Package 'RHMS'

May 28, 2018

Type Package

Title Hydrologic Modelling System for R Users

Version 1.4

Depends R (>= 3.0.0), graphics, stats, pso, Hmisc, network, GGally

Date 2018-05-27

Author Rezgar Arabzadeh ; Shahab Araghinejad

Maintainer Rezgar Arabzadeh <rezgararabzadeh@ut.ac.ir>

Description Hydrologic modelling system is an object oriented tool which enables R users to simulate and analyze hydrologic events. The package proposes functions and methods for construction, simulation, visualization, and calibration of hydrologic systems.

License GPL-2

Imports ggplot2

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-28 18:35:55 UTC

R topics documented:

`RHMS-package` *Hydrologic Modelling System for R Users*

Description

The RHMS package provides tools to R users for simulation of hydrologic events. The packages includes functions and methods for building, simulation, visualization, and calibration of hydrologic systems.

Details

Package:	RHMS
Type:	Package
Version:	1.4
Date:	2018-05-28
License:	GPL-3

the package include three major types of functions as follows:

1- functions for construction and manipulatation of hydrologic features.

a) createBasin. constructor for basin

b) createJunction. constructor for junction

c) createReach. constructor for reach, rivers, and channels

d) createReservoir. constructor for reservoirs

e) createSubbasin. constructor for sub-bains

f) createDiversion. constructor for diversions

g) addObjectToBasin. adds objects form mentioned above constructors to a basin inherited from class of createBasin

2- functions for analysis and simulation of hydrologic events.

a) reachRouting. routes a flood in a channel or river

b) reservoirRouting. routes a flood in a reservoir

c) transform. trnasforms a rainfall event to runoff

d) loss. computes excess rainfall and loss depths

e) baseFlowSeparation. separates baseflow from a given discharge series

e) sim. simulates an objects inherited from class of createBasin

3- functions for tunning, summerizing, and visualization.

a) plot.sim. plots the objects inherited from class of sim

b) plot.createBasin. plots the objects inherited from class of createBasin

c) summary.sim. summerzies the simulation results in the tabular form for every objects existing in the basin

d) tune. calibrates an objects inherited from class of createBasin

Author(s)

Rezgar Arabzadeh ; Shahab Araghinejad

Maintainer: Rezgar Arabzadeh <rezgararabzadeh@ut.ac.ir>

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

sim

addObjectToBasin *adds an object to basin*

Description

adds an object inherited from a hydrologic feature class to a basin instantiated from class of createBasin.

Usage

```
addObjectToBasin(object, basin)
```

Arguments

object	an object to be added to the basin inherited from one of the following classes: 'createReservoir','createReach','createSubbasin','createJunction'
basin	an object inherited from class of createBasin

Value

an object from class of createBasin

Author(s)

Rezgar Arabzadeh

See Also

sim

Examples

```
Junc1<-createJunction(name = "Junc1", downstream=6,
                      label=5, inflow = NA, delayInflow = 1)
R1<-createReach(name="Reach1",label=3,downstream=5,
                routingParams=list(k=3,x=0.2))
R2<-createReach(name="Reach2",label=4,downstream=5,
                routingParams=list(k=3,x=0.2))
R3<-createReach(name="Reach3",label=6,downstream=7,
                routingMethod="muskingumcunge",
                routingParams=list(bedWith=100,
                                   sideSlope=2,
                                   channelSlope=0.01,
                                   manningRoughness=0.05,
                                   riverLength=120))
S1<-createSubbasin(name="Sub1",Area=500,label=1,downstream=8,
                   precipitation=round(sin(seq(0,pi,length.out=24))*20),
                   transformMethod="SCS",lossMethod="SCS",BFSMethod='recession',
```

```
                transformParams=list(Tlag=4),lossParams=list(CN=70),BFSParams=list(k=1.1))
S2<-createSubbasin(name="Sub2",Area=650,label=2,downstream=4,
                   precipitation=round(sin(seq(0,pi,length.out=24))*20),
                   transformMethod="snyder",lossMethod="horton",
                   transformParams=list(Cp=0.17,Ct=1.5,L=140,Lc=30),
                   lossParams=list(f0=5,f1=1,k=1))
D1<-createDiversion(name="Diversion1",label=8,downstream=3,
                    divertTo=7,capacity=80)
ratingCurve1<-data.frame(s=0:100*10,h=100:200)
dischargeCurve1<-data.frame(q=seq(0,5000,length.out=10),h=seq(180,200,length.out=10))
Res1<-createReservoir(name = "Reservoir1", inflow = NA, ratingCurve=ratingCurve1,
                      dischargeCurve=dischargeCurve1, initialStorage=800,
                      capacity=800, delayInflow = 1, label=7, downstream = NA)

basin1<-createBasin(name = "Ghezil_Ozan", simPeriod=200, interval=3600)

basin1<-addObjectToBasin(Junc1, basin1)
basin1<-addObjectToBasin(R1, basin1)
basin1<-addObjectToBasin(R2, basin1)
basin1<-addObjectToBasin(R3, basin1)
basin1<-addObjectToBasin(S1, basin1)
basin1<-addObjectToBasin(S2, basin1)
basin1<-addObjectToBasin(Res1, basin1)
basin1<-addObjectToBasin(D1, basin1)

## Not run: plot(basin1)

object<-sim(basin1)

## Not run: plot(object)

summary(object)
```

`baseFlowSeparation` *Parametric methods for separating baseflow*

Description

This function calculates baseflow for agiven discharge series, `Q`, using a number of method proposed in `BFSMethod`.

