Package 'EGRET'

August 29, 2016

Type Package

Title Exploration and Graphics for RivEr Trends (EGRET)

Version 2.6.0

Description Statistics and graphics for streamflow history, water quality trends, and the statistical modeling algorithm: Weighted Regressions on Time, Discharge, and Season (WRTDS).

License CC0

Date 2016-07-15

Depends R (>= 3.0)

Imports dataRetrieval (>= 2.0.1), survival, fields, lubridate, methods, utils, graphics, stats, grDevices, truncnorm

Suggests xtable, knitr, rmarkdown, extrafont, testthat, rkt

LazyLoad yes

LazyData yes

VignetteBuilder knitr

BuildVignettes true

URL http://pubs.usgs.gov/tm/04/a10/, https://github.com/USGS-R/EGRET/wiki

Copyright This software is in the public domain because it contains materials that originally came from the United States Geological Survey, an agency of the United States Department of Interior. For more information, see the official USGS copyright policy at http://www.usgs.gov/visual-id/credit_usgs.html#copyright

RoxygenNote 5.0.1

NeedsCompilation no

Author Robert Hirsch [aut], Laura DeCicco [aut, cre]

Maintainer Laura DeCicco <ldecicco@usgs.gov>

Repository CRAN

Date/Publication 2016-07-27 02:53:18

R topics documented:

`EGRET-package` *EGRET package includes WRTDS and flowHistory*

Description

Package:	EGRET
Type:	Package
License:	Unlimited for this package, dependencies have more restrictive licensing.
Copyright:	This software is in the public domain because it contains materials that originally came from the United States G
LazyLoad:	yes

Details

Collection of functions to do WRTDS and flowHistory analysis, and produce graphs and tables of data and results from these analyses.

Author(s)

Robert M. Hirsch <rhirsch@usgs.gov>, Laura De Cicco <ldecicco@usgs.gov>

References

Hirsch, R.M., and De Cicco, L.A., 2014, User guide to Exploration and Graphics for RivEr Trends (EGRET) and dataRetrieval: R packages for hydrologic data: U.S. Geological Survey Techniques and Methods book 4, chap. A10, 94 p., http://dx.doi.org/10.3133/tm4A10

`as.egret` *Create named list for EGRET analysis*

Description

Create a named list with the INFO, Daily, and Sample dataframes, and surface matrix. If any of these are not available, an NA should be

Usage

```
as.egret(INFO, Daily, Sample = NA, surfaces = NA)
```

Arguments

INFO	dataframe containing the INFO dataframe
Daily	dataframe containing the daily data
Sample	dataframe containing the sample data
surfaces	matrix returned from modelEstimation. Default is NA.

Value

eList named list with Daily, Sample, and INFO dataframes, along with the surfaces matrix. Any of these values can be NA, not all EGRET functions will work with missing parts of the named list eList.

See Also

readNWISDaily, readNWISSample

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
INFO <- getInfo(eList)
eList_flowHistory <- as.egret(INFO, Daily)
plotFlowSingle(eList_flowHistory, 1)
Sample <- getSample(eList)
surfaces <- getSurfaces(eList)
eList_full <- as.egret(INFO, Daily, Sample, surfaces)
plotFluxQ(eList_full)
```

---

blankTime *Deletes the computed values during periods of time when there is no sample data*

Description

This function is used when the data analyst believes that a gap in the sample data record is so long that estimates during that period are not reliable. This is only unsed for periods of several years in duration. For this period, the values of Conc, Flux, FNConc and FNFlux are all converted to NA.

Usage

```
blankTime(eList, startBlank, endBlank)
```

Arguments

eList	named list with at least the Daily dataframe
startBlank	character specifying starting date of blank period, input in quotes in yyyy-mm-dd format
endBlank	character specifying the ending date of blank period, input in quotes in yyyy-mm-dd format

Value

eList named list with modified Daily data frame.

Examples

```
startBlank = "2004-10-01"
endBlank = "2006-09-30"
eList <- Choptank_eList
eList <- blankTime(eList, startBlank, endBlank)
```

---

boxConcMonth *Box plot of the water quality data by month*

Description

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
boxConcMonth(eList, printTitle = TRUE, cex = 0.8, cex.axis = 1.1,
  cex.main = 1.1, las = 1, logScale = FALSE, tcl = 0.5,
  tinyPlot = FALSE, customPar = FALSE, showYLabels = TRUE,
  showXLabels = TRUE, showXAxis = TRUE, showYAxis = TRUE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
las	numeric in 0,1,2,3; the style of axis labels, see ?par
logScale	logical if TRUE y plotted in log axis

tcl	number defaults to 0.5, specifies length of tick marks as fraction of height of a line of text
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-plot figure, default is FALSE.
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function
showYLabels	logical defaults to TRUE. If FALSE, the y axis label is not plotted
showXLabels	logical defaults to TRUE. If FALSE, the x axis label is not plotted
showXAxis	logical defaults to TRUE. If FALSE, the x axis is not plotted
showYAxis	logical defaults to TRUE. If FALSE, the y axis is not plotted
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

boxplot

Examples

```
eList <- Choptank_eList
# Water year:
boxConcMonth(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
boxConcMonth(eList)
```

---

boxConcThree *Three box plots side-by-side*

Description

This function is used to compare the distribution of concentration in the sample and predicted data set.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
boxConcThree(eList, tinyPlot = FALSE, printTitle = TRUE,
  moreTitle = "WRTDS", customPar = FALSE, font.main = 2, cex = 0.8,
  cex.main = 1.1, cex.axis = 1.1, ...)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-plot figure, default is FALSE.
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
moreTitle	character specifying some additional information to go in figure title, typically some information about the specific estimation method used, default is no additional information
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function
font.main	font to be used for plot main titles
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.main	magnification to be used for main titles relative to the current setting of cex
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

boxplot

Examples

```
eList <- Choptank_eList
# Water year:
boxConcThree(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
boxConcThree(eList)
```

---

boxQTwice *Two box plots side-by-side, discharge on sample days, and discharge on all days*

Description

This function is used to compare the distribution of discharges in the sample data set and the discharges in the full daily data set. Note that discharge is plotted on a logarithmic axis. The data is logged before the statistics are performed to determine the output of the boxplot.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
boxQTwice(eList, printTitle = TRUE, qUnit = 2, cex = 0.8,
  cex.main = 1.1, logScale = TRUE, cex.axis = 1.1, tcl = 0.5, las = 1,
  tinyPlot = FALSE, customPar = FALSE, ...)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.main	magnification to be used for main titles relative to the current setting of cex
logScale	logical if TRUE y plotted in log axis. Defaults to TRUE.
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
tcl	number defaults to 0.5, specifies length of tick marks as fraction of height of a line of text
las	numeric in 0,1,2,3; the style of axis labels, see ?par
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-plot figure, default is FALSE.
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

boxplot

Examples

```
eList <- Choptank_eList
# Water year:
boxQTwice(eList)
boxQTwice(eList, qUnit=1)
boxQTwice(eList, qUnit='cfs')
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
boxQTwice(eList)
```

boxResidMonth *A box plot of WRTDS residuals by month*

Description

This function produces a boxplot of the residuals from WRTDS, expressed in natural log concentration units. It provides an alternative for viewing the standardized residuals, where the each residual is divided by its estimated standard error. The monthly boxplot widths are proportional to the square root of the sample size. The residuals for a censored value are determined as the difference between the natural log of the average of the upper and lower. bounds on the sample value, minus the log space estimate of concentration.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata

Usage

```
boxResidMonth(eList, stdResid = FALSE, las = 1, printTitle = TRUE,
  cex = 0.8, cex.axis = 1.1, cex.main = 1.1, font.main = 2,
  tinyPlot = FALSE, customPar = FALSE, randomCensored = FALSE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
stdResid	logical variable, if TRUE it uses the standardized residual, if FALSE it uses the actual, default is FALSE
las	numeric in 0,1,2,3; the style of axis labels
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
font.main	font to be used for plot main titles
tinyPlot	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function
randomCensored	logical. Show censored residuals as randomized.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

boxplot

Examples

```
eList <- Choptank_eList
# Water year:
boxResidMonth(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
boxResidMonth(eList)
```

---

calculateMonthlyResults

Calculates monthly values of Q, Conc, Flux, FNConc, and FNFlux for the entire record

---

Description

Computes the monthly mean values of discharge, concentration, flux, flow-normalized concentration and flow-normalized flux (Q, Conc, Flux, FNConc, and FNFlux) in SI units (For discharge they are in m3/s, concentration is mg/L, and flux is kg/day). It returns a data frame containing month, year, decimal year, and mean values of DecYear, Q, Conc, Flux, FNConc, and FNFlux.

Usage

```
calculateMonthlyResults(eList)
```

Arguments

eList	named list with at least the Daily dataframes

Value

MonthlyResults data frame of numeric values describing the monthly average values

Examples

```
eList <- Choptank_eList
monthlyResults <- calculateMonthlyResults(eList)
```

censoredSegments *Generic plotting function to create censored line segments*

Description

Basic plotting framework for EGRET dot plots. Graphical parameters default to values that work well with most plots, but all can be re-assigned. See ?par for complete definitions of most optional input variables.

Usage

```
censoredSegments(yBottom, yLow, yHigh, x, Uncen, col = "black", lwd = 1)
```

Arguments

yBottom	number specifying minimum flux (required)
yLow	vector specifying the x data (required), such as ConcLow
yHigh	vector specifying the x data (required), such as ConcHigh
x	vector x data (required)
Uncen	vector that defines whether the values are censored (0) or not (1)
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width

See Also

segments

Examples

```
x <- c(1,2,3,4,5,6)
y <- c(1,3,4,3.3,4.4,7)
xlim <- c(min(x)*.75,max(x)*1.25)
ylim <- c(0,1.25*max(y))
xlab <- "Date"
ylab <- "Concentration"
xTicks <- pretty(xlim)
yTicks <- pretty(ylim)
genericEGRETDotPlot(x=x, y=y,
                    xlim=xlim, ylim=ylim,
                    xlab=xlab, ylab=ylab,
                    xTicks=xTicks, yTicks=yTicks,
                    plotTitle="Test"
)
yBottom <- 0
yLow <- c(NA,3,4,3.3,4,7)
yHigh <- c(1,3,4,3.3,5,NA)
Uncen <- c(0,1,1,1,0,0)
censoredSegments(yBottom=yBottom,yLow=yLow,yHigh=yHigh,x=x,Uncen=Uncen)
```

checkStartEndDate *checkStartEndDate*

Description

Checks that the start date is before the end date. If not, it will give the user the opportunity to correct, otherwise will create a warning.

Usage

```
checkStartEndDate(startDate, endDate, interactive = TRUE)
```

Arguments

startDate	character
endDate	character
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

vector where first value is startDate, second is endDate

Examples

```
startDate <- '1985-01-01'
endDate <- '1990-01-01'
checkStartEndDate(startDate, endDate)
```

Choptank_eList *Example eList*

Description

Example data representing data from the Choptank River at Greensboro, MD, USGS data Data is a named list of the Daily, Sample, INFO dataframes, and the surface matrix.

Examples

```
head(Choptank_eList$Daily)
head(Arkansas_eList$Daily)
```

compressData *Compress sample data frame*

Description

Using raw data that has at least dateTime, value, code, populates the measured data portion of the Sample dataframe used in EGRET. ConcLow = Lower bound for an observed concentration ConcHigh = Upper bound for an observed concentration ConcAve = Average of ConcLow and ConcHigh. If ConcLow is NA, then ConcAve = ConcHigh/2 Uncen = 1 if uncensored, 0 if censored

Usage

```
compressData(data, interactive = TRUE)
```

Arguments

data	dataframe contains at least dateTime, value, code columns
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

dataframe returnDataFrame data frame containing dateTime, ConcHigh, ConcLow, Uncen, ConcAve

Examples

```
dateTime <- c('1985-01-01', '1985-01-02', '1985-01-03')
comment1 <- c("","","")
value1 <- c(1,2,3)
comment2 <- c("","<","")
value2 <- c(2,3,4)
comment3 <- c("","","<")
value3 <- c(3,4,5)
dataInput <- data.frame(dateTime, comment1, value1,
      comment2, value2,
      comment3, value3, stringsAsFactors=FALSE)
compressData(dataInput)
```

Constants *Constants included with EGRET*

Description

- fluxConstFlux conversion object
- qConstFlow conversion object
- monthInfoMonth object

Examples

```
fluxConst
fluxConst[['kgDay']]
fluxConst[['kgDay']]@unitName
qConst
qConst[['cfs']]
qConst[['cfs']]@qUnitName
```

dataOverview *Data Overview for WRTDS*

Description

Gives a summary of data to be used for WRTDS analysis

Usage

```
dataOverview(Daily, Sample)
```

Arguments

Daily	dataframe
Sample	dataframe

See Also

mergeReport

Examples

```
eList <- Choptank_eList
exDaily <- getDaily(eList)
exSample <- getSample(eList)
dataOverview(Daily = exDaily, Sample = exSample)
```

dateFormatCheck *Check date format*

Description

Checks to see if format is YYYY-MM-DD. Also performs a few other date checks.

Usage

```
dateFormatCheck(date)
```

Arguments

date	character

Value

condition logical if TRUE,

Examples

```
date <- '1985-01-01'
dateFormatCheck(date)
dateWrong <- '1999/1/7'
dateFormatCheck(dateWrong)
```

estCrossVal *Jack-Knife cross validation of the WRTDS (Weighted Regressions on Time, Discharge, and Season)*

Description

This function fits the WRTDS model n times (where n is the number of observations). For each fit, the data value being estimated is eliminated from the record. This gives predictions that do not depend on knowing the actual result for that day. Thus it provides for a more "honest" estimate of model performance than a traditional error analysis that uses all the data.

Usage

```
estCrossVal(numDays, DecLow, DecHigh, Sample, windowY = 7, windowQ = 2,
  windowS = 0.5, minNumObs = 100, minNumUncen = 50, edgeAdjust = TRUE)
```

Arguments

numDays	number of days in the Daily record
DecLow	number specifying minimum decimal year
DecHigh	number specifying maximum decimal year
Sample	data frame containing the sample values, default is Sample
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 7
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE.

Value

SampleCrossV data frame containing the sample data augmented by the results of the cross-validation exercise

Examples

```
eList <- Choptank_eList
Sample <- getSample(eList)
Daily <- getDaily(eList)
numDays <- length(Daily$DecYear)
DecLow <- Daily$DecYear[1]
DecHigh <- Daily$DecYear[numDays]
## Not run:
SampleCrossV <- estCrossVal(numDays,DecLow,DecHigh,Sample)

## End(Not run)
```

---

estDailyFromSurfaces *Estimates all daily values of Concentration, Flux, Flow-Normalized Concentration, and Flow Normalized Flux*

Description

Uses the surfaces estimated in estSurfaces to estimate these four time series in addition to the time series for standard error and yHat (estimated log concentration). The results are stored in an augmented version of the Daily data frame, which is returned as part of an EGRET object.

