Package: smwrQW (via r-universe)

November 12, 2026

Version 0.7.14

Date 2017-07-24

Title Tools for censored data analysis

Depends dataRetrieval, smwrBase, smwrGraphs, smwrStats

Imports boot, coin, lubridate, methods, mvtnorm, survival, XML, zCompositions

Suggests smwrData, cluster, NADA, psych, knitr

Description This package provides tools for the analysis and graphing of censored data, focussing on water-quality data. It provides a uniform user interface to many different methods for the analysis.

License CC0

LazyLoad yes

LazyData yes

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

Collate 'qw-class.R' 'lcens-class.R' 'mcens-class.R' 'Arith-censored.R' 'Compare-censored.R' 'Math-censored.R' 'View.r' 'add.R' 'as.character.R' 'as.data.frame.R' 'as.double.R' 'as.lcens.R' 'as.mcens.R' 'as.qw.R' 'atrazine.df.R' 'atrazine.dl.R' 'atrazine.spk.R' 'boxPlot.lcens.R' 'boxPlotCensStats.R' 'c.R' 'censCor.R' 'censKSample.test.R' 'censMean.CI.R' 'censMove.1.R' 'censMulticomp.test.R' 'censPPCC.test.R' 'censQtiles.CI.R' 'censQuantile.R' 'censReg.R' 'censReg.fit.R' 'censReg.pred.R' 'censStats.R' 'censSumStats.R' 'censor.R' 'censoring.R' 'censorlevels.R' 'censpp.R' 'code01.R' 'codeU.R' 'coef.censReg.R' 'convert2qw.R' 'convertFqw.R' 'dlimit.R' 'dotPlot-censored.R' 'ecdfPlot.R' 'extractAIC.censReg.R' 'fillIn.R' 'fitted.censReg.R' 'format.R' 'gehanScores.R' 'genWilcox.test.R' 'importNWISqw.R' 'importQW.R' 'imputeLessThans.R' 'is.na.R' 'kendallATS.test.R' 'length.R' 'logLik.censReg.R' 'mImputeLessThans.R' 'makeColNames.R'

'mcenKM.R' 'mcenMLE.R' 'mcenROS.R' 'mdlAMLE.R' 'mdlKM.R'
'mdlKMstats.R' 'mdlMLE.R' 'mdlROS.R' 'mean.qw.R' 'meanStats.R'
'medianStats.R' 'multiply.R' 'nobs.censReg.R'
'pairedPratt.test.R' 'pcodeColData.R' 'pcodeNWISqw.R'
'pctCens.R' 'percentile.lcens.R' 'plot.censReg.R'
'plot.htest.R' 'plot.qw.R' 'plot.summary.censReg.R' 'pow.R'
'ppw.test.R' 'predict.censReg.R' 'print.censMCT.R'
'print.censQuantile.R' 'print.censReg.R' 'print.censStats.R'
'print.summary.censReg.R' 'printData.R' 'probPlot.R' 'qqPlot.R'
'quantile.R' 'qw2mcens.R' 'qwCoalesce.R' 'qwRPD.R' 'ratio.R'
'rbindQW.R' 'readNWQLdl.R' 'residuals.censReg.R'
'rmse.censReg.R' 'sdlFill.R' 'show-censored.R'
'smwrQW-package.R' 'sort.censored.R' 'splitQual.R' 'str.R'
'subset.R' 'summary.R' 'summary.censReg.R'
'timePlot-censored.R' 'vcov.censReg.R' 'vif.censReg.R'
'xyPlot-censored.R' 'z%eq%.R'

RoxygenNote 6.0.1

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository https://ldecicco-usgs.r-universe.dev

Date/Publication 2022-10-05 20:04:04 UTC

RemoteUrl https://github.com/USGS-R/smwrQW

RemoteRef HEAD

RemoteSha 7f9d3f24c9ec072da5dba9f3f3ad19b0c394854d

Contents

smwrQW-package 5
[.lcens 5
%~=% 7
add 8
Arith-censored 8
as.character.lcens 10
as.data.frame.lcens 10
as.double.qw 11
as.lcens 12
as.mcens 14
as.qw 16
atrazine.df 18
atrazine.dl 19
atrazine.spk 20
boxPlot.lcens 21
boxPlotCensStats 23
c.lcens 24
censCor 24
censKSample.test 26

censMean.CI . . . 27
censMove.1 . . . 29
censMulticomp.test . . . 30
censor . . . 31
censoring . . . 32
censorLevels . . . 33
censpp . . . 34
censPPCC.test . . . 36
censQtiles.CI . . . 37
censQuantile . . . 38
censReg . . . 40
censReg_AMLE.fit . . . 42
censReg_AMLE.pred . . . 42
censStats . . . 43
censSumStats . . . 44
code01 . . . 45
codeU . . . 46
coef.censReg . . . 46
Compare-censored . . . 47
convert2qw . . . 48
convertFqw . . . 49
dlimit . . . 50
dotPlot-censored . . . 52
ecdfPlot.lcens . . . 54
extractAIC.censReg . . . 55
fillIn . . . 56
fitted.censReg . . . 57
format.lcens . . . 58
gehanScores . . . 59
genWilcox.test . . . 60
importNWISqw . . . 62
importQW . . . 63
imputeLessThans . . . 65
is.na.lcens . . . 66
kendallATS.test . . . 67
lcens-class . . . 68
length.lcens . . . 69
logLik.censReg . . . 70
makeColNames . . . 71
Math-censored . . . 71
mcenKM . . . 72
mcenMLE . . . 73
mcenROS . . . 73
mcens-class . . . 74
mdlAMLE . . . 75
mdlKM . . . 75
mdlKMstats . . . 76
mdlMLE . . . 76

mdlROS . . . 77
mean.lcens . . . 78
mean.qw . . . 79
median.lcens . . . 80
mImputeLessThans . . . 81
multiply . . . 82
nobs.censReg . . . 83
pairedPratt.test . . . 83
pcodeColData . . . 85
pcodeNWISqw . . . 86
pctCens . . . 87
percentile.lcens . . . 88
plot.censReg . . . 89
plot.ppw . . . 90
plot.qw . . . 91
plot.summary.censReg . . . 91
pow . . . 93
ppw.test . . . 93
predict.censReg . . . 95
print.censMCT . . . 96
print.censQuantile . . . 97
print.censReg . . . 98
print.censStats . . . 98
print.lcens . . . 99
print.summary.censReg . . . 100
probPlot.lcens . . . 100
qqPlot.lcens . . . 102
quantile.lcens . . . 104
qw-class . . . 105
qw2mcens . . . 107
qwCoalesce . . . 107
qwRPD . . . 108
ratio . . . 109
rbindQW . . . 110
readNWQLdl . . . 110
residuals.censReg . . . 111
rmse.censReg . . . 112
sdlFill . . . 113
show-censored . . . 113
sort.lcens . . . 114
splitQual . . . 115
str.lcens . . . 115
summary.censReg . . . 116
summary.lcens . . . 117
timePlot-censored . . . 118
vcov.censReg . . . 119
View . . . 120
vif.censReg . . . 120

xyPlot-censored 121

Index **123**

`smwrQW-package` *Tools for censored water-quality data analysis*

Description

A collection of tools for the analysis of left- and multiply-censored data that focuses on the analysis of water-quality data.