Usage

```
baseFlowSeparation(Q,BFSMethod,BFSParams,plot)
```

Arguments

`Q`	a vector: The flow time series (cms)
`BFSMethod`	a string: The method of base flow separation. Available methods: `'nathan'`, `'chapman'`, `'eckhardt'`, `'recession'`

BFSParams	a list including parameters associated with the method coerced in 'BFSMethod'. alpha is in [0, 1] interval required for 'nathan', 'chapman', and 'eckhardt' methods; base flow index,BFI, is in [0, 1] interval required for 'eckhardt'; k in [0, Inf] interval and timeInterval in daily scale needed for 'recession' method
plot	logical: whether to plot the result or not

Value

a list: an object from class of baseFlowSeparation consisting matrix of results available at object$operation.

Author(s)

Rezgar Arabzadeh

References

Chapman, Tom. "A comparison of algorithms for stream flow recession and baseflow separation." Hydrological Processes 13.5 (1999): 701-714.

See Also

baseFlowSeparation

Examples

```
Q<-Q<-(dnorm(seq(-3,4,length.out=200),-.3,1)+dnorm(seq(-1,7,length.out=200),4.5,1)*2)*1200
BFSMethod<-c('nathan','chapman','eckhardt','recession')
BFSParams<-list(alpha=0.6,BFI=0.3,k=1.1,timeInterval=15*60)
baseFlowSeparation(Q,BFSMethod[1],BFSParams)
baseFlowSeparation(Q,BFSMethod[2],BFSParams)
baseFlowSeparation(Q,BFSMethod[3],BFSParams)
baseFlowSeparation(Q,BFSMethod[4],BFSParams)
```

---

baseFlowSeparation.base

base function for class of baseFlowSeparation

Description

base function of methods separating baseflow for a given flow discharge.

Usage

```
## S3 method for class 'base'
baseFlowSeparation(Q,BFSMethod,BFSParams,plot)
```

Arguments

Q	a vector: The flow time series (cms)
BFSMethod	a string: The method of base flow separation. Available methods: 'nathan', 'chapman', 'eckhardt', 'recession'
BFSParams	a list including parameters associated with the method coerced in 'BFSMethod'. alpha is in [0, 1] interval required for 'nathan', 'chapman', and 'eckhardt' methods; BFI in [0, 1] interval required for 'eckhardt'; k in [0, 1] interval and timeInterval in day required for 'recession' method
plot	logical: whether to plot the result or not

Value

a matrix: A matrix of results including computed separated flow for Q series

Author(s)

Rezgar Arabzadeh

See Also

baseFlowSeparation

---

baseFlowSeparation.default

default function for class of baseFlowSeparation

Description

default function of methods separating baseflow for a given flow discharge

Usage

```
## Default S3 method:
baseFlowSeparation(Q,BFSMethod='none'                ,
                                    BFSParams=list(alpha=NULL        ,
                                                   BFI=NULL          ,
                                                   k=NULL            ,
                                                   timeInterval=NULL),
                                  plot=TRUE)
```

Arguments

Q	a vector: The flow time series (cms)
BFSMethod	a string: The method of base flow separation. Available methods: 'nathan', 'chapman', 'eckhardt', 'recession'
BFSParams	a list including parameters associated with the method coerced in 'BFSMethod'. alpha is in [0, 1] interval required for 'nathan', 'chapman', and 'eckhardt' methods; BFI in [0, 1] interval required for 'eckhardt'; k in [0, 1] interval and timeInterval in day required for 'recession' method
plot	logical: whether to plot the result or not

Value

a list: an object from class of baseFlowSeparation consisting matrix of results available at object$operation.

Author(s)

Rezgar Arabzadeh

See Also

createSubbasin

---

createBasin *creates a basin*

Description

instantiates an object from class of createBasin

Usage

```
createBasin(name, simPeriod, interval)
```

Arguments

name	a string: a name for the basin
simPeriod	the simulation period in seconds
interval	number of simulation time steps

Value

a list: an object from class of creatBasin

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

---

createBasin.base *base function for class of* createBasin

Description

instantiates an object from class of createBasin

Usage

```
## S3 method for class 'base'
createBasin(name, simPeriod, interval)
```

Arguments

name	a string: a name for the basin
simPeriod	the simulation period in seconds
interval	number of simulation time steps

Value

a list: an object from class of creatBasin

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

---

createBasin.default *default function for class of* createBasin

---

Description

instantiates an object from class of createBasin

Usage

```
## Default S3 method:
createBasin(name = "Untittled", simPeriod, interval)
```

Arguments

name	a string: a name for the basin
simPeriod	the simulation period in seconds
interval	number of simulation time steps

Value

a list: an object from class of creatBasin

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

---

createDiversion *creates a diversion object*

Description

instantiates an object from class of createDiversion

Usage

```
createDiversion(name,label,downstream,divertTo,capacity)
```

Arguments

name	a string: the name of diversion to be instantiated
downstream	an integer: the code of downstream object
label	an integer: a unique number, as label
divertTo	an integer: the code of an object at the diversion outlet
capacity	diversion capacity (cms)

Value

a list: an object from class of list instantiated by createDiversion

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createDiversion.base *base function for class of* createDiversion

Description

instantiates an object from class of createDiversion

Usage

```
## S3 method for class 'base'
createDiversion(name,label,downstream,divertTo,capacity)
```

Arguments

name	a string: the name of diversion to be instantiated
downstream	an integer: the code of downstream object
label	an integer: a unique number, as label
divertTo	an integer: the code of an object at the diversion outlet
capacity	diversion capacity (cms)

