Package: reservoir (via r-universe)

September 15, 2024

Type Package

Title Tools for Analysis, Design, and Operation of Water Supply Storages

Version 1.1.6

Date 2021-06-06

URL https://cran.r-project.org/package=reservoir

Description Measure single-storage water supply system performance using resilience, reliability, and vulnerability metrics; assess storage-yield- reliability relationships; determine no-fail storage with sequent peak analysis; optimize release decisions for water supply, hydropower, and multi-objective reservoirs using deterministic and stochastic dynamic programming; generate inflow replicates using parametric and non-parametric models; evaluate inflow persistence using the Hurst coefficient.

License GPL (>= 2)

LazyData yes

Imports gtools, stats, graphics, moments

RoxygenNote 7.1.1

Repository https://critical-infrastructure-systems-lab.r-universe.dev

RemoteUrl https://github.com/critical-infrastructure-systems-lab/reservoir

RemoteRef HEAD

RemoteSha 8805b5b55fbb2b723ddc472f80bde68ee0466826

Contents

`bootcast` *Develop seasonal streamflow forecasts for an observed period ("hindcasts") using the k nearest neighbour (kNN) bootstrap in forecast mode.*

Description

For seasonal streamflow "hindcasts" based solely on persistence.

Usage

```
bootcast(Q, H, start_yr, end_yr, k, d, sampling_mode, plot)
```

Arguments

`Q`	time series object - seasonal streamflow rate or streamflow totals.
`H`	integer giving the number of seasons in the forecast horizon (e.g., H = 3 means a three month forecast if the frequency of Q is 12).
`start_yr`	start year for the forecast to be generated.
`end_yr`	end_year (if omitted, forecast will be for last year of input)
`k`	integer. The k parameter of the kNN Bootstrap (i.e., number of nearest neighbors from which to sample). If left blank k = n ^ 0.5., where n is the number of years in the input data.
`d`	integer. The d parameter of the kNN Bootstrap (i.e., number of previous time periods to inform the model). If left blank d = 1.

sampling_mode
: used to define period of observed streamflow data (Q) from which to sample the forecasts. "past" will use the period of data prior to the forecast start year. "all" will use all data within Q. "adapt" (the default) updates the sample each year to all years prior to forecast release.

plot
: logical. If TRUE (the default) the function will return a plot of the forecasts against observed flow during the period start_yr : end_yr.

Value

Returns the critical drawdown period of the reservoir, giving the shortest time from full to empty by default.

Examples

```
# prepare three month ahead forecasts of resX$Q_Mm3 for the period 1995 to 2000...
Q <- resX$Q_Mm3
Q_fcast <- bootcast(Q, H = 3, start_yr = 1995, end_yr = 2000)
```

---

critPeriod *Critical (drawdown) period.*

Description

For computing the critical period of a reservoir from its storage time series. The critical period is defined as the length of time for the reservoir to go from full to empty (without spilling in between). Input storage should fill and empty at least once for correct calculation of critical period.

Usage

```
critPeriod(x, report)
```

Arguments

x
: vector or time series object giving the reservoir storage behaviour. The sequence should contain at least one zero values for the critical period to be estimated.

report
: character string giving the critical period to report... "shortest" (default) returns shortest of all critical periods (i.e., the most severe drought); "average" returns the mean of all critical periods; "longest" returns the longest critical drawdown period; "all" returns all critical drawdown periods in the time series.

Value

Returns the critical drawdown period of the reservoir, giving the shortest time from full to empty by default.

Examples

```
storage_behavior <- simRes(Q=resX$Q_Mm3, target = 0.9*mean(resX$Q_Mm3),
                                     capacity=1000, plot=FALSE)$storage
plot(storage_behavior)

## longest critical period
critPeriod(storage_behavior) # or critPeriod(x, "longest")

## shortest critical period
critPeriod(storage_behavior, "shortest")

## average critical period
critPeriod(storage_behavior, "average")

## all critical periods
critPeriod(storage_behavior, "all")
```

---

dirtyreps — *Quick and dirty stochastic generation of seasonal streamflow replicates for a single site.*

Description

Generates seasonal time series using either the kNN Bootstrap (non-parametric) or a numerically-fitted PARMA(1,1) (parametric) model. For the parametric model, the function automatically transforms the seasonal sub-series to normal and deseasonalizes prior to model fitting.

Usage

```
dirtyreps(Q, reps, years, k, d, adjust, parameters, method = "kNNboot")
```

Arguments

`Q`	time series object with seasonal resolution (e.g., frequency = 2, 3, 4, 6 or 12 for monthly data).
`reps`	integer. The number of replicates to be generated. The default is 100.
`years`	integer. The length of each replicate in years. The default is equal to the number of complete years given in Q.
`k`	integer. The k parameter of the kNN Bootstrap (i.e., number of nearest neighbors from which to sample). If left blank k = n ^ 0.5., where n is the number of years in the input data.
`d`	integer. The d parameter of the kNN Bootstrap (i.e., number of previous time periods to inform the model). If left blank d = 1.
`adjust`	logical. If TRUE (the default) the final output time series X will be coerced for 0 <= X <= 1.2*max(Q). Applies only if the PARMA method is used.

`parameters`	logical. If TRUE the output will be given as a list including the replicate samples and relevant model parameters (k and d for kNNboot and phi, theta and standard deviation of residuals for PARMA). The default is FALSE.
`method`	character object giving the method used to generate the data. Defaults to "kNN-boot" - the k Nearest Neighbour Bootstrap. See references for detail on the two methods available.

Value

Returns a multi time series object containing synthetic streamflow replicates.

References

kNN Bootstrap method: Lall, U. and Sharma, A., 1996. A nearest neighbor bootstrap for resampling hydrologic time series. Water Resources Research, 32(3), pp.679-693.

PARMA method: Salas, J.D. and Fernandez, B., 1993. Models for data generation in hydrology: univariate techniques. In Stochastic Hydrology and its Use in Water Resources Systems Simulation and Optimization (pp. 47-73). Springer Netherlands.