Usage

```
estDailyFromSurfaces(eList)
```

Arguments

eList	named list with at least the Daily and INFO dataframes, and the surface matrix

Value

egret object with altered Daily dataframe

Examples

```
eList <- Choptank_eList
################################################
# This is usually done in modelEstimation:
Daily <- getDaily(eList)
surfaceIndexParameters<-surfaceIndex(Daily)
INFO <- eList$INFO
INFO$bottomLogQ<-surfaceIndexParameters[1]
INFO$stepLogQ<-surfaceIndexParameters[2]
INFO$nVectorLogQ<-surfaceIndexParameters[3]
INFO$bottomYear<-surfaceIndexParameters[4]
INFO$stepYear<-surfaceIndexParameters[5]
INFO$nVectorYear<-surfaceIndexParameters[6]
eList$INFO <- INFO
################################################
## Not run:
Daily <- estDailyFromSurfaces(eList)

## End(Not run)
```

---

estSurfaces — *Estimate the three surfaces (for yHat, SE and ConcHat) as a function of DecYear and logQ and store in the three-dimensional object called surfaces*

Description

This function uses weighted survival regression to estimate three surfaces that cover the complete range of DecYear and log(Q) values in the Daily data set. These surfaces are: (1) is the estimated log concentration (yHat), (2) is the estimated standard error (SE), (3) is the estimated concentration (ConcHat). They are mapped as an array that covers the complete space of daily discharge and time. The first index is discharge, layed out in 14 equally spaced levels of log(Q). The second index is time, layed out as 16 increments of the calendar year, starting January 1. It returns the 3 dimensional array called surfaces. This array will be used to estimate these 3 quantities for any given day in the daily values record.

Usage

```
estSurfaces(eList, windowY = 7, windowQ = 2, windowS = 0.5,
  minNumObs = 100, minNumUncen = 50, edgeAdjust = TRUE)
```

Arguments

eList	named list with at least the Sample and Daily dataframes
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 7
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. Default is TRUE.

Value

surfaces array containing the three surfaces estimated, array is 3 dimensional

Examples

```
eList <- Choptank_eList
## Not run: surfaces <- estSurfaces(eList)
```

---

fixSampleFrame *Update Sample dataframe*

Description

Used for updating the Sample dataframe if ConcLow or ConcHigh is manually adjusted. Adjusts ConcAve and Uncen columns.

Usage

```
fixSampleFrame(eList)
```

Arguments

eList	named list with at least the Sample dataframes

Value

localSample data frame

Examples

```
eList <- Choptank_eList
Sample <- eList$Sample
Sample[1,c("ConcLow","ConcHigh")] <- c(NA, 0.01) # Adjusted to left-censored
Sample[2,c("ConcLow","ConcHigh")] <- c(1.1, 1.3) # Adjusted to interval-censored
Sample[3,c("ConcLow","ConcHigh")] <- c(1.3, 1.3) # Simple adjustment
eList$Sample <- Sample
eList <- fixSampleFrame(eList)
eList$Sample[1:3,]
```

---

flowDuration *Computes several values of the flow duration curve for streamflow centered on a specific date of the year*

Description

This function is useful for helping the analyst determine the empirical probability distribution of streamflow for a particular part of the year or for the whole year. This is particularly useful in setting up discharge scales for various other plots in this package.

Usage

```
flowDuration(eList, centerDate = "09-30", qUnit = 2, span = 365)
```

Arguments

eList	named list with at least Daily and INFO dataframes
centerDate	character specifying the center date of the part of the year for which the flow duration is to be calculated, it is in the form "mm-dd" (it must be in quotes), default is "09-30"
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
span	number this is the half-width of the window over which the discharge values are to be used in constructing the flow-duration curve. If the full year is desired any value greater than 182 will provide serve. Note that for a window of about 2-months width, a span value shoud be about 30.

Value

qDuration A named vector with flow duration information.

Examples

```
eList <- Choptank_eList
# for a window of 30 days either side of June 25 expressed in units of cfs:
flowDuration(eList,"06-25", qUnit=1,span=30)
# for a flow-duration curve covering the whole year, expressed in units of csf:
flowDuration(eList, "01-01", qUnit=2)
```

`fluxBiasMulti` *Produces 8-panel plot that is useful for determining if there is a flux bias problem*

Description

These plots use the jack-knife estimates from WRTDS to investigate the potential flux bias problem. It can also be used for estimates constructed by other methods (such as LOADEST) if the results are stored in a data frame organized like the Sample data frame. It allows additional label information to indicate what method is used.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata

Usage

```
fluxBiasMulti(eList, qUnit = 2, fluxUnit = 3, moreTitle = "WRTDS",
  cex = 0.7, cex.axis = 1.1, cex.main = 1.1, randomCensored = FALSE,
  col = "black", lwd = 1, ...)
```

Arguments

`eList`	named list with at least Sample, Daily, and INFO dataframes
`qUnit`	object of qUnit class. `printqUnitCheatSheet`, or numeric represented the short code, or character representing the descriptive name.
`fluxUnit`	object of fluxUnit class. `printFluxUnitCheatSheet`, or numeric represented the short code, or character representing the descriptive name.
`moreTitle`	character specifying some additional information to go in figure title, typically some information about the specific estimation method used, default is no additional information
`cex`	numerical value giving the amount by which plotting symbols should be magnified
`cex.axis`	magnification to be used for axis annotation relative to the current setting of cex
`cex.main`	magnification to be used for main titles relative to the current setting of cex
`randomCensored`	logical. Show censored residuals as randomized.
`col`	color of points on plot, see ?par 'Color Specification'
`lwd`	number line width
`...`	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

Examples

```
eList <- Choptank_eList
fluxBiasMulti(eList)
# Water year:
## Not run:
pdf("fluxBiasMulti.pdf", height=9, width=8)
fluxBiasMulti(eList)
dev.off()
# Graphs consisting of Jun-Aug
eList <- setPA(eList,paStart=6,paLong=3)
pdf("fluxBiasMultiSummer.pdf", height=9, width=8)
fluxBiasMulti(eList)
dev.off()

## End(Not run)
```

---

fluxBiasStat *Compute the flux bias statistic: (mean of estimated flux - mean of observed flux) / mean of observed flux*

Description

Computes three versions of the flux bias: The first where all censored values are set to their miniumum. The second where all censored values are set to their maximum. The third which is the average of the other two. In practice there is rarely a noticable difference among them.

Usage

```
fluxBiasStat(localSample)
```

Arguments

localSample	data frame that contains the concentration data, default name is Sample

Value

fluxBias a vector of three numerical values, a lower bound, upper bound and an average estimate of the ratio of (mean estimated flux - mean observed flux) / mean estimated flux. Typically one should use fluxBias[3]

Examples

```
eList <- Choptank_eList
Sample <- getSample(eList)
fluxBias <- fluxBiasStat(Sample)
```

fluxUnit-class *fluxUnit class*

Description

Some details about the fluxUnit class

Details

shortName A character specifying the short name.

unitFactor A numeric representing the conversion factor

unitName A character specifying the full name.

unitExpress An expression specifying the full name starting with Observed.

unitExpressTiny An expression specifying the abbreviated name starting with Observed.

unitEstimate An expression specifying the full name starting with Estimated.

unitEstimateTiny An expression specifying the abbreviated name starting with Estimated.

unitUSGS A character specifying flux with full text.

shortCode A number for quick lookup

formatCheckDate *formatCheckDate*

Description

Response to the date format checker. If the date is not formated correctly, it will give the user the opportunity to correct, otherwise will create a warning.

Usage

```
formatCheckDate(Date, dateString, interactive = TRUE)
```

Arguments

Date	character
dateString	character used in either error message or interactive message. An example would be "startDate"
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

condition logical if TRUE,

Examples

```
Date <- '1985-01-01'
dateString <- 'startDate'
formatCheckDate(Date, dateString, interactive = FALSE)
```

`formatCheckParameterCd` *formatCheckParameterCd*

Description

Checks that the parameter code is 5 digits. If it is less, it will pad the character with zeros. If more, ask the user to re-enter.

Usage

```
formatCheckParameterCd(parameterCd, interactive = TRUE)
```

Arguments

`parameterCd`	character to check
`interactive`	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

parameterCd character

Examples

```
pCode <- '01234'
formatCheckParameterCd(pCode)
```

`generalAxis` *Axis generation for log discharge*

Description

Discharge axis tick generation

Usage

```
generalAxis(x, maxVal, minVal, units = NA, logScale = FALSE,
  tinyPlot = FALSE, padPercent = 5, concentration = TRUE,
  prettyDate = TRUE)
```

Arguments

x	vector to create scale about
maxVal	number maximum value on returned scale
minVal	number minimum value on returned scale
units	character concentration units. Typically found in INFO$param.units.
logScale	logical whether or not to return a log scale
tinyPlot	logical
padPercent	number used to pad the max and min if not specified
concentration	logical if concentration=TRUE, labels returned as concentration units, otherwise flux units.
prettyDate	logical use 'pretty' limits for date axis if TRUE, or force the yearStart/yearEnd as limits if FALSE

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
INFO <- getInfo(eList)
x <- Daily$Q
max <- max(x)
min <- 0
units <- INFO$param.units
generalAxis(x, max, min, units)
min <- min(x)
generalAxis(x, max, min, units, log=TRUE)
```

---

genericEGRETDotPlot *Generic EGRET plotting function*

Description

Basic plotting framework for EGRET dot plots. Graphical parameters default to values that work well with most plots, but all can be re-assigned. See ?par for complete definitions of most optional input variables.

Usage

```
genericEGRETDotPlot(x, y, xlim, ylim, xTicks, yTicks, printTitle = TRUE,
  xaxs = "i", xlab = "", yaxs = "i", ylab = "", plotTitle = "",
  pch = 20, cex = 0.7, cex.main = 1.3, font.main = 2, cex.lab = 1.2,
  tcl = 0.5, cex.axis = 1, las = 1, xDate = FALSE, tinyPlot = FALSE,
  hLine = FALSE, oneToOneLine = FALSE, rmSciX = FALSE, rmSciY = FALSE,
  customPar = FALSE, col = "black", lwd = 1, showXLabels = TRUE,
  showYLabels = TRUE, showXAxis = TRUE, showYAxis = TRUE,
  removeFirstX = FALSE, removeLastX = FALSE, removeFirstY = FALSE,
  removeLastY = FALSE, ...)
```

Arguments

x	vector specifying the x data (required)
y	vector specifying the y data (required)
xlim	vector specifying the x plotting range (required)
ylim	vector specifying the y plotting range (required)
xTicks	vector specifying x axis tick placement (required)
yTicks	vector specifying y axis tick placement (required)
printTitle	logical defaults to TRUE, plotting parameter to control whether to have title
xaxs	character defaults to "i", defines the style of x-axis interval calculation. Possible values are i, r, e, s, d.
xlab	character defaults to "", defines the x label
yaxs	character defaults to "i", defines the style of y-axis interval calculation. Possible values are i, r, e, s, d.
ylab	character defaults to "", defines the y label
plotTitle	character defaults to "", defines the plot title
pch	number defaults to 20, specifies plot symbol
cex	number defaults to 0.7, specifies plotting text magnification
cex.main	number defaults to 1.3, specifies title text magnification
font.main	number defaults to 2, specifies which font to use for text
cex.lab	number defaults to 1.2 specifies label text magnification
tcl	number defaults to 0.5, specifies length of tick marks as fraction of height of a line of text.
cex.axis	number defaults to 1, specifies axis text magnification
las	number represents style of axis labels
xDate	logical defaults to FALSE, changes x label to "year-month" format if set to TRUE and total years less than 4.
tinyPlot	logical defaults to FALSE, if TRUE, changes defaults to be appropriate for multi-plot
hLine	logical defaults to FALSE, inserts horizontal line at zero
oneToOneLine	logical defaults to FALSE, inserts 1:1 line
rmSciX	logical defaults to FALSE, changes x label from scientific to fixed
rmSciY	logical defaults to FALSE, changes y label from scientific to fixed
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
showXLabels	logical defaults to TRUE. If FALSE, the x axis label is not plotted
showYLabels	logical defaults to TRUE. If FALSE, the y axis label is not plotted

showXAxis	logical defaults to TRUE. If FALSE, the x axis is not plotted
showYAxis	logical defaults to TRUE. If FALSE, the y axis is not plotted
removeFirstX	logical defaults to FALSE. If TRUE, removes the first x axis label. This can be handy for plotting mutliple plots.
removeLastX	logical defaults to FALSE. If TRUE, removes the last x axis label. This can be handy for plotting mutliple plots.
removeFirstY	logical defaults to FALSE. If TRUE, removes the first y axis label. This can be handy for plotting mutliple plots.
removeLastY	logical defaults to FALSE. If TRUE, removes the last y axis label. This can be handy for plotting mutliple plots.
...	additional graphical parameters can be adjusted

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
x <- Daily$Date
y <- Daily$Q
xlim <- c(min(x),max(x))
ylim <- c(min(y),1.05*max(y))
xlab <- "Date"
ylab <- "Flow"
xTicks <- pretty(xlim)
yTicks <- pretty(ylim)
genericEGRETDotPlot(x=x, y=y,
                    xlim=xlim, ylim=ylim,
                    xlab=xlab, ylab=ylab,
                    xTicks=xTicks, yTicks=yTicks,
                    plotTitle="Test"
)
```

---

getDaily *Get Daily dataframe from EGRET object*

Description

From a named list or EGRET object, extract the Daily dataframe

Usage

```
getDaily(x, ...)

## S3 method for class 'egret'
getDaily(x, ...)

## Default S3 method:
getDaily(x, ...)
```

Arguments

x	EGRET object or named list
...	additional parameters

Value

Daily dataframe

See Also

readNWISDaily, readNWISSample

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
```

---

getInfo *Get INFO dataframe from EGRET object*

Description

From a named list or EGRET object, extract the INFO dataframe

Usage

```
getInfo(x, ...)

## S3 method for class 'egret'
getInfo(x, ...)

## Default S3 method:
getInfo(x, ...)
```

Arguments

x	EGRET object or named list
...	additional parameters

Value

INFO dataframe

See Also

readNWISDaily, readNWISSample

Examples

```
eList <- Choptank_eList
INFO <- getInfo(eList)
```

`getSample` *Get Sample dataframe from EGRET object*

Description

From a named list or EGRET object, extract the Sample dataframe

Usage

```
getSample(x, ...)

getSample(x, ...)

getSample.default(x, ...)
```

Arguments

`x`	EGRET object or named list
`...`	additional parameters

Value

Sample dataframe

See Also

`readNWISDaily`, `readNWISSample`

Examples

```
eList <- Choptank_eList
Sample <- getSample(eList)
```

getSurfaces *Get surfaces matrix from EGRET object*

Description

From a named list or EGRET object, extract the surfaces matrix

Usage

```
getSurfaces(x, ...)