Value

a list: an object from class of list instantiated by createDiversion

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createDiversion.default *default function for class of* createDiversion

Description

instantiates an object from class of createDiversion

Usage

```
## Default S3 method:
createDiversion(name="Unttitled",label,downstream=NA,divertTo,capacity)
```

Arguments

name	a string: the name of diversion to be instantiated
downstream	an integer: the code of downstream object
label	an integer: a unique number, as label
divertTo	an integer: the code of an object at the diversion outlet
capacity	diversion capacity (cms)

Value

a list: an object from class of list instantiated by createDiversion

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

---

createJunction *creates a junction object*

Description

instantiates an object from class of createJunction

Usage

```
createJunction(name, downstream,label,
               inflow, delayInflow)
```

Arguments

name	a string: the name of junction to be instantiated
downstream	an integer: the downstream object code
label	an integer: a unique number, as label
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series

Value

a list: an object from class createJunction

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

---

createJunction.base *base function for class of* createJunction

---

Description

instantiates an object from class of createJunction

Usage

```
## S3 method for class 'base'
createJunction(name , downstream,
                              label, inflow , delayInflow )
```

Arguments

name	a string: the name of junction to be instantiated
downstream	an integer: the downstream object code
label	an integer: a unique number, as label
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series

Value

a list: an object from class of createJunction

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createJunction.default
default function for class of createJunction

Description

instantiates an object from class of createJunction

Usage

```
## Default S3 method:
createJunction(name = "Unttitled", downstream=NA,
                                   label, inflow = NA, delayInflow = 1)
```

Arguments

name	a string: the name of junction to be instantiated
downstream	an integer: the code of downstream object
label	an integer: a unique number, as label
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series

Value

a list: an object from class of createJunction

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createReach *creates a reach object*

Description

instantiates an object from class of createReach

Usage

```
createReach(name,routingMethod,inflow,routingParams,delayInflow,label,downstream)
```

Arguments

name	a string: the name of reach to be instantiated
routingMethod	a string: the method of channel routing. available types: "muskingum", and "muskingumcunge"
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
routingParams	a list : parameters associated to the routingMethod: k and x for "muskingum", bedWith (m), sideSlope (m/m), channelSlope (m/m), manningRoughness, riverLength (Km) for "muskingumcunge"
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object

Value

a list: an object from class of createReach

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createReach.base *base function for class of* createReach

Description

instantiates an object from class of createReach

Usage

```
## S3 method for class 'base'
createReach(name,routingMethod,inflow,
                            routingParams,
                            delayInflow,label,downstream)
```

Arguments

name	a string: the name of reach to be instantiated
routingMethod	a string: the method of channel routing. available types: "muskingum", and "muskingumcunge"
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
routingParams	a list : parameters associated to the routingMethod: k and x for "muskingum", bedWith (m), sideSlope (m/m), channelSlope (m/m), manningRoughness, riverLength (Km) for "muskingumcunge"
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object

Value

a list: an object from class of list instantiated by createReach

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createReach.default *default function for class of* createReach

Description

instantiates an object from class of createReach

Usage

```
## Default S3 method:
createReach(name="Unttitled",routingMethod="muskingum",inflow=NA,
                              routingParams=list(k=3,x=0.2,bedWith=NULL,
                                                 sideSlope=2,channelSlope=NULL,
                                           manningRoughness=0.025,riverLength=NULL),
                              delayInflow=1,label,downstream=NA)
```

Arguments

name	a string: the name of reach to be instantiated
routingMethod	a string: the method of channel routing. available types: "muskingum", and "muskingumcunge"
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
routingParams	a list : parameters associated to the routingMethod: k and x for "muskingum", bedWith (m), sideSlope (m/m), channelSlope (m/m), manningRoughness, riverLength (Km) for "muskingumcunge"
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object

Value

a list: an object from class of createReach

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createReservoir *creates a reservoir object*

Description

instantiates an object from class of createReservoir

Usage

```
createReservoir(name , inflow , ratingCurve,
                dischargeCurve, initialStorage, capacity,
                delayInflow , label, downstream )
```

Arguments

name	a string: the name of reservoir to be instantiated
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
ratingCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents equivalant volume to the height at first collumn (MCM)
dischargeCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents equivalant discharge rate to the height at first collumn (cms)
initialStorage	the initial storage of reservoir at the first time step of simulation (MCM)
capacity	the maximum volume of reservoir capacity (MCM)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object

Value

a list: an object from class of createReservoir

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createReservoir.base *base function for class of* createReservoir

Description

instantiates an object from class of createReservoir

Usage

```
## S3 method for class 'base'
createReservoir(name , inflow , ratingCurve,
                                dischargeCurve, initialStorage, capacity,
                                delayInflow , label, downstream )
```

Arguments

name	a string: the name of reservoir to be instantiated
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
ratingCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents equivalant volume to the height at first collumn (MCM)
dischargeCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents equivalant discharge rate to the height at first collumn (cms)
initialStorage	the initial storage of reservoir at the first time step of simulation (MCM)
capacity	the maximum volume of reservoir capacity (MCM)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object

Value

a list: an object from class of createReservoir

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createReservoir.default

default function for class of createReservoir

Description

instantiates an object from class of createReservoir

Usage

```
## Default S3 method:
createReservoir(name = "Unttitled", inflow = NA, ratingCurve,
                                    dischargeCurve, initialStorage, capacity,
                                    delayInflow = 1, label, downstream = NA)
```