Examples

```
Q <- resX$Q_Mm3
replicates <- dirtyreps(Q, reps = 3)
mean(replicates); mean(Q)
sd(replicates); sd(Q)
plot(replicates)
```

---

dp   *Dynamic Programming (Deprecated function; use 'dp_supply' instead)*

Description

Determines the optimal sequence of releases from the reservoir to minimise a penalty cost function based on water supply defict.

Usage

```
dp(
  Q,
  capacity,
  target,
  S_disc = 1000,
  R_disc = 10,
  loss_exp = 2,
  S_initial = 1,
  plot = TRUE,
  rep_rrv = FALSE
)
```

Arguments

Q	vector or time series object. Net inflows to the reservoir.
capacity	numerical. The reservoir storage capacity (must be the same volumetric unit as Q and the target release).
target	numerical. The target release constant.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
loss_exp	numeric. The exponent of the penalty cost function–i.e., Cost[t] <- ((target - release[t]) / target) ^ **loss_exp**). Default value is 2.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.
rep_rrv	logical. If TRUE then reliability, resilience and vulnerability metrics are computed and returned.

Value

Returns the time series of optimal releases and, if requested, the reliability, resilience and vulnerability of the system.

References

Loucks, D.P., van Beek, E., Stedinger, J.R., Dijkman, J.P.M. and Villars, M.T. (2005) Water resources systems planning and management: An introduction to methods, models and applications. Unesco publishing, Paris, France.

See Also

sdp for Stochastic Dynamic Programming

---

dp_hydro *Dynamic Programming for hydropower reservoirs*

Description

Determines the optimal sequence of turbined releases to maximise the total energy produced by the reservoir.

Usage

```
dp_hydro(
  Q,
  capacity,
  capacity_live = capacity,
  surface_area,
  evap,
  installed_cap,
  head,
  qmax,
  max_depth,
  efficiency = 0.9,
  S_disc = 1000,
  R_disc = 10,
  S_initial = 1,
  r2g,
  plot = TRUE
)
```

Arguments

`Q`	time series object. Net inflows to the reservoir. Must be in volumetric units of Mm^3.
`capacity`	numerical. The total reservoir storage capacity (including unusable "dead" storage). Must be in Mm^3.
`capacity_live`	numerical. The volume of usable water in the reservoir ("live capacity" or "active storage"). capacity_live <= capacity. Default capacity_live = capacity. Must be in Mm^3.
`surface_area`	numerical. The reservoir surface area at full capacity. Must be in square kilometers (km^2), or Mm^2.
`evap`	vector or time series object of length Q, or a numerical constant, representing evaporation loss potential from reservoir surface. Varies with level if depth and surface_area parameters are specified. Must be in meters, or kg/m2 * 10 ^ -3.
`installed_cap`	numerical. The hydropower plant electric capacity (MW).
`head`	numerical. The maximum hydraulic head of the hydropower plant (m). Can be omitted and estimated if qmax is supplied.
`qmax`	numerical. The maximum flow into the hydropower plant. Can be omitted and estimated if head is supplied. Must be in volumetric units of Mm^3.
`max_depth`	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
`efficiency`	numerical. The hydropower plant efficiency. Default is 0.9, but, unless user specifies an efficiency, it will be automatically re-estimated if head and qmax are supplied.
`S_disc`	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.

R_disc	integer. Release discretization. Default = 10 divisions.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
r2g	vector. Optional end-state cost-to-go.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

Value

Returns the time series of optimal releases and simulated storage, evaporation, depth, uncontrolled spill, and power generated. Total energy generated is also returned.

See Also

sdp_hydro for Stochastic Dynamic Programming for hydropower reservoirs.

Examples

```
layout(1:4)
dp_hydro(resX$Q_Mm3, resX$cap_Mm3, surface_area = resX$A_km2,
installed_cap = resX$Inst_cap_MW, qmax = mean(resX$Q_Mm3), S_disc = 100)
```

---

dp_multi *Dynamic Programming with multiple objectives (supply, flood control, amenity)*

Description

Determines the optimal sequence of releases from the reservoir to minimise a penalty cost function based on water supply, spill, and water level. For water supply: Cost[t] = ((target - release[t]) / target) ^ loss_exp[1]). For flood control: Cost[t] = (Spill[t] / quantile(Q, spill_targ)) ^ loss_exp[2]. For amenity: Cost[t] = abs(((storage[t] - (vol_targ * capacity)) / (vol_targ * capacity))) ^ loss_exp[3].

Usage

```
dp_multi(
  Q,
  capacity,
  target,
  surface_area,
  max_depth,
  evap,
  R_max = 2 * target,
  spill_targ = 0.95,
  vol_targ = 0.75,
  weights = c(0.7, 0.2, 0.1),
  loss_exp = c(2, 2, 2),
```

```
  S_disc = 1000,
  R_disc = 10,
  S_initial = 1,
  c2g,
  plot = TRUE
)
```

Arguments

Q	vector or time series object. Net inflow totals to the reservoir. Recommended units: Mm^3 (Million cubic meters).
capacity	numerical. The reservoir storage capacity (must be the same volumetric unit as Q and the target release).
target	numerical. The target release constant. Recommended units: Mm^3 (Million cubic meters).
surface_area	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).
max_depth	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
evap	vector or time series object of length Q, or a numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
R_max	numerical. The maximum controlled release, in the same units as target.
spill_targ	numerical. The quantile of the inflow time series used to standardise the "minimise spill" objective.
vol_targ	numerical. The target storage volume constant (as proportion of capacity).
weights	vector of length 3 indicating weighting to be applied to release, spill and water level objectives respectively.
loss_exp	vector of length 3 indicating the exponents on release, spill and water level deviations from target. Default exponents are c(2,2,2).
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
c2g	vector. Optional end-state cost-to-go.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

Value

Returns reservoir simulation output (storage, release, spill), total penalty cost associated with the objective function, and, if requested, the reliability, resilience and vulnerability of the system.

