A single-row data.frame of predictor values.
type	Character. Decomposition method: "shap" (default, averages over random orderings) or "break_down" (greedy, fixed order).
B	Integer. Number of random orderings for SHAP-type breakdown. Default: 25.

Value

An object of class c("hydro_breakdown", "predict_parts").

References

Biecek, P., & Burzykowski, T. (2021). *Explanatory Model Analysis*. Chapman and Hall/CRC. https://ema.drwhy.ai/

See Also

plot_breakdown_hydro

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 120, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  bd  <- hydro_breakdown(exp, X[3, , drop = FALSE])
  plot_breakdown_hydro(bd)
}
```

---

hydro_explainer — *Create a hydrology-aware DALEX explainer*

Description

Wraps explain with hydrology-specific defaults, variable labelling, and input validation. The returned explainer object is compatible with all xaiHydro analysis functions and can also be passed directly to any DALEX function.

For automatic model-class detection use the trained model directly; all standard S3 predict methods are auto-detected by DALEX.

Usage

```
hydro_explainer(
  model,
  data,
  y,
  variable = "streamflow",
  units = "",
  label = "hydro_model",
  predict_function = NULL,
  colorize = TRUE,
  ...
)
```

Arguments

`model`	A trained model object. Supported interfaces include `randomForest`, `ranger`, `xgboost`, `gbm`, `caret`, and any object for which a `predict()` method exists.
`data`	A `data.frame` of predictor variables (no response column).
`y`	Numeric vector of observed response values, length equal to `nrow(data)`.
`variable`	Character string naming the hydro-climate target variable. Used for axis labelling. Examples: `"streamflow"`, `"drought_index"`, `"evapotranspiration"`. Default: `"streamflow"`.
`units`	Character string for the response unit label (e.g. `"m3/s"`, `"mm/day"`). Default: `""`.
`label`	Character string for the model label in plots. Default: `"hydro_model"`.
`predict_function`	Optional custom predict function with signature `function(model, newdata)` returning a numeric vector. If `NULL`, `explain` auto-detects the method. Default: `NULL`.
`colorize`	Logical. Colour-coded console output during explainer creation. Default: `TRUE`.
`...`	Additional arguments passed to `explain`.

Value

An object of class `c("hydro_explainer", "explainer")` – a list containing the DALEX explainer plus xaiHydro metadata:

`hydro_variable` Target variable name string.

`hydro_units` Response unit string.

`n_obs` Number of observations in `data`.

`n_features` Number of predictor columns.

References

Biecek, P. (2018). DALEX: Explainers for complex predictive models in R. *Journal of Machine Learning Research*, 19(84), 1–5.

Biecek, P., & Burzykowski, T. (2021). *Explanatory Model Analysis*. Chapman and Hall/CRC. https://ema.drwhy.ai/

See Also

hydro_shap, hydro_lime, hydro_pdp, hydro_importance

Examples

```
df     <- sim_streamflow_data(n = 80, seed = 42)
y      <- df$streamflow
X      <- df[, setdiff(names(df), "streamflow")]
lm_mod <- lm(streamflow ~ ., data = df)
exp    <- hydro_explainer(lm_mod, X, y,
                          variable = "streamflow",
                          units    = "m3/s",
                          colorize = FALSE)
print(exp)
```

---

hydro_importance *Permutation-based variable importance for hydro-climate models*

Description

Computes permutation feature importance using model_parts. Each feature's importance is the mean increase in RMSE when that feature's values are randomly shuffled across B repetitions (Breiman, 2001).

Usage

```
hydro_importance(
  explainer,
  loss_function = DALEX::loss_root_mean_square,
  B = 10L,
  ...
)
```

Arguments

explainer	A hydro_explainer object.
loss_function	A DALEX loss function object or a custom function(observed, predicted). Default: DALEX::loss_root_mean_square (RMSE).
B	Integer. Permutation repetitions. Default: 10.
...	Additional arguments forwarded to model_parts.

Details

The `loss_function` argument must be a DALEX loss function object such as `DALEX::loss_root_mean_square` or a custom `function(observed, predicted)` returning a scalar numeric. Passing a character string (e.g. `"rmse"`) is not supported in DALEX >= 2.4 and will trigger a warning followed by automatic fallback to RMSE.

Value

An object of class `c("hydro_importance", "data.frame")`.

References

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. doi:10.1023/A:1010933404324

See Also

`plot_importance_hydro`

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  imp <- hydro_importance(exp, B = 5L)
  plot_importance_hydro(imp)
}
```

`hydro_lime` *Local LIME explanation for a single observation*

Description

Applies Local Interpretable Model-agnostic Explanations (LIME) to a single observation using a self-contained locally weighted linear surrogate (Ribeiro et al., 2016). A Gaussian radial basis function kernel weights perturbed samples by their proximity to the target observation. The top `n_features` predictors are selected by coefficient magnitude. No external packages beyond the base xaiHydro dependencies are required.