## S3 method for class 'egret'
getSurfaces(x, ...)

## Default S3 method:
getSurfaces(x, ...)
```

Arguments

x	EGRET object or named list
...	additional parameters

Value

Sample dataframe

See Also

readNWISDaily, readNWISSample

Examples

```
eList <- Choptank_eList
surfaces <- getSurfaces(eList)
```

INFOdataframe *Import metadata to create INFO data frame*

Description

Populates INFO data frame from either NWIS (readNWISInfo), Water Quality Portal (readWQPInfo), or user-supplied files (readUserInfo).

Usage

```
readNWISInfo(siteNumber, parameterCd, interactive = TRUE)

readWQPInfo(siteNumber, parameterCd, interactive = TRUE)

readUserInfo(filePath, fileName, hasHeader = TRUE, separator = ",",
  interactive = TRUE)
```

Arguments

siteNumber	character site number. For readNWISInfo, this is usually an 8 digit number, for readWQPInfo, it is usually a longer code. For instance, a USGS site number in the Water Quality Portal would be in the form 'USGS-XXXXXXXX'. If the siteNumber is left blank (an empty string), the interactive option allows users to enter required information by hand, otherwise those fields are left blank.
parameterCd	character USGS parameter code (a 5 digit number) or characteristic name (if using readWQPInfo). If the parameterCd is left blank (an empty string), the interactive option allows users to enter required information by hand, otherwise those fields are left blank.
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.
filePath	character specifying the path to the file (used in readUserInfo)
fileName	character name of file to open (used in readUserInfo)
hasHeader	logical true if the first row of data is the column headers (used in readUserInfo)
separator	character that separates data cells (used in readUserInfo)

Value

INFO data frame. Any metadata can be stored in INFO. However, there are 8 columns that EGRET uses by name in some functions:

Required column	Used in function	Description
param.units***	All concentration plotting functions	The units as listed in this field are used to create the concentration a
shortName	All plotting functions	Station short name, used to label plots
paramShortName	All plotting functions	Parameter short name, used to label plots
drainSqKm	plotFlowSingle, printSeries	Calculate runoff
constitAbbrev	saveResults	Parameter abbrieviation, used to auto-name workspace
staAbbrev	saveResults	Station abbrieviation, used to auto-name workspace
paStart	Most EGRET functions	Starting month of period of analysis. Defaults to 10
paLong	Most EGRET functions	Length in number of months of period of analysis. Defaults to 12

*** Additionally, EGRET assumes that all concentrations are saved in mg/l. If some variation of 'mg/l' is not found in INFO$param.units, functions that calculate flux will issue a warning. This is because the conversion from mg/l to the user-specified flux unit (e.g., kg/day) uses hard-coded conversion factors.

See Also

readNWISsite, readNWISpCode

whatWQPsites

Examples

```
# These examples require an internet connection to run
# Automatically gets information about site 05114000 and temperature
## Not run:
INFO <- readNWISInfo('05114000','00010')

## End(Not run)
# These examples require an internet connection to run
# Automatically gets information about site 01594440 and temperature, no interaction with user
nameToUse <- 'Specific conductance'
pcodeToUse <- '00095'
## Not run:
INFO <- readWQPInfo('USGS-04024315',pcodeToUse)

INFO2 <- readWQPInfo('WIDNR_WQX-10032762',nameToUse)
# To adjust the label names:
INFO$shortName <- "Little"
INFO$paramShortName <- "SC"

## End(Not run)
filePath <- system.file("extdata", package="EGRET")
filePath <- paste(filePath,"/",sep="")
fileName <- 'infoTest.csv'
INFO <- readUserInfo(filePath,fileName, separator=",",interactive=FALSE)
```

---

is.egret *Check for EGRET object*

Description

Checks object to see if it is an EGRET object

Usage

```
is.egret(x)
```

Arguments

x	object to check

Value

logical

Examples

```
eList <- Choptank_eList
is.egret(eList)
```

`logPretty1` *Sets up tick marks for an axis with a log scale, where the graph is small*

Description

Axis tick marks for a log scale for cases where the data cover many orders of magnitude and the graph is small. These tick marks are designed to progress by factors of 10.

Usage

```
logPretty1(xMin, xMax)
```

Arguments

`xMin`	A numeric value for the minimum value to be plotted, it must be >0
`xMax`	A numeric value for the maximum value to be plotted, it must be >xMax

Value

xTicks A vector representing the values for each of the tick marks

Examples

```
xMin<-0.7
xMax<-990000
logPretty1(xMin,xMax)
xMin<-3
xMax<-15
logPretty1(xMin,xMax)
```

`logPretty3` *Sets up tick marks for an axis with a log scale*

Description

Axis tick marks for a log scale. These tick marks are designed to progress with 3 tick marks for every factor of 10. For example: 2,5,10,20,50,100,200,500.

Usage

```
logPretty3(xMin, xMax)
```

Arguments

xMin	A numeric value for the minimum value to be plotted, it must be >0
xMax	A numeric value for the maximum value to be plotted, it must be >xMax

Value

xTicks A vector representing the values for each of the tick marks

Examples

```
logPretty3(0.7,990000)
logPretty3(3,15)
```

---

makeAnnualSeries *Produces annual series of 8 streamflow statistics (and a lowess smooth of them) from daily streamflow data*

Description

Part of the flowHistory system. The data come from Daily and INFO data frames. Note that the function setPA must be run before this to establish the period of analysis (e.g. water year).

Usage

```
makeAnnualSeries(eList, edgeAdjust = TRUE)
```

Arguments

eList	named list with at least Daily and INFO dataframes
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE, but a logical in INFO$edgeAdjust will override the default.

Details

istat	Name
1	minimum 1-day daily mean discharge
2	minimum 7-day mean of the daily mean discharges
3	minimum 30-day mean of the daily mean discharges
4	median of the daily mean discharges
5	mean of the daily mean discharges
6	maximum 30-day mean of the daily mean discharges
7	maximum 7-day mean of the daily mean discharges
8	maximum 1-day daily mean discharge

Value

annualSeries data frame that contains the annual series of streamflow statistics

Examples

```
eList <- Choptank_eList
annualSeries <- makeAnnualSeries(eList)
```

`makeAugmentedSample` *Create Randomized Residuals and Observations*

Description

This function is used to add two columns to the Sample data frame: rResid and rObserved. rResid is the randomized residual value computed in log concentration units, and rObserved is the randomized 'observed' value of concentration in concentration units.

Usage

```
makeAugmentedSample(eList)
```

Arguments

`eList`	named list with at least the Sample dataframe

Value

eList named list with modified Sample data frame.

`mergeReport` *Merge Sample and Daily Data into EGRET object*

Description

Merges the flow data from the daily record into the sample record, then creates a named list with the Daily, Sample, and INFO dataframe. The Sample dataframe in the global enviornment does not update with the flow information. To extract the new Sample dataframe, use the command: `Sample <- eList$Sample`.

Usage

```
mergeReport(INFO, Daily, Sample, surfaces = NA, interactive = TRUE)
```

Arguments

INFO	dataframe containing the INFO dataframe
Daily	dataframe containing the daily data
Sample	dataframe containing the sample data
surfaces	matrix returned from modelEstimation. Default is NA.
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

eList named list with Daily, Sample, and INFO dataframes, along with the surfaces matrix. Any of these values can be NA, not all EGRET functions will work with missing parts of the named list eList.

See Also

readNWISDaily, readNWISSample

Examples

```
## Not run:
siteNumber <- '01594440'
pCode <- '01075'
Daily <- readNWISDaily(siteNumber,'00060', '1985-01-01', '1990-03-31')
Sample <- readNWISSample(siteNumber,pCode, '1985-01-01', '1990-03-31')
INFO <- readNWISInfo(siteNumber,pCode,interactive=FALSE)
eList <- mergeReport(INFO, Daily, Sample)
Sample <- eList$Sample

## End(Not run)
```

---

modelEstimation *Estimation process for the WRTDS (Weighted Regressions on Time, Discharge, and Season)*

Description

This one function does a jack-knife cross-validation of a WRTDS model, fits the surface (concentration as a function of discharge and time), estimates daily values of concentration and flux, and flow normalized values. It returns a named list with the following dataframes: Daily, INFO, Sample, and the matrix: surfaces.

Usage

```
modelEstimation(eList, windowY = 7, windowQ = 2, windowS = 0.5,
  minNumObs = 100, minNumUncen = 50, edgeAdjust = TRUE)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 7
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE.

Value

eList named list with Daily, Sample, and INFO dataframes, along with the surfaces matrix. Any of these values can be NA, not all EGRET functions will work with missing parts of the named list eList.

Examples

```
eList <- Choptank_eList
## Not run:

#Run an estimation adjusting windowQ from default:
eList <- modelEstimation(eList, windowQ=5)

## End(Not run)
```

---

monthLabel-class *monthLabel class*

Description

Some details about the monthLabel class

Details

monthAbbrev A character specifying the abbreviated month name.

monthFull A character specifying the full month name

monthSingle A character specifying the single letter of the month.

`multiPlotDataOverview` *Produces a 4 panel plot that gives an overview of the data set prior to any processing*

Description

The four plots produced are 1) log concentration versus log discharge, 2) log concentration versus time 3) a boxplot of log concentration by month, and 4) a side-by-side boxplot of the sampled discharges and all daily discharges. To save space, the graphic is labeled only at the top of the 4 graph display.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
multiPlotDataOverview(eList, qUnit = 2, cex.main = 1.2,
  randomCensored = FALSE, logScaleConc = TRUE, logScaleQ = TRUE)
```

Arguments

`eList`	named list with at least Daily, Sample, and INFO dataframes
`qUnit`	object of qUnit class `printqUnitCheatSheet`, or numeric represented the short code, or character representing the descriptive name.
`cex.main`	magnification to be used for main titles relative to the current setting of cex
`randomCensored`	logical. Show censored values as randomized.
`logScaleConc`	logical if TRUE y in concentration graphs plotted in log axis. Default is TRUE.
`logScaleQ`	logical if TRUE y in streamflow graphs plotted in log axis. Default is TRUE.

See Also

`plotConcQ`, `boxConcMonth`, `plotConcTime`, `boxQTwice`

Examples

```
eList <- Choptank_eList
# Water year:
multiPlotDataOverview(eList, qUnit=1)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
multiPlotDataOverview(eList, qUnit=1)
```

plot15 *Makes 15 graphs of streamflow statistics on a single page*

Description

Part of flowHistory system.

Usage

```
plot15(eList, yearStart, yearEnd)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
yearStart	A numeric value for year in which the graph should start, default is NA, which indicates that the graph should start with first annual value
yearEnd	A numeric value for year in which the graph should end, default is NA, which indicates that the graph should end with last annual value

See Also

plot1of15

Examples

```
eList <- Choptank_eList
## Not run:
pdf("plot15.pdf",heigh=10,width=8)
plot15(eList, yearStart=1990,yearEnd=2000)
dev.off()

## End(Not run)
```

plot1of15 *plots 1 of the 15 graphs of streamflow statistics on a single page*

Description

Part of the flowHistory system. The 15 graphs include annual and four seasonal graphs for each of 3 flow statistics: 1-day maximum, mean, and 7-day minimum

Usage

```
plot1of15(eList, yearStart, yearEnd, qf, istat, isBottom = FALSE)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
yearStart	A numeric value for the year in which the graph should start
yearEnd	A numeric value for the year in which the graph should end
qf	a scale factor to convert discharge in cubic feet per second to mm/day
istat	A numeric value selecting the flow statistic to be plotted, must be an integer from 1 to 8
isBottom	logical, if TRUE the graph is from the bottom row and thus needs x axis labels, if FALSE it does not need labels

Examples

```
eList <- Choptank_eList
plot1of15(eList, 1990, 2000, 0.2938476,5)
```

---

plotConcHist *Graph of annual concentration and flow normalized concentration versus year*

Description

Data come from named list, which contains a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

The annual concentrations are "time-weighted" mean concentrations (as opposed to "flow-weighted"). The annual results reported are for a specified "period of analysis" which can be an entire water year, a calendar, a season or even an individual month. User specifies this period of analysis in the call to setupYears.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
plotConcHist(eList, yearStart = NA, yearEnd = NA, concMax = NA,
  printTitle = TRUE, tinyPlot = FALSE, plotFlowNorm = TRUE, cex = 0.8,
  cex.axis = 1.1, cex.main = 1.1, lwd = 2, col = "black",
  col.pred = "green", customPar = FALSE, ...)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
yearStart	numeric is the calendar year containing the first estimated annual value to be plotted, default is NA (which allows it to be set automatically by the data)
yearEnd	numeric is the calendar year just after the last estimated annual value to be plotted, default is NA (which allows it to be set automatically by the data)

concMax	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
printTitle	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure)
tinyPlot	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
plotFlowNorm	logical variable if TRUE flow normalized line is plotted, if FALSE not plotted
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
lwd	number magnification of line width.
col	color of points on plot, see ?par 'Color Specification'
col.pred	color of flow normalized line on plot, see ?par 'Color Specification'
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

setupYears, genericEGRETDotPlot

Examples

```
yearStart <- 2001
yearEnd <- 2010
eList <- Choptank_eList
# Water year:
plotConcHist(eList, yearStart, yearEnd)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotConcHist(eList, yearStart, yearEnd)
```

---

plotConcPred *Plot of Observed Concentration versus Estimated Concentration*

Description

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
plotConcPred(eList, concMax = NA, logScale = FALSE, printTitle = TRUE,
  tinyPlot = FALSE, cex = 0.8, cex.axis = 1.1, cex.main = 1.1,
  customPar = FALSE, col = "black", lwd = 1, randomCensored = FALSE,
  ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
concMax	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
logScale	logical, default TRUE, TRUE indicates y axis is in log scale, "xy" indicates both x and y in log scale, "x" is only x
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
tinyPlot	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
randomCensored	logical. Show censored values as randomized.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotConcPred(eList)
plotConcPred(eList, logScale=TRUE)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotConcPred(eList)
```

plotConcQ *Plot of Observed Concentration versus Discharge*

Description

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata. Discharge is plotted on a log scale.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
plotConcQ(eList, qUnit = 2, tinyPlot = FALSE, logScale = FALSE,
  randomCensored = FALSE, concMax = NA, concMin = NA, printTitle = TRUE,
  cex = 0.8, cex.axis = 1.1, cex.main = 1.1, rmSciX = FALSE,
  rmSciY = FALSE, customPar = FALSE, col = "black", lwd = 1, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-part figure, default is FALSE.
logScale	logical if TRUE x and y plotted in log axis
randomCensored	logical. Show censored values as randomized.
concMax	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
concMin	numeric value for lower limit on concentration shown on the vertical log graph, default is NA (which causes the lower limit to be set automatically, based on the data). This value is ignored for linear scales, using 0 as the minimum value for the concentration axis.
printTitle	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure)
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
rmSciX	logical defaults to FALSE, changes x label from scientific to fixed
rmSciY	logical defaults to FALSE, changes y label from scientific to fixed
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.