See Also

sdp_multi for Stochastic Dynamic Programming

Examples

```
layout(1:3)
dp_multi(resX$Q_Mm3, cap = resX$cap_Mm3, target = 0.2 * mean(resX$Q_Mm3), S_disc = 100)
```

---

dp_supply *Dynamic Programming for water supply reservoirs*

Description

Determines the optimal sequence of releases from the reservoir to minimise a penalty cost function based on water supply defict.

Usage

```
dp_supply(
  Q,
  capacity,
  target,
  surface_area,
  max_depth,
  evap,
  S_disc = 1000,
  R_disc = 10,
  loss_exp = 2,
  S_initial = 1,
  c2g,
  plot = TRUE,
  rep_rrv = FALSE
)
```

Arguments

Q	vector or time series object. Net inflow totals to the reservoir. Recommended units: Mm^3 (Million cubic meters).
capacity	numerical. The reservoir storage capacity. Recommended units: Mm^3 (Million cubic meters).
target	numerical. The target release constant. Recommended units: Mm^3 (Million cubic meters).
surface_area	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).

max_depth	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
evap	vector or time series object of length Q, or a numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
loss_exp	numeric. The exponent of the penalty cost function–i.e., Cost[t] <- ((target - release[t]) / target) ^ **loss_exp**). Default value is 2.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
c2g	vector. Optional end-state cost-to-go.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.
rep_rrv	logical. If TRUE then reliability, resilience and vulnerability metrics are computed and returned.

Value

Returns the reservoir simulation output (storage, release, spill), total penalty cost associated with the objective function, and, if requested, the reliability, resilience and vulnerability of the system.

See Also

sdp_supply for Stochastic Dynamic Programming for water supply reservoirs

Examples

```
layout(1:3)
dp_supply(resX$Q_Mm3, capacity = resX$cap_Mm3, target = 0.3 * mean(resX$Q_Mm3), S_disc = 100)
```

Hurst *Hurst coefficient estimation*

Description

Hurst coefficient estimation.

Usage

```
Hurst(Q)
```

Arguments

`Q`	vector or annualized time series object. Net inflows or streamflow totals.

Value

Returns an estimate of the Hurst coefficient, H (0.5 < H < 1).

Examples

```
Q_annual <- aggregate(resX$Q_Mm3) #convert monthly to annual data
Hurst(Q_annual)
```

`keystats` *Generate a range of key water supply reservoir variables*

Description

For quuickly analyzing a range of stats relating to inflows, outflows, storage dynamics and performance.

Usage

```
keystats(Q, target, capacity)
```

Arguments

`Q`	time series object representing the release time series of a reservoir.
`target`	numerical. The constant target water delivery.
`capacity`	numerical. The active capacity of the reservoir.

Value

Returns a wide range of statistics relating to the dynamics and performance of the reservoir.

Examples

```
keystats(resX$Q_Mm3, target = 50, capacity = resX$cap_Mm3)
```

reservoir — *reservoir: Tools for Analysis, Design, and Operation of Water Supply Storages*

Description

Measure single reservoir performance using resilience, reliability, and vulnerability metrics; compute storage-yield-reliability relationships; determine no-fail Rippl storage with sequent peak analysis; optimize release decisions for water supply, hydropower, and multi-objective reservoirs using deterministic and stochastic dynamic programming; evaluate inflow characteristics.

Analysis and design functions

The `Rippl` function executes the sequent peak algorithm [Thomas and Burden, 1963] to determine the no-fail storage [Rippl, 1883] for given inflow and release time series. The `storage` function gives the design storage for a specified time-based reliability and yield. Similarly, the `yield` function computes the reliability yield given the storage capacity. The `simRes` function simulates a reservoir under standard operating policy, or using an optimised policy produced by `sdp_supply`. The `rrv` function returns three reliability measures, resilience, and dimensionless vulnerability for given storage, inflow time series, and target release [McMahon et al, 2006]. Users can assume Standard Operating Policy, or can apply the output of `sdp_supply` to determine the RRV metrics under different operating objectives. The `Hurst` function estimates the Hurst coefficient [Hurst, 1951] for an annualized inflow time series, using the method proposed by Pfaff [2008].

Optimization functions

The Dynamic Programming functions find the optimal sequence of releases for a given reservoir. The Stochastic Dynamic Programming functions find the optimal release policy for a given reservoir, based on storage, within-year time period and, optionally, current-period inflow. For single-objective water supply reservoirs, users may specify a loss exponent parameter for supply deficits and then optimize reservoir release decisions to minimize summed penalty costs over the operating horizon. This can be done using `dp_supply` or `sdp_supply`. There is also an option to simulate the output of `sdp_supply` using the `rrv` function to validate the policy under alternative inflows or analyze reservoir performance under different operating objectives. The optimal operating policy for hydropower operations can be found using `dp_hydro` or `sdp_hydro`. The operating target is to maximise total energy output over the duration of the input time series of inflows. The `dp_multi` and `sdp_multi` functions allow users to optimize for three weighted objectives representing water supply deficit, flood control, and amenity.

Storage-depth-area relationships

All reservoir analysis and optimization functions, with the exception of `Rippl`, `storage`, and `yield`, allow the user to account for evaporation losses from the reservoir surface. The package incorporates two storage-depth-area relationships for adjusting the surface area (and therefore evaporation potential) with storage. The simplest is based on the half-pyramid method [Liebe et al, 2005], requiring the user to input the surface area of the reservoir at full capacity via the `surface_area` parameter. A more nuanced relationship [Kaveh et al., 2013] is implemeted if the user also provides the maximum depth of the reservoir at full capacity via the `max_depth` parameter. Users must use the recommended units when implementing evaporation losses.