Usage

```
hydro_lime(
  explainer,
  new_obs,
  n_features = 6L,
  n_samples = 1000L,
  kernel_width = 0.75,
  seed = 42L
)
```

Arguments

explainer	A hydro_explainer object created by hydro_explainer.
new_obs	A single-row data.frame of predictor values.
n_features	Integer. Features to retain in the local linear model. Default: 6.
n_samples	Integer. Perturbed samples to generate. Default: 1000.
kernel_width	Numeric. Gaussian kernel bandwidth (in scaled units). Default: 0.75.
seed	Integer random seed. Default: 42.

Value

An object of class c("hydro_lime", "data.frame") with columns feature, effect, feature_value, and r_squared.

References

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. doi:10.1145/2939672.2939778

See Also

plot_lime_hydro, hydro_shap

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  lr  <- hydro_lime(exp, new_obs = X[1, , drop = FALSE])
  plot_lime_hydro(lr)
}
```

hydro_pdp *Compute partial dependence or ALE profiles*

Description

Computes marginal (PDP, Friedman 2001) or conditional (ALE, Apley and Zhu 2020) profiles for one or more predictor variables using `model_profile`. ALE is preferred when predictors are correlated (e.g. precipitation and soil moisture in monsoon hydrology).

Usage

```
hydro_pdp(
  explainer,
  variable = NULL,
  type = c("partial", "accumulated"),
  grid_points = 50L,
  ...
)
```

Arguments

`explainer`	A `hydro_explainer` object.
`variable`	Character vector. Predictor names to profile. Default: all predictors in the explainer dataset.
`type`	Character. `"partial"` (PDP) or `"accumulated"` (ALE). Default: `"partial"`.
`grid_points`	Integer. Grid points per variable. Default: `50`.
`...`	Additional arguments forwarded to `model_profile`.

Value

An object of class `c("hydro_pdp", "list")`.

References

Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B*, 82(4), 1059–1086. doi:10.1111/rssb.12377

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. doi:10.1214/aos/1013203451

See Also

`plot_pdp_hydro`

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  pdp <- hydro_pdp(exp, variable = "precipitation", grid_points = 20L)
  plot_pdp_hydro(pdp)
}
```

`hydro_residuals` *Four-panel residual diagnostics for hydro-climate models*

Description

Computes model residuals directly from the explainer's stored observations and predict function, then produces a four-panel diagnostic figure: (1) residual histogram, (2) residuals vs predicted, (3) observed vs predicted (1:1 line), (4) residuals over observation index (temporal autocorrelation check).

Usage

```
hydro_residuals(explainer, title = NULL)
```

Arguments

`explainer`	A `hydro_explainer` object.
`title`	Character. Panel title. Default: auto-generated.

Value

A `patchwork` panel of four `ggplot2` objects, invisibly. The panel is printed to the active graphics device.

References

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3), 885–900. doi:10.13031/2013.23153

Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I. *Journal of Hydrology*, 10(3), 282–290. doi:10.1016/00221694(70)902556

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 120, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  hydro_residuals(exp)
}
```

---

hydro_shap *Compute SHAP values for a hydro-climate model*

Description

Calculates SHapley Additive exPlanations (SHAP) values using a permutation-based Monte Carlo estimator (Strumbelj and Kononenko, 2014). No additional packages beyond the base xaiHydro dependencies are required. Returns a tidy data frame suitable for `plot_shap_summary` and `plot_shap_waterfall`.

Usage

