Package 'EGRETci'

October 12, 2022

Type Package

Title Exploration and Graphics for RivEr Trends Confidence Intervals

Version 2.0.4

Description Collection of functions to evaluate uncertainty of results from water quality analysis using the Weighted Regressions on Time Discharge and Season (WRTDS) method. This package is an add-on to the EGRET package that performs the WRTDS analysis. The WRTDS modeling method was initially introduced and discussed in Hirsch et al. (2010) <doi:10.1111/j.1752-1688.2010.00482.x>, and expanded in Hirsch and De Cicco (2015) <doi:10.3133/tm4A10>. The paper describing the uncertainty and confidence interval calculations is Hirsch et al. (2015) <doi:10.1016/j.envsoft.2015.07.017>.

License CC0

Depends R (>= 3.5.0)

Imports EGRET(>= 3.0.5), binom, stats, graphics, utils

Suggests knitr, testthat, doParallel, iterators, rmarkdown, pkgdown, covr

LazyLoad yes

LazyData yes

VignetteBuilder knitr

BuildVignettes true

URL https://github.com/USGS-R/EGRETci

BugReports https://github.com/USGS-R/EGRETci/issues

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 7.1.1

NeedsCompilation no

Author Laura DeCicco [aut, cre] (<https://orcid.org/0000-0002-3915-9487>),
Robert Hirsch [aut] (<https://orcid.org/0000-0002-4534-075X>),
Jennifer Murphy [ctb]

Maintainer Laura DeCicco <ldecicco@usgs.gov>

Repository CRAN

Date/Publication 2021-04-13 15:10:11 UTC

R topics documented:

`EGRETci-package` *EGRETci package for bootstrap hypothesis tests and confidence interval analysis for WRTDS (Weighted Regressions on Time, Discharge, and Season) statistical models. This package is designed to be used in conjunction with the EGRET package, which estimates and describes WRTDS models.*

Description

Package:	EGRETci
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

Collection of functions to evaluate uncertainty of results from water quality analysis using the Weighted Regressions on Time Discharge and Season (WRTDS) method. This package is an add-on to the EGRET package that performs the WRTDS analysis.

Author(s)

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

References

Hirsch, R.M., and De Cicco, L.A., 2015, 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., doi: 10.3133/tm4A10

Hirsch, R.M., Archfield, S.A., and De Cicco, L.A., 2015, A bootstrap method for estimating uncertainty of water quality trends. Environmental Modelling & Software, 73, 148-166. https://www.sciencedirect.com/science/article/pii/S1364815215300220

`blockSample` *blockSample*

Description

Get a bootstrap replicate of the Sample data frame based on the user-specified blockLength. The bootstrap replicate is made up randomly selected blocks of data from Sample data frame. Each block includes all the samples in a standard period of time (the blockLength measured in days). The blocks are created based on the random selection (with replacement) of starting dates from the full Sample data frame. The bootstrap replicate has the same number of observations as the original Sample, but some observations are included once, some are included multiple times, and some are not included at all.

Usage

```
blockSample(localSample, blockLength, startSeed = NA)
```

Arguments

localSample	Sample data frame
blockLength	integer size of subset, expressed in days. 200 days has been found to be a good choice.
startSeed	setSeed value. This is used to make repeatable output. Default = NA.

Value

newSample data frame in same format as Sample data frame. It has the same number of rows as the Sample data frame.

Examples

```
library(EGRET)
eList <- Choptank_eList
Sample <- eList$Sample
bsReturn <- blockSample(Sample, 200)
```

---

bootAnnual *Single confidence interval bootstrap run*

Description

One bootstrap run used in calculating confidence interval bands.