Details

Package:	smwrQW
Type:	Package
License:	CC0
Depends:	dataRetrieval smwrBase smwrGraphs smwrStats
Imports:	boot coin lubridate methods mvtnorm survival XML zCompositions
Suggests:	smwrData cluster NADA psych

Modern water-quality information requires much more that a value and remark code to characterize the data. The tools in this library focus on the "qw" class, which store the value and the remark code, but also information about the reporting level, the units, the methods of analysis analysis, and the analyte itself.
Some specialized analytical tools for left-only censored data, requiring objects of class "lcens" have been developed and are included in this library. Other analytical tools for multiply- or any censored data are alos included. The tools include automatic conversion of "qw" data into the "best" (left-only or multiply censored) format.
Need a general description of the tools by group–a brief synopsis of the OFR.

References

Lorenz, D.L., in preparation.

`[.lcens` *Subset an Object*

Description

Extracts or subsets a vector: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
x[i, ...]

## S3 method for class 'mcens'
x[i, ...]

## S3 method for class 'qw'
x[i, ...]

## S3 method for class 'qw'
subset(x, subset, ...)
```

Arguments

x	any R object to subset.
i	index specifying elements to extract.
...	not used, required for other methods.
subset	an logical expression involving a slot name of the qw object.

Details

The values for i can be either numeric indexes or logical.
Valid slot names are "remark.codes," "reporting.level," "reporting.method," "reporting.units," "analyte.method," "analyte.name," and "unique.code."

Value

A subset of the object x.

Note

In general, assignment of specific values should not be done using the "[" extraction operator because it does not modify the meta data assiciated with a value. But any value can be set to NA by treating the value as a matrix and setting the equivalent of the row to NA. See the example below.

See Also

[.qw

Examples

```
Xcens <- as.lcens(c(1, 2, 3, 4), 2)
# Extract the 3rd value
Xcens[3]
# Change the 3rd value to NA and print the data
Xcens[3,] <- NA
Xcens
```

`%~=%` *Possible Equality*

Description

Checks for the possibility that two values are equal. If there is any overlap in range, they are considered possibly equal and `TRUE` is returned, otherwise `FALSE` is returned.

Usage

```
e1 %~=% e2
```

Arguments

`e1`	any vector that can be converted to class "mcens."
`e2`	any vector that can be converted to class "mcens."

Details

Missing values are permitted in either `e1` or `e2`. wherever either are missing the returned element is set to `NA`.

Value

A logical vector that matches the length of the longer of `e1` or `e2`, set to `TRUE` is there is overlap in the values, and `FALSE` otherwise.

See Also

`Compare,qw,numeric-method`

Examples

```
as.mcens(1, 2) %~=% 0.5
as.mcens(1, 2) %~=% 1.5
as.mcens(1, 2) %~=% 2.5
```

add *Add Water-Quality Data*

Description

Adds or subtracts water-quality data while trying to maintain the integrity of the data.

Usage

```
add(..., analyte, pcode = "", gt0 = TRUE)
```

Arguments

...	any number of water-quality (class "qw") or numeric vectors. The length of the water-quality data must all agree, but numeric values are replicated to the length of the water-quality data. See **Details**.
analyte	the name of the analyte, if not supplied, generated from ....
pcode	the parameter code of the analyte or any othe unique identifier the user chooses.
gt0	logical, if TRUE, then force values to be strictly non-negative on subtraction, otherwise negative values are permitted.

Details

The ... arguments must not be named in the call. Only objects of class "qw" or "numeric" are allowed. To perform a subtraction, simply preceed the argument with the unary - sign in the call. The first argument must be a positive valued object of class "qw."

Value

An object of class "qw" that is the result of the requested operations.

Arith-censored *Arithmetic Methods for* lcens*,* mcens*, and* qw *objects*

Description

Some limited arithmetic methods are possible and well-defined for censored data. Water-quality data require specialized functions to maintain data integrity.

Usage

```
## S4 method for signature 'lcens,numeric'
Arith(e1, e2)

## S4 method for signature 'numeric,lcens'
Arith(e1, e2)

## S4 method for signature 'lcens,lcens'
Arith(e1, e2)

## S4 method for signature 'mcens,numeric'
Arith(e1, e2)

## S4 method for signature 'numeric,mcens'
Arith(e1, e2)

## S4 method for signature 'mcens,mcens'
Arith(e1, e2)

## S4 method for signature 'qw,numeric'
Arith(e1, e2)

## S4 method for signature 'numeric,qw'
Arith(e1, e2)

## S4 method for signature 'qw,qw'
Arith(e1, e2)
```

Arguments

e1, e2	numeric, censored, or water-quality data. Missing values are permitted in either argument and result in a missing value in the output.

Value

An object of the appropriate class for the data.

See Also

add, ratio

Examples

```
as.lcens(c(1, 3), 2) + 1
as.lcens(c(1, 3), 2) * 2
```

as.character.lcens *Character Representations*

Description

Creates character representations of censored and water-quality data: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
as.character(x, ...)

## S3 method for class 'mcens'
as.character(x, ...)

## S3 method for class 'qw'
as.character(x, ...)
```

Arguments

x	any R object to coerce to character
...	further arguments passed to or from other methods.

Value

Creates a character representation of x.

as.data.frame.lcens *Coerce to a Data Frame*

Description

Creates data frames from censored and water-quality data: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
as.data.frame(x, row.names = NULL, optional = FALSE, ...,
  nm = deparse(substitute(x)))

## S3 method for class 'mcens'
as.data.frame(x, row.names = NULL, optional = FALSE, ...,
  nm = deparse(substitute(x)))
```

```
## S3 method for class 'qw'
as.data.frame(x, row.names = NULL, optional = FALSE, ...,
  nm = deparse(substitute(x)), expand = FALSE)
```

Arguments

x	any R object to put into a data frame.
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
optional	logical. If TRUE, set column names to nm.
...	additional arguments to be passed to or from methods.
nm	set as column name if optional is TRUE.
expand	create a data frame from all of the slots from the object instead of creating a data frame containting the object.

Value

A data frame constructred from x.

See Also

data.frame

---

as.double.qw *Numeric Representations*

Description

Converts data to numeric values: method for "qw" data.

Usage

```
## S3 method for class 'qw'
as.double(x, ...)
```

Arguments

x	the object to be converted.
...	further arguments passed to or from other methods.

Value

A numeric representation of x, using a method commonly refered to as simple substitution.

Note

The process of simple substitution proceeds in 4 steps for each kind of censoring: none, interval, left and right. First uncensored data are retained as is. Interval-censored data are converted by computing the mid rnge. Left-censored data are converted sequentially from the lowest censoring level, which is set to 1/2 the reporting level, any subsequent levels are replaced by the mean of all values less than the current reporting level. Conversion of right-censored data are not implemented in the current version and result in an error.

The function `as.double` is executed for `as.numeric`.

See Also

`fillIn`

Examples

```
## Create a simple qw object
Test.qw <- as.qw(c(1,2,3,2,4,4), c("<", "<", "<", " ", " ", " "), "", NA_real_,
"Miss", "None", "Mine", "TEST", "")
as.double(Test.qw)
as.numeric(Test.qw)
```

`as.lcens` *Left-censored Data Conversion*

Description

Converts data to a left-censored data (`lcens`) object: various methods.

Usage

```
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'lcens,missing,missing'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'numeric,missing,missing'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'numeric,numeric,missing'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'numeric,numeric,logical'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'numeric,numeric,character'
```