col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotConcQ(eList)
plotConcQ(eList, logScale=TRUE)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotConcQ(eList)
```

---

plotConcQSmooth *Plot up to three curves representing the concentration versus discharge relationship. Each curve is a different point in time.*

Description

These plots are like a vertical slice of the estimated concentration surface that is seen in the plotContours function. These plots show how the concentration-discharge relationship is changing over time. Typically the time points selected would be in three years at the same time of year spaced out over the period of record. But that is not necessary. Another possibility is to use this to explore seasonal differences. In this case the three dates would be in the same year but different times during the year.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Usage

```
plotConcQSmooth(eList, date1, date2, date3, qLow, qHigh, qUnit = 2,
  legendLeft = 0, legendTop = 0, concMax = NA, concMin = NA,
  bw = FALSE, printTitle = TRUE, printValues = FALSE, minNumObs = 100,
  minNumUncen = 50, colors = c("black", "red", "green"),
  printLegend = TRUE, windowY = 7, windowQ = 2, windowS = 0.5,
  tinyPlot = FALSE, customPar = FALSE, lwd = 2, cex = 0.8,
  cex.axis = 1.1, cex.main = 1.1, cex.legend = 1.2, lineVal = c(1, 1,
  1), logScale = FALSE, edgeAdjust = TRUE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
date1	character specifying the date for the first curve on the graph, it is in the form "yyyy-mm-dd" (must be in quotes)
date2	character specifying the date for the second curve on the graph, it is in the form "yyyy-mm-dd" (must be in quotes). If only one curve is wanted this should be NA
date3	character specifying the date for the third curve on the graph, it is in the form "yyyy-mm-dd" (must be in quotes). If a third curve is not wanted this should be NA
qLow	numeric value for the lowest discharge to be considered, expressed in the units of discharge that are being used (as specified in qUnit)
qHigh	numeric value for the highest discharge to be considered, expressed in the units of discharge that are being used (as specified in qUnit)
qUnit	object of qUnit class. printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
legendLeft	numeric which represents the left edge of the legend in the units of the plot.
legendTop	numeric which represents the top edge of the legend in the units of the plot.
concMax	numeric value for upper limit on concentration shown on the graph, default = NA (which causes the upper limit to be set automatically, based on the data)
concMin	numeric value for lower limit on concentration shown on the vertical log graph, default is NA (which causes the lower limit to be set automatically, based on the data). This value is ignored for linear scales, using 0 as the minimum value for the concentration axis.
bw	logical if TRUE graph is produced in black and white, default is FALSE (which means it will use color)
printTitle	logical variable if TRUE title is printed, if FALSE not printed
printValues	logical variable if TRUE the results shown on the graph are also printed to the console and returned in a dataframe (this can be useful for quantifying the changes seen visually in the graph), default is FALSE (not printed)
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50
colors	color vector of lines on plot, see ?par 'Color Specification'. Defaults to c("black","red","green")
printLegend	logicalif TRUE, legend is included
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 7
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5

tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-part figure, default is FALSE.
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
lwd	number line width
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
cex.legend	magnification to be used for legend annotation relative to the current setting of cex
lineVal	vector of line types. Defaults to c(1,1,1) which is a solid line for each line. Options: 0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash
logScale	logical whether or not to use a log scale in the y axis.
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

genericEGRETDotPlot, runSurvReg

Examples

```
date1<-"2001-06-01"
date2<-"2005-06-01"
date3<-"2010-06-01"
qLow<-1
qHigh<-100
eList <- Choptank_eList
plotConcQSmooth(eList, date1,date2,date3,qLow,qHigh)
plotConcQSmooth(eList, date1,date2,date3,qLow,qHigh,logScale=TRUE)
```

---

plotConcTime *Plot of Observed Concentration versus Time*

Description

This function allows the user to plot all of the data, but also to limit it in two ways. The data can be limited to only those observed concentrations collected in a specified discharge range. The data can also be limited to only those observed in certain months of the year. These two selection criteria can be combined. For example, we may only want to plot data for discharges between 100 and 500 cubic feet per second in the months of March, April and May.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Usage

```
plotConcTime(eList, qUnit = 2, yearStart = NA, yearEnd = NA,
  qLower = NA, qUpper = NA, randomCensored = FALSE, tinyPlot = FALSE,
  concMax = NA, concMin = NA, printTitle = TRUE, logScale = FALSE,
  cex = 0.8, cex.axis = 1.1, cex.main = 1.1, customPar = FALSE,
  col = "black", lwd = 1, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
yearStart	numeric is the calendar year containing the first estimated annual value to be plotted, default is NA (which allows it to be set automatically by the data)
yearEnd	numeric is the calendar year just after the last estimated annual value to be plotted, default is NA (which allows it to be set automatically by the data)
qLower	numeric the lower bound on values of discharge used to select the data points to be plotted, units are those specified by qUnit, default = NA which is equivalent to a lower bound of zero but if the desired lower bound is zero use qLower = NA
qUpper	numeric the upper bound on values of discharge for selection of data points to be plotted, units are those specified by qUnit, default = NA which is equivalent to an upper bound of infinity
randomCensored	logical. Show censored values as randomized.
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-part figure, default is FALSE.
concMax	numeric value for the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
concMin	numeric value for lower limit on concentration shown on the vertical log graph, default is NA (which causes the lower limit to be set automatically, based on the data). This value is ignored for linear scales, using 0 as the minimum value for the concentration axis.
printTitle	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure).

logScale	logical, default FALSE, FALSE creates a linear scale y-axis, TRUE creates a y-axis is in log scale.
cex	numerical value giving the amount by which plotting symbols should be magnified.
cex.axis	magnification to be used for axis annotation relative to the current setting of cex.
cex.main	magnification to be used for main titles relative to the current setting of cex.
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function. (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width.
...	arbitrary functions sent to the generic plotting function. See ?par for details on possible parameters.

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotConcTime(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotConcTime(eList, qUnit = 1, qLower = 100, qUpper = 10000)
plotConcTime(eList, logScale=TRUE)
plotConcTime(eList, qUnit = 1, qLower = 100, qUpper = 10000, randomCensored = TRUE)
```

---

plotConcTimeDaily *Plot of the time series of daily concentration estimates and the sample values for the days that were sampled*

Description

This plot is useful for visual examination of the ability of the WRTDS, or other model, to fit the data, seen in a time-series perspective. The graph is most useful when it covers a period of just a few years and not the complete record but a complete record can be done by repeated use over a series of segments.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotConcTimeDaily(eList, yearStart = NA, yearEnd = NA, tinyPlot = FALSE,
  concMax = NA, printTitle = TRUE, cex = 0.8, cex.axis = 1.1,
  randomCensored = FALSE, cex.main = 1.1, customPar = FALSE,
  col = "black", lwd = 1, prettyDate = TRUE, ...)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
yearStart	numeric specifying the starting date (expressed as decimal years, for example 1989.0) for the plot
yearEnd	numeric specifiying the ending date for the plot
tinyPlot	logical variable, if TRUE plot is designed to be short and wide, default is FALSE.
concMax	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
printTitle	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure)
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
randomCensored	logical. Show censored values as randomized.
cex.main	magnification to be used for main titles relative to the current setting of cex
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
prettyDate	logical use 'pretty' limits for date axis if TRUE, or force the yearStart/yearEnd as limits if FALSE
...	arbitrary functions sent to the generic plotting function. See ?par for details on possible parameters

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotConcTimeDaily(eList)
plotConcTimeDaily(eList, yearStart=1998,yearEnd=2001)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotConcTimeDaily(eList)
```

plotConcTimeSmooth *Plot up to three curves representing the concentration versus time relationship, each curve representing a different flow.*

Description

These plots show how the concentration-time relationship is changing over flow.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data and an INFO dataframe with metadata.

Usage

```
plotConcTimeSmooth(eList, q1, q2, q3, centerDate, yearStart, yearEnd,
  qUnit = 2, legendLeft = 0, legendTop = 0, concMax = NA,
  concMin = NA, bw = FALSE, printTitle = TRUE, colors = c("black",
  "red", "green"), printValues = FALSE, tinyPlot = FALSE, minNumObs = 100,
  minNumUncen = 50, windowY = 10, windowQ = 2, windowS = 0.5,
  cex.main = 1.1, lwd = 2, printLegend = TRUE, cex.legend = 1.2,
  cex = 0.8, cex.axis = 1.1, customPar = FALSE, lineVal = c(1, 1, 1),
  logScale = FALSE, edgeAdjust = TRUE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
q1	numeric This is the discharge value for the first curve to be shown on the plot. It is expressed in units specified by qUnit.
q2	numeric This is the discharge value for the second curve to be shown on the plot. It is expressed in units specified by qUnit. If you don't want a second curve then the argument must be q2=NA
q3	numeric This is the discharge value for the third curve to be shown on the plot. It is expressed in units specified by qUnit. If you don't want a third curve then the argument must be q3=NA
centerDate	character This is the time of year to be used as the center date for the smoothing. It is expressed as a month and day and must be in the form "mm-dd"
yearStart	numeric This is the starting year for the graph. The first value plotted for each curve will be at the first instance of centerDate in the year designated by yearStart.
yearEnd	numeric This is the end of the sequence of values plotted on the graph.The last value will be the last instance of centerDate prior to the start of yearEnd. (Note, the number of values plotted on each curve will be yearEnd-yearStart.)
qUnit	object of qUnit class. printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
legendLeft	numeric which represents the left edge of the legend in the units of the plot.

legendTop	numeric which represents the top edge of the legend in the units of the plot.
concMax	numeric value for upper limit on concentration shown on the graph, default = NA (which causes the upper limit to be set automatically, based on the data)
concMin	numeric value for lower limit on concentration shown on the vertical log graph, default is NA (which causes the lower limit to be set automatically, based on the data). This value is ignored for linear scales, using 0 as the minimum value for the concentration axis.
bw	logical if TRUE graph is produced in black and white, default is FALSE (which means it will use color)
printTitle	logical variable if TRUE title is printed, if FALSE not printed
colors	color vector of lines on plot, see ?par 'Color Specification'. Defaults to c("black","red","green")
printValues	logical variable if TRUE the results shown on the graph are printed to the console and returned in a dataframe (this can be useful for quantifying the changes seen visually in the graph), default is FALSE (not printed)
tinyPlot	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 10
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5
cex.main	magnification to be used for main titles relative to the current setting of cex
lwd	line width, a positive number, defaulting to 1
printLegend	logicalif TRUE, legend is included
cex.legend	number magnification of legend
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
lineVal	vector of line types. Defaults to c(1,1,1) which is a solid line for each line. Options: 0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash
logScale	logical whether or not to use a log scale in the y axis.
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE.

...	arbitrary functions sent to the generic plotting function. See ?par for details on possible parameters

See Also

genericEGRETDotPlot, runSurvReg

Examples

```
q1 <- 10
q2 <- 25
q3 <- 75
centerDate <- "07-01"
yearStart <- 2000
yearEnd <- 2010
eList <- Choptank_eList
plotConcTimeSmooth(eList, q1, q2, q3, centerDate, yearStart, yearEnd)
plotConcTimeSmooth(eList, q1, q2, q3, centerDate, yearStart, yearEnd,logScale=TRUE)
```

---

plotContours *Color contour plot of the estimated surfaces as a function of discharge and time (surfaces include log concentration, standard error, and concentration)*

Description

These plots are normally used for plotting the estimated concentration surface (whatSurface=3) but can be used to explore the estimated surfaces for the log of concentration or for the standard error (in log space) which is what determines the bias correction. The plots are often more interpretable when the time limits are only about 4 years apart. To explore changes over a long time period it is best to do this multiple times, for various time slices of 4 years (for example) or to use the function plotDiffContours.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotContours(eList, yearStart, yearEnd, qBottom = NA, qTop = NA,
  whatSurface = 3, qUnit = 2, contourLevels = NA, span = 60,
  pval = 0.05, printTitle = TRUE, vert1 = NA, vert2 = NA, horiz = NA,
  tcl = 0.1, flowDuration = TRUE, customPar = FALSE, yTicks = NA,
  tick.lwd = 2, lwd = 1, cex.main = 1, cex.axis = 1,
  color.palette = colorRampPalette(c("white", "gray", "blue", "red")), ...)
```

Arguments

eList	named list with at least the Daily and INFO dataframes, and surfaces matrix
yearStart	numeric value for the starting date for the graph, expressed as decimal year (typically whole number such as 1989.0)
yearEnd	numeric value for the ending date for the graph, expressed as decimal year, (for example 1993.0)
qBottom	numeric value for the bottom edge of the graph, expressed in the units of discharge that are being used (as specified in qUnit). NA will choose a "pretty" lower limit nearest to the 5% of discharge. If yTicks are specified, then the first value of yTicks becomes the lowest discharge shown on the figure.
qTop	numeric value for the top edge of the graph, expressed in the units of discharge that are being used (as specified in qUnit). NA will choose a "pretty" upper limit nearest to the 95% of discharge. If yTicks are specified, then the last value of yTicks becomes the highest discharge shown on the figure.
whatSurface	numeric value, can only accept 1, 2, or 3; whatSurface=1 is yHat (log concentration), whatSurface=2 is SE (standard error of log concentration), and whatSurface=3 is ConcHat (unbiased estimate of concentration), default = 3.
qUnit	object of qUnit class. printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
contourLevels	numeric vector containing the contour levels for the contour plot, arranged in ascending order, default is NA (which causes the contour levels to be set automatically, based on the data)
span	numeric, it is the half-width (in days) of the smoothing window for computing the flow duration information, default = 60
pval	numeric, the probability value for the lower flow frequency line on the graph
printTitle	logical variable if TRUE title is printed, if FALSE not printed
vert1	numeric, the location in time for a black vertical line on the figure, yearStart<vert1<yearEnd, default is NA (vertical line is not drawn)
vert2	numeric, the location in time for a black vertical line on the figure, yearStart<vert2<yearEnd, default is NA (vertical line is not drawn)
horiz	numeric, the location in discharge for a black horizontal line on the figure, qBottom<vert1<qTop, default is NA (no horizontal line is drawn)
tcl	numeric, length of tick marks in inches, default is 0.1
flowDuration	logical variable if TRUE plot the flow duration lines (5 and 95 flow percentiles), if FALSE do not plot them, default = TRUE
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins.
yTicks	vector of yTick labels and marks that will be plotted in log space. (for example yTicks = c(3, 5, 10, 20, 50, 100, 200, 400). The first and last values determine the range of the y axis. If NA, the tick marks will be automatically generated.
tick.lwd	line width for axis ticks, default is 2

lwd	numeric, line width of flowDuration curve, default is 1
cex.main	magnification to be used for main titles relative to the current setting of cex
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
color.palette	a function that creates a color palette for the contour plot. Default goes from white to gray to blue to red using the function colorRampPalette(c("white","gray","blue","red")) A few preset options are heat.colors, topo.colors, and terrain.colors.
...	arbitrary functions sent to the generic plotting function. See ?par for details on possible parameters

Examples