Usage

```
bootAnnual(eList, blockLength = 200, startSeed = 494817, verbose = FALSE,
  jitterOn = FALSE, V = 0.2)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
blockLength	integer default value is 200.
startSeed	setSeed value. Defaults to 494817. This is used to make repeatable output.
verbose	logical specifying whether or not to display progress message.
jitterOn	logical, if TRUE, adds "jitter" to the data in an attempt to avoid some numerical problems. Default = FALSE. See Details below.
V	numeric a multiplier for addition of jitter to the data, default = 0.2.

Details

In some situations numerical problems are encountered in the bootstrap process, resulting in highly unreasonable spikes in the confidence intervals. The use of "jitter" can often prevent these problems, but should only be used when it is clearly needed. It adds a small amount of random "jitter" to the explanatory variables of the WRTDS model. The V parameter sets the scale of variation in the log discharge values. The standard deviation of the added jitter is V * standard deviation of Log Q. The default for V is 0.2. Larger values should generally be avoided, and smaller values may be sufficient.

Examples

```
library(EGRET)
eList <- Choptank_eList
## Not run:
annualResults <- bootAnnual(eList)

## End(Not run)
```

Choptank_eBoot *Example eBoot*

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 xConc, and xFlux vectors.

---

ciBands *Confidence Interval Band Calculations*

Description

Computes confidence intervals for Flow-Normalized Concentration and Flow-Normalized Flux for a WRTDS model.

Usage

```
ciBands(eList, repAnnualResults, probs = c(0.05, 0.95))
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
repAnnualResults	named list returned from bootstrapping process.
probs	numeric vector low and high confidence interval frequencies, default = c(0.05, 0.95) (which results in a 90% confidence interval).

Examples

```
library(EGRET)
eList <- Choptank_eList
nBoot <- 100
blockLength <- 200
## Not run:

repAnnualResults <- vector(mode = "list", length = nBoot)
for(n in 1:nBoot){
   annualResults <- bootAnnual(eList, blockLength, startSeed = n)
   repAnnualResults[[n]] <- annualResults
}

CIAnnualResults <- ciBands(eList, repAnnualResults)


## End(Not run)
```

`ciCalculations` *ciCalculations*

Description

Interactive function to calculate confidence bands for flow normalized concentration or flow normalized flux. It returns the data frame CIAnnualResults, which is used as input to the functions plotConcHistBoot( ), and plotFluxHistBoot( ) which produce the graphical output.

Usage

```
ciCalculations(eList, startSeed = 494817, verbose = TRUE,
  jitterOn = FALSE, V = 0.2, ...)
```

Arguments

`eList`	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running `modelEstimation`.
`startSeed`	setSeed value. Defaults to 494817. This is used to make repeatable output.
`verbose`	logical specifying whether or not to display progress messag, default = TRUE
`jitterOn`	logical, if TRUE, adds "jitter" to the data in an attempt to avoid some numerical problems. Default = FALSE. See Details below.
`V`	numeric a multiplier for addition of jitter to the data, default = 0.2. See Details below.
`...`	optionally include nBoot, blockLength, or widthCI

Details

In some situations numerical problems are encountered in the bootstrap process, resulting in highly unreasonable spikes in the confidence intervals. The use of "jitter" can often prevent these problems, but should only be used when it is clearly needed. It adds a small amount of random "jitter" to the explanatory variables of the WRTDS model. The V parameter sets the scale of variation in the log discharge values. The standard deviation of the added jitter is V * standard deviation of Log Q. The default for V is 0.2. Larger values should generally be avoided, and smaller values may be sufficient.

Argument values suggested. To test the code nBoot = 10 is sufficient, but for meaningful results nBoot = 100 or even nBoot = 500 are more appropriate. blockLength = 200 widthCI = 90 (90% confidence interval)

Value

CIAnnualResults a data frame with the following columns Year, mean decYear value for the year being reported FNConcLow, the lower confidence limit for flow normalized concentration, in mg/L FNConcHigh, the upper confidence limit for flow normalized concentration, in mg/L FNFluxLow, the lower confidence limit for flow normalized flux, in kg FNFluxLow, the lower confidence limit for flow normalized flux, in kg

Examples

```
library(EGRET)
eList <- Choptank_eList
## Not run:
# If run interactively, using stationary flow normalization
# in this format it will prompt for nBoot, blockLength and widthCI.
# CIAnnualResults <- ciCalculations(eList)

# run in batch mode, using non-stationary flow normalization
# In this example nBoot is set very small, useful for an initial trial run.
# A meaningful application would use nBoot values such as 100 or even 500.
seriesOut_2 <- runSeries(eList, windowSide = 11)
CIAnnualResults <- ciCalculations(seriesOut_2,
                     nBoot = 10,
                     blockLength = 200,
                     widthCI = 90)

 plotConcHistBoot(seriesOut_2, CIAnnualResults)


## End(Not run)
```

---

genDailyBoot *WRTDSKalman Bootstrapping*

Description

Function to get multiple bootstrap replicates at a daily time step using the WRTDS_K method. It is done by doing bootstrap resampling of the original Sample data frame. The number of these replicate samples that are created is called nBoot and in each case the WRTDS model is estimated. Then, for each of these models, there are nKalman time series of daily values computed, using all of the sample values in the original Sample data frame. The total number of replicates of the complete process is nBoot * nKalman. For example we might generate 500 replicates by setting nBoot = 20 and nKalman = 25.