Stochastic generation of synthetic streamflow replicates

The `dirtyreps` function provides quick and dirty generation of stochastic streamflow replicates (seasonal input data, such as monthly or quarterly, only). Two methods are available: the non-parametric kNN bootstrap [Lall and Sharma, 1996] and the parametric periodic Autoregressive Moving Average (PARMA). The PARMA is fitted for p = 1 and q = 1, or PARMA(1,1). Fitting is done numerically by the least-squares method [Salas and Fernandez, 1993]. When using the PARMA model, users do not need to transform or deseasonalize the input data as this is done automatically within the algorithm. The kNN bootstrap is non-parametric, so no intial data preparation is required here either.

References

Hurst, H.E. (1951) Long-term storage capacity of reservoirs, Transactions of the American Society of Civil Engineers 116, 770-808.

Kaveh, K., H. Hosseinjanzadeh, and K. Hosseini. (2013) A new equation for calculation of reservoir's area-capacity curves, KSCE Journal of Civil Engineering 17(5), 1149-1156.

Liebe, J., N. Van De Giesen, and Marc Andreini. (2005) Estimation of small reservoir storage capacities in a semi-arid environment: A case study in the Upper East Region of Ghana, Physics and Chemistry of the Earth, 30(6), 448-454.

Loucks, D.P., van Beek, E., Stedinger, J.R., Dijkman, J.P.M. and Villars, M.T. (2005) Water resources systems planning and management: An introduction to methods, models and applications. Unesco publishing, Paris, France.

McMahon, T.A., Adeloye, A.J., Zhou, S.L. (2006) Understanding performance measures of reservoirs, Journal of Hydrology 324 (359-382)

Nicholas E. Graham and Konstantine P. Georgakakos, 2010: Toward Understanding the Value of Climate Information for Multiobjective Reservoir Management under Present and Future Climate and Demand Scenarios. J. Appl. Meteor. Climatol., 49, 557-573.

Pfaff, B. (2008) Analysis of integrated and cointegrated time series with R, Springer, New York. [p.68]

Rippl, W. (1883) The capacity of storage reservoirs for water supply, In Proceedings of the Institute of Civil Engineers, 71, 270-278.

Thomas H.A., Burden R.P. (1963) Operations research in water quality management. Harvard Water Resources Group, Cambridge

kNN Bootstrap method: Lall, U. and Sharma, A. (1996). A nearest neighbor bootstrap for resampling hydrologic time series. Water Resources Research, 32(3), pp.679-693.

PARMA method: Salas, J.D. and Fernandez, B. (1993). Models for data generation in hydrology: univariate techniques. In Stochastic Hydrology and its Use in Water Resources Systems Simulation and Optimization (pp. 47-73). Springer Netherlands.

Examples

```
# 1. Express the distribution of Rippl storage for a known inflow process...
layout(1:4)
# a) Assume the inflow process follows a lognormal distribution
# (meanlog = 0, sdlog = 1):
x <- rlnorm(1200)

# b) Convert to a 100-year, monthly time series object beginning Jan 1900
x <- ts(x, start = c(1900, 1), frequency = 12)

# c) Begin reservoir analysis... e.g., compute the Rippl storage
x_Rippl <- Rippl(x, target = mean(x) * 0.9)
no_fail_storage <- x_Rippl$Rippl_storage

# d) Resample x and loop the procedure multiple times to get the
# distribution of no-failure storage for the inflow process assuming
# constant release (R) equal to 90 percent of the mean inflow.
no_fail_storage <- vector("numeric", 100)
for (i in 1:length(no_fail_storage)){
  x <- ts(rlnorm(1200), start = c(1900, 1), frequency = 12)
  no_fail_storage[i] <- Rippl(x, target = mean(x) * 0.9 ,plot = FALSE)$No_fail_storage
}
hist(no_fail_storage)


# 2. Trade off between annual reliability and vulnerability for a given system...
layout(1:1)
# a) Define the system: inflow time series, storage, and target release.
inflow_ts <- resX$Q_Mm3
storage_cap <- resX$cap_Mm3
demand <- 0.3 * mean(resX$Q_Mm3)

# b) define range of loss exponents to control preference of high reliability
# (low loss exponent) or low vulnerability (high loss exponent).
loss_exponents <- c(1.0, 1.5, 2)

# c) set up results table
pareto_results <- data.frame(matrix(ncol = 2, nrow = length(loss_exponents)))
names(pareto_results) <- c("reliability", "vulnerability")
row.names(pareto_results) <- loss_exponents

# d) loop the sdp function through all loss exponents and plot results
for (loss_f in loss_exponents){
 sdp_temp <- sdp_supply(inflow_ts, capacity = storage_cap, target = demand, rep_rrv = TRUE,
 S_disc = 100, R_disc = 10, plot = FALSE, loss_exp = loss_f, Markov = TRUE)
 pareto_results$reliability[which(row.names(pareto_results)==loss_f)] <- sdp_temp$annual_reliability
 pareto_results$vulnerability[which(row.names(pareto_results)==loss_f)] <- sdp_temp$vulnerability
 }
plot (pareto_results$reliability,pareto_results$vulnerability, type = "b", lty = 3)
```

`resX` *Reservoir X inflow time series and reservoir detail*

Description

Reservoir X inflow time series and reservoir detail

Format

list object

Source

http://atlas.gwsp.org

References

Lehner, B., R-Liermann, C., Revenga, C., Vorosmarty, C., Fekete, B., Crouzet, P., Doll, P. et al.: High resolution mapping of the world's reservoirs and dams for sustainable river flow management. Frontiers in Ecology and the Environment. Source: GWSP Digital Water Atlas (2008). Map 81: GRanD Database (V1.0).

`Rippl` *Rippl analysis*

Description

Computes the Rippl no-failure storage for given time series of inflows and releases using the sequent peak algorithm.

Usage

```
Rippl(Q, target, R, double_cycle = FALSE, plot = TRUE)
```

Arguments

`Q`	vector or time series object. Net inflow totals to the reservoir.
`target`	a target release constant in same volumteric units as Q. Can be omitted if R is given.
`R`	a time series or vector of target releases (volumetric). Must be the same length as Q.
`double_cycle`	logical. If TRUE the Q and R time series will be replicated and placed end-to-end to double the simulation. Recommended if the critical period occurs at the end of the sequence.
`plot`	logical. If TRUE (the default) the storage behavior diagram is plotted.