```
yearStart <- 2001
yearEnd <- 2010
qBottom <- 0.5
qTop<- 22
clevel <- seq(0,3.5,0.5)
eList <- Choptank_eList
plotContours(eList, yearStart,yearEnd,qBottom,qTop, contourLevels = clevel)
plotContours(eList, yearStart,yearEnd,qBottom=0.1,qTop=NA, contourLevels = clevel)
yTicksModified <- c(.1,1,10,25)
plotContours(eList, yearStart,yearEnd,qBottom,qTop,
      contourLevels = clevel,yTicks=yTicksModified,flowDuration=FALSE)
colors <- colorRampPalette(c("white","red"))
plotContours(eList, yearStart,yearEnd,qBottom,qTop,
       contourLevels = clevel,yTicks=yTicksModified,
       color.palette=colors,flowDuration=FALSE)
colors2 <- heat.colors # Some other options: topo.colors, terrain.colors, cm.colors
plotContours(eList, yearStart,yearEnd,qBottom,qTop,
       contourLevels = clevel,color.palette=colors2,lwd=2,flowDuration=FALSE)
plotContours(eList, yearStart,yearEnd,qBottom,qTop,
       contourLevels = clevel,cex.axis=2,flowDuration=FALSE)
par(mar=c(5,8,5,8))
plotContours(eList, yearStart,yearEnd,qBottom,qTop,
       contourLevels = clevel,customPar=TRUE,printTitle=FALSE,flowDuration=FALSE)
```

---

plotDiffContours *Plots the difference between two years from a contour plot created by plotContours*

Description

These plots are normally used for plotting changes in the estimated concentration surface (whatSurface=3) but can be used to explore the changes in estimated surfaces for the log of concentration or for the standard error (in log space) which is what determines the bias correction.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotDiffContours(eList, year0, year1, qBottom = NA, qTop = NA,
  maxDiff = NA, whatSurface = 3, tcl = 0.1, qUnit = 2, span = 60,
  pval = 0.05, printTitle = TRUE, plotPercent = FALSE, vert1 = NA,
  vert2 = NA, horiz = NA, flowDuration = TRUE, yTicks = NA,
  tick.lwd = 2, lwd = 1, cex.main = 0.95, cex.axis = 1,
  customPar = FALSE, color.palette = colorRampPalette(c("blue", "white",
  "red")), ...)
```

Arguments

eList	named list with at least the Daily and INFO dataframes, and surfaces matrix
year0	numeric value for the calendar year that is the first year of the pair of years for the analysis, should be a whole number
year1	numeric value for the calendar year that is the second year of the pair of years for the analysis, should be a whole number
qBottom	numeric value for the bottom edge of the graph, expressed in the units of discharge that are being used (as specified in qUnit). NA will choose a "pretty" lower limit nearest to the 5% of discharge. If yTicks are specified, then the first value of yTicks becomes the lowest discharge shown on the figure.
qTop	numeric value for the top edge of the graph, expressed in the units of discharge that are being used (as specified in qUnit). NA will choose a "pretty" upper limit nearest to the 95% of discharge. If yTicks are specified, then the last value of yTicks becomes the highest discharge shown on the figure.
maxDiff	numeric value which is the absolute value of the largest change in concentration that will be shown on the figure. Alternatively, a vector with the minimum and maximum values in the change in concentration scale. If NA, the scale will be set from 5% to 95% of the concentration difference.
whatSurface	numeric value, can only accept 1, 2, or 3; whatSurface=1 is yHat (log concentration), whatSurface=2 is SE (standard error of log concentration), and whatSurface=3 is ConcHat (unbiased estimate of concentration), default = 3
tcl	numeric, length of tick marks in inches, default is 0.1
qUnit	object of qUnit class. printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
span	numeric, it is the half-width (in days) of the smoothing window for computing the flow duration information, default = 60
pval	numeric, the probability value for the lower flow frequency line on the graph
printTitle	logical variable if TRUE title is printed, if FALSE not printed
plotPercent	logical. If TRUE, plots percent difference, if FALSE, plots absolute differences. Defaults to FALSE.
vert1	numeric, the location in time for a black vertical line on the figure, yearStart < vert1 < yearEnd, default is NA (vertical line is not drawn)
vert2	numeric, the location in time for a black vertical line on the figure, yearStart < vert2 < yearEnd, default is NA (vertical line is not drawn)

horiz	numeric, the location in discharge for a black horizontal line on the figure, qBottom<vert1<qTop, default is NA (no horizontal line is drawn)
flowDuration	logical variable if TRUE plot the flow duration lines (5 and 95 flow percentiles), if FALSE do not plot them, default = TRUE
yTicks	vector of yTick labels and marks that will be plotted in log space. (for example yTicks = c(3, 5, 10, 20, 50, 100, 200, 400). The first and last values determine the range of the y axis. If NA, the tick marks will be automatically generated.
tick.lwd	line width for axis ticks, default is 2
lwd	numeric, line width of flowDuration curve, default is 1
cex.main	magnification to be used for main titles relative to the current setting of cex
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins.
color.palette	a function that creates a color palette for the contour plot. Default goes from blue to white to red using the function colorRampPalette(c("blue","white","red")). A few preset options are heat.colors, topo.colors, and terrain.colors.
...	arbitrary functions sent to the generic plotting function. See ?par for details on possible parameters

Examples

```
year0<-2001
year1<-2009
qBottom<-0.33
qTop<-22
maxDiff<-0.5
eList <- Choptank_eList
plotDiffContours(eList, year0,year1)
plotDiffContours(eList, year0,year1,maxDiff=maxDiff)
plotDiffContours(eList, year0,year1,qBottom,qTop,maxDiff)

yTicksModified <- c(.1,1,10,25)
plotDiffContours(eList, year0, year1,qBottom,qTop,maxDiff,
       yTicks=yTicksModified,flowDuration=FALSE)
colors <-colorRampPalette(c("blue","white","red"))
plotDiffContours(eList, year0,year1,qBottom,qTop,maxDiff,
       color.palette=colors,flowDuration=FALSE)
colors2 <- heat.colors # Some other options: topo.colors, terrain.colors, cm.colors
plotDiffContours(eList, year0,year1,qBottom,qTop,maxDiff,
       lwd=2,color.palette=colors2,flowDuration=FALSE)
plotDiffContours(eList, year0,year1,qBottom,qTop,maxDiff,cex.lab=2,flowDuration=FALSE)
par(mar=c(5,8,5,8))
plotDiffContours(eList, year0,year1,qBottom,qTop,maxDiff,
       customPar=TRUE,flowDuration=FALSE)
```

plotFlowSingle *Creates a plot of a time series of a particular flow statistic and a lowess smooth of that flow statistic*

Description

A part of the flowHistory system. The index of the flow statistics is istat. These statistics are: (1) 1-day minimum, (2) 7-day minimum, (3) 30-day minimum, (4) median (5) mean, (6) 30-day maximum, (7) 7-day maximum, and (8) 1-day maximum

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotFlowSingle(eList, istat, yearStart = NA, yearEnd = NA, qMax = NA,
  printTitle = TRUE, tinyPlot = FALSE, customPar = FALSE,
  runoff = FALSE, qUnit = 1, printStaName = TRUE, printPA = TRUE,
  printIstat = TRUE, cex = 0.8, cex.axis = 1.1, cex.main = 1.1,
  lwd = 2, col = "black", ...)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
istat	A numeric value for the flow statistic to be graphed (possible values are 1 through 8)
yearStart	A numeric value for year in which the graph should start, default is NA, which indicates that the graph should start with first annual value
yearEnd	A numeric value for year in which the graph should end, default is NA, which indicates that the graph should end with last annual value
qMax	A numeric value for the maximum value to be used for y-axis of graph, default is NA means that graph is self-scaling
printTitle	logical variable, if TRUE title is printed, if FALSE title is not printed, default is TRUE
tinyPlot	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
runoff	logical variable, if TRUE the streamflow data are converted to runoff values in mm/day
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.

printStaName	logical variable, if TRUE station name is printed in title, if FALSE not printed, default is TRUE
printPA	logical variable, if TRUE Period of Analysis information is printed in title, if FALSE not printed, default is TRUE
printIstat	logical variable, if TRUE print the statistic name is printed in title, if FALSE not printed, default is TRUE
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
lwd	number line width
col	color of points on plot, see ?par 'Color Specification'
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

makeAnnualSeries, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotFlowSingle(eList, 1)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotFlowSingle(eList, 1)
## Not run:
siteNumber <- '01010000'
StartDate <- ''
EndDate <- '2014-10-01'
Daily <- readNWISDaily(siteNumber, '00060', StartDate, EndDate)
INFO <- readNWISInfo(siteNumber, '00060', interactive = FALSE)
eList <- as.egret(INFO, Daily)
plotFlowSingle(eList, 5)

## End(Not run)
```

---

plotFluxHist *Graph of annual flux and flow normalized flux versus year*

Description

The annual results reported are for a specified "period of analysis" which can be an entire water year, a calendar, a season or even an individual month. The user specifies this period of analysis in the call to setupYears.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotFluxHist(eList, yearStart = NA, yearEnd = NA, fluxUnit = 9,
  fluxMax = NA, printTitle = TRUE, plotFlowNorm = TRUE,
  tinyPlot = FALSE, col = "black", col.pred = "green", cex = 0.8,
  cex.axis = 1.1, cex.main = 1.1, lwd = 2, customPar = FALSE, ...)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
yearStart	numeric is the calendar year containing the first estimated annual value to be plotted, default is NA (which allows it to be set automatically by the data)
yearEnd	numeric is the calendar year just after the last estimated annual value to be plotted, default is NA (which allows it to be set automatically by the data)
fluxUnit	number representing entry in pre-defined fluxUnit class array. printFluxUnitCheatSheet
fluxMax	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
printTitle	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure)
plotFlowNorm	logical variable if TRUE the flow normalized line is plotted, if FALSE not plotted
tinyPlot	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
col	color of points on plot, see ?par 'Color Specification'
col.pred	color of flow normalized line on plot, see ?par 'Color Specification'
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
lwd	number line width
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

setupYears

Examples

```
yearStart <- 2001
yearEnd <- 2010
eList <- Choptank_eList
# Water year:
plotFluxHist(eList)
plotFluxHist(eList, yearStart, yearEnd, fluxUnit = 1)
plotFluxHist(eList, yearStart, yearEnd, fluxUnit = 'kgDay')
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotFluxHist(eList)
```

---

plotFluxPred *Graph of observed versus estimated flux*

Description

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Usage

```
plotFluxPred(eList, fluxUnit = 3, fluxMax = NA, printTitle = TRUE,
  oneToOneLine = TRUE, customPar = FALSE, col = "black", lwd = 1,
  cex = 0.8, cex.axis = 1.1, cex.main = 1.1, tinyPlot = FALSE,
  logScale = FALSE, randomCensored = FALSE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
fluxUnit	number representing entry in pre-defined fluxUnit class array. printFluxUnitCheatSheet
fluxMax	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
oneToOneLine	inserts 1:1 line
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
col	color of points on plot, see ?par 'Color Specification'

lwd	number line width
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
tinyPlot	logical variable if TRUE plot is designed to be small, if FALSE it is designed for page size, default is FALSE (not fully implemented yet)
logScale	logical if TRUE x and y plotted in log axis
randomCensored	logical. Show censored values as randomized.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotFluxPred(eList)
plotFluxPred(eList, fluxUnit = 'poundsDay')
plotFluxPred(eList, logScale=TRUE)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotFluxPred(eList)
```

---

plotFluxQ *Sample data plot: observed log flux vs log discharge*

Description

Concentration and discharge data used to compute flux come from a data frame named Sample which contains the sample data. The metadata come from a data frame named INFO.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Usage

```
plotFluxQ(eList, qUnit = 2, logScale = TRUE, fluxUnit = 3,
  tinyPlot = FALSE, fluxMax = NA, fluxMin = NA, col = "black",
  lwd = 1, printTitle = TRUE, cex = 0.8, cex.axis = 1.1,
  cex.main = 1.1, customPar = FALSE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
qUnit	object of qUnit class. printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
logScale	logical, default TRUE, TRUE creates a log-log scale, FALSE creates an arithmatic scale.
fluxUnit	object of fluxUnit class. printFluxUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
tinyPlot	logical variable if TRUE plot is designed to fit into a multi-plot array, default is FALSE
fluxMax	numeric specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
fluxMin	numeric specifying the minimum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotFluxQ(eList, qUnit = 1, fluxUnit = 1)
plotFluxQ(eList, fluxUnit = 'kgDay')
plotFluxQ(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotFluxQ(eList)
```

`plotFluxTimeDaily` *Plot of the time series of daily flux estimates and the sample values for the days that were sampled*

Description

This plot is useful for visual examination of the ability of the WRTDS, or other model, to fit the data, as seen in a time-series perspective.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotFluxTimeDaily(eList, yearStart = NA, yearEnd = NA, tinyPlot = FALSE,
  fluxUnit = 3, fluxMax = NA, printTitle = TRUE, cex = 0.8,
  cex.axis = 1.1, cex.main = 1.1, customPar = FALSE, col = "black",
  lwd = 1, prettyDate = TRUE, ...)
```