Usage

```
genDailyBoot(eList, nBoot = 10, nKalman = 10, rho = 0.9, setSeed = NA,
  jitterOn = FALSE, V = 0.2)
```

Arguments

eList	is the data with a fitted model already done. Note that the eList$Sample may have multiple values on a given day and it can also have censored values.
nBoot	number of times the bootstrap resampling and model estimating is done.
nKalman	number of different realizations of the daily time series for each re-estimated model.
rho	numeric the lag one autocorrelation. Default is 0.9.
setSeed	value. Defaults is NA, which will not specify a randomized seed. This can be used to make repeatable output.
jitterOn	logical, if TRUE, adds "jitter" to the data in an attempt to avoid some numerical problems. Default = FALSE. See Details below.
V	numeric a multiplier for addition of jitter to the data, default = 0.2. See Details below.

Details

In some situations numerical problems are encountered in the bootstrap process, resulting in highly unreasonable spikes in the confidence intervals. The use of "jitter" can often prevent these problems, but should only be used when it is clearly needed. It adds a small amount of random "jitter" to the explanatory variables of the WRTDS model. The V parameter sets the scale of variation in the log discharge values. The standard deviation of the added jitter is V * standard deviation of Log Q. The default for V is 0.2. Larger values should generally be avoided, and smaller values may be sufficient.

Value

dailyBootOut a matrix of daily flux values (in kg/day). The number of columns of the matrix is the number of replicates produced which is nBoot * nKalman The number of rows is the number of days in the record. The set of days simulated is the same set of days that are in the eList$Daily data frame.

Examples

```
eList <- EGRET::Choptank_eList
# Very long running function:
## Not run:
dailyBootOut <- genDailyBoot(eList,
                             nBoot = 20,
                             nKalman = 25)

## End(Not run)
```

---

makeAnnualPI *Make Annual Prediction Intervals*

Description

This function takes the output from genDailyBoot and calculates the quantiles for an annual (based on paStart/paLong) aggregation. This means that the function can be used for seasons.

Usage

```
makeAnnualPI(dailyBootOut, eList, paLong = 12, paStart = 10, fluxUnit = 3)
```

Arguments

dailyBootOut	data frame returned from genDailyBoot
eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
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
fluxUnit	number representing entry in pre-defined fluxUnit class array. printFluxUnitCheatSheet

Value

a list of 2 data frames, one for average concentration, in mg/L and one for flux (unit depends on fluxUnit argument) In each data frame the first column is DecYear. The remaining columns are quantiles of the flux or concentration (depending on the data frame).

Examples

```
eList <- EGRET::Choptank_eList
# This example is only based on 4 iterations
# Actual prediction intervals should be calculated on
# a much larger number of iterations (several hundred).
dailyBoot <- Choptank_dailyBootOut
annualPcts <- makeAnnualPI(dailyBoot, eList)
head(annualPcts[["flux"]])
head(annualPcts[["conc"]])
```

---

makeDailyPI *Make Daily Prediction Intervals*

Description

This function takes the output from genDailyBoot and calculates the quantiles for a daily aggregation.

Usage

```
makeDailyPI(dailyBootOut, eList, fluxUnit = 3)
```

Arguments

dailyBootOut	data frame returned from genDailyBoot
eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
fluxUnit	number representing entry in pre-defined fluxUnit class array. printFluxUnitCheatSheet

Value

a list of 2 data frames, one for average concentration, in mg/L and one for flux (unit depends on fluxUnit argument) In each data frame the first column is Date. The remaining columns are quantiles of the flux or concentration (depending on the data frame).