Value

Returns the no-fail storage capacity and corresponding storage behaviour time series.

Examples

```
# define a release vector for a constant release equal to 90 % of the mean inflow
no_fail_storage <- Rippl(resX$Q_Mm3, target = 0.9 * mean(resX$Q_Mm3))$No_fail_storage
```

`rrv` *Reliability, resilience, and vulnerability analysis for water supply reservoirs*

Description

Computes time-based, annual, and volumetric reliability, as well as resilience and dimensionless vulnerability for a single reservoir.

Usage

```
rrv(x, target)
```

Arguments

`x`	time series object representing the release time series of a reservoir.
`target`	numerical constant (or a vector of length = length(x)). If omitted then the trigger constant will be < max(X).

Value

Returns reliability, resilience and vulnerability metrics based on supply deficits.

Examples

```
# Compare reliability, resilience and vulnerability for two operating policies (SOP and SDP).
```

sdp *Stochastic Dynamic Programming (Deprecated function; use 'sdp_supply' instead)*

Description

Derives the optimal release policy based on storage state, inflow class and within-year period.

Usage

```
sdp(
  Q,
  capacity,
  target,
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  loss_exp = 2,
  S_initial = 1,
  plot = TRUE,
  tol = 0.99,
  rep_rrv = FALSE
)
```

Arguments

Q	time series object. Net inflows to the reservoir.
capacity	numerical. The reservoir storage capacity (must be the same volumetric unit as Q and the target release).
target	numerical. The target release constant.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
Q_disc	vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).
loss_exp	numeric. The exponent of the penalty cost function–i.e., Cost[t] <- ((target - release[t]) / target) ^ **loss_exp**). Default value is 2.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.
tol	numerical. The tolerance for policy convergence. The default value is 0.990.
rep_rrv	logical. If TRUE then reliability, resilience and vulnerability metrics are computed and returned.

Value

Returns a list that includes: the optimal policy as an array of release decisions dependent on storage state, month/season, and current-period inflow class; the Bellman cost function based on storage state, month/season, and inflow class; the optimized release and storage time series through the training inflow data; the flow discretization (which is required if the output is to be implemented in the rrv function); and, if requested, the reliability, resilience, and vulnerability of the system under the optimized policy.

References

Loucks, D.P., van Beek, E., Stedinger, J.R., Dijkman, J.P.M. and Villars, M.T. (2005) Water resources systems planning and management: An introduction to methods, models and applications. Unesco publishing, Paris, France.

Gregory R. Warnes, Ben Bolker and Thomas Lumley (2014). gtools: Various R programming tools. R package version 3.4.1. http://CRAN.R-project.org/package=gtools

See Also

`sdp` for deterministic Dynamic Programming

---

sdp_hydro — *Stochastic Dynamic Programming for hydropower reservoirs*

Description

Determines the optimal policy of turbined releases to maximise the total energy produced by the reservoir. The policy can be based on season and storage level, or season, storage level, and current-period inflow.

Usage

```
sdp_hydro(
  Q,
  capacity,
  capacity_live = capacity,
  surface_area,
  max_depth,
  evap,
  installed_cap,
  head,
  qmax,
  efficiency = 0.9,
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  S_initial = 1,
```

```
  plot = TRUE,
  tol = 0.99,
  Markov = FALSE,
  envFlow = FALSE
)
```

Arguments

Q	time series object. Net inflows to the reservoir. Must be in volumetric units of Mm^3.
capacity	numerical. The total reservoir storage capacity (including unusable "dead" storage). Must be in Mm^3.
capacity_live	numerical. The volume of usable water in the reservoir ("live capacity" or "active storage"). capacity_live <= capacity. Default capacity_live = capacity. Must be in Mm^3.
surface_area	numerical. The reservoir surface area at full capacity. Must be in square kilometers (km^2), or Mm^2.
max_depth	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
evap	vector or time series object of length Q, or a numerical constant, representing evaporation loss potential from reservoir surface. Varies with level if depth and surface_area parameters are specified. Must be in meters, or kg/m2 * 10 ^ -3.
installed_cap	numerical. The hydropower plant electric capacity (MW).
head	numerical. The maximum hydraulic head of the hydropower plant (m). Can be omitted if qmax is supplied.
qmax	numerical. The maximum flow into the hydropower plant. Can be omitted and estimated if head is supplied. Must be in volumetric units of Mm^3.
efficiency	numerical. The hydropower plant efficiency. Default is 0.9, but, unless user specifies an efficiency, it will be automatically re-estimated if head and qmax are supplied.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
Q_disc	vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.
tol	numerical. The tolerance for policy convergence. The default value is 0.990.
Markov	logical. If TRUE the current period inflow is used as a hydrological state variable and inflow persistence is incorporated using a first-order, periodic Markov chain. The default is FALSE.
envFlow	logical. If TRUE an environmental flow constraint is applied in simulation.

Value

Returns the optimal release policy, associated Bellman function, simulated storage, release, evaporation, depth, uncontrolled spill, and power generated, and total energy generated.

See Also

dp_hydro for deterministic Dynamic Programming for hydropower reservoirs.

Examples

```
layout(1:4)
sdp_hydro(resX$Q_Mm3, resX$cap_Mm3, surface_area = resX$A_km2,
installed_cap = resX$Inst_cap_MW, qmax = mean(resX$Q_Mm3))
sdp_hydro(resX$Q_Mm3, resX$cap_Mm3, surface_area = resX$A_km2,
installed_cap = resX$Inst_cap_MW, qmax = mean(resX$Q_Mm3), Markov = TRUE)
```

---

sdp_multi — *Stochastic Dynamic Programming with multiple objectives (supply, flood control, amenity)*

Description

Determines the optimal sequence of releases from the reservoir to minimise a penalty cost function based on water supply, spill, and water level. For water supply: Cost[t] = ((target - release[t]) / target) ^ loss_exp[1]). For flood control: Cost[t] = (Spill[t] / quantile(Q, spill_targ)) ^ loss_exp[2]. For amenity: Cost[t] = abs(((storage[t] - (vol_targ * capacity)) / (vol_targ * capacity))) ^ loss_exp[3].