Arguments

name	a string: the name of reservoir to be instantiated
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
ratingCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents equivalant volume to the height at first collumn (MCM)
dischargeCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents equivalant discharge rate to the height at first collumn (cms)
initialStorage	the initial storage of reservoir at the first time step of simulation (MCM)
capacity	the maximum volume of reservoir capacity (MCM)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object

Value

a list: an object from class of createReservoir

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

createSubbasin *creates a sub-basin object*

Description

instantiates an object from class of createSubbasin

Usage

```
createSubbasin(name,precipitation,
     inflow,Area,delayInflow,label,downstream,
     transformMethod,lossMethod,BFSMethod,UH,
     transformParams,lossParams,BFSParams)
```

Arguments

name	a string: the name of sub-basin to be instantiated
precipitation	a vector : a time series of precipitation hytograph (mm)
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
Area	the area of basin (Km^2)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object
transformMethod	a string: the type of transformation method. Available types: "SCS", "snyder", and "user" for user defined unit hydrograph
lossMethod	a string: the type of loss method. Available types: "SCS" and "horton"
BFSMethod	a string: The method of base flow separation. Available methods: 'nathan', 'chapman', 'eckhardt', 'recession'
UH	a data.frame: including the ordinates of user UH. the HU first collumn indicates time (Hr) and second collumn include flow rates (cms)
transformParams	a list: list of parameters associated to the selcted type of transformMethod:Tlag for "SCS" and Ct, Cp, L, and Lc other for "snyder"
lossParams	a list: list of parameters associated to the selcted type of lossMethod: CN for "SCS" and f0, f1, k other for "horton"
BFSParams	a list including parameters associated with the method coerced in 'BFSMethod'. alpha is in [0, 1] interval required for 'nathan', 'chapman', and 'eckhardt' methods; BFI in [0, 1] interval required for 'eckhardt'; k in [0, 1] interval and timeInterval in day required for 'recession' method

Value

a list: an object from class of `createSubbasin`

Author(s)

Rezgar Arabzadeh

See Also

`addObjectToBasin`

`createSubbasin.base` *base function for class of* `createSubbasin`

Description

instantiates an object from class of `createSubbasin`

Usage

```
## S3 method for class 'base'
createSubbasin(name,precipitation,
      inflow,Area,delayInflow,label,downstream,
      transformMethod,lossMethod,BFSMethod,UH,
      transformParams,lossParams,BFSParams)
```

Arguments

`name`	a string: the name of sub-basin to be instantiated
`precipitation`	a vector : a time series of precipitation hytograph (mm)
`inflow`	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
`Area`	the area of basin (Km^2)
`delayInflow`	an integer (optional): presenting the time steps to delay direct inflow time series
`label`	an integer: a unique number, as label
`downstream`	an integer: the code of downstream object
`transformMethod`	a string: the type of transformation method. Available types: `"SCS"`, `"snyder"`, and `"user"` for user defined unit hydrograph
`lossMethod`	a string: the type of loss method. Available types: `"SCS"` and `"horton"`
`BFSMethod`	a string: The method of base flow separation. Available methods: `'nathan'`, `'chapman'`, `'eckhardt'`, `'recession'`
`UH`	a data.frame: including the ordinates of user UH. the HU first collumn indicates time (Hr) and second collumn include flow rates (cms)

transformParams
: a list: list of parameters associated to the selcted type of transformMethod:Tlag for "SCS" and Ct, Cp, L, and Lc other for "snyder"

lossParams
: a list: list of parameters associated to the selcted type of lossMethod: CN for "SCS" and f0, f1, k other for "horton"

BFSParams
: a list including parameters associated with the method coerced in 'BFSMethod'. alpha is in [0, 1] interval required for 'nathan', 'chapman', and 'eckhardt' methods; BFI in [0, 1] interval required for 'eckhardt'; k in [0, 1] interval and timeInterval in day required for 'recession' method

Value

a list: a list features for the constructed sub-basin

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

---

createSubbasin.default

default function for class of createSubbasin

Description

instantiates an object from class of createSubbasin

Usage

```
## Default S3 method:
createSubbasin(name="Unttitled",
       precipitation,inflow=NA,Area,delayInflow=1,
       label,downstream=NA,
       transformMethod="SCS",lossMethod="none",BFSMethod='none',
       UH=NA,
       transformParams=list(Tlag=NULL,Cp=NULL,Ct=NULL,L=NULL,Lc=NULL),
       lossParams=list(CN=NULL,f0=NULL,f1=NULL,k=NULL),
       BFSParams=list(alpha=NULL,BFI=NULL,k=NULL))
```

Arguments

name	a string: the name of sub-basin to be instantiated
precipitation	a vector : a time series of precipitation hytograph (mm)
inflow	a vector (optional): a time series of direct inflow rather than flows comming from upstream (cms)
Area	the area of basin (Km^2)
delayInflow	an integer (optional): presenting the time steps to delay direct inflow time series
label	an integer: a unique number, as label
downstream	an integer: the code of downstream object
transformMethod	a string: the type of transformation method. Available types: "SCS", "snyder", and "user" for user defined unit hydrograph
lossMethod	a string: the type of loss method. Available types: "SCS" and "horton"
BFSParams	a list including parameters associated with the method coerced in 'BFSMethod'. alpha is in [0, 1] interval required for 'nathan', 'chapman', and 'eckhardt' methods; BFI in [0, 1] interval required for 'eckhardt'; k in [0, 1] interval and timeInterval in day required for 'recession' method
BFSMethod	a string: The method of base flow separation. Available methods: 'nathan', 'chapman', 'eckhardt', 'recession'
UH	a data.frame: including the ordinates of user UH. the HU first collumn indicates time (Hr) and second collumn include flow rates (cms)
transformParams	a list: list of parameters associated to the selcted type of transformMethod:Tlag for "SCS" and Ct, Cp, L, and Lc other for "snyder"
lossParams	a list: list of parameters associated to the selcted type of lossMethod: CN for "SCS" and f0, f1, k other for "horton"