```
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'numeric,missing,logical'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'numeric,missing,character'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'qw,missing,missing'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'qw,numeric,missing'
as.lcens(values, detlim, censor.codes)

## S4 method for signature 'character,missing,missing'
as.lcens(values, detlim, censor.codes)
```

Arguments

`values`	numeric values representing "raw" values. Missing values are permitted. A vector of character strings is allowed when the remark code is combined with the value; blank values are treated as missing values, any other value is converted to a missing value with a warning. See **Examples**.
`detlim`	the corresponding detection limit of the sensor. Missing values are permitted. Detection limits are required for each non-missing value in `values`; they will be imputed if missing. The imputation scheme is fairly sophisticated, see Lorenz (in preparation) for details, so it is better to set to missing, rather than provide an arbitrary detection limit.
`censor.codes`	optional codes indicating a left-censored value. If class "logical," then `TRUE` indicates left-censored. If class "character," then "<" indicates left-censored and anything other than "" or " " generates a warning.

Value

An S4 object of class "lcens." These objects are the basis for the analysis of left-censored data.

Note

All methods force `values` to be no less than the corresponding `detlim`. All `values` that are less than `detlim` are treated as left-censored values at `detlim`. This ensures unbiased analytical results.

References

Lorenz, D.L., in preparation

See Also

`as.mcens`

Examples

```
## The first value is censored at 2
as.lcens(c(1,3), 2)
## Set the first value to censored at that level and the detection limit is
# carried forward
as.lcens(c(1,3), censor.codes=c("<", ""))
# For combined remark and values:
as.lcens(c("<1", "1", "<1", "1", "2"))
```

---

as.mcens *Multiply-censored Data Conversion*

Description

Converts data to a multiply-censored data (mcens) object: various methods.

Usage

```
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'mcens,missing,missing'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'lcens,missing,missing'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'numeric,missing,missing'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'numeric,numeric,missing'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'numeric,missing,numeric'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'numeric,missing,integer'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'numeric,missing,character'
as.mcens(lower.val, upper.val,
  censor.codes)

## S4 method for signature 'qw,missing,missing'
as.mcens(lower.val, upper.val, censor.codes)

## S4 method for signature 'character,missing,missing'
```

```
as.mcens(lower.val, upper.val,
  censor.codes)
```

Arguments

`lower.val`	The lower limit of the actual value, the special value of `-Inf` or `NA` can be used to indicate left-censoring. A vector of character strings is allowed when the remark code is combined with the value; blank values are treated as missing values, any other value is converted to a missing value with a warning. See **Examples**.
`upper.val`	The upper limit of the actual value, the special value of `Inf` or `NA` can be used to indicate right-censoring.
`censor.codes`	optional codes if `upper.val` is missing. Any numeric value less than 0 indicates left-censored, any value greater than 0 indicates right-censored, and 0 indicates an observed value. The character value "<" indicates left-censored, ">" indicates right-censored, any anything else indicates an observed value.

Details

In keeping with the logic of `Surv`, `NA` is permitted to indicate left- or right-censored data. If both are `NA`, then the observation is treated as missing.

Value

An S4 object of class "mcens." These objects are the basis for the analysis of censored data that are not strictly left censored.

References

Lorenz, D.L., in preparation

See Also

`Surv`

Examples

```
## Create one of each type of censoring, including uncensored
# the last value is missing
as.mcens(c(-Inf, 2, 2, 5, NA), c(1, 2, 3, Inf, NA))
# For combined remark and values:
as.mcens(c("<1", "1", ">3", "1", "2"))
```

as.qw *Water-quality Data Conversion*

Description

Converts data and meta data to a water-quality data (qw) object: various methods.

Usage

```
as.qw(values, remark.codes, value.codes, reporting.level, reporting.method,
  reporting.units, analyte.method, analyte.name, unique.code, value2)


  ## S4 method for signature
## 'numeric,
##   character,
##   character,
##   numeric,
##   character,
##   character,
##   character,
##   character,
##   character,
##   missing'
as.qw(values,
  remark.codes, value.codes, reporting.level, reporting.method, reporting.units,
  analyte.method, analyte.name, unique.code, value2)


  ## S4 method for signature
## 'numeric,
##   character,
##   character,
##   numeric,
##   character,
##   character,
##   character,
##   character,
##   character,
##   numeric'
as.qw(values,
  remark.codes, value.codes, reporting.level, reporting.method, reporting.units,
  analyte.method, analyte.name, unique.code, value2)
```

Arguments

values	the uncensored values and optionally the censored values.

`remark.codes` the remarks codes: " " or "" indicated uncensored, "<" indicates left-censoring, ">" indicates right censoring. The special remark code "M" is created when the value is rounded to 0 by NWIS. In this case, the value is set to less than the reporting level if the reporting level is not the missing value `NA`, otherwise the value is set to missing `NA` and "M" is retained as the remark code. Other codes have a special meaning that can be modified by `value.codes`.

`value.codes` special value qualifier codes. See Lorenz and others, 2014 for a description of these codes and the special `remark.codes`.

`reporting.level` the reporting level associated with each value. Note that if the reporting level is not known, then `NA_real_` should be used instead of using any arbitrary value.

`reporting.method` the method used to compute the reporting level. These can be arbitrary codes, known only to the user.

`reporting.units` the reporting units. A few defined units such as "mg/l" are recognized and used by the software.

`analyte.method` the method used to compute the concentration. These can be arbitrary codes, known only to the user.

`analyte.name` the name of the analyte. A few names are recongnized such as "chloride" and used by the software.

`unique.code` an arbitrary code identifying unique instances of data. For data retreived from the US Geological Survey, this should be the 5-digit parameter code.

`value2` for interval censored data, the upper value.

Details

Two methods for constructing water-quality data are defined in the current version. The first method works well for un-, left-, and right-censored data where the remark code and value can completely define the numeric values. The second method works for any censored data, including interval-censored data, where an upper and lower range of values are required. For the second method, `values` is the lower limit of the range and for left-censored data, may be either `-Inf`, `NA`, or `0`. If `-Inf`, then the value is treated as strictly left-censored, otherwise, the lower limit is set to `0`, which corresponds to the traditional interpretation of left-censored water-quality data. For the second method, `value2` is the upper limit and for right-censored data will always be set to `Inf`.

Value

An object of class "qw" that contains the numeric data in a two-column matrix and the meta data as additional slot information.

Note

The class "qw" is intended to be a flexible and user-friendly way to store and manage water-quality data. The meta data that are stored with the numeric and remark data are essential for unbiased statistical analysis and for reducing other errors in interpretation or computation.

Objects of class "qw" also have a slot named "rounding" that controls how the data are roudend when printed. That slot is an integer vector of length 2, the first element is the maximum number of significant figures and the second is the maximum number of decimal places to show. The default value used by `as.qw` is c(2L,3L). It can be changed by specifically modifying the slot value.

References

Lorenz, D.L., in preparation

See Also

`importNWISqw`

`atrazine.df` *Atrazine Data A dataset of atrazine samples, replicates, and blanks collected by the U.S. Geological Survey from October 1, 2001 through September 30, 2012.*

Description

Atrazine Data A dataset of atrazine samples, replicates, and blanks collected by the U.S. Geological Survey from October 1, 2001 through September 30, 2012.

Usage