Arguments

`eList`	named list with at least the Daily, Sample, and INFO dataframes
`yearStart`	numeric specifying the starting date (expressed as decimal years, for example 1989.0) for the plot
`yearEnd`	numeric specifiying the ending date for the plot
`tinyPlot`	logical variable, if TRUE plot is designed to be short and wide, default is FALSE.
`fluxUnit`	number representing in pre-defined fluxUnit class array. `printFluxUnitCheatSheet`
`fluxMax`	number specifying the maximum value to be used on the vertical axis, default is NA (which allows it to be set automatically by the data)
`printTitle`	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure)
`cex`	numerical value giving the amount by which plotting symbols should be magnified
`cex.axis`	magnification to be used for axis annotation relative to the current setting of cex
`cex.main`	magnification to be used for main titles relative to the current setting of cex
`customPar`	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
`col`	color of points on plot, see ?par 'Color Specification'
`lwd`	number line width
`prettyDate`	logical use 'pretty' limits for date axis if TRUE, or force the yearStart/yearEnd as limits if FALSE
`...`	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotFluxTimeDaily(eList)
plotFluxTimeDaily(eList, 2001,2009)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotFluxTimeDaily(eList)
```

---

plotFour *Makes four graphs of streamflow statistics on a single page*

Description

Part of the flowHistory system. The four statistics are 1-day maximum, annual mean, annual 7-day minimum, and the running standard deviation of the log daily discharge values.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotFour(eList, yearStart = NA, yearEnd = NA, printTitle = TRUE,
  runoff = FALSE, qUnit = 1, window = 15, cex = 0.8, cex.axis = 1.2,
  cex.main = 1.2, col = "black", lwd = 1, ...)
```

Arguments

eList	named list with at least Daily and INFO dataframes
yearStart	A numeric value for year in which the graph should start, default is NA, which indicates that the graph should start with first annual value
yearEnd	A numeric value for year in which the graph should end, default is NA, which indicates that the graph should end with last annual value
printTitle	logical variable, if TRUE title is printed, if FALSE title is not printed, default is TRUE
runoff	logical variable, if TRUE the streamflow data are converted to runoff values in mm/day
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
window	numeric which is the full width, in years, of the time window over which the standard deviation is computed, default = 15

cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

plotFlowSingle

Examples

```
eList <- Choptank_eList
## Not run:
#Water year:
plotFour(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList,paStart=6,paLong=3)
plotFour(eList)

## End(Not run)
```

---

plotFourStats *Makes four graphs of annual streamflow statistics on a single page*

Description

Part of the flowHistory system. The four statistics are 1-day maximum, annual mean, annual median, and annual 7-day minimum. Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotFourStats(eList, yearStart = NA, yearEnd = NA, printTitle = TRUE,
  runoff = FALSE, cex.main = 1.2, qUnit = 1, cex.axis = 1.2,
  cex = 0.8, col = "black", lwd = 1, ...)
```

Arguments

eList	named list with at least Daily and INFO dataframes
yearStart	A numeric value for year in which the graph should start, default is NA, which indicates that the graph should start with first annual value
yearEnd	A numeric value for year in which the graph should end, default is NA, which indicates that the graph should end with last annual value
printTitle	logical variable, if TRUE title is printed, if FALSE title is not printed, default is TRUE
runoff	logical variable, if TRUE the streamflow data are converted to runoff values in mm/day
cex.main	magnification to be used for main titles relative to the current setting of cex
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex	numerical value giving the amount by which plotting symbols should be magnified
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

plotFlowSingle

Examples

```
eList <- Choptank_eList
## Not run:
# Water year:
plotFourStats(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList,paStart=6,paLong=3)
plotFourStats(eList)

## End(Not run)
```

`plotQTimeDaily` *Plot of the discharge time series*

Description

Part of flowHistory component. Allows discharge record to only show those discharges above a given threshold

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotQTimeDaily(eList, yearStart = NA, yearEnd = NA, qLower = NA,
  qUnit = 1, logScale = FALSE, tinyPlot = FALSE, printTitle = TRUE,
  lwd = 3, col = "red", cex.main = 1.2, cex.lab = 1.2,
  customPar = FALSE, prettyDate = TRUE, ...)
```

Arguments

`eList`	named list with at least the Daily and INFO dataframes
`yearStart`	numeric indicating the starting year for the graph
`yearEnd`	numeric indicating the ending year for the graph (should be a time in decimal years that is after the last observations to be plotted)
`qLower`	numeric specifying the lower bound on discharges that are to be plotted, must be in the units specified by qUnit, default is NA (lower bound is zero)
`qUnit`	object of qUnit class. `printqUnitCheatSheet`, or numeric represented the short code, or character representing the descriptive name. Default is qUnit=1 (cubic feet per second)
`logScale`	logical whether or not to use a log scale in the y axis. Default is FALSE.
`tinyPlot`	logical variable, if TRUE plot is designed to be short and wide, default is FALSE.
`printTitle`	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure)
`lwd`	line width, a positive number, defaulting to 1
`col`	specification for the default plotting color
`cex.main`	magnification to be used for main titles relative to the current setting of cex
`cex.lab`	magnification to be used for x and y labels relative to the current setting of cex
`customPar`	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
`prettyDate`	logical use 'pretty' limits for date axis if TRUE, or force the yearStart/yearEnd as limits if FALSE
`...`	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotQTimeDaily(eList)
plotQTimeDaily(eList, yearStart=1990, yearEnd=2000,qLower=1500)
plotQTimeDaily(eList, prettyDate=FALSE)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotQTimeDaily(eList)
```

---

plotResidPred — *Plot of the residuals from WRTDS versus the estimated values (all in log concentration units)*

Description

This function produces a plot of the residuals from WRTDS, expressed in natural log concentration units versus the estimated values, also in natural log concentration units. These estimates are the log-space estimates prior to bias-correction. The function provides an alternative for viewing the standardized residuals, where the each residual is divided by its estimated standard error.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Usage

```
plotResidPred(eList, stdResid = FALSE, tinyPlot = FALSE,
  printTitle = TRUE, col = "black", lwd = 1, cex = 0.8,
  cex.axis = 1.1, cex.main = 1.1, customPar = FALSE,
  randomCensored = FALSE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
stdResid	logical variable, if TRUE it uses the standardized residual, if FALSE it uses the actual, default is FALSE
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-part figure, default is FALSE.
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
col	color of points on plot, see ?par 'Color Specification'

lwd	number line width
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for x and y labels relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
randomCensored	logical. Show censored residuals as randomized.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotResidPred(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotResidPred(eList)
```

---

plotResidQ — *Plot of the residuals from WRTDS (in log concentration units) versus the discharge*

Description

This function produces a plot of the residuals from WRTDS, expressed in natural log concentration units versus the discharge shown on a log scale. The function also provides an alternative for viewing the standardized residuals, where the each residual is divided by its estimated standard error

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Usage

```
plotResidQ(eList, qUnit = 2, tinyPlot = FALSE, stdResid = FALSE,
  printTitle = TRUE, col = "black", lwd = 1, cex = 0.8,
  cex.axis = 1.1, cex.main = 1.1, rmSciX = FALSE, customPar = FALSE,
  randomCensored = FALSE, ...)
```

Arguments

eList	named list with at least the Sample and INFO dataframes
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
tinyPlot	logical variable, if TRUE plot is designed to be plotted small as part of a multi-part figure, default is FALSE.
stdResid	logical variable, if TRUE it uses the standardized residual, if FALSE it uses the actual, default is FALSE
printTitle	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
col	color of points on plot, see ?par 'Color Specification'
lwd	number line width
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
cex.main	magnification to be used for main titles relative to the current setting of cex
rmSciX	logical defaults to FALSE, changes x label from scientific to fixed
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
randomCensored	logical. Show censored residuals as randomized.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotResidQ(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotResidQ(eList)
```

`plotResidTime` *Plot of the residuals from WRTDS (in log concentration units) versus time*

Description

This function produces a plot of the residuals from WRTDS, expressed in natural log concentration units versus time. It also provides an alternative for viewing the standardized residuals, where the each residual is divided by its estimated standard error.

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Sample dataframe with the sample data, and an INFO dataframe with metadata.

Usage

```
plotResidTime(eList, stdResid = FALSE, printTitle = TRUE, hLine = TRUE,
  tinyPlot = FALSE, col = "black", lwd = 1, cex = 0.8, cex.axis = 1.1,
  cex.main = 1.1, customPar = FALSE, randomCensored = FALSE, ...)
```

Arguments

`eList`	named list with at least the Sample and INFO dataframes
`stdResid`	logical variable, if TRUE it uses the standardized residual, if FALSE it uses the actual, default is FALSE
`printTitle`	logical variable if TRUE title is printed, if FALSE not printed (this is best for a multi-plot figure)
`hLine`	inserts horizontal line at zero
`tinyPlot`	logical variable, if TRUE plot is designed to be plotted small, as a part of a multipart figure, default is FALSE
`col`	color of points on plot, see ?par 'Color Specification'
`lwd`	number line width
`cex`	numerical value giving the amount by which plotting symbols should be magnified
`cex.axis`	magnification to be used for axis annotation relative to the current setting of cex
`cex.main`	magnification to be used for main titles relative to the current setting of cex
`customPar`	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
`randomCensored`	logical. Show censored residuals as randomized.
`...`	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
# Water year:
plotResidTime(eList)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotResidTime(eList)
```

---

plotSDLogQ *Graph of the standard deviation of the log of daily discharge versus year*

Description

Graph of the standard deviation of the log of daily discharge versus year

Although there are a lot of optional arguments to this function, most are set to a logical default.

Data come from named list, which contains a Daily dataframe with the daily flow data, and an INFO dataframe with metadata.

Usage

```
plotSDLogQ(eList, yearStart = NA, yearEnd = NA, window = 15, sdMax = NA,
  printTitle = TRUE, tinyPlot = FALSE, printStaName = TRUE,
  printPA = TRUE, cex = 0.8, cex.main = 1.1, cex.axis = 1.1, lwd = 2,
  customPar = FALSE, ...)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
yearStart	numeric is the calendar year of the first value to be included in graph, default is NA, which plots from the start of the period of record
yearEnd	numeric is the calendar year of the last value to be included in graph, default is NA, which plots to the end of the period of record
window	numeric which is the full width, in years, of the time window over which the standard deviation is computed, default = 15
sdMax	numeric is the maximum value to be used on the vertical axis of the graph, default is NA (which allows it to be set automatically by the data)
printTitle	logical variable if TRUE title is printed, if FALSE title is not printed (this is best for a multi-plot figure), default is TRUE
tinyPlot	logical variable if TRUE plot is designed to be small, if FALSE it is designed for page size, default is FALSE (not fully implemented yet)

printStaName	logical variable, if TRUE print the station name, if FALSE do not, default is TRUE
printPA	logical variable, if TRUE print the period of analysis information in the plot title, if FALSE leave it out, default is TRUE
cex	numerical value giving the amount by which plotting symbols should be magnified
cex.main	magnification to be used for main titles relative to the current setting of cex
cex.axis	magnification to be used for axis annotation relative to the current setting of cex
lwd	line width, a positive number, defaulting to 1
customPar	logical defaults to FALSE. If TRUE, par() should be set by user before calling this function (for example, adjusting margins with par(mar=c(5,5,5,5))). If customPar FALSE, EGRET chooses the best margins depending on tinyPlot.
...	arbitrary graphical parameters that will be passed to genericEGRETDotPlot function (see ?par for options)

See Also

selectDays, genericEGRETDotPlot

Examples

```
eList <- Choptank_eList
## Not run:
# Water year:
plotSDLogQ(eList)
plotSDLogQ(eList, 1998,2000)
# Graphs consisting of Jun-Aug
eList <- setPA(eList, paStart=6,paLong=3)
plotSDLogQ(eList)

## End(Not run)
```

---

populateConcentrations

Populate Concentration Columns

Description

Creates ConcLow, ConcHigh, Uncen (0 if censored, 1 if uncensored) columns for Sample data frame for WRTDS study.

Usage

```
populateConcentrations(rawData)
```

Arguments

rawData	vector with value and code columns

Value

concentrationColumns dataframe

Examples

```
code <- c("","<","")
value <- c(1,2,3)
dataInput <- data.frame(value, code, stringsAsFactors=FALSE)
concentrationDF <- populateConcentrations(dataInput)
```

---

populateDaily *Populate Daily data frame*

Description

Using raw data that has at least dateTime, value, code, populates the rest of the basic Daily data frame used in WRTDS

Usage

```
populateDaily(rawData, qConvert, interactive = TRUE)
```

Arguments

rawData	dataframe contains at least dateTime, value, code columns
qConvert	character conversion to cubic meters per second
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

A data frame 'Daily' with the following columns:

Name	Type	Description
Q	numeric	Discharge in m^3/s
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850
Qualifier	character	Qualifying code
i	integer	Index of days, starting with 1
LogQ	numeric	Natural logarithm of Q

Q7	numeric	7 day running average of Q
Q30	numeric	30 day running average of Q

Author(s)

Robert M. Hirsch <rhirsch@usgs.gov>

See Also

readNWISDaily, readUserDaily

Examples

```
dateTime <- as.character(seq(as.Date("2001/1/1"),
         as.Date("2001/12/31"), by = "day"))
value <- 1:365
code <- rep("",365)
dataInput <- data.frame(dateTime, value, code, stringsAsFactors=FALSE)
Daily <- populateDaily(dataInput, 2)
```

---

populateDateColumns *Populate Date Columns*

Description

Creates various date columns for WRTDS study.

Usage

```
populateDateColumns(rawData)
```

Arguments

rawData	vector with dateTime

Value

DateFrame dataframe

Examples

```
dateTime <- c('1984-02-28 13:56', '1984-03-01', '1986-03-01',"1986-10-15")
expandedDateDF <- populateDateColumns(dateTime)
```

`populateParameterINFO` *Populate Parameter Information Columns*

Description

Populates INFO data frame with additional user-supplied information concerning the measured parameter.

Usage

```
populateParameterINFO(parameterCd, INFO, interactive = TRUE)
```

Arguments

`parameterCd`	character USGS parameter code
`INFO`	dataframe with value and code columns. Default is INFO
`interactive`	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

INFO dataframe

Examples

```
## Not run:
library(dataRetrieval)
INFO <- readNWISsite('01594440')
parameterCd <- "01075"
parameterData <- readNWISpCode(parameterCd)
INFO$param.nm <- parameterData$parameter_nm
INFO$param.units <- parameterData$parameter_units
INFO$paramShortName <- parameterData$srsname
INFO$paramNumber <- parameterData$parameter_cd

INFO <- populateParameterINFO(parameterCd, INFO)

## End(Not run)
```

populateSampleColumns *Populate Sample Columns*

Description

Creates ConcAve and ConcLow based on Uncen. Removes any samples with NA values in ConcHigh

Usage

```
populateSampleColumns(rawData)
```

Arguments

rawData	dataframe with dateTime, ConcLow, ConcHigh, Uncen

Value

Sample2 dataframe

Examples

```
dateTime <- c('1985-01-01', '1985-01-02', '1985-01-03')
ConcLow <- c(1,2,0)
ConcHigh <- c(1,2,3)
Uncen <- c(1,1,0)
dataInput <- data.frame(dateTime, ConcLow, ConcHigh, Uncen, stringsAsFactors=FALSE)
Sample <- populateSampleColumns(dataInput)
```

populateSiteINFO *Populate Site Information Columns*

Description

Populates INFO data frame with additional user-supplied information. Also removes fields not related to WRTDS study.