Value

a list: an object from class of createSubbasin

Author(s)

Rezgar Arabzadeh

See Also

addObjectToBasin

loss *Excess rainfall computation*

Description

this function provides parametric methods (e.g. "horton" and "SCS") to compute loss and direct runoff depths

Usage

```
loss(precipitation,lossParams,
simulation,lossMethod)
```

Arguments

precipitation	a vector of precipitation time series(mm)
lossParams	a list: list of parameters associated to the selcted type of lossMethod: CN for "SCS" and f0, f1, k for "horton"
simulation	a vector of simulation interval (in second) and time steps (an integer)
lossMethod	a string including the type of lossMethod: "SCS" and "horton"

Value

a dataframe: including precipitation, loss, and exess rainfall depth

Author(s)

Rezgar Arabzadeh

See Also

transform

Examples

```
precipitation<-sin(seq(0.1,pi-0.1,length.out=10))*30
lossParams<-list(f0=20,f1=5,k=2,CN=65)
simulation<-c(interval=3600,period=NA)
lossMethod<-c("horton","SCS")
Horton_loss<-loss(precipitation,lossParams,simulation,lossMethod[1])
SCS_loss<-loss(precipitation,lossParams,simulation,lossMethod[2])
```

loss.base *base function for class of* reachRouting

Description

this function provides parametric methods (e.g. "horton" and "SCS") to compute loss and direct runoff depths

Usage

```
## S3 method for class 'base'
loss(precipitation,lossParams,
simulation,lossMethod)
```

Arguments

precipitation	a vector of precipitation time series(mm)
lossParams	a list: list of parameters associated to the selcted type of lossMethod: CN for "SCS" and f0, f1, k for "horton"
simulation	a vector of simulation interval (in second) and time steps (an integer)
lossMethod	a string including the type of lossMethod: "SCS" and "horton"

Value

a dataframe: including precipitation, loss, and exess rainfall depth

Author(s)

Rezgar Arabzadeh

See Also

loss

loss.default *default function for class of* loss

Description

this function provides parametric methods (e.g. "horton" and "SCS") to compute loss and direct runoff depths

Usage

```
## Default S3 method:
loss(precipitation,lossParams=list(f0=NULL,f1=NULL,k=NULL,CN=NULL),
                        simulation=c(interval=3600,period=NA),lossMethod)
```

Arguments

precipitation	a vector of precipitation time series(mm)
lossParams	a list: list of parameters associated to the selcted type of lossMethod: CN for "SCS" and f0, f1, k for "horton"
simulation	a vector of simulation interval (in second) and time steps (an integer)
lossMethod	a string including the type of lossMethod: "SCS" and "horton"

Value

a dataframe: including precipitation, loss, and exess rainfall depth

Author(s)

Rezgar Arabzadeh

See Also

loss

---

plot.createBasin *plots basin layout*

Description

plot method for objects inherited from class of createBasin

Usage

```
## S3 method for class 'createBasin'
plot(x,...)
```

Arguments

x	an object from class of createBasin
...	other objects that can be passed to plot function

Author(s)

Rezgar Arabzadeh

See Also

sim

plot.sim *plot method for an RHMS object*

Description

plot method for objects inherited from class of sim

Usage

```
## S3 method for class 'sim'
plot(x,...)
```

Arguments

x	an object from class of sim
...	other objects that can be passed to plot function

Author(s)

Rezgar Arabzadeh

See Also

sim

reachRouting *channel routing computation*

Description

function for flood routing using parameteric Muskingum and muskingum-cunge techniques.

Usage

```
reachRouting(inflow,routingMethod,
             routingParams,simulation)
```

Arguments

inflow	a vector of runoff time series (cms) representing a runoff event generated by excess rainfall computed by loss methods or an object inherited from any of the following classes :transform ; reachRouting ; reservoirRouting.
routingMethod	a string: the type of channel routing method: "muskingum" or "muskingumcunge"
routingParams	a list : parameters associated to the routingMethod: k and x for "muskingum", bedWith (m), sideSlope (m/m), channelSlope (m/m), manningRoughness, riverLength (Km) for "muskingumcunge"
simulation	a vectors: number of simulation time steps and the simulation period in seconds.

Value

a data.frame: including inflow time series routing resaults and simulation details

Author(s)

Rezgar Arabzadeh

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

reservoirRouting

Examples

```
inflow<-c(100,500,1500,2500,5000,11000,22000,28000,28500,26000,
          22000,17500,14000,10000,7000,4500,2500,1500,1000,500,100)
routingMethod<-c("muskingum","muskingumcunge")
routingParams<-list(k=3,x=0.2,bedWith=50,sideSlope=2,channelSlope=0.0001,
                    manningRoughness=0.01,riverLength=100)
simulation<-c(interval=3600*1,period=100)

reachRouting(inflow,routingMethod[1],routingParams,simulation)
reachRouting(inflow,routingMethod[2],routingParams,simulation)
```

---

reachRouting.base *base function for class of* reachRouting

Description

function for flood routing using parameteric Muskingum and muskingum-cunge techniques.