Examples

```
eList <- EGRET::Choptank_eList
# This example is only based on 4 iterations
# Actual prediction intervals should be calculated on
# a much larger number of iterations (several hundred).
dailyBoot <- Choptank_dailyBootOut

dailyPcts <- makeDailyPI(dailyBoot, eList)
head(dailyPcts[["flux"]])
head(dailyPcts[["conc"]])
```

makeMonthPI *Make Monthly Prediction Intervals*

Description

Month statistics using WRTDSKalman bootstrapping approach. The input to this function is the dailyBootOut matrix which contains nReplicate sets of daily flux values for the period of interest. The results are in the form of quantiles of concentration and of flux for each of these months.

Usage

```
makeMonthPI(dailyBootOut, eList, fluxUnit = 3)
```

Arguments

dailyBootOut	data frame returned from genDailyBoot
eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
fluxUnit	number representing entry in pre-defined fluxUnit class array. printFluxUnitCheatSheet

Value

a list of 2 data frames, one for average concentration, in mg/L and one for flux (unit depends on fluxUnit argument) In each data frame the first column is monthSeq that corresponds to the months in the "MonthSeq" column in the eList$Daily data frame. The remaining columns are quantiles of the flux or concentration (depending on the data frame).

Examples

```
eList <- EGRET::Choptank_eList
# This example is only based on 4 iterations
# Actual prediction intervals should be calculated on
# a much larger number of iterations (several hundred).
dailyBoot <- Choptank_dailyBootOut
monthPcts <- makeMonthPI(dailyBoot, eList)
head(monthPcts[["flux"]])
head(monthPcts[["conc"]])
```

monthSeqToDec *monthSeqToDec*

Description

Convert a sequence of month integers into their decimal years.

Usage

```
monthSeqToDec(monthSeq)
```

Arguments

monthSeq	integer vector of months. Month 1 is considered Jan. 1850.

Examples

```
months <- 1558:1600
monthSeqToDec(months)
```

plotConcHistBoot *Graph of annual concentration, flow normalized concentration, and confidence bands for flow normalized concentrations*

Description

Uses the output of modelEstimation in the EGRET package (results in the named list eList), and the data frame CIAnnualResults (produced by the function ciCalculations in the EGRETci package using scripts described in the EGRETci vignette) to produce a graph of annual concentration, flow normalized concentration, and confidence bands for flow-normalized concentrations. In addition to the arguments listed below, it will accept any additional arguments that are listed for the EGRET function plotConcHist.

Usage

```
plotConcHistBoot(eList, CIAnnualResults, yearStart = NA, yearEnd = NA,
  plotFlowNorm = TRUE, col.pred = "green", concMax = NA,
  plotAnnual = TRUE, plotGenConc = FALSE, cex = 0.8, cex.axis = 1.1,
  lwd = 2, col = "black", col.gen = "red", customPar = FALSE,
  printTitle = TRUE, cex.main = 1.1, ...)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
CIAnnualResults	data frame generated from ciBands (includes nBoot, probs, and blockLength attributes).
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).
plotFlowNorm	logical variable if TRUE flow normalized concentration line is plotted, if FALSE not plotted, default is TRUE.
col.pred	character color of line for flow-normalized concentration and for the confidence limits, default is "green".
concMax	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).
plotAnnual	logical variable if TRUE, annual mean concentration points from WRTDS output are plotted, if FALSE not plotted.
plotGenConc	logical variable. If TRUE, annual mean concentration points from WRTDS_K output are plotted, if FALSE not plotted.
cex	numeric value giving the amount by which plotting symbols should be magnified, default = 0.8.
cex.axis	numeric value of magnification to be used for axis annotation relative to the current setting of cex, default = 1.1.
lwd	numeric magnification of line width, default = 2.
col	color of annual mean points on plot, see ?par 'Color Specification', default = "black".
col.gen	color of annual mean points for WRTDS_K output on plot, see ?par 'Color Specification', default = "red".
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, EGRETci chooses the best margins.
printTitle	logical print title of the plot, default = TRUE.
cex.main	numeric value of magnification to be used for plot title, default = 1.1.
...	graphical parameters

Examples

```
library(EGRET)
eList <- Choptank_eList
CIAnnualResults <- Choptank_CIAnnualResults
plotConcHistBoot(eList, CIAnnualResults)
plotConcHistBoot(eList, CIAnnualResults, yearStart=1990, yearEnd=2002)
```