Usage

```
populateSiteINFO(INFO, siteNumber, interactive = TRUE)
```

Arguments

INFO	dataframe with value and code columns
siteNumber	character USGS site number
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

INFO dataframe

Examples

```
## Not run:
library(dataRetrieval)
INFO <- readNWISsite('01594440')
siteNumber <- "01594440"
siteINFO <- populateSiteINFO(INFO, siteNumber)

## End(Not run)
```

`print.egret` *Prints EGRET object*

Description

Print function for EGRET object

Usage

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

Arguments

`x`	EGRET object
`...`	additional parameters

Value

logical

Examples

```
eList <- Choptank_eList
eList
```

`printFluxUnitCheatSheet`
Available Flux Unit properties

Description

Cheat sheet to print out pre-defined flux unit properties from fluxUnit class Flux units included:

Number	ObjectName	shortName	unitFactor	unitName
1	POUNDS_DAY	lbs/day	2.204623	pounds/day
2	TONS_DAY	tons/day	0.001102	tons/day
3	KG_DAY	kg/day	1	kg/day
4	THOUSAND_KG_DAY	10^3 kg/day	0.001	thousands of kg/day
5	TONS_YEAR	tons/yr	0.402619	tons/year
6	THOUSAND_TONS_YEAR	10^3 tons/yr	0.000402619	thousands of tons/year
7	MILLION_TONS_YEAR	10^6 tons/yr	4.02619e-07	millions of tons/year
8	THOUSAND_KG_YEAR	10^3 kg/yr	0.36525	thousands of kg/year
9	MILLION_KG_YEAR	10^6 kg/yr	0.00036525	millions of kg/year
10	BILLION_KG_YEAR	10^9 kg/yr	3.6525e-07	billions of kg/year
11	thousandTonsDay	10^3 tons/day	1.102e-06	thousands of tons/day
12	millionKgDay	10^6 kg/day	1e-06	millions of kg/day
13	kgYear	kg/year	365.25	kg/year

Usage

```
printFluxUnitCheatSheet()
```

Examples

```
printFluxUnitCheatSheet()
```

`printqUnitCheatSheet` *Available Flow Unit properties*

Description

Cheat sheet to print out pre-defined qUnit properties from qUnit class. Flow units included:

Number	ObjectName	shortName	unitFactor
1	cfs	Cubic Feet per Second	35.31467
2	cms	Cubic Meters per Second	1
3	thousandCfs	Thousand Cubic Feet per Second	0.03531467
4	thousandCms	Thousand Cubic Meters per Second	0.001
5	mmDay	mm per day	
6	mmYear	mm per year	

Usage

```
printqUnitCheatSheet()
```

Examples

```
printqUnitCheatSheet()
```

printSeries *Print annual results for a given streamflow statistic*

Description

Part of the flowHistory system. The index of the flow statistics is istat. These statistics are: (1) 1-day minimum, (2) 7-day minimum, (3) 30-day minimum, (4) median (5) mean, (6) 30-day maximum, (7) 7-day maximum, and (8) 1-day maximum.

Usage

```
printSeries(eList, istat, qUnit = 1, runoff = FALSE)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
istat	A numeric value for the flow statistic to be graphed (possible values are 1 through 8)
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
runoff	logical variable, if TRUE the streamflow data are converted to runoff values in mm/day

Value

data frame with:

years	integer	year
qActual	numeric	Actual flow statistic (based on istat)
qSmooth	numeric	Smoothed flow statistic

Examples

```
eList <- Choptank_eList
printReturn <- printSeries(eList, 5)
```

processQWData *Processing of USGS NWIS Water Quality Data*

Description

Processes water quality portal data. This function looks at detection limit and detection conditions to determine if a value is left censored or not. Censored values are given the qualifier "<". The dataframe is also converted from a long to wide format.

Usage

```
processQWData(data, pCode = TRUE)
```

Arguments

data	dataframe from Water Quality Portal
pCode	logical if TRUE, assume data came from a pCode search, if FALSE, characteristic name.

Value

data dataframe with first column dateTime, and at least one qualifier and value columns (subsequent qualifier/value columns could follow depending on the number of parameter codes)

See Also

readWQPqw

Examples

```
## Not run:
library(dataRetrieval)

rawSample <- readWQPqw('USGS-01594440','', '', '')
rawSampleSelect <- processQWData(rawSample)

rawWQP <- readWQPqw('21FLEECO_WQX-IMPRGR80','Phosphorus', '', '')
Sample2 <- processQWData(rawWQP)

## End(Not run)
```

---

qUnit-class *qUnit class*

Description

Some details about the qUnit class

Details

qshortName A character specifying the short name.

qUnitFactor A numeric representing the conversion factor

qUnitName A character specifying the full name.

qUnitExpress An expression specifying the full name.

qUnitTiny An expression specifying the abbreviated name.

shortCode A number for quick lookup

readDataFromFile *Basic Data Import for Water Flow Data*

Description

Imports data from user-supplied data file. Specifically used to import water flow data for use in the EGRET package. For EGRET usage, the first column is expected to be dates. If the data is daily data, then next column is expected to be the measured values. If the data is sampled data, the next column is remark codes, and the third column is values.

Usage

```
readDataFromFile(filePath, fileName, hasHeader = TRUE, separator = ",")
```

Arguments

filePath	character specifying the path to the file
fileName	character name of file to open
hasHeader	logical true if the first row of data is the column headers
separator	character character that separates data cells

Value

retval dataframe with dateTime, value, and code columns

Examples

```
filePath <- system.file("extdata", package="EGRET")
fileName <- 'ChoptankRiverFlow.txt'
ChopData <- readDataFromFile(filePath,fileName, separator="\t")
```

readNWISDaily *Import NWIS Daily Data for EGRET analysis*

Description

Imports data from NWIS web service. This function gets the data from here: http://waterservices.usgs.gov/ A list of parameter codes can be found here: http://nwis.waterdata.usgs.gov/nwis/pmcodes/ A list of statistic codes can be found here: http://nwis.waterdata.usgs.gov/nwis/help/?read_file=stat&format=table

Usage

```
readNWISDaily(siteNumber, parameterCd = "00060", startDate = "",
  endDate = "", interactive = TRUE, convert = TRUE)
```

Arguments

siteNumber	character USGS site number. This is usually an 8 digit number
parameterCd	character USGS parameter code. This is usually an 5 digit number.
startDate	character starting date for data retrieval in the form YYYY-MM-DD.
endDate	character ending date for data retrieval in the form YYYY-MM-DD.
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.
convert	logical Option to include a conversion from cfs to cms (35.314667). The default is TRUE, which is appropriate for using NWIS data in the EGRET package. Set this to FALSE to not include the conversion. If the parameter code is not 00060 (NWIS discharge), there is no conversion applied.

Value

A data frame 'Daily' with the following columns:

Name	Type	Description
Date	Date	Date
Q	numeric	Discharge in m^3/s
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850
Qualifier	character	Qualifying code
i	integer	Index of days, starting with 1
LogQ	numeric	Natural logarithm of Q
Q7	numeric	7 day running average of Q
Q30	numeric	30 day running average of Q

See Also

readNWISdv, populateDaily

Examples

```
## Not run:

Daily <- readNWISDaily('01594440','00060', '1985-01-01', '1985-03-31')
DailySuspSediment <- readNWISDaily('01594440','80154', '1985-01-01', '1985-03-31',convert=FALSE)

## End(Not run)
```

`readNWISSample` *Import NWIS Sample Data for EGRET analysis*

Description

Imports data from NWIS web service. This function gets the data from here: http://nwis.waterdata.usgs.gov/nwis/qwdata/ A list of parameter and statistic codes can be found here: http://help.waterdata.usgs.gov/codes-and-parameters For raw data, use readNWISqw from the dataRetrieval package. This function will retrieve the raw data, and compress it (summing constituents). See section 3.2.4 of the vignette for more details.

Usage

```
readNWISSample(siteNumber, parameterCd, startDate = "", endDate = "",
  interactive = TRUE)
```

Arguments

`siteNumber`	character USGS site number. This is usually an 8 digit number
`parameterCd`	character USGS parameter code. This is usually an 5 digit number.
`startDate`	character starting date for data retrieval in the form YYYY-MM-DD.
`endDate`	character ending date for data retrieval in the form YYYY-MM-DD.
`interactive`	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

A data frame 'Sample' with the following columns:

Name	Type	Description
Date	Date	Date
ConcLow	numeric	Lower limit of concentration
ConcHigh	numeric	Upper limit of concentration
Uncen	integer	Uncensored data (1=TRUE, 0=FALSE)
ConcAve	numeric	Average concentration
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850
SinDY	numeric	Sine of the DecYear
CosDY	numeric	Cosine of the DecYear

See Also

compressData, populateSampleColumns, readNWISqw

Examples

```
## Not run:
# These examples require an internet connection to run

Sample_01075 <- readNWISSample('01594440','01075', '1985-01-01', '1985-03-31')
Sample_All2 <- readNWISSample('05114000',c('00915','00931'), '1985-01-01', '1985-03-31')
Sample_Select <- readNWISSample('05114000',c('00915','00931'), '', '')

## End(Not run)
```

`readUserDaily` *Import user daily data for EGRET analysis*

Description

Imports data from a user-supplied file, and converts it to a Daily data frame, appropriate for WRTDS calculations.

Usage

```
readUserDaily(filePath, fileName, hasHeader = TRUE, separator = ",",
  qUnit = 1, interactive = TRUE)
```

Arguments

`filePath`	character specifying the path to the file
`fileName`	character name of file to open
`hasHeader`	logical true if the first row of data is the column headers
`separator`	character character that separates data cells
`qUnit`	number 1 is cubic feet per second, 2 is cubic meters per second, 3 is 10^3 cubic feet per second, and 4 is 10^3 cubic meters per second
`interactive`	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

A data frame 'Daily' with the following columns:

Name	Type	Description
Date	Date	Date
Q	numeric	Discharge in m^3/s
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850

Qualifier	character	Qualifying code
i	integer	Index of days, starting with 1
LogQ	numeric	Natural logarithm of Q
Q7	numeric	7 day running average of Q
Q30	numeric	30 day running average of Q

Examples

```
filePath <- system.file("extdata", package="EGRET")
fileName <- "ChoptankRiverFlow.txt"
Daily <- readUserDaily(filePath,fileName,separator="\t")
```

`readUserSample` *Import user sample data for EGRET analysis*

Description

Imports data from a user-supplied file, and converts it to a Sample data frame (including summing multiple constituents), appropriate for EGRET analysis. First column is date, second is remark code, and third is value. If multiple constituents are to be combinded with interval censoring, additional columns can be inserted, each starting with remark code (specifically looking for <), and values.

Usage

```
readUserSample(filePath, fileName, hasHeader = TRUE, separator = ",",
  interactive = TRUE)
```

Arguments

`filePath`	character specifying the path to the file
`fileName`	character name of file to open
`hasHeader`	logical true if the first row of data is the column headers
`separator`	character character that separates data cells
`interactive`	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

A data frame 'Sample' with the following columns:

Name	Type	Description
Date	Date	Date
ConcLow	numeric	Lower limit of concentration
ConcHigh	numeric	Upper limit of concentration
Uncen	integer	Uncensored data (1=TRUE, 0=FALSE)

ConcAve	numeric	Average concentration
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850
SinDY	numeric	Sine of the DecYear
CosDY	numeric	Cosine of the DecYear

See Also

compressData, populateSampleColumns

Examples

```
filePath <- system.file("extdata", package="EGRET")
fileName <- 'ChoptankRiverNitrate.csv'
Sample <- readUserSample(filePath,fileName, separator=";",interactive=FALSE)
```

---

readWQPSample *Import Sample Data for WRTDS*

Description

Imports data from the Water Quality Portal, so it could be STORET, NWIS, or . This function gets the data from: http://www.waterqualitydata.us For raw data, use readWQPdata. This function will retrieve the raw data, and compress it (summing constituents). See chapter 7 of the EGRET user guide for more details, then converts it to the Sample dataframe structure.

Usage

```
readWQPSample(siteNumber, characteristicName, startDate, endDate,
  interactive = TRUE)
```

Arguments

siteNumber	character site number. If USGS, it should be in the form :'USGS-XXXXXXXXX...'
characteristicName	character
startDate	character starting date for data retrieval in the form YYYY-MM-DD.
endDate	character ending date for data retrieval in the form YYYY-MM-DD.
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

A data frame 'Sample' with the following columns:

Name	Type	Description
Date	Date	Date
ConcLow	numeric	Lower limit of concentration
ConcHigh	numeric	Upper limit of concentration
Uncen	integer	Uncensored data (1=TRUE, 0=FALSE)
ConcAve	numeric	Average concentration
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850
SinDY	numeric	Sine of the DecYear
CosDY	numeric	Cosine of the DecYear

See Also

readWQPdata, whatWQPsites, readWQPqw, compressData, populateSampleColumns

Examples

```
# These examples require an internet connection to run
## Not run:
Sample_01075 <- readWQPSample('USGS-01594440','Chloride', '', '')
Sample_All <- readWQPSample('WIDNR_WQX-10032762','Specific conductance', '', '')

## End(Not run)
```

---

removeDuplicates *Remove Duplicates*

Description

Removes observations from the data frame Sample when the observation has the identical date and value as another observation

Usage

```
removeDuplicates(Sample)
```

Arguments

Sample	dataframe with at least DecYear and ConcHigh, default name is Sample

Value

A data frame 'Sample' with the following columns:

Name	Type	Description
Date	Date	Date
ConcLow	numeric	Lower limit of concentration
ConcHigh	numeric	Upper limit of concentration
Uncen	integer	Uncensored data (1=TRUE, 0=FALSE)
ConcAve	numeric	Average concentration
Julian	integer	Number of days since Jan. 1, 1850
Month	integer	Month of the year [1-12]
Day	integer	Day of the year [1-366]
DecYear	numeric	Decimal year
MonthSeq	integer	Number of months since January 1, 1850
SinDY	numeric	Sine of the DecYear
CosDY	numeric	Cosine of the DecYear

Examples

```
DecYear <- c('1985.01', '1985.01', '1985.02', '1985.02', '1985.03')
ConcHigh <- c(1,2,3,3,5)
dataInput <- data.frame(DecYear, ConcHigh, stringsAsFactors=FALSE)
Sample <- removeDuplicates(dataInput)
```

---

runSurvReg *Run the weighted survival regression for a set of estimation points (defined by DecYear and Log(Q))*

Description

This function runs the survival regression which is the concentration estimation method of WRTDS. It uses sample data from the data frame Sample. It does the estimation for a set of data points defined by two vectors: estPtYear and estPtLQ. It returns an array of results for the estimation points. The array returned contains yHat, SE and ConcHat (in that order).

Usage