```
hydro_shap(explainer, nsim = 50L, newdata = NULL, seed = 42L)
```

Arguments

`explainer`	A `hydro_explainer` object created by `hydro_explainer`.
`nsim`	Integer. Monte Carlo replicates for SHAP estimation. `nsim` = `50` is adequate for feature ranking; `nsim` >= `100` recommended for publication figures. Default: `50`.
`newdata`	Optional `data.frame` of observations to explain. If `NULL` the background dataset stored in the explainer is used. Default: `NULL`.
`seed`	Integer random seed for reproducibility. Default: `42`.

Value

An object of class `c("hydro_shap", "data.frame")` with columns `id`, `feature`, `shap_value`, `feature_value`, and `mean_abs_shap`.

References

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. doi:10.48550/arXiv.1705.07874

Strumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3), 647–665. doi:10.1007/s101150130679x

See Also

plot_shap_summary, plot_shap_waterfall

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, variable = "streamflow",
                         colorize = FALSE)
  shv <- hydro_shap(exp, nsim = 10, seed = 42)
  head(shv)
}
```

---

hydro_xai_report — *Generate a full XAI diagnostic report panel*

Description

Runs SHAP attribution, permutation importance, and partial dependence profiles and assembles them into a single multi-panel wrap_plots figure. Optionally saves a high-resolution PNG.

Usage

```
hydro_xai_report(
  explainer,
  nsim = 30L,
  top_n = 6L,
  new_obs = NULL,
  seed = 42L,
  save_path = NULL,
  width = 14,
  height = 10,
  dpi = 300L
)
```

Arguments

explainer	A hydro_explainer object.
nsim	Integer. SHAP simulation replicates. Default: 30.
top_n	Integer. Top features per panel. Default: 6.
new_obs	Optional single-row data.frame for a breakdown panel. Default: NULL.
seed	Integer random seed. Default: 42.

`save_path`	Character file path for PNG, or `NULL`. Default: `NULL`.
`width, height`	Numeric. Figure dimensions in inches. Defaults: `14` and `10`.
`dpi`	Integer. PNG resolution. Default: `300`.

Value

A `patchwork` object, invisibly. Always printed.

References

Islam, S., & Kaushal, R. (2026). Explainable Artificial Intelligence for Hydro-Climatic Modelling: Methods, Applications, and Implementation Using the xaiHydro R Package. In S. K. Chandniha et al. (Eds.), *Hydro-Climate Analytics: Remote Sensing, AI and Geospatial Modelling*. Springer.

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 150, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  hydro_xai_report(exp, nsim = 10, top_n = 4)
}
```

`plot_breakdown_hydro` *Plot prediction breakdown for a single observation*

Description

Plot prediction breakdown for a single observation

Usage

```
plot_breakdown_hydro(bd_result, title = NULL, max_features = 8L)
```

Arguments

`bd_result`	A `hydro_breakdown` object from `hydro_breakdown`.
`title`	Character. Plot title. Default: auto-generated.
`max_features`	Integer. Maximum features shown individually; remainder collapsed to "Other". Default: `8`.

Value

A `ggplot2` object.

References

Biecek, P., & Burzykowski, T. (2021). *Explanatory Model Analysis*. Chapman and Hall/CRC. https://ema.drwhy.ai/

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 120, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  bd  <- hydro_breakdown(exp, X[3, , drop = FALSE])
  plot_breakdown_hydro(bd)
}
```

---

plot_importance_hydro *Plot permutation variable importance*

Description

Plot permutation variable importance

Usage

```
plot_importance_hydro(imp_result, top_n = 10L, title = NULL)
```

Arguments

imp_result	A hydro_importance object from hydro_importance.
top_n	Integer. Features to display. Default: 10.
title	Character. Plot title; auto-generated if NULL.

Value

A ggplot2 object.

References

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. doi:10.1023/A:1010933404324

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  imp <- hydro_importance(exp, B = 5L)
  plot_importance_hydro(imp, top_n = 6L)
}
```

---

plot_lime_hydro — *Plot LIME feature effects for a single observation*

Description

Plot LIME feature effects for a single observation

Usage

```
plot_lime_hydro(lime_result, title = NULL)
```

Arguments

`lime_result`	A `hydro_lime` object from `hydro_lime`.
`title`	Character. Plot title; auto-generated if `NULL`.

Value

A `ggplot2` object.

References

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. doi:10.1145/2939672.2939778

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  lr  <- hydro_lime(exp, X[1, , drop = FALSE])
```

```
  plot_lime_hydro(lr)
}
```

`plot_pdp_hydro` *Plot PDP or ALE profiles*

Description

Plot PDP or ALE profiles

Usage

```
plot_pdp_hydro(pdp_result, variables = NULL, ncol = 2L, title = NULL)
```

Arguments

`pdp_result`	A hydro_pdp object from `hydro_pdp`.
`variables`	Character vector. Subset of variables to plot. Default: all computed variables.
`ncol`	Integer. Number of facet columns. Default: `2`.
`title`	Character. Plot title; auto-generated if `NULL`.