```
atrazine.df
```

Format

Data frame with 13106 rows and 12 columns

Name	Type	Description
staid	character	USGS station number
Date	Date	Sample date
MEDIUM_CD	character	Sample medium code, see **Details**
SAMP_TYPE_CD	integer	Sample type code, see **Details**
sched	character	Lab schedule
RESULT_VA	numeric	Numeric value of the concentration
REMARK_CD	character	Remark code for the value
RPT_LEV_VA	numeric	Reporting level
RPT_LEV_CD	character	Reporting level code
Units	character	Units of the concentration
QW_METHOD_CD	character	Lab method code
PARAMETER_CD	character	Parameter code

Details

The sample medium codes are regular water-quality sample, WS; replicate water-quality sample, WSQ; and field blank, OAQ. The sample type codes are regular sample with no QAQC sample, 9; regular sample with QAQC or the replicate sample, 7; and field blank , 2.

Source

Data retreived from the national water information system and processed by Jeffery Martin and Laura Medalie of the U.S. Geological Survey.

Examples

```
data(atrazine.df)
# How many different kinds of samples?
with(atrazine.df, table(MEDIUM_CD))
# Now many stations?
with(atrazine.df, table(staid))
```

---

atrazine.dl *Atrazine Reporting Data A dataset of the reporting and detection levels for atrazine from the U.S. Geological Survey National Water Quality Lab.*

Description

Atrazine Reporting Data A dataset of the reporting and detection levels for atrazine from the U.S. Geological Survey National Water Quality Lab.

Usage

```
atrazine.dl
```

Format

Data frame with 23 rows and 9 columns

Name	Type	Description
ConstituentName	character	Name of the constituent
TestID	character	Parameter code and method
LabCode	character	Lab code for method
StartDate	Date	Start date for the reporting and detection levels
EndDate	Date	End date for the reporting and detection levels
ReportLevelCode	character	Reporting level code
DetectionLevel	numeric	Detection level
ReportingLevel	numeric	Reporting level
Units	character	Units of the concentration

Source

Data retrieved from the U.S. Geological Survey National Water Quality Lab reporting level database on January 4, 2016.

Examples

```
data(atrazine.dl)
# What methods; 39632 is the parameter code for atrazine
with(atrazine.dl, unique(TestID))
```

`atrazine.spk` *Atrazine Spike Data A dataset of atrazine environmental and spike samples collected by the U.S. Geological Survey from October 1, 2001 through September 30, 2012.*

Description

Atrazine Spike Data A dataset of atrazine environmental and spike samples collected by the U.S. Geological Survey from October 1, 2001 through September 30, 2012.

Usage

```
atrazine.spk
```

Format

Data frame with 596 rows and 22 columns

Name	Type	Description
pstaid	character	Not sure why we need this
suid	character	Study Unit Identifier, see **Details**
Date	Date	Date of sample
PARAMETER_CD	character	Parameter code
PARAMETER_NM	character	Parameter name
spkTime	character	Time of spike sample
spkMEDIUM_CD	character	Spike sample medium code
spkREMARK_CD	character	Spike sample remark code for value
spkRESULT_VA	numeric	Spike sample result value
spkSched	character	Spike sample schedule code
envTime	character	Time of environmental sample
envMEDIUM_CD	character	Environmental sample medium code
envREMARK_CD	character	Environmental sample remark code for value
envRESULT_VA	numeric	Environmental sample result value

envSched	character	Environmental sample schedule code
spkLotNo	numeric	Spike sample lot number
spkConc	numeric	Spike sample concentration
spkVol	numeric	Spike sample volume
envVol	numeric	Environmental sample volume
Recovery	numeric	Spike recovery, in percent
sname	character	USGS station name
staid	character	USGS station number

Details

A map of the study units can be found at https://water.usgs.gov/nawqa/nawqamap.html. Some of the study unit codes represent combined study units from that map.

Source

Data retreived from the national water information system and processed by Jeffery Martin of the U.S. Geological Survey.

Examples

```
data(atrazine.spk)
# The mean spike recovery
mean(atrazine.spk$Recovery)
```

---

boxPlot.lcens *Box Plot*

Description

Produces an estimated or censored style box plot for truncated, Tukey, simple, or extended type box plot: methods for "lcens" and "qw" data. The Tukey type box plot requires an estimated style box plot.

Usage

```
## S3 method for class 'lcens'
boxPlot(..., group = NULL, Box = list(type = "truncated",
  show.counts = TRUE, censorbox = NA, censorstyle = "censored", nobox = 5, width
  = "Auto", fill = "none", truncated = c(10, 90)), yaxis.log = TRUE,
  yaxis.range = c(NA, NA), ylabels = "Auto", xlabels = "Auto",
  xlabels.rotate = FALSE, xtitle = "", ytitle = "", caption = "",
  margin = c(NA, NA, NA, NA))

## S3 method for class 'qw'
boxPlot(..., group = NULL, Box = list(type = "truncated",
  show.counts = TRUE, censorbox = NA, censorstyle = "censored", nobox = 5, width
```

```
  = "Auto", fill = "none", truncated = c(10, 90)), yaxis.log = TRUE,
  yaxis.range = c(NA, NA), ylabels = "Auto", xlabels = "Auto",
  xlabels.rotate = FALSE, xtitle = "", ytitle = "", caption = "",
  margin = c(NA, NA, NA, NA))

## S3 method for class 'data.frame'
boxPlot(..., group = NULL, Box = list(type =
  "truncated", show.counts = TRUE, censorbox = NA, censorstyle = "censored",
  nobox = 5, width = "Auto", fill = "none", truncated = c(10, 90)),
  yaxis.log = FALSE, yaxis.range = c(NA, NA), ylabels = "Auto",
  xlabels = "Auto", xlabels.rotate = FALSE, xtitle = "", ytitle = "",
  caption = "", margin = c(NA, NA, NA, NA))
```

Arguments

`...`	the data to plot.
`group`	create groups of a single left-censored data vector. Invalid for anything else
`Box`	control parameters for the box. See `boxPlot`
`yaxis.log`	logical: log transform y axis?
`yaxis.range`	set y-axis range.
`ylabels`	set up y-axis labels. See `linearPretty` for details.
`xlabels`	set up x-axis labels.
`xlabels.rotate`	rotate x-axis labels 90 degrees?
`xtitle`	the x-axis title.
`ytitle`	the y-axis title.
`caption`	the figure caption.
`margin`	set up the plot area margins.

Details

If group is numeric, then the boxes will be plotted along a continuous numeric axis. Otherwise the x-axis will be discrete groups.

Value

Information about the graph.

Note

The censored style box plot truncates the boxplot at the largest value of the reporting level. The estimated style box plot estimates values for left-censored data and uses those estimates to construct the full boxplot; the range of estimated values is shown in gray. The maximum reporting level, except for some simple heuristics for identifying elevated reporting levels, is used for the censoring level when there are multiple reporting levels.

The `fillIn` function is used to estimate values that are censored. If `yaxis.log` is `TRUE`, then the "log ROS" method is used to estimate those values, otherwise, the "ROS" method is used.

Occasionally, estimated values by `fillIn` will estimate values for censored data that are greater than the censoring level. In those cases, a red bar is drawn on the box plot at the largest estimated value.

The default method for data frames in smwrGraphs is replaced with a method that supports data frames that contain columns of class "qw" or "lcens."

See Also

`fillIn`

Examples

```
## Not run:
set.seed(1932)
Xu <- rlnorm(32)
setPage("sq") # required page set up
boxPlot(as.lcens(Xu, 1.0))