```
runSurvReg(estPtYear, estPtLQ, numDays, DecLow, DecHigh, Sample, windowY = 7,
  windowQ = 2, windowS = 0.5, minNumObs = 100, minNumUncen = 50,
  interactive = TRUE, edgeAdjust = TRUE)
```

Arguments

estPtYear	numeric vector of Decimal Year values at the estimation points
estPtLQ	numeric vector of ln(Q) values at the estimation points, must be the same length as estPtYear
numDays	number of days in the Daily record

DecLow	number specifying minimum decimal year
DecHigh	number specifying maximum decimal year
Sample	dataframe created for EGRET analysis
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 7
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50
interactive	logical specifying whether or not to display progress message
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE.

Value

resultSurvReg numeric array containing the yHat, SE, and ConcHat values array dimensions are (numEstPts,3)

Examples

```
eList <- Choptank_eList
estPtYear<-c(2001.0,2005.0,2009.0)
estPtLQ<-c(1,1,1)
Sample <- getSample(eList)
numDays <- Sample$Julian[nrow(Sample)] - Sample$Julian[1] + 1
DecLow <- Sample$DecYear[1]
DecHigh <- Sample$DecYear[nrow(Sample)]
resultSurvReg <- runSurvReg(estPtYear,estPtLQ,numDays,DecLow,DecHigh,Sample)
```

---

saveResults

A utility program for saving the contents of the workspace This function saves the workspace. It assigns the file a name using the abbreviations for station and constituent.

Description

A utility program for saving the contents of the workspace

This function saves the workspace. It assigns the file a name using the abbreviations for station and constituent.

Usage

```
saveResults(savePath, eList)
```

Arguments

savePath	character specifying the full pathname of the folder where the file is to be saved ending with the final slash
eList	named list with at least the INFO dataframe

Examples

```
eList <- Choptank_eList
savePath <- "~/"
## Not run: saveResults(savePath, eList)

#To load:
load(paste(savePath,"Chop.nitrogen.RData",sep=""))
## End(Not run)
```

---

selectDays *Creates a subset data frame that only contains data for the specified period of analysis*

Description

This function uses the user-defined 'period of analysis', and subsets the data. If you want to examine your data set as a time series of water years, then the period of analysis is October through September. If you want to examine the data set as calendar years then the period of analysis is January through December. You might want to examine the winter season, which you could define as December through February, then those 3 months become the period of analysis. The only constraints on the definition of a period of analysis are these: it must be defined in terms of whole months; it must be a set of contiguous months (like March-April-May), and have a length that is no less than 1 month and no more than 12 months. Define the PA by using two arguments: paLong and paStart. paLong is the length of the period of analysis, and paStart is the starting month.

Usage

```
selectDays(df, paLong, paStart)
```

Arguments

df	dataframe with Q
paLong	a numeric value for the length of the period of analysis, must be an integer from 1 to 12
paStart	a numeric value for the starting month of the period of analysis, must be an integer from 1 to 12

Value

localDaily a data frame containing the daily data but only for the period of analysis (not all months)

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
DailySubset <- selectDays(Daily, 4, 11)
```

`setPA` *Sets up the period of analysis*

Description

Period of analysis is defined by the starting month (paStart) and length in months (paLong). paStart and paLong are constrained to be integers from 1 to 12. for example, a water year would be paStart = 10 and paLong = 12. for example, the winter season, defined by Dec,Jan,Feb would be paStart = 12 and paLong =3.

Usage

```
setPA(eList, paStart = 10, paLong = 12, window = 20)
```

Arguments

`eList`	named list with at least the INFO dataframe
`paStart`	A numeric value for the starting month of the Period of Analysis, default is 10
`paLong`	A numeric value for the length of the Period of Analysis in months, default is 12
`window`	A numeric value for the half-width of a smoothing window for annual streamflow values, default is 20

Value

eList named list at least the INFO dataframe, along any other part of the list that was input. Any of these values can be NA, not all EGRET functions will work with missing parts of the named list eList.

Examples

```
eList <- Choptank_eList
eList <- setPA(eList, paStart=12, paLong=3)
```

`setSeasonLabel` *Create a character character that describes the period of analysis, when period of analysis has already been set in AnnualResults*

Description

The period of analysis can be of any length from 1 month to 12 months. The period of analysis can have any starting month from 1 (January) through 12 (December). This function produces a character character that describes this period of analysis. For example "water year", "calendar year", "year starting with April", or "Season consisting of April, May, June". There is an alternative version of this function for the case where AnnualResults does not exist. This might arise in a call from plotConcTime or plotLogConcTime. That function is called setSeasonLabelByUser.

Usage

```
setSeasonLabel(localAnnualResults)
```

Arguments

`localAnnualResults`
: data frame that contains the annual results, default is AnnualResults

Value

periodName character which describes the period of analysis

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
AnnualResults <- setupYears(Daily)
setSeasonLabel(AnnualResults)
```

`setSeasonLabelByUser` *Creates a character character that describes the period of analysis, when the period of analysis is being set by the user and not from AnnualResults*

Description

The period of analysis can be of any length from 1 month to 12 months. The period of analysis can have any starting month from 1 (January) through 12 (December). This function produces a character that describes this period of analysis. For example "water year", "calendar year", "year starting with April", or "Season consisting of April, May, June". There is an alternative version of this function for the case where AnnualResults exists. And we want to use the period of analysis defined there. That function is called setSeasonLabel.

Usage

```
setSeasonLabelByUser(paStartInput = 10, paLongInput = 12)
```

Arguments

paStartInput	numeric the month which is the start of the period of analysis, default is 10 which would be the case if the period of analysis is the water year
paLongInput	numeric the length of the the period of analysis, in months, default is 12 which would be the case if the period of analysis is the water year

Value

periodName character which describes the period of analysis

Examples

```
setSeasonLabelByUser(paStartInput=1,paLongInput=12)
setSeasonLabelByUser(paStartInput=4,paLongInput=3)
```

---

setUpEstimation *setUpEstimation*

Description

Set up the INFO data frame for a modelEstimation

Usage

```
setUpEstimation(eList, windowY = 7, windowQ = 2, windowS = 0.5,
  minNumObs = 100, minNumUncen = 50, edgeAdjust = TRUE,
  interactive = TRUE)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
windowY	numeric specifying the half-window width in the time dimension, in units of years, default is 7
windowQ	numeric specifying the half-window width in the discharge dimension, units are natural log units, default is 2
windowS	numeric specifying the half-window with in the seasonal dimension, in units of years, default is 0.5
minNumObs	numeric specifying the miniumum number of observations required to run the weighted regression, default is 100
minNumUncen	numeric specifying the minimum number of uncensored observations to run the weighted regression, default is 50

edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record. The modified method tends to reduce curvature near the start and end of record. Default is TRUE.
interactive	logical Option for interactive mode. If true, there is user interaction for error handling and data checks.

Value

eList named list with Daily, Sample, and INFO dataframes.

Examples

```
eList <- Choptank_eList
eList <- setUpEstimation(eList)
```

---

setupYears *Creates the AnnualResults data frame from the Daily data frame*

Description

This function aggregates the results stored on a daily basis in the Daily data frame and stores the average values of these in the new data frame called AnnualResults. The "annual values" can be a full 12 months, or they can be shorter. See manual to understand paLong and paStart arguments. The simplest case, a Water Year (October through September), would have paLong=12, and paStart=10. A calendar year would be paLong=12 and paStart=1. A winter season of Dec, Jan, Feb would be paLong=3 and paStart=12

Usage

```
setupYears(localDaily, paLong = 12, paStart = 10)
```

Arguments

localDaily	data frame containing the daily values, default is Daily
paLong	numeric integer specifying the length of the period of analysis, in months, 1<=paLong<=12, default is 12
paStart	numeric integer specifying the starting month for the period of analysis, 1<=paStart<=12, default is 10

Value

AnnualResults data frame with one row per year

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
AnnualResults <- setupYears(Daily, 4, 10)
```

surfaceIndex *Compute the 6 parameters needed to lay out the grid for the surfaces computed in estSurfaces*

Description

The code here is a repetition of the first part of the code for estSurfaces

Usage

```
surfaceIndex(Daily)
```

Arguments

Daily	data frame containing the daily values, default is Daily

Value

surfaceIndexParameters a numeric vector of length 6, defining the grid for the surfaces

Examples

```
eList <- Choptank_eList
Daily <- getDaily(eList)
surfaceIndex(Daily)
```

tableChange *Create a table of the changes in flow-normalized values between various points in time in the record*

Description

These tables describe trends in flow-normalized concentration and in flow-normalized flux. They are described as changes in real units or in percent and als as slopes in real units per year or in percent per year. They are computed over pairs of time points. These time points can be user-defined or they can be set by the program to be the final year of the record and a set of years that are multiple of 5 years prior to that.

Usage

```
tableChange(eList, fluxUnit = 9, yearPoints = NA)
```

Arguments

eList	named list with at least the Daily and INFO dataframes
fluxUnit	object of fluxUnit class. printFluxUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
yearPoints	numeric vector listing the years for which the change or slope computations are made, they need to be in chronological order. For example yearPoints=c(1975,1985,1995,2005), default is NA (which allows the program to set yearPoints automatically)

Examples

```
eList <- Choptank_eList
# Water Year:
tableChange(eList, fluxUnit=6,yearPoints=c(2001,2005,2008,2009))
tableChange(eList, fluxUnit=9)
# Winter:
eList <- setPA(eList, paStart=12,paLong=3)
tableChange(eList, fluxUnit=6,yearPoints=c(2001,2005,2008,2009))
```

---

tableChangeSingle *Create a table of the changes in flow-normalized concentration or flux values between various points in time in the record*

Description

This table describe trends in flow-normalized concentration or flux depending on if flux is defined as TRUE or FALSE. The results are described as changes in real units or in percent and als as slopes in real units per year or in percent per year. They are computed over pairs of time points (Year1 to Year2). These time points can be user-defined or they can be set by the program to be the final year of the record and a set of years that are multiple of 5 years prior to that.

Usage

```
tableChangeSingle(eList, fluxUnit = 9, yearPoints = NA, flux = FALSE)
```

Arguments

eList	named list with at least Daily and INFO dataframes
fluxUnit	object of fluxUnit class. printFluxUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
yearPoints	numeric vector listing the years for which the change or slope computations are made, they need to be in chronological order. For example yearPoints=c(1975,1985,1995,2005), default is NA (which allows the program to set yearPoints automatically)
flux	logical if TRUE results are returned in flux, if FALSE concentration. Default is set to FALSE.

Value

resultsDF dataframe describing trends in flow-normalized concentration or flux if returnDataFrame is TRUE

dataframe with Year1, Year2, change[mg/L], slope[mg/L], change[percent], slope[percent] columns. The data in each row is the change or slope calculated from Year1 to Year2

Examples

```
eList <- Choptank_eList
# Water Year:
#This returns concentration ASCII table in the console:
tableChangeSingle(eList, fluxUnit=6,yearPoints=c(2001,2005,2008,2009), flux=FALSE)
#This returns flux values ASCII table in the console
tableChangeSingle(eList, fluxUnit=6,yearPoints=c(2001,2005,2008,2009), flux=TRUE)
# Winter:
eList <- setPA(eList, paStart=12,paLong=3)
tableChangeSingle(eList, fluxUnit=6,yearPoints=c(2001,2005,2008,2009), flux=FALSE)
```

---

tableFlowChange *Prints table of change metrics for a given streamflow statistic*

Description

Part of the flowHistory system. The index of the flow statistics is istat. These statistics are: (1) 1-day minimum, (2) 7-day minimum, (3) 30-day minimum, (4) median (5) mean, (6) 30-day maximum, (7) 7-day maximum, and (8) 1-day maximum. A dataframe is returned, as well as a printout in the R console.

Usage

```
tableFlowChange(eList, istat, qUnit = 1, runoff = FALSE, yearPoints = NA)
```

Arguments

eList	named list with at least Daily and INFO dataframes
istat	A numeric value for the flow statistic to be graphed (possible values are 1 through 8)
qUnit	object of qUnit class printqUnitCheatSheet, or numeric represented the short code, or character representing the descriptive name.
runoff	logical variable, if TRUE the streamflow data are converted to runoff values in mm/day
yearPoints	A vector of numeric values, specifying the years at which change metrics are to be calculated, default is NA (which allows the function to set these automatically), yearPoints must be in ascending order

Examples

```
eList <- Choptank_eList
tableFlowChange(eList, istat=5,yearPoints=c(2001,2005,2009))
df <- tableFlowChange(eList, istat=5,yearPoints=c(2001,2005,2009))
```

`tableResults` *Table of annual results for discharge, concentration and flux*

Description

Produce an ASCII table showing: year, mean discharge, mean concentration, flow-normalized concentration, mean flux, and flow-normalized flux.

Usage

```
tableResults(eList, qUnit = 2, fluxUnit = 9)
```

Arguments

`eList`	named list with at least Daily and INFO dataframes
`qUnit`	object of qUnit class. `printqUnitCheatSheet`, or numeric represented the short code, or character representing the descriptive name.
`fluxUnit`	object of fluxUnit class. `printFluxUnitCheatSheet`, or numeric represented the short code, or character representing the descriptive name.

Value

results dataframe, if returnDataFrame=TRUE

dataframe with year, discharge, concentration, flow-normalized concentration, flux, and flow-normalized concentration columns.

Examples

```
eList <- Choptank_eList
# Water Year:
tableResults(eList, fluxUnit = 1)
tableResults(eList, fluxUnit = 'kgDay', qUnit = 'cms')
returnedTable <- tableResults(eList, fluxUnit = 1)
# Winter:
eList <- setPA(eList, paLong=3,paStart=12)
tableResults(eList, fluxUnit = 1)
```

triCube *Tricube weight function*

Description

Computes the tricube weight function on a vector of distance values (d), based on a half-window width of h, and returns a vector of weights that range from zero to 1.

Usage

```
triCube(d, h)
```

Arguments

d	numeric vector of distances from the point of estimation to the given sample value
h	numeric value, the half-window width, measured in the same units as d

Value

w numeric vector of weights, all 0<=w<=1

Examples

```
h<-10
d<-c(-11,-10,-5,-1,-0.01,0,5,9.9,10,20)
triCube(d,h)
```

yPretty *Sets up tick marks for an axis for a graph with an arithmetic scale which starts at zero*

Description

Axis tick marks that run from zero to some specified maximum, creates about 4 to 8 ticks marks.

Usage

```
yPretty(yMax)
```

Arguments

yMax	A numeric value for the maximum value to be plotted, it must be >0

Value

yTicks A numeric vector representing the values for each of the tick marks

Examples

```
yPretty(7.8)
yPretty(125)
```

Index