```
# Very long-running function:
## Not run:
CIAnnualResults <- ciCalculations(eList, nBoot = 100, blockLength = 200)
plotConcHistBoot(eList, CIAnnualResults)

## End(Not run)
```

`plotFluxHistBoot` *Graph of annual flux, flow normalized flux, and confidence bands for flow normalized flux*

Description

Uses the output of `modelEstimation` in the EGRET package (results in the named list eList), and the data frame CIAnnualResults (produced by EGRETci package using scripts described in the vignette) to produce a graph of annual flux, flow normalized flux, and confidence bands for flow-normalized flux. In addition to the arguments listed below, it will accept any additional arguments that are listed for the EGRET function `plotFluxHist`.

Usage

```
plotFluxHistBoot(eList, CIAnnualResults, yearStart = NA, yearEnd = NA,
  fluxUnit = 9, fluxMax = NA, plotFlowNorm = TRUE, col.pred = "green",
  plotAnnual = TRUE, plotGenFlux = FALSE, cex = 0.8, cex.axis = 1.1,
  lwd = 2, col = "black", col.gen = "red", cex.main = 1.1,
  printTitle = TRUE, customPar = FALSE, ...)
```

Arguments

`eList`	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running `modelEstimation`.
`CIAnnualResults`	data frame from ciBands (needs nBoot, probs, and blockLength attributes).
`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`	integer representing entry in pre-defined fluxUnit class array. `printFluxUnitCheatSheet`
`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), uses units specificed by fluxUnit.
`plotFlowNorm`	logical variable if TRUE flow normalized flux line is plotted, if FALSE not plotted, default is TRUE.
`col.pred`	character color of line for flow-normalized flux and for the confidence limits, default is "green".

plotAnnual	logical variable if TRUE, annual mean flux points from WRTDS output are plotted, if FALSE not plotted.
plotGenFlux	logical variable. If TRUE, annual mean flux points from WRTDS_K output are plotted, if FALSE not plotted.
cex	numeric value giving the amount by which plotting symbols should be magnified, default = 0.8.
cex.axis	numeric magnification to be used for axis annotation relative to the current setting of cex, default = 1.1.
lwd	numeric magnification of line width, default = 2.
col	color of annual mean points on plot, see ?par 'Color Specification', default = "black".
col.gen	color of annual mean points for WRTDS_K output on plot, see ?par 'Color Specification', default = "red".
cex.main	numeric title scale
printTitle	logical print title of the plot, 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.
...	graphical parameters

Examples

```
library(EGRET)
eList <- Choptank_eList
CIAnnualResults <- Choptank_CIAnnualResults
plotFluxHistBoot(eList, CIAnnualResults, fluxUnit=5)

## Not run:
CIAnnualResults <- ciCalculations(eList, nBoot = 100, blockLength = 200)
plotFluxHistBoot(eList, CIAnnualResults, fluxUnit=5)

## End(Not run)
```

---

plotHistogramTrend *plotHistogramTrend*

Description

Produces a histogram of trend results from bootstrap process. The histogram shows the trend results expressed as percentage change between the first year (or first period) and the second year (or second period). It shows the zero line (no trend) and also shows the WRTDS estimate of the trend in percent. It is based on the output of either wBT or runPairsBoot.

Usage

```
plotHistogramTrend(eList, eBoot, caseSetUp, flux = TRUE, xMin = NA,
  xMax = NA, xStep = NA, printTitle = TRUE, cex.main = 1.1,
  cex.axis = 1.1, cex.lab = 1.1, col.fill = "grey", ...)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
eBoot	named list. Returned from wBT or from runPairsBoot.
caseSetUp	data frame. Returned from trendSetUp, or if runPairsBoot was used, need to specify caseSetUp = NA.
flux	logical if TRUE, plots flux results, if FALSE plots concentration results.
xMin	minimum bin value for histogram, it is good to have the xMin and xMax arguments straddle zero, default is NA (value set from the data).
xMax	maximum bin value for histogram, default is NA (value set from the data).
xStep	step size, typically multiples of 10 or 20, default is NA (value set from the data).
printTitle	logical if TRUE, plot includes title.
cex.main	numeric magnification of font size for title, default is 1.1.
cex.axis	numeric magnification of font size for axis, default is 1.1.
cex.lab	numeric magnification of font size for axis labels, default is 1.1.
col.fill	character fill color for histogram, default is "grey".
...	base R graphical parameters that can be passed to the hist function

Details

For any given set of results (from eBoot) it is best to run it first with the arguments xMin = NA, xMax = NA, and xStep = NA. Then, observing the range the histogram covers it can be run again with values of these three arguments selected by the user to provide for a more readable version of the histogram.