## End(Not run)
```

`boxPlotCensStats` *Not exported*

Description

Computes the statistics for a left-censored data boxplot.

Usage

```
boxPlotCensStats(x, Box, yaxis.log)
```

Arguments

`x`	list of the lcens data
`Box`	the Box parameters
`yaxis.log`	is it?

Value

Info to actually plot the boxplot.

c.lcens *Combine Values into a Vector*

Description

Combines arguments to form a vector for censored or water-quality data: methods for "cens," "mcens," and "qw" data. All arguments are coerced to a common type which is the type of the first argument, except for the qw method,

Usage

```
## S3 method for class 'lcens'
c(..., recursive = FALSE)

## S3 method for class 'mcens'
c(..., recursive = FALSE)

## S3 method for class 'qw'
c(..., recursive = FALSE)
```

Arguments

...	any objects that can be coerced to the class of the first one.
recursive	not used.

Value

A vector of an appropriate class.

Examples

```
c(as.lcens(c(1,3), 2), 2)
```

censCor *Correlation*

Description

Computes the maximum likelihood estimate of the correlation between two possibly left-censored vectors. It is equivalent the the Pearson product-moment correlation for uncensored data.

Usage

```
censCor(x, y, Full = TRUE, na.rm = TRUE)
```

Arguments

x	any data that can be converted to a left-censored data object.
y	any data that can be converted to a left-censored data object.
Full	how to compute the mean and standard deviation of x and y. See **Details**.
na.rm	logical, remove missing values before computing the correlation?

Details

Full may be either logical or a numeric vector. If Full is TRUE, then estimate the means and standard deviations for x and y. If Full is FALSE, use the initial maximum likelihood estimate for those statistics. Otherwise Full can be a named vector containing mnx, the mean for x; sdx, the standard deviation for x; mny, the mean for y; sdy, the standard deviation for y. Full can be set to FALSE if the optimization fails at large censoring levels or to improve processing speed for large sample sizes.

Value

A vector with these names:

cor	the correlation between x and y.
mnx	the mean of x.
sdx	the standard deviation of x.
mny	the mean of y.
sdy	the standard deviation of y.
cx	the proportion of censored values of x.
cy	the proportion of censored values of y.
cxy	the proportion of censored values common to x and y.
n	the number of observations.
ll0	the log likelihood for cor=0
llcor	the log likelihood for cor=cor

References

Lyles, R.H., Williams, J.K., and Chuachoowong R., 2001, Correlating two viral load assays with known detection limits: Biometrics, v. 57 no. 4, p. 1238–1244.

Examples

```
# Simple no censoring
set.seed(450)
tmp.X <- rnorm(25)
tmp.Y <- tmp.X/2 + rnorm(25)
cor(tmp.X, tmp.Y)
censCor(tmp.X, tmp.Y)
# Some censoring
censCor(as.lcens(tmp.X, -1), as.lcens(tmp.Y, -1))
```

censKSample.test *Test for difference in left-censored samples*

Description

Performs a test for differences in left-censored samples from 2 or more groups.

Usage

```
censKSample.test(x, groups, type = "Peto", data.names)
```

Arguments

x	the samples from each group. Forced to class "lcens." Missing values are removed before the analysis.
groups	the group identifier for each sample. Missing values are removed before the analysis.
type	the string "Peto" or "log-rank," depending on the desired type of test. Default is "Peto."
data.names	character string to be used to explain the data. Default names are derived from the data arguments.

Value

An object of class "htest."

Null Hypothesis

The null hypothesis is that the distributions in each group are not different from one another.

Note

The censKSample.test function use the survdiff function to test for differences. Helsel (2012) describes flipping the left-censored data so that small values become large and left-censored values become right-censored values and adapt nonparametric techniques from survival analysis.

Tableman and others (2004) state, "the Peto test emphasizes the beginning of the survival curve in that earlier failures receive larger weights. The log-rank test emphasizes the tail of the survival curve in that it gives equal weight to each failure." For flipped data, earlier failures translates to larger uncensored values. In general, the Peto test gives results that are more similar to the Wilcoxon test if all data are uncensored.

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

Peto, R., and Peto, J., 1972, Asymptotically efficient rank invariant test procedures (with discussion): Journal of the Royal Statistical Society, Series A v. 135, p. 185-206.

Prentice, R.L. 1978, Linear rank tests with right-censored data: Biometika, v 65, p 167-179.

Prentice, R.L., and Marke, P., 1979, A qualitative discrepancy between censored data rank tests: Biometrika, v. 35, p. 861-867.

Tableman, M., and Kim, J.S., 2004, Survival analysis using S—analysis of time-to-event data: Boca Raton, Fla., Chapman and Hall, 280 p.

See Also

survdiff, lcens-class

Examples

```
# Compare uncensored results
set.seed(69)
Xu <- rlnorm(22, 0, 1)
Yu <- rlnorm(22, .4, 1)
Zu <- rlnorm(22, 1, 1)
Gu <- factor(rep(c("X", "Y", "Z"), each=22))
censKSample.test(c(Xu, Yu, Zu), Gu)
kruskal.test(c(Xu, Yu, Zu), Gu)
# Censor the data at 1
censKSample.test(as.lcens(c(Xu, Yu, Zu), 1), Gu)
```

---

censMean.CI *Mean with Confidence Limits*

Description

Computes the mean and confidence limits of left-censored data.

Usage

```
censMean.CI(x, method = "log AMLE", CI = 0.9, bound = c("two.sided",
  "upper", "lower"), alpha = 0.4)
```

Arguments

x	any R object than can be converted to class "lcens," or "mcens" for multiply censored data, to compute the mean. Missing values are permitted and are removed before computing the mean.
method	the method to use for computing the statistics. See **Details**.
CI	the minimum desired confidence interval for each level specifed in probs.
bound	a character string indicating the desired bounds, "two.sided" means the two-sided interval, "upper" means the upper bound of the interval, and "lower" means the lower bound of the interval. Only a single character is needed. The lower confidence limit is -Inf or 0 when bound is "upper" and the upper confidence limit is Inf when bound is "lower."
alpha	the offset fraction to be used, depending on method; typically in [0,1].

Details

The methods available in the current version are "log MLE," "MLE," "log ROS," "ROS," "log AMLE," and "AMLE." The methods "log ROS," "log MLE," and "log AMLE" are described by Helsel (2012) and Helsel and Cohn (1988). The methods "ROS," "MLE," and "AMLE" are similar to the previous except that no log- and back-transforms are made on the data. For "log" methods, if any value in x is negative, then a warning is generated and the returned value contains missing values.

Value

A matrix of the mean, the lower confidence limit, the upper confidence limit, and the probability represented by the confidence interval.

Note

Helsel (2012) describes computing the confidence interval for the mean of normally dsitributed data in Sections 7.4.1 and 7.4.2; those approaches are used for method "MLE" and "AMLE." Helsel (2012) describes computing the confidence interval for the mean of lognormally dsitributed data in Sections 7.5.1 and 7.5.2; those approaches are used for method "log MLE" and "log AMLE." Helsel (2012) describes bootstrapping the confidence interval for the mean when method is "ROS" or "log ROS" in Section 7.6. For this function, the number of replicates is set to a value to guarantee at least 50 replicate values lie outside of the requested confidence interval. The percentage type confidence interval is reported.

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

See Also

censStats

Examples