Usage

```
## S3 method for class 'base'
reachRouting(inflow,routingMethod,
             routingParams,simulation)
```

Arguments

inflow	a vector of runoff time series (cms) representing a runoff event generated by excess rainfall computed by loss methods or an object inherited from any of the following classes :transform ; reachRouting ; reservoirRouting.
routingMethod	a string: the type of channel routing method: "muskingum" or "muskingumcunge"

routingParams	a list : parameters associated to the routingMethod: k and x for "muskingum", bedWith (m), sideSlope (m/m), channelSlope (m/m), manningRoughness, riverLength (Km) for "muskingumcunge"
simulation	a vectors: number of simulation time steps and the simulation period in seconds.

Value

a data.frame: including inflow time series routing resaults and simulation details

Author(s)

Rezgar Arabzadeh

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

reachRouting

---

reachRouting.default *default function for class of* reachRouting

---

Description

function for flood routing using parameteric Muskingum and muskingum-cunge techniques.

Usage

```
## Default S3 method:
reachRouting(inflow,routingMethod="muskingum",
             routingParams=list(k=3,
                                x=0.2,
                                bedWith=NULL,
                                sideSlope=2,
                                channelSlope=NULL,
                                manningRoughness=0.025,
                                riverLength=NULL),
             simulation=c(interval=3600*1,period=NA))
```

Arguments

inflow	a vector of runoff time series (cms) representing a runoff event generated by excess rainfall computed by loss methods or an object inherited from any of the following classes :transform ; reachRouting ; reservoirRouting.
routingMethod	a string: the type of channel routing method: "muskingum" or "muskingumcunge"
routingParams	a list : parameters associated to the routingMethod: k and x for "muskingum", bedWith (m), sideSlope (m/m), channelSlope (m/m), manningRoughness, riverLength (Km) for "muskingumcunge"
simulation	a vectors: number of simulation time steps and the simulation period in seconds.

Value

a list: including inflow time series routing resaults and simulation details

Author(s)

Rezgar Arabzadeh

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

reachRouting

---

reservoirRouting *reservoir routing*

Description

function for routing flood through a reservoir using classical Muskingum technique

Usage

```
reservoirRouting(inflow, ratingCurve, dischargeCurve,
                 initialStorage, capacity,
                 simulation)
```

Arguments

inflow	a vector of runoff time series (cms) representing a runoff event generated by excess rainfall computed by loss methods or an object inherited from any of the following classes :transform ; reachRouting ; reservoirRouting.
ratingCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents volume corresponding to the height at first collumn (MCM)

dischargeCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents discharge rate corresponding to the height at first collumn (cms)
initialStorage	the initial storage of reservoir at the first time step of simulation (MCM)
capacity	the maximum volume of reservoir capacity (MCM)
simulation	a vectors: number of simulation time steps and the simulation period (seconds)

Value

a data.frame: including inflow time series and routing resaults

Author(s)

Rezgar Arabzadeh

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

reachRouting

Examples

```
inflow1<-sin(seq(0,pi,length.out=50))*1000
ratingCurve1<-data.frame(s=0:49*2,h=100:149)
dischargeCurve1<-data.frame(q=0:9*250,h=140:149)
reservoir_sim<-reservoirRouting(inflow=inflow1, ratingCurve=ratingCurve1,
                                dischargeCurve=dischargeCurve1,
                                initialStorage=80, capacity=80,
                                simulation = c(interval = 3600 * 1, period = 60))
plot(reservoir_sim$operation[,2],typ="o",
     ylab="Discharge rate (cms)",
     xlab="Time step")
lines(reservoir_sim$operation[,4])
```

---

reservoirRouting.base *base function for class of* reservoirRouting

---

Description

function for routing flood through a reservoir using classical Muskingum technique

Usage

```
## S3 method for class 'base'
reservoirRouting(inflow, ratingCurve, dischargeCurve,
                 initialStorage, capacity,
                 simulation)
```

Arguments

inflow	a vector of runoff time series (cms) representing a runoff event generated by excess rainfall computed by loss methods or an object inherited from any of the following classes :transform ; reachRouting ; reservoirRouting.
ratingCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents volume corresponding to the height at first collumn (MCM)
dischargeCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents discharge rate corresponding to the height at first collumn (cms)
initialStorage	the initial storage of reservoir at the first time step of simulation (MCM)
capacity	the maximum volume of reservoir capacity (MCM)
simulation	a vectors: number of simulation time steps and the simulation period (seconds)

Value

a data.frame: including inflow time series and routing resaults

Author(s)

Rezgar Arabzadeh

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

reservoirRouting

---

reservoirRouting.default

default function for class of reservoirRouting

Description

function for routing flood through a reservoir using classical Muskingum technique

Usage

```
## Default S3 method:
reservoirRouting(inflow, ratingCurve, dischargeCurve,
                 initialStorage, capacity,
                 simulation = c(interval = 3600 * 1, period = NA))
```

Arguments

inflow	a vector of runoff time series (cms) representing a runoff event generated by excess rainfall computed by loss methods or an object inherited from any of the following classes :transform ; reachRouting ; reservoirRouting.
ratingCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents volume corresponding to the height at first collumn (MCM)
dischargeCurve	a data frame: a data frame at which its first collumn includes height (masl) and second collums presents discharge rate corresponding to the height at first collumn (cms)
initialStorage	the initial storage of reservoir at the first time step of simulation (MCM)
capacity	the maximum volume of reservoir capacity (MCM)
simulation	a vectors: number of simulation time steps and the simulation period (seconds)

Value

a data.frame: including inflow time series and routing resaults

Author(s)

Rezgar Arabzadeh

References

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

See Also

reservoirRouting

---

sim *RHMS simulation function*

Description

simulates an object inherited form class of createBasin

Usage

```
sim(object)
```

Arguments

object	an object from class of createBasin

Value

a list: the same as objects inherited from class of createBasin

Author(s)

Rezgar Arabzadeh

References

NRCS, U. (1986). Urban hydrology for small watersheds-Technical Release 55 (TR55). Water Resources Learning Center. Washington DC.