Value

A `ggplot2` object.

References

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. doi:10.1214/aos/1013203451

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  pdp <- hydro_pdp(exp,
                   variable    = c("precipitation", "temperature"),
                   grid_points = 20L)
  plot_pdp_hydro(pdp)
}
```

plot_shap_summary *SHAP beeswarm summary plot*

Description

Produces a beeswarm-style SHAP summary plot showing global feature importance and directionality. Features are ordered by mean |SHAP| (most important at top). Colour encodes the raw feature value (red = high, blue = low).

Usage

```
plot_shap_summary(
  shap_result,
  top_n = 10L,
  point_size = 1.5,
  alpha = 0.6,
  title = NULL
)
```

Arguments

`shap_result`	A `hydro_shap` object from `hydro_shap`.
`top_n`	Integer. Number of top features to display. Default: `10`.
`point_size`	Numeric. Jitter point size. Default: `1.5`.
`alpha`	Numeric. Point transparency (0–1). Default: `0.6`.
`title`	Character. Plot title; auto-generated if `NULL`.

Value

A `ggplot2` object.

References

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67. doi:10.1038/s4225601901389

See Also

`hydro_shap`, `plot_shap_waterfall`

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  shv <- hydro_shap(exp, nsim = 10)
  plot_shap_summary(shv, top_n = 6)
}
```

---

plot_shap_waterfall *SHAP waterfall plot for a single observation*

Description

Shows the additive SHAP contributions for one observation, starting from the global baseline (mean prediction) and accumulating to the final predicted value. Red bars push above baseline; blue bars pull below.

Usage

```
plot_shap_waterfall(shap_result, obs_id = 1L, top_n = 8L, title = NULL)
```

Arguments

`shap_result`	A hydro_shap object from `hydro_shap`.
`obs_id`	Integer. Row index of the observation to plot. Default: `1`.
`top_n`	Integer. Features shown individually; remainder collapsed to "Other features". Default: `8`.
`title`	Character. Plot title; auto-generated if `NULL`.

Value

A `ggplot2` object.

References

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. doi:10.48550/arXiv.1705.07874

See Also

`hydro_shap`, `plot_shap_summary`

Examples

```
if (requireNamespace("randomForest", quietly = TRUE)) {
  df  <- sim_streamflow_data(n = 100, seed = 42)
  y   <- df$streamflow
  X   <- df[, setdiff(names(df), "streamflow")]
  rf  <- randomForest::randomForest(x = X, y = y, ntree = 50)
  exp <- hydro_explainer(rf, X, y, colorize = FALSE)
  shv <- hydro_shap(exp, nsim = 10)
  plot_shap_waterfall(shv, obs_id = 1)
}
```

---

sim_drought_data *Synthetic monthly drought index dataset*

Description

Generates a fully synthetic monthly dataset for demonstrating XAI methods on drought prediction models. The response variable is the Standardised Precipitation-Evapotranspiration Index (SPEI; Vicente-Serrano et al., 2010). **No real observations are included.** Predictor ranges are calibrated to published Indian hydro-climatic summaries (Mishra & Singh, 2010).

Usage

```
sim_drought_data(n = 240L, seed = 2026L)
```

Arguments

`n`	Integer. Number of monthly observations. Default: `240` (20 synthetic years).
`seed`	Integer random seed. Default: `2026`.

Value

A `data.frame` with `n` rows and eight columns:

`precip_anom` Monthly precipitation anomaly (mm).

`temp_anom` Monthly temperature anomaly (degrees C).

`pdsi` Palmer Drought Severity Index (dimensionless; Palmer, 1965). Negative = drought.

`ndvi_anom` NDVI anomaly (dimensionless). Negative = vegetation stress.

`spi_3` SPI at 3-month accumulation (McKee et al., 1993). Negative = below-average precipitation.

`spi_12` SPI at 12-month accumulation (McKee et al., 1993).