```
## Generate a random lognormal sample
set.seed(221)
XX.rn <- rlnorm(30)
# Compare log AMLE and log ROS
censMean.CI(XX.rn, method="log AMLE", CI=.80)
censMean.CI(XX.rn, method="log ROS", CI=.80)
# check upper bound for AMLE
censMean.CI(XX.rn, method="log AMLE", CI=.90, bound="upper")
```

---

censMove.1 *Censored Maintenance of Variance Extension, Type 1*

Description

Calculates the Maintenance Of Variance Extension, Type 1 (MOVE.1) for record extension by fitting a Line of Organic Correlation (LOC) regression model to left-censored data.

Usage

```
censMove.1(formula, data, subset, na.action, distribution = "normal")
```

Arguments

formula	a formula with a single response variable on the left and a single explanatory variable on the right. See **Details**.
data	the data to search for the variables in formula.
subset	an expression to select a subset of the data.
na.action	what to do with missing values.
distribution	the distribution of the data, either "normal," "lognormal," or "commonlog." See **Details**.

Details

The left- or right-hand side of the formula may be any numeric variable, just as with move.1 or a variable of class "lcens." The response variable must be the name of a column and not converted using as.lcens in the call.

If distribution is "normal," then the data in x and y are assumed to have a bivariate normal distribution. Otherwise, they are assumed to have a bivariate log-normal distribution and a logarithmic transform is applied to both x and y before analysis. The natural logarithm is used if distribution is "lognormal" and the commmon logarithm is used if distribution is "commonlog."

Value

An object of class "move.1" as decribed in the documentation for move.1. Except, there are no fitted.values or residuals components, the values of x and y include estimated values (See **Note**), and components cx and cy, which indicate left-censoring are included.

Note

Because predictions are made using the `predict` method for class "move.1," and do not expect censored values, the explanatory variable in `formula` must refer to a column in `data` and not converted using the `as.lcens` function; the user must convert the explanatory variable column to class "lcens" before using `censMove.1`.

The components `x` and `y` are the actual values if the data are uncensored. If `y` is censored and `x` is not censored, then `y` is the expected value of `y`, given its censored value and the value of `x` and likewise for `x` censored and `y` not censored. If both are cesnored, then the values are computed from their expected values given the censored value without respect to any condition of the other.

References

will need some.

See Also

`move.1`

Examples

```
## Not run:
library(smwrData)
data(IonBalance)
# Build model for non missing Alkalinity
move.1(Anion_sum ~ Cation_sum, data=IonBalance,
 subset=abs(Pct_Diff) < 10)
# Compare to censored
censMove.1(Anion_sum ~ Cation_sum, data=IonBalance,
 subset=abs(Pct_Diff) < 10)
# The difference in standard deviations is due to using sd/MLE

## End(Not run)
```

---

censMulticomp.test *Multiple Comparisons*

Description

Performs a multiple nonparametric comparison test among groups of left-censored data.

Usage

```
censMulticomp.test(x, groups, method = "holm", alpha = 0.05)
```

Arguments

x	the observations the data are forced to class "lcens." Missing values (NAs) are allowed and removed before the test is performed.
groups	any group vector for the observations. Missing values (NAs) are allowed and removed before the test is performed.
method	the method for adjustment of p-value. See p.adjust for options and details.
alpha	the significance level of the test.

Value

An object of class "censMCT."

Note

need some kind of general description.

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

See Also

censKSample.test

Examples

```
set.seed(69)
Xu <- rlnorm(22, 0, 1)
Yu <- rlnorm(22, .4, 1)
Zu <- rlnorm(22, 1, 1)
Gu <- factor(rep(c("X", "Y", "Z"), each=22))
# Censor the data at 1
censKSample.test(as.lcens(c(Xu, Yu, Zu), 1), Gu)
censMulticomp.test(as.lcens(c(Xu, Yu, Zu), 1), Gu)
```

---

censor *Multiply-censored Data Conversion*

Description

Censors a vector at user-set limits for left- or right-censoring.

Usage

```
censor(x, left = -Inf, right = Inf)
```

Arguments

x	a numeric vector. Missing values are permitted.
left	a vector of left-censoring thresholds. If the length is greater than 1, then they are replicated to the length of x.
right	a vector of right-censoring thresholds. If the length is greater than 1, then they are replicated to the length of x.

Value

An S4 object of class "lcens" if right is Inf, otherwise an S4 object of class "mcens."

Note

Missing velues are generated wherever right is less than or equal to left.
The censor function is provided as a simple function to create either left- or multiply-censored data from numeric values, primarily for simulation pruposes. The functions as.lcens or as.mcens offer greater flexibility and should be used for real data.

References

Lorenz, D.L., in preparation

See Also

as.lcens, as.mcens

Examples

```
## Create a random vector
set.seed(651)
x.tmp <- rnorm(5)
censor(x.tmp, 0, .75)
```

---

censoring *Describe Censoring*

Description

Returns the type of censoring ("none," "left," "multiple") for an object.

Usage

```
censoring(x)

## Default S3 method:
censoring(x)

## S3 method for class 'lcens'
censoring(x)

## S3 method for class 'mcens'
censoring(x)

## S3 method for class 'qw'
censoring(x)
```

Arguments

x	the object to get the type of censoring.

Value

A character string "none," "left," or "multiple" describing the type of censoring present in the object.

Note

This function is mostly used within other functions to determine the 'best' technique to use for analysis.

See Also

censorLevels

Examples

```
censoring(2.3) # a simple number
censoring(as.lcens(1, 2)) # left-censored
```

---

censorLevels *Describe Censoring*

Description

Returns all censoring levels for an object.

Usage

```
censorLevels(x, type = "left")

## Default S3 method:
censorLevels(x, type = "left")

## S3 method for class 'lcens'
censorLevels(x, type = "left")

## S3 method for class 'mcens'
censorLevels(x, type = "left")

## S3 method for class 'qw'
censorLevels(x, type = "left")
```

Arguments

x	the object to get the levels of censoring.
type	a character string indicating the type of censoring levels, must be either "left" or "right."

Value

A numeric vector of the sorted censoring levels. If all data are uncensored, then -Inf is returned for type = "left" and Inf is returned for type = "right."

Note

Interval censoring is not characterized by a single value, so it is not an option for type.

See Also

censoring

Examples

```
censorLevels(2.3) # a simple number
censorLevels(as.lcens(1, 2)) # left-censored
```

---

censpp *Probability Plotting Positions*

Description

Generates the sequence of probability points for uncensored and censored values

Usage

```
censpp(x, a = 0.44)

## Default S3 method:
censpp(x, a = 0.44)

## S3 method for class 'lcens'
censpp(x, a = 0.44)

## S3 method for class 'mcens'
censpp(x, a = 0.44)
```

Arguments

x	a vector of observations
a	the offset fraction to be used; typically in [0, 0.5].

Value

A list with these components:

Note that xrcn and pprcn are provided only if there are right-censored values.

x	sorted uncensored observations.
pp	corresponding plotting positions for uncensored observations.
xcen	sorted left-censored observations.
ppcen	corresponding plotting positions for left-censored observations.
xrcn	sorted right-censored observations.
pprcn	corresponding plotting positions for right-censored observations.

Note

The left-censored plotting positions are adjusted by the method of Helsel and Cohn (1988), which produces a mean of 0.5 for all data.

References

Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp.1997-2004

See Also

ppoints

Examples