Usage

```
sdp_multi(
  Q,
  capacity,
  target,
  surface_area,
  max_depth,
  evap,
  R_max = 2 * target,
  spill_targ = 0.95,
  vol_targ = 0.75,
  Markov = FALSE,
  weights = c(0.7, 0.2, 0.1),
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  loss_exp = c(2, 2, 2),
  S_initial = 1,
  plot = TRUE,
  tol = 0.99
)
```

Arguments

`Q`	time series object. Net inflow to the reservoir.
`capacity`	numerical. The reservoir storage capacity (must be the same volumetric unit as Q and the target release).
`target`	numerical. The target release constant. Recommended units: Mm^3 (Million cubic meters).
`surface_area`	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).
`max_depth`	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
`evap`	vector or time series object of length Q, or a numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
`R_max`	numerical. The maximum controlled release.
`spill_targ`	numerical. The quantile of the inflow time series used to standardise the "minimise spill" objective.
`vol_targ`	numerical. The target storage volume constant (as proportion of capacity).
`Markov`	logical. If TRUE the current period inflow is used as a hydrological state variable and inflow persistence is incorporated using a first-order, periodic Markov chain. The default is FALSE.
`weights`	vector of length 3 indicating weighting to be applied to release, spill and water level objectives respectively.
`S_disc`	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
`R_disc`	integer. Release discretization. Default = 10 divisions.
`Q_disc`	vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).
`loss_exp`	vector of length 3 indicating the exponents on release, spill and water level deviations from target. Default exponents are c(2,2,2).
`S_initial`	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
`plot`	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.
`tol`	numerical. The tolerance for policy convergence. The default value is 0.990.

Value

Returns a list that includes: the optimal policy as an array of release decisions dependent on storage state, month/season, and current-period inflow class; the Bellman cost function based on storage state, month/season, and inflow class; the optimized release and storage time series through the training inflow data; the flow discretization (which is required if the output is to be implemented in the rrv function); and, if requested, the reliability, resilience, and vulnerability of the system under the optimized policy.

See Also

dp_multi for deterministic Dynamic Programming.

Examples

```
layout(1:3)
sdp_multi(resX$Q_Mm3, cap = resX$cap_Mm3, target = 0.2 * mean(resX$Q_Mm3))
```

---

sdp_supply — *Stochastic Dynamic Programming for water supply reservoirs*

Description

Derives the optimal release policy based on either season and storage level, or season, storage level, and current-period inflow.

Usage

```
sdp_supply(
  Q,
  capacity,
  target,
  surface_area,
  max_depth,
  evap,
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  loss_exp = 2,
  S_initial = 1,
  plot = TRUE,
  tol = 0.99,
  Markov = FALSE,
  rep_rrv = FALSE
)
```

Arguments

Q	vector or time series object. Net inflow totals to the reservoir. Recommended units: Mm^3 (Million cubic meters).
capacity	numerical. The reservoir storage capacity. Recommended units: Mm^3 (Million cubic meters).
target	numerical. The target release constant. Recommended units: Mm^3 (Million cubic meters).
surface_area	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).

max_depth	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
evap	vector or time series object of length Q, or a numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
Q_disc	vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).
loss_exp	numeric. The exponent of the penalty cost function–i.e., Cost[t] <- ((target - release[t]) / target) ^ **loss_exp**). Default value is 2.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.
tol	numerical. The tolerance for policy convergence. The default value is 0.990.
Markov	logical. If TRUE the current period inflow is used as a hydrological state variable and inflow persistence is incorporated using a first-order, periodic Markov chain. The default is FALSE.
rep_rrv	logical. If TRUE then reliability, resilience and vulnerability metrics are computed and returned.

Value

Returns a list that includes: the optimal policy as an array of release decisions dependent on storage state, month/season, and current-period inflow class; the Bellman cost function based on storage state, month/season, and inflow class; the optimized release and storage time series through the training inflow data; the flow discretization (which is required if the output is to be implemented in the rrv function); and, if requested, the reliability, resilience, and vulnerability of the system under the optimized policy.

See Also

dp_supply for deterministic Dynamic Programming for water supply reservoirs

Examples

```
layout(1:3)
sdp_supply(resX$Q_Mm3, capacity = resX$cap_Mm3, target = 0.3 *mean(resX$Q_Mm3))
sdp_supply(resX$Q_Mm3, capacity = resX$cap_Mm3, target = 0.3 *mean(resX$Q_Mm3), Markov = TRUE)
```

simcast_hydro	*Simulate water supply reservoir operations informed by seasonal forecasts.*

Description

For simulating a water supply reservoir operated with rolling horizon, adaptive control (Model Predictive Control).

Usage

```
simcast_hydro(
  Q,
  forecast,
  start_yr,
  capacity,
  capacity_live = capacity,
  surface_area,
  max_depth,
  evap,
  installed_cap,
  head,
  qmax,
  efficiency = 0.9,
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  S_initial = 1,
  plot = TRUE
)
```

Arguments

`Q`	time series object. Observed reservoir inflow totals. Recommended units: Mm^3 (Million cubic meters).
`forecast`	matrix: N * H, where N is the number of forecast-issue periods and H is the forecast horizon (i.e., number of periods) .
`start_yr`	the start year of the forecast. If the 'Q' and 'forecast' parameters have the same start year then leave blank.
`capacity`	numerical. The reservoir storage capacity. Recommended units: Mm^3 (Million cubic meters).
`capacity_live`	numerical. The volume of usable water in the reservoir ("live capacity" or "active storage"). capacity_live <= capacity. Default capacity_live = capacity. Must be in Mm^3.
`surface_area`	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).

max_depth	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.
evap	time series object of equal length to Q, vector of length frequency(Q), or numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
installed_cap	numerical. The hydropower plant electric capacity (MW).
head	numerical. The maximum hydraulic head of the hydropower plant (m). Can be omitted if qmax is supplied.
qmax	numerical. The maximum flow into the hydropower plant. Can be omitted and estimated if head is supplied. Must be in volumetric units of Mm^3.
efficiency	numerical. The hydropower plant efficiency. Default is 0.9, but, unless user specifies an efficiency, it will be automatically re-estimated if head and qmax are supplied.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
Q_disc	vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

Value

Returns a list of reservoir variables as time series for the forecast period. Also returns penalty cost during operating period and cost savings relative to operations without forecasts.