Examples

```
library(EGRET)
eList <- Choptank_eList
eBoot <- Choptank_eBoot
caseSetUp <- Choptank_caseSetUp
plotHistogramTrend(eList, eBoot, caseSetUp, flux = FALSE)

## Not run:
# Using wBT:
caseSetUp <- trendSetUp(eList)
eBoot <- wBT(eList,caseSetUp)
plotHistogramTrend(eList, eBoot, caseSetUp,
                   flux = FALSE, xMin = -20, xMax = 60, xStep = 5)
plotHistogramTrend(eList, eBoot, caseSetUp,
```

```
                    flux = TRUE, xMin = -20, xMax = 60, xStep = 5)

# Using runPairs followed by runPairsBoot:
year1 <- 1985
year2 <- 2009
pairOut_2 <- runPairs(eList, year1, year2, windowSide = 7)
boot_pair_out <- runPairsBoot(eList, pairOut_2, nBoot = 10)

plotHistogramTrend(eList, boot_pair_out, caseSetUp = NA,
                    flux = TRUE, xMin = -20, xMax = 60, xStep = 5)

## End(Not run)
```

---

pVal *pVal*

Description

Computes the two-sided p value for the null hypothesis, where the null hypothesis is that the slope is zero. It is based on the binomial distribution. Note that the result does not depend on the magnitude of the individual slope values only depends on the number of positive slopes and number of negative slopes.

Usage

```
pVal(s)
```

Arguments

s	numeric vector of slope values from the bootstrap

Value

pVal numeric value

Examples

```
s <- c(-1.0, 0, 0.5, 0.55, 3.0)
pValue <- pVal(s)
```

runGroupsBoot *The bootstrap uncertainty analysis for runGroups results*

Description

This function that does the uncertainty analysis for determining the change between two groups of years. The process is virtually identical to what is used for runPairsBoot which looks at a change between a pair of years.

Usage

```
runGroupsBoot(eList, groupResults, nBoot = 100, startSeed = 494817,
  blockLength = 200, jitterOn = FALSE, V = 0.2)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
groupResults	data frame returned from runGroups
nBoot	the maximum number of bootstrap replicates to be used, typically 100
startSeed	setSeed value. Defaults to 494817. This is used to make repeatable output.
blockLength	days, typically 200 is a good choice
jitterOn	logical, if TRUE, adds "jitter" to the data in an attempt to avoid some numerical problems. Default = FALSE. See Details below.
V	numeric a multiplier for addition of jitter to the data, default = 0.2.

Details

In some situations numerical problems are encountered in the bootstrap process, resulting in highly unreasonable spikes in the confidence intervals. The use of "jitter" can often prevent these problems, but should only be used when it is clearly needed. It adds a small amount of random "jitter" to the explanatory variables of the WRTDS model. The V parameter sets the scale of variation in the log discharge values. The standard deviation of the added jitter is V * standard deviation of Log Q. The default for V is 0.2. Larger values should generally be avoided, and smaller values may be sufficient.

Value

eBoot, a named list with bootOut, wordsOut, xConc, xFlux, pConc, pFlux values.

- bootOut is a data frame with the results of the bootstrap test.
- wordsOut is a character vector describing the results.
- xConc and xFlux are vectors of length iBoot, of the change in flow normalized concentration and flow normalized flux computed from each of the bootstrap replicates.
- pConc and pFlux are vectors of length iBoot, of the change in flow normalized concentration or flow normalized flux computed from each of the bootstrap replicates expressed as % change.

See Also

runPairsBoot, runGroups

Examples

```
library(EGRET)
eList <- Choptank_eList

## Not run:
groupResults <- runGroups(eList,
                          group1firstYear = 1995,
                          group1lastYear = 2004,
                          group2firstYear = 2005,
                          group2lastYear = 2014,
                          windowSide = 7, wall = TRUE,
                          sample1EndDate = "2004-10-30",
                          paStart = 4, paLong = 2,
                          verbose = FALSE)

boot_group_out <- runGroupsBoot(eList, groupResults)

plotHistogramTrend(eList, boot_group_out, caseSetUp=NA)