`et_deficit` ET deficit (mm). Always non-negative.

`spei` SPEI (Vicente-Serrano et al., 2010). Response variable. Below -1 = moderate drought.

Data status

Fully synthetic. No proprietary or copyrighted records are used.

References

McKee, T. B., Doesken, N. J., & Kleist, J. (1993). The relationship of drought frequency and duration to time scales. *Proceedings of the Eighth Conference on Applied Climatology*, 179–183.

Mishra, A. K., & Singh, V. P. (2010). A review of drought concepts. *Journal of Hydrology*, 391(1–2), 202–216. doi:10.1016/j.jhydrol.2010.07.012

Palmer, W. C. (1965). *Meteorological Drought* (Research Paper No. 45). U.S. Department of Commerce, Weather Bureau.

Vicente-Serrano, S. M., Begueria, S., & Lopez-Moreno, J. I. (2010). A multiscalar drought index sensitive to global warming: The Standardised Precipitation Evapotranspiration Index. *Journal of Climate*, 23(7), 1696–1718. doi:10.1175/2009JCLI2909.1

See Also

`sim_streamflow_data`, `hydro_explainer`

Examples

```
df <- sim_drought_data(n = 120, seed = 42)
head(df)
## Distribution of SPEI
hist(df$spei, breaks = 20, xlab = "SPEI",
     main = "Synthetic SPEI distribution")
abline(v = -1, col = "red", lty = 2)  # moderate drought threshold
```

`sim_streamflow_data` *Synthetic daily streamflow dataset*

Description

Generates a fully synthetic dataset of daily hydro-meteorological predictor variables and a corresponding streamflow response calibrated to representative Indian river-basin conditions.

All values are simulated: no real observations are used. The predictor ranges follow published statistics for central India: precipitation, soil moisture, NDVI and evapotranspiration are drawn from ranges typical of gridded climatology and satellite retrievals for the region. The values are illustrative only and are not taken from, nor calibrated against, any specific product or gauge record.

The streamflow response is the non-linear function

$$Q(t) = 2 + 0.8P + 1.2P_{t-1} + 0.5\bar{P}_3 - 0.3\mathrm{ET} + 10\,\mathrm{SM} - 0.1\,\mathrm{SWE} + 0.5(\mathrm{NDVI} \times P) + \varepsilon$$

where all terms are defined in the **Value** section below.

Usage

```
sim_streamflow_data(n = 365L, seed = 2026L)
```

Arguments

n	Integer. Number of daily observations to generate. Default: 365 (one synthetic year).
seed	Integer random seed for reproducibility. Default: 2026.

Value

A data.frame with n rows and nine columns:

- precipitation Daily rainfall (mm). Seasonal mean 5-8 mm/day during monsoon; calibrated to CHIRPS climatology for central India.
- temperature Mean daily air temperature (degrees C). Seasonal cycle centred at 20 degrees C; calibrated to ERA5 and IMD gridded data.
- evapotranspiration Reference ET (mm/day). Analogous to MODIS MOD16.
- soil_moisture Volumetric root-zone moisture (m3/m3, range 0.05-0.50). Analogous to SMAP L3.
- snow_water_equiv Snow water equivalent (mm). Analogous to MODIS MOD10A1.
- ndvi NDVI (dimensionless, 0-1). Analogous to MODIS MOD13A3.
- lag1_precip Previous-day precipitation (mm).
- lag3_precip 3-day moving-mean precipitation (mm).
- streamflow Simulated streamflow (m3/s). Response variable; always positive.

Data status

Fully synthetic. No proprietary or copyrighted observational data are included. Results are illustrative; they should not be extrapolated to specific real catchments without calibration against observed records.

References

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, 377(1–2), 80–91. doi:10.1016/j.jhydrol.2009.08.003

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3), 885–900. doi:10.13031/2013.23153

Zounemat-Kermani, M., & Kheimi, M. (2026). Explainable artificial intelligence in hydrology: A review. *Water Resources Management*, 40(3), 106. doi:10.1007/s11269025044359

See Also

sim_drought_data, hydro_explainer

Examples

```
df <- sim_streamflow_data(n = 200, seed = 42)
head(df)
summary(df)
## Seasonal streamflow pattern
plot(df$streamflow, type = "l",
     xlab = "Day", ylab = "Streamflow (m3/s)",
     main = "Synthetic daily streamflow")
```

Index