Examples

```
Q <- resX$Q_Mm3
forecastQ <- bootcast(Q, start_yr = 1980, H = 2, plot = FALSE)
layout(1:4)
simQ <- simcast_hydro(Q, forecast = forecastQ, start_yr=1980,
resX$cap_Mm3, surface_area = resX$A_km2, installed_cap = resX$Inst_cap_MW,
head = resX$y_m, S_disc = 200)
```

simcast_multi *Simulate water supply reservoir operations informed by seasonal forecasts.*

Description

For simulating a water supply reservoir operated with rolling horizon, adaptive control (Model Predictive Control).

Usage

```
simcast_multi(
  Q,
  forecast,
  start_yr,
  capacity,
  target,
  surface_area,
  max_depth,
  evap,
  R_max = 2 * target,
  spill_targ = 0.95,
  vol_targ = 0.75,
  weights = c(0.7, 0.2, 0.1),
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  loss_exp = c(2, 2, 2),
  S_initial = 1,
  plot = TRUE
)
```

Arguments

`Q`	time series object. Observed reservoir inflow totals. Recommended units: Mm^3 (Million cubic meters).
`forecast`	matrix: N * H, where N is the number of forecast-issue periods and H is the forecast horizon (i.e., number of periods) .
`start_yr`	the start year of the forecast. If the 'Q' and 'forecast' parameters have the same start year then leave blank.
`capacity`	numerical. The reservoir storage capacity. Recommended units: Mm^3 (Million cubic meters).
`target`	numerical. The target release constant. Recommended units: Mm^3 (Million cubic meters).
`surface_area`	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).

max_depth — numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.

evap — time series object of equal length to Q, vector of length frequency(Q), or numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.

R_max — numerical. The maximum controlled release.

spill_targ — numerical. The quantile of the inflow time series used to standardise the "minimise spill" objective.

vol_targ — numerical. The target storage volume constant (as proportion of capacity).

weights — vector of length 3 indicating weighting to be applied to release, spill and water level objectives respectively.

S_disc — integer. Storage discretization–the number of equally-sized storage states. Default = 1000.

R_disc — integer. Release discretization. Default = 10 divisions.

Q_disc — vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).

loss_exp — vector of length 3 indicating the exponents on release, spill and water level deviations from target. Default exponents are c(2,2,2).

S_initial — numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.

plot — logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

Value

Returns a list of reservoir variables as time series for the forecast period. Also returns penalty cost during operating period and cost savings relative to operations without forecasts.

Examples

```
Q <- resX$Q_Mm3
forecastQ <- bootcast(Q, start_yr = 1980, H = 3, plot = FALSE)
layout(1:3)
simQ <- simcast_multi(Q, resX$cap_Mm3, target = 0.3*mean(Q),
forecast = forecastQ, start_yr=1980, S_disc = 200)
```

simcast_supply *Simulate water supply reservoir operations informed by seasonal forecasts.*

Description

For simulating a water supply reservoir operated with rolling horizon, adaptive control (Model Predictive Control).

Usage

```
simcast_supply(
  Q,
  forecast,
  start_yr,
  capacity,
  target,
  surface_area,
  max_depth,
  evap,
  S_disc = 1000,
  R_disc = 10,
  Q_disc = c(0, 0.2375, 0.475, 0.7125, 0.95, 1),
  loss_exp = 2,
  S_initial = 1,
  plot = TRUE
)
```

Arguments

`Q`	time series object. Observed reservoir inflow totals. Recommended units: Mm^3 (Million cubic meters).
`forecast`	matrix: N * H, where N is the number of forecast-issue periods and H is the forecast horizon (i.e., number of periods) .
`start_yr`	the start year of the forecast. If the 'Q' and 'forecast' parameters have the same start year then leave blank.
`capacity`	numerical. The reservoir storage capacity. Recommended units: Mm^3 (Million cubic meters).
`target`	numerical. The target release constant. Recommended units: Mm^3 (Million cubic meters).
`surface_area`	numerical. The reservoir water surface area at maximum capacity. Recommended units: km^2 (square kilometers).
`max_depth`	numerical. The maximum water depth of the reservoir at maximum capacity. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only. Recommended units: meters.

evap	time series object of equal length to Q, vector of length frequency(Q), or numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
S_disc	integer. Storage discretization–the number of equally-sized storage states. Default = 1000.
R_disc	integer. Release discretization. Default = 10 divisions.
Q_disc	vector. Inflow discretization bounding quantiles. Defaults to five inflow classes bounded by quantile vector c(0.0, 0.2375, 0.4750, 0.7125, 0.95, 1.0).
loss_exp	numeric. The exponent of the penalty cost function–i.e., Cost[t] <- ((target - release[t]) / target) ^ **loss_exp**). Default value is 2.
S_initial	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

Value

Returns a list of reservoir variables as time series for the forecast period. Also returns penalty cost during operating period and cost savings relative to operations without forecasts.

Examples

```
Q <- resX$Q_Mm3
forecastQ <- bootcast(Q, start_yr = 1980, H = 3, plot = FALSE)
layout(1:3)
simQ <- simcast_supply(Q, resX$cap_Mm3, target = 0.3*mean(Q),
forecast = forecastQ, start_yr=1980, S_disc = 200)
```

---

simRes *Simulate a water supply reservoir with specified operating policy.*

Description

Simulates a reservoir for a given inflow time series and assuming Standard Operating Policy (meet target at all times, unless constrained by available water in reservoir plus incoming flows) or an optimised policy deived using sdp_supply.