## End(Not run)
```

---

runPairsBoot *The bootstrap uncertainty analysis for runPairs results*

Description

The function that does the uncertainty analysis for determining the change between any pair of years. It is very similar to the wBT function that runs the WRTDS bootstrap test. It differs from wBT in that it runs a specific number of bootstrap replicates, unlike the wBT approach that will stop running replicates based on the status of the test statistics along the way. Also, this code can be used with generalized flow normalization, which handles non-stationary discharge, whereas wBT does not.

Usage

```
runPairsBoot(eList, pairResults, nBoot = 100, startSeed = 494817,
  blockLength = 200, jitterOn = FALSE, V = 0.2)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes
pairResults	data frame returned from runPairs
nBoot	the maximum number of bootstrap replicates to be used, typically 100
startSeed	setSeed value. Defaults to 494817. This is used to make repeatable output.

blockLength	days, typically 200 is a good choice
jitterOn	logical, if TRUE, adds "jitter" to the data in an attempt to avoid some numerical problems. Default = FALSE. See Details below.
V	numeric a multiplier for addition of jitter to the data, default = 0.2.

Details

In some situations numerical problems are encountered in the bootstrap process, resulting in highly unreasonable spikes in the confidence intervals. The use of "jitter" can often prevent these problems, but should only be used when it is clearly needed. It adds a small amount of random "jitter" to the explanatory variables of the WRTDS model. The V parameter sets the scale of variation in the log discharge values. The standard deviation of the added jitter is V * standard deviation of Log Q. The default for V is 0.2. Larger values should generally be avoided, and smaller values may be sufficient.

Value

eBoot, a named list with bootOut, wordsOut, xConc, xFlux, pConc, pFlux values.

- bootOut is a data frame with the results of the bootstrap test.
- wordsOut is a character vector describing the results.
- xConc and xFlux are vectors of length iBoot, of the change in flow normalized concentration and flow normalized flux computed from each of the bootstrap replicates.
- pConc and pFlux are vectors of length iBoot, of the change in flow normalized concentration or flow normalized flux computed from each of the bootstrap replicates expressed as % change.

See Also

runGroupsBoot, runPairs

Examples

```
library(EGRET)
eList <- Choptank_eList
year1 <- 1985
year2 <- 2009

## Not run:
pairOut_2 <- runPairs(eList, year1, year2, windowSide = 7)

boot_pair_out <- runPairsBoot(eList, pairOut_2)

plotHistogramTrend(eList, boot_pair_out, caseSetUp = NA)

## End(Not run)
```

saveEGRETci *Save EGRETci workspace after running wBT (the WRTDS bootstrap test)*

Description

Saves critical information in a EGRETci workflow when analyzing trends between a starting and ending year.

Usage

```
saveEGRETci(eList, eBoot, caseSetUp, fileName = "")
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
eBoot	named list. Returned from wBT.
caseSetUp	data frame. Returned from trendSetUp.
fileName	character. If left blank (empty quotes), the function will interactively ask for a name to save.

Value

A .RData file containing three objects: eList, eBoot, and caseSetUp

See Also

wBT, trendSetUp, modelEstimation

Examples

```
eList <- EGRET::Choptank_eList
## Not run:
caseSetUp <- trendSetUp(eList)
eBoot <- wBT(eList,caseSetUp)
saveEGRETci(eList, eBoot, caseSetUp)

## End(Not run)
```

setForBoot *Allows user to set window parameters for the WRTDS model prior to running the bootstrap procedure*

Description

Adds window parameters to INFO file in eList.

Usage

```
setForBoot(eList, caseSetUp, windowY = 7, windowQ = 2, windowS = 0.5,
  edgeAdjust = TRUE)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
caseSetUp	data frame returned from trendSetUp.
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.
edgeAdjust	logical specifying whether to use the modified method for calculating the windows at the edge of the record, default is TRUE.

Value

eList list with Daily,Sample, INFO data frames and surface matrix.