Chow, V. T., Maidment, D. R., & Mays, L. W. (1988). Applied hydrology.

Examples

```
data(Zaab)

Zaab_sim<-sim(Zaab)

## Not run: plot(Zaab_sim)

summary(Zaab_sim)
```

---

sim.base *base function for class of* sim

Description

simulates an object inherited form class of createBasin

Usage

```
## S3 method for class 'base'
sim(object)
```

Arguments

object	an object from class of createBasin

Author(s)

Rezgar Arabzadeh

See Also

sim

---

sim.default *default function for class of* sim

Description

simulates an object inherited form class of createBasin

Usage

```
## Default S3 method:
sim(object)
```

Arguments

object	an object from class of createBasin

Author(s)

Rezgar Arabzadeh

See Also

sim

---

summary.sim *summary method for RHMS objects*

Description

summary method for objects inherited from class of sim

Usage

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

Arguments

object	an object from class of sim
...	other objects that can be passed to summary function

Value

a matrix: including inflow and outflow volumes and peaks rates respectively

Author(s)

Rezgar Arabzadeh

See Also

sim

---

transform *Transforms a rainfall event to runoff*

Description

This function transforms an excess rainfall event to a direct runoff hydorgraph.

Usage

```
transform(rainfall,
          transformParams,
          Area,simulation,UH,
          transformMethod)
```

Arguments

rainfall	an object inherited from loss function
transformParams	a list: list of parameters associated to the selcted type of transformMethod:Tlag for "SCS" and Ct, Cp, L, and Lc other for "snyder"
Area	the area of drainage basin (Km^2)
simulation	a vector of simulation interval (in second) and time steps (an integer)
UH	a data.frame: must be provided when transformMethod is set to "user". UH is the ordinates of a user defined UH by the which its first collumn is time (Hr) and the second collumn includes flow rates (cms)
transformMethod	a string: the type of transformation method. available types: "SCS", "snyder", and "user"

Value

Hydrogaph of direct runoff

Author(s)

Rezgar Arabzadeh

See Also

sim

Examples

```
Area=200
lossMethod<-"SCS"
lossParams<-list(CN=65)
transformMethod<-c("snyder","SCS","user")
simulation<-c(interval=3600,period=100)
precipitation<-sin(seq(0.1,pi-0.1,length.out=10))*20
transformParams=list(Tlag=4,Cp=0.15,Ct=2,L=100,Lc=15)
UH<-data.frame(t=1:20,q=sin(seq(0,pi,length.out=20))*1)

SCS_loss<-loss(precipitation,lossParams,simulation,lossMethod)

snyder_transformation<-transform(rainfall=SCS_loss,
                                 transformParams,
                                 Area,
                                 simulation,
                                 transformMethod=transformMethod[1])
SCS_transformation<-transform(rainfall=SCS_loss,
                                 transformParams,
                                 Area,
                                 simulation,
                                 transformMethod=transformMethod[2])
user_transformation<-transform(rainfall=SCS_loss,
                                 transformParams,
                                 Area,
                                 simulation,
                                 UH,
                                 transformMethod=transformMethod[3])
```

---

transform.base *base function for class of* transform

Description

This function transforms an excess rainfall event to a direct runoff hydorgraph.

Usage

```
## S3 method for class 'base'
transform(rainfall,
          transformParams,
          Area,simulation,UH,
          transformMethod)
```

Arguments

rainfall	an object inherited from loss function
transformParams	a list: list of parameters associated to the selcted type of transformMethod:Tlag for "SCS" and Ct, Cp, L, and Lc other for "snyder"
Area	the area of drainage basin (Km^2)
simulation	a vector of simulation interval (in second) and time steps (an integer)
UH	a data.frame: must be provided when transformMethod is set to "user". UH is the ordinates of a user defined UH by the which its first collumn is time (Hr) and the second collumn includes flow rates (cms)
transformMethod	a string: the type of transformation method. available types: "SCS", "snyder", and "user"

Value

Hydrogaph of direct runoff

Author(s)

Rezgar Arabzadeh

See Also

transform

---

transform.default *default function for class of* transform

Description

This function transforms an excess rainfall event to a direct runoff hydorgraph.