```
# This example demonstrates the probability adjustments made for censoring
# The raw, uncensored data:
ppoints(7, .44)
# censor ar 0.5 (the lowest 3 values)
censpp(as.lcens(ppoints(7, .44), 0.5))
```

`censPPCC.test` *Test for Normality*

Description

Computes the probability plot correlation coefficient test for departures from normality for left-censored data.

Usage

```
censPPCC.test(x)
```

Arguments

`x`	a vector of left-censored observations, missing values are removed before computation.

Value

An object of class "htest" having these components:

`statistic`	the value of the test statistic.
`p.value`	the attained p-value for the test.
`data.name`	a character string describing the name of the data used in the test.
`method`	a description of the method.
`alternative`	a description of the null and alternate hypotheses.

Note

The PPCC test is attractive because it has a simple, graphical interpretation: it is a measure of the correlation in a Q-normal plot of the uncensored data and is related to the Shapiro-Francia test for normality described by Royston (1993). The extension to multiple detection limit data is described in the help files in Millard (2001). The attained p-value for multiple detection limit data is only approximate.

References

Millard, S.P., 2001, EnvironmentalStats for Spotfire S+ Help, Version 2.0. Probability, Statistics & Information, Seattle, WA.
Royston, P., 1993, A toolkit for testing for non-normality in complete and censored samples: The Statistician, v. 42, p. 37–43.

See Also

`ppcc.test`, `censpp`

`censQtiles.CI` *Quantiles with Confidence Limits*

Description

Computes sample quantiles and confidence limits of left-censored data for specified probabilities.

Usage

```
censQtiles.CI(x, probs = 0.5, CI = 0.9, bound = c("two.sided", "upper",
  "lower"))
```

Arguments

`x`	any R object than can be converted to class "lcens" to compute the sample quantiles. Missung values are permitted and are removed before computing the quantiles.
`probs`	numeric vector of desired probabilities with values between 0 and 1.
`CI`	the minimum desired confidence interval for each level specifed in probs.
`bound`	a character string indicating the desired bounds, "two.sided" means the two-sided interval, "upper" means the upper bound of the interval, and "lower" means the lower bound of the interval. Only a single character is needed. The lower confidence limit is `-Inf` when bound is "upper" and the upper confidence limit is `Inf` when bound is "lower."

Details

The `survfit` function in the `survival` package is used to compute the flipped Kaplan-Meier statistics. The type of confidence interval is "plain," which produces symmetric confidence intervals that maintain consistency when back flipping the estimates and is consisten with the B-C method described by Helsel (2012).

Value

A matrix of sample quantiles, the lower confidence limit, the upper confidence limit, and the probability represented by the confidence interval corresponding to the probs levels in the sorted x data. Missing values denote values less than the minimum (either censored or uncensored) value. The minimum value is included as the "minimum" attribute of the matrix. The maximum value is included as the "maximum" attribute of the matrix.

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

See Also

censQuantile

Examples

```
## Generate a random sample
set.seed(222)
XX.rn <- rexp(32)
# Compare uncensored to the binomial method
censQtiles.CI(XX.rn, probs=c(.25, .5, .75))
qtiles.CI(XX.rn, probs=c(.25, .5, .75))
```

---

censQuantile *Sample Quantiles*

Description

Computes sample or estimated quantiles corresponding to the given probabilities. The smallest observation (censored or uncensored) corresponds to a probability of 0 and the largest to a probability of 1.

Usage

```
censQuantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE, method = "ROS",
  type = 2, alpha = 0.4)

## Default S3 method:
censQuantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  method = "", type = 2, alpha = 0.4)

## S3 method for class 'lcens'
censQuantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  method = "ROS", type = 2, alpha = 0.4)
```

```
## S3 method for class 'mcens'
censQuantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  method = "flipped K-M", type = 2, alpha = 0.4)

## S3 method for class 'qw'
censQuantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  method = "log MLE", type = 2, alpha = 0.4)
```

Arguments

x	an object of a censored-data class whose sample quantiles are wanted.
probs	numeric vector of probabilities with values in [0,1].
na.rm	logical; if TRUE, any NA and NaNs are removed from x before the quantiles are computed.
method	the method to use for computing quantiles. See **Details**.
type	an integer between 1 and 9 selecting one of the nine quantile algorithms detailed below to be used. See quantile for a description of the types.
alpha	the offset fraction to be used, depending on method; typically in [0,1].

Details

The methods available in the current version are "flipped K-M," "log ROS," "ROS," "log MLE," and "MLE." The method "flipped K-M" produces quantiles using the Kaplan-Meier method on fliiped data described by Helsel (2012). The methods "log ROS" and "log MLE" are described by Helsel, 2012 and Helsel and Cohn (1988). The methods "ROS" and "MLE" are similar to "log ROS" and "log MLE" except that no log- and back-transforms are made on the data.

Value

An object of class "censQuantile" with one named component for each level requested in probs. The values returned for method="flipped K-M" can be of class "lcens."

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.
Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp. 1997-2004

See Also

quantile, quantile.lcens

Examples

```
set.seed(28)
Xu <- rnorm(23)
censQuantile(Xu) # Uncensored (true) quartiles
censQuantile(as.lcens(Xu, 0)) # How'd ROS do?
```

```
censQuantile(as.lcens(Xu, 0), method="MLE") # How'd ROS do?
```

---

censReg *Censored Data Regression*

Description

Builds a linear regression model for censored response data.

Usage

```
censReg(formula, data, subset, weights, na.action, dist = "normal")
```

Arguments

formula	a formula describing the regression model. See **Details**.
data	the data to search for the variables in formula.
subset	an expression to select a subset of the data.
weights	a variable to use for weights.
na.action	what to do with missing values.
dist	the distribution of the data, either "normal," "lognormal," or "commonlog." See **Details**.

Details

The left-hand side of the formula may be any numeric variable, just as with lm or a variable of class "lcens," "mcens," or "qw."
For un- or left-censored data, AMLE is used unless weights are specified in the model, then MLE is used, through a call to survreg. For any other censored data, MLE is used.
If dist is "normal," then the regression analysis assumes that the residuals are normally distributed. If dist is "lognormal," then the regression analysis assumes that the residuals are lognormally distributed. In this case, predicted values are back transformed and optionally bias corrected to represent the expected mean. If dist is "commonlog," then the response data are transformed using log10 and those residuals are assumed to be normally distributed. No back transformation is made for predicted values.

Value

An object of class "censReg."

Note

A special feature of censReg is the ability to expand the formula passed as a variable rather than as an expressly entered formula. This feature is intended to facilitate certain kinds of scripts that construct a formula from combinations of variables in a dataset.

Objects of class "censReg" have these methods:

coef extract the regression coefficents

extractAIC compute another information criterion (AIC) value

fitted extract the fitted values

logLik compute the log-likelihood of the fitted model

nobs extract the number of observations used to fit the model

plot create some basic diagnostic plots

predict predict new values from the model

print print the model

residuals extract the residuals from the model

rmse compute the root mean squared error (residual standard error)

summary compute summary information about the model

vcov extract the variance-covariance matrix from the model

vif compute the variance inflation factors for the explanatory variables in the model

The AMLE method has arbitrary limitations on the size of the data: 24 explanatory variables and 5000 observations. Exceeding these will generate a warning message with the error code in the output object.

References

Lorenz, 2015, smwrQW.
Breen, R., 1996, Regression models: censored, sample selected, or truncated data: Sage University Paper series on Quantitative Applications in the Social Sciences, 07-111, Thousand Oaks, CA,
Cohn, T.A., 1988, Adjusted maximum likelihood estimation of moments of lognormal populations from type I censored samples: U.S. Geological Survey Open-File Report 88-350, 34 p.

See Also

lm, survreg

Examples