Examples

```
eList <- EGRET::Choptank_eList

caseSetUp <- trendSetUp(eList,
  year1=1985,
  year2=2005,
  nBoot = 50,
  bootBreak = 39,
  blockLength = 200)

bootSetUp <- setForBoot(eList,caseSetUp)
```

trendSetUp *Interactive setup for running wBT, the WRTDS Bootstrap Test*

Description

Walks user through the set-up for the WRTDS Bootstrap Test. Establishes start and end year for the test period. Sets the minimum number of bootstrap replicates to be run, the maximum number of bootstrap replicates to be run, and the block length (in days) for the block bootstrapping. The test is designed to evaluate the uncertainty about the trend between any pair of years.

Usage

```
trendSetUp(eList, ...)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
...	additional arguments to bring in to reduce interactive options (year1, year2, nBoot, bootBreak, blockLength)

Value

caseSetUp data frame with columns year1, yearData1, year2, yearData2, numSamples, nBoot, bootBreak, blockLength, confStop. These correspond to:

Column Name	Manuscript Variable
year1	y_s
year2	y_e
nBoot	M_max
bootBreak	M_min
blockLength	B

See Also

setForBoot, wBT

Examples

```
eList <- EGRET::Choptank_eList

# Completely interactive:
# caseSetUp <- trendSetUp(eList)
# Semi-interactive:
# caseSetUp <- trendSetUp(eList, nBoot = 100, blockLength = 200)

# fully scripted:
```

```
caseSetUp <- trendSetUp(eList,
  year1=1985,
  year2=2005,
  nBoot = 50,
  bootBreak = 39,
  blockLength = 200)
```

---

wBT *Run the WBT (WRTDS Bootstrap Test)*

Description

Runs the WBT for a given data set to evaluate the significance level and confidence intervals for the trends between two specified years. The trends evaluated are trends in flow normalized concentration and flow normalized flux. Function produces text outputs and a named list (eBoot) that contains all of the relevant outputs. Check out runPairsBoot and runGroupsBoot for more bootstrapping options. The wBT only runs stationary flow normalization (i.e. making the assumption that discharge is stationary). The runPairsBoot and runGroupsBoot allow for generalized flow normalization (i.e. non-stationary discharge).

Usage

```
wBT(eList, caseSetUp, saveOutput = TRUE, fileName = "temp.txt",
  startSeed = 494817, jitterOn = FALSE, V = 0.2)
```

Arguments

eList	named list with at least the Daily, Sample, and INFO dataframes. Created from the EGRET package, after running modelEstimation.
caseSetUp	data frame. Returned from trendSetUp.
saveOutput	logical. If TRUE, a text file will be saved in the working directory.
fileName	character. Name to save the output file if saveOutput=TRUE.
startSeed	setSeed value. Defaults to 494817. This is used to make repeatable output.
jitterOn	logical, if TRUE, adds "jitter" to the data in an attempt to avoid some numerical problems. Default = FALSE. See Details below.
V	numeric a multiplier for addition of jitter to the data, default = 0.2. See Details below.

Details

In some situations numerical problems are encountered in the bootstrap process, resulting in highly unreasonable spikes in the confidence intervals. The use of "jitter" can often prevent these problems, but should only be used when it is clearly needed. It adds a small amount of random "jitter" to the explanatory variables of the WRTDS model. The V parameter sets the scale of variation in the log discharge values. The standard deviation of the added jitter is V * standard deviation of Log Q. The default for V is 0.2. Larger values should generally be avoided, and smaller values may be sufficient.

Value

eBoot, a named list with bootOut, wordsOut, xConc, xFlux, pConc, pFlux values.

Object	Description
bootOut	a data frame with the results of the bootstrap test.
wordsOut	a character vector describing the results.
xConc and xFlux	vectors of length iBoot, of the change in flow normalized concentration and flow normalized flux compute
pConc and pFlux	vectors of length iBoot, of the change in flow normalized concentration or flow normalized flux computed

See Also

trendSetUp, setForBoot, runGroupsBoot, runPairsBoot

Examples

```
eList <- EGRET::Choptank_eList
caseSetUp <- trendSetUp(eList,
                        year1 = 1985,
                        year2 = 2005,
                        nBoot = 50,
                        bootBreak = 39,
                        blockLength = 200)
# Very long-running function:
## Not run:
eBoot <- wBT(eList,caseSetUp)

## End(Not run)
```

Index