Usage

```
## Default S3 method:
transform(rainfall,
          transformParams=list(Tlag=NULL,
                               Cp  =NULL,
                               Ct  =NULL,
                               L   =NULL,
                               Lc  =NULL),
          Area,simulation=c(interval=3600*1,period=NA),UH,
          transformMethod)
```

Arguments

`rainfall`	an object inherited from `loss` function
`transformParams`	a list: list of parameters associated to the selcted type of `transformMethod`:`Tlag` for `"SCS"` and `Ct`, `Cp`, `L`, and `Lc` other for `"snyder"`
`Area`	the area of drainage basin (Km^2)
`simulation`	a vector of simulation interval (in second) and time steps (an integer)
`UH`	a data.frame: must be provided when `transformMethod` is set to `"user"`. `UH` is the ordinates of a user defined UH by the which its first collumn is time (Hr) and the second collumn includes flow rates (cms)
`transformMethod`	a string: the type of transformation method. available types: `"SCS"`, `"snyder"`, and `"user"`

Value

Hydrogaph of direct runoff

Author(s)

Rezgar Arabzadeh

See Also

`transform`

---

tune *tunning an RHMS model*

Description

a function for tunning an RHMS model based on a set of observed time series, using *particle swarm optimization*

Usage

```
tune(object, observationTS, labelTS, delay=0,tuneLabels,maxiter,
                  transformBandWith=list(ct=c(1  ,2.5),
                                         cp=c(0.1,0.3),
                                         cn=c(25 ,85 ),
                                         k =c(0.1,2  )),
                  routingBandWith=list(manning = c(0.0001, 0.1),
                                       x       = c(0.2   , 0.6),
                                       k       =c(1      , 5  )),
                  update=FALSE)
```

Arguments

`object`	an object from class of `createBasin`
`observationTS`	a vector: an observed time series (cms)
`labelTS`	an integer: label of the object associated to the `observationTS`
`delay`	an integer: presenting the number of time steps to delay direct inflow time series
`tuneLabels`	a vector of integers: the label(s) of element(s) to be tunned
`maxiter`	an integer: maximum number of iterations
`transformBandWith`	an list: a list of vector(s), including upper and lower limit of parameters of tansformation methods. Each parameter search domain is set as a two-value vector, whose first element includes lower limit and second elemnt is upper limit. `Ct`=[1, 2.5] and `Cp`=[0.1, 0.3] are parameters for `"Snyder"` Unit Hydrograph (SUH), `cn`=[25, 85] curve number for `"SCS"` loss method, and `k` for `"horton"` loss method.
`routingBandWith`	an list: a list of vector(s), including upper and lower limit of parameters of routing methods. Each parameter search domain is set as a two-value vector, whose first element includes lower limit and second elemnt is upper limit. `manning`=[0.0001, 0.1] is a parameter used `"muskingumcunge"` method, and `x` = [0.2, 0.6] and `k`=[1, 5] belong to `"muskingum"` channel routing method.
`update`	logical: If FALSE, the optimized parameter(s) are returned,If TRUE, the calibrated object from class of `createBasin` is returned

Value

a vector of tunned parameters or an object from class of `createBasin`

Author(s)

Rezgar Arabzadeh

References

Kennedy, J. (1997). "The particle swarm: social adaptation of knowledge". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 303-308

Examples

```
S1<-createSubbasin(name = "S1",
                   precipitation=sin(seq(0,pi,length.out=20))*40,
                   Area=100,label=1, downstream=3,
                   transformMethod="SCS",lossMethod="SCS",
                   transformParams=list(Tlag=4),lossParams=list(CN=60))
S2<-createSubbasin(name = "S2",
                   precipitation=sin(seq(0,pi,length.out=20))*30,
                   Area=300,label=2, downstream=4,
                   transformMethod="snyder",lossMethod="horton",
                   transformParams=list(Cp=0.17,Ct=2,L=30,Lc=15),
                   lossParams=list(f0=10,f1=4,k=1))

R1<-createReach(name="R1",routingMethod="muskingum",
                routingParams=list(k=3,x=0.2),
                label=3,downstream=5)
R2<-createReach(name="R2",routingMethod="muskingumcunge",
                routingParams=list(bedWith=50,
                                   sideSlope=2,
                                   channelSlope=0.0005,
                                   manningRoughness=0.025,
                                   riverLength=100),
                label=4,downstream=5)
J1<-createJunction (name="J1",downstream=NA,label=5)

basin1<-createBasin(name = "Ghezil_Ozan", simPeriod=100, interval=3600)
basin1<-addObjectToBasin(S1, basin1)
basin1<-addObjectToBasin(S2, basin1)
basin1<-addObjectToBasin(R1, basin1)
basin1<-addObjectToBasin(R2, basin1)
basin1<-addObjectToBasin(J1, basin1)

## Not run:
plot(basin1)

## End(Not run)

simulated<-sim(basin1)

observationTS1<-simulated$operation$junctions[[1]]$outflow
set.seed(1)
observationTS1<-observationTS1+rnorm(length(observationTS1),0,200)
y<-observationTS1; x<-1:length(observationTS1)
observationTS1<-predict(loess(y~x),x)
observationTS1[which(observationTS1<0)]<-0
observationTS<-observationTS1

maxiter <- 20
transformBandWith=list(ct=c(1  ,2.5),
                       cp=c(0.1,0.3),
                       cn=c(25 ,85) ,
```

```
                        k =c(0.1,2))
routingBandWith=list(maning = c(0.0001,0.1),
                     x      = c(0.2   ,0.6),
                     k      = c(1     ,5))
labelTS<-5 ; tuneLabels<-1:4
## Not run:
tune(object=basin1,
     labelTS=labelTS,
     maxiter=maxiter,
     tuneLabels=tuneLabels,
     observationTS=observationTS,
     routingBandWith=routingBandWith,
     transformBandWith=transformBandWith)

## End(Not run)
```

Zaab *datasets for Zaab subbasin, a subbasin in Kurdistan, Iran.*

Description

an object inherited from class of `createBasin`. including features, of a sub-basin in Kurditan known as Zaab, such as: reservoirs, reachs, subbasins, and junctions.

Usage

```
data(Zaab)
```

Source

Iran Water Resources Management Company (2015)

Examples

```
data(Zaab)
```

Index