Usage

```
simRes(
  Q,
  target,
  capacity,
  surface_area,
  max_depth,
```

```
  evap,
  double_cycle = FALSE,
  plot = TRUE,
  S_initial = 1,
  policy
)
```

Arguments

Q	vector or time series object. Net inflow totals to the reservoir. Mm^3 (Million cubic meters).
target	numerical constant, or a time series or vector of the target releases. Must be the same length as Q is given as a vector or time series. Mm^3 (Million cubic meters).
capacity	numerical. The reservoir capacity. Should be same volumetric unit as Q. Mm^3 (Million cubic meters).
surface_area	numerical. The reservoir surface area at full capacity. Must be in square kilometers (km^2), or Mm^2.
max_depth	numerical. The maximum water depth of the reservoir at maximum capacity. Must be in meters. If omitted, the depth-storage-area relationship will be estimated from surface area and capacity only.
evap	vector or time series object of length Q, or a numerical constant. Evaporation from losses from reservoir surface. Varies with level if depth and surface_area parameters are specified. Recommended units: meters, or kg/m2 * 10 ^ -3.
double_cycle	logical. If TRUE the Q and R time series will be replicated and placed end-to-end to double the simulation. Recommended if the critical period occurs at the end of the sequence.
plot	logical. If TRUE (the default) the storage and release time series are plotted.
S_initial	numerical. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
policy	list. The output of the SDP function. If omitted, Standard Operating Policy is assumed.

Value

Returns the no-fail storage capacity and corresponding storage behaviour time series.

Examples

```
# simulate a reservoir assuming standard operating policy, then compare with SDP-derived policy
#trained on historical flows.

# DEFINE RESERVOIR SPECS AND MODEL INPUTS
res_cap <- 1500 #Mm3
targ <- 150 #Mm3
area <- 40 #km2
max_d <- 40 #m
```

```
ev = 0.2 #m
Q_pre1980 <- window(resX$Q_Mm3, end = c(1979, 12), frequency = 12)
Q_post1980 <- window(resX$Q_Mm3, start = c(1980, 1), frequency = 12)

# SIMULATE WITH SOP
layout(1:3)
simSOP <- simRes(Q_post1980, capacity = res_cap, target = targ,
surface_area = area, max_depth = max_d, evap = ev)

# TRAIN SDP POLICY ON HISTORICAL FLOWS
policy_x <- sdp_supply(Q_pre1980, capacity = res_cap, target = targ,
surface_area = area, max_depth = max_d, evap = ev, Markov = TRUE, plot = FALSE, S_disc = 100)

# SIMULATE WITH SDP-DERIVED POLICY
simSDP <- simRes(Q_post1980, capacity = res_cap, target = targ,
surface_area = area, max_depth = max_d, evap = ev, policy = policy_x)
```

---

storage *Storage-Reliability-Yield (SRY) relationships: Storage computation*

Description

Returns the required storage for given inflow time series, yield, and target time-based reliability. Assumes standard operating policy. Storage is computed iteratively using the bi-section method.

Usage

```
storage(
  Q,
  yield,
  reliability,
  demand_profile,
  plot = TRUE,
  S_initial = 1,
  max_iterations = 50,
  double_cycle = FALSE
)
```

Arguments

Q	vector or time series object. Net inflow totals to the reservoir. Recommended units: Mm^3 (Million cubic meters).
yield	the required yield. Must be same volumetric units as Q.
reliability	numerical. The required time-based reliability.
demand_profile	a vector of factors with length = frequency(Q). Represents within-year demand profile. Defaults to constant release if left blank.
plot	logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

`S_initial`	numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.
`max_iterations`	Maximum number of iterations for yield computation.
`double_cycle`	logical. If TRUE the input series will be replicated and placed end-to-end to double the simulation. (Recommended if the critical period occurs at the end of the recorded inflow time series)

Value

Returns the required storage capacity necessary to supply specified yield with specified reliability.

Examples

```
# Determine the required storage for 95 % reliability and yield equal to 80 % of the mean inflow.
layout(1:3)
storage(resX$Q_Mm3 * 20, yield = 0.9 * mean(resX$Q_Mm3), reliability = 0.95)
```

---

yield — *Storage-Reliability-Yield (SRY) relationships: Yield computation*

Description

Returns the yield for given inflow time series, reservoir capacity, and required time-based reliability. Assumes standard operating policy. Yield is computed iteratively using the bi-section method.

Usage

```
yield(
  Q,
  capacity,
  reliability,
  demand_profile,
  plot = TRUE,
  S_initial = 1,
  max_iterations = 50,
  double_cycle = FALSE
)
```

Arguments

`Q`	vector or time series object. Net inflow totals to the reservoir.
`capacity`	numerical. The reservoir storage capacity. Must be in the same volumetric units as Q.
`reliability`	numerical. The required time-based reliability.
`demand_profile`	a vector of factors with length = frequency(Q). Represents within-year demand profile. Defaults to constant release if left blank.

plot
: logical. If TRUE (the default) the storage behavior diagram and release time series are plotted.

S_initial
: numeric. The initial storage as a ratio of capacity (0 <= S_initial <= 1). The default value is 1.

max_iterations
: Maximum number of iterations for yield computation.

double_cycle
: logical. If TRUE the input series will be replicated and placed end-to-end to double the simulation. (Recommended if the critical period occurs at the end of the recorded inflow time series)

Value

Returns yield of a reservoir with specified storage capacity and time-based reliability.

Examples

```
# Compute yield for 0.95 reliability
layout(1:3)
yield_ResX <- yield(resX$Q_Mm3, capacity = 500, reliability = 0.95)
# Compute yield for quarterly time series with seasonal demand profile

quart_ts <- aggregate(resX$Q_Mm3, nfrequency = 4)
yld <- yield(quart_ts,
capacity = 500, reliability = 0.9, demand_profile = c(0.8, 1.2, 1.2, 0.8))
```

Index