```
set.seed(345)
X <- runif(24, 1, 5)
Y <- X/2 + rnorm(24)
lm(Y ~ X) # the uncensored regression
censReg(as.lcens(Y, 1) ~ X) # censored at 1
```

censReg_AMLE.fit *Support Functions for Censored Regression*

Description

These are support functions for fitting censored regression. They perform either AMLE or MLE regression.

Usage

```
censReg_AMLE.fit(Y, X, dist = "normal")

censReg_MLE.fit(Y, X, Wt, dist = "normal")
```

Arguments

Y	the response data.
X	a matrix of the explanatory variables, including the intercept term.
dist	the distribuiton, either "normal" or "lognormal."
Wt	observation weights.

Value

a list containing the output from the FORTRAN code,or converted from survreg.

See Also

censReg, survreg

censReg_AMLE.pred *Support Function for Censored Regression*

Description

Support function for censored regression to predict values from a censored regression model.

Usage

```
censReg_AMLE.pred(fit, XPRED)
```

Arguments

fit	the output from censReg_AMLE.fit.
XPRED	a matrix of the prediction variables.

Value

A list containing the output from the FORTRAN code.

See Also

censReg,

---

censStats *Mean and Standard Deviation*

Description

Computes the mean and standard deviation of censored data.

Usage

```
censStats(x, method = "MLE", na.rm = FALSE, alpha = 0.4)

## Default S3 method:
censStats(x, method = "MLE", na.rm = FALSE, alpha = 0.4)

## S3 method for class 'lcens'
censStats(x, method = "MLE", na.rm = FALSE, alpha = 0.4)

## S3 method for class 'mcens'
censStats(x, method = "MLE", na.rm = FALSE, alpha = 0.4)

## S3 method for class 'qw'
censStats(x, method = "log MLE", na.rm = FALSE, alpha = 0.4)
```

Arguments

x	an object of a censored-data class whose sample mean and standard deviation are needed.
method	the method to use for computing the statistics. See **Details**.
na.rm	logical; if TRUE, any NA and NaN's are removed from x before the quantiles are computed.
alpha	the offset fraction to be used, depending on method; typically in [0,1].

Details

The methods available in the current version are "log MLE," "MLE," "log ROS," "ROS," "log AMLE," "AMLE," and "flipped K-M." The method "flipped K-M" produces statistics using the Kaplan-Meier method on flipped data described by Helsel (2012). The methods "log ROS," "log MLE," and "log AMLE" are described by Helsel (2012) and Helsel and Cohn (1988). The methods "ROS," "MLE," and "AMLE" are similar to the previous except that no log- and back-transforms are made on the data. For "log" methods, if any value in x is negative, then a warning is generated and the returned list contains missing values.

Value

A list with one component for mean and sd. If "log" methods are specified, then the meanlog amd meansd components are also included. The values returned for `method="flipped K-M"` can be of class "mcens."

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp.1997–2004

See Also

`mdlAMLE`, `mdlKM`, `mdlKMstats`, `mdlMLE`, `mdlROS`

Examples

```
set.seed(936)
Y <- rlnorm(32)
# Uncensored statistics
censStats(Y, method="log MLE")
# Censored statistics, at 1 (37.5 percent)
censStats(as.lcens(Y, 1), method="log MLE")
```

`censSumStats` *Summary Statistics*

Description

Computes selected summary statistics for censored data.

Usage

```
censSumStats(..., group = NULL, Nums = TRUE, Stats = TRUE,
  Probs = (0:4)/4, na.rm = TRUE, method = "log MLE")
```

Arguments

`...`	any number of vectors or a data frame of the data to be summarized. If `...` is a data frame, then only numeric data, including "lcens," "mcens," and "qw" data types, will be summarized.
`group`	the data to group the data in `....`
`Nums`	logical, include the number of observations and numbers of censored values in the output data.frame.
`Stats`	logical, include the mean and standard deviation in the output data.frame.

Probs	the argument to quantile specifying the desired probabilities in the output data.frame
na.rm	logical, remove missing values before processing?
method	the method to use for computing the statistics. Must be one of "log MLE," "MLE," "log ROS," "ROS," or "flipped K-M." The method "flipped K-M" is only applicaple to un- or left-censored data.

Value

A data.frame with the selected statistcs for each variable in ... grouped by each unique value in group.

---

code01 *Code Censored Values*

Description

Recodes censored data to a binary coding system with 0 indicating less than some critical value and 1 indicating greater than or equal to that critical value.

Usage

```
code01(..., criteria)

## Default S3 method:
code01(..., criteria)

## S3 method for class 'data.frame'
code01(..., criteria)
```

Arguments

...	either a data frame that contains columns of class "qw" or any combination of individual vectors of class "integer," "numeric," "lcens," or "qw." Missing values are removed before processing.
criteria	a vector indicating the critical values for each repsective value in .... May be a named vector containing critical values for the named argument in ... or not supplied. For any argument in ... that does not have a critical value, the critical value is the largest censoring level or if the data have no censored values, the median. The critical value can never be smaller than the largest censoring level and will be changed with a warning message. A special case is for integer values that have a minimum of 0—for those data the default criterion is to return 0 if the value is 0 and return 1 for any other value.

Value

A data frame containing the binary coded values. Each column has an attribute called "critical.value" that reports the critical value used to recode the values in that column. The column names are taken from ...; the rownames are derived from the sequential numbers of the original data.

---

codeU *Code Censored Values*

Description

Recodes censored data to u-scores, which is the sum of the sign of the differences beween each value and all other values. It is equivalent to the rank but scaled so the the mean is 0.

Usage

```
codeU(...)

## Default S3 method:
codeU(...)

## S3 method for class 'data.frame'
codeU(...)
```

Arguments

...	either a data frame that contains columns of class "qw" or any combination of individual vectors of class "numeric," "lcens," or "qw." Missing values are removed before processing.

Value

A data frame containing the u-coded values. The column names are taken from ...; the rownames are derived from the sequential numbers of the original data.

---

coef.censReg *Extract Model Coefficients*

Description

Extracts the model coefficients from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
coef(object, summary = FALSE, ...)
```

Arguments

object	the output from censReg.
summary	include standard errors and other information?
...	further arguments passed to or from other methods.

Value

Either a names vector of the coefficients, if summary is FALSE or a matrix of the coefficients, their standard errors, z-scores, and attained p-values, if summary is TRUE.

Note

The attained p-values are computed from the log-likelihood test for AMLE regression and from a Wald chi-square test for MLE regression.

See Also

censReg,

Compare-censored *Comparison Methods for* lcens, mcens, *and* qw *objects*

Description

Comprisons are possible and well-defined between numeric and censored or water-quality data. Comparisons between censored data types are not supported in this version.

Usage

```
## S4 method for signature 'lcens,numeric'
Compare(e1, e2)

## S4 method for signature 'mcens,numeric'
Compare(e1, e2)

## S4 method for signature 'qw,numeric'
Compare(e1, e2)

## S4 method for signature 'numeric,lcens'
Compare(e1, e2)

## S4 method for signature 'numeric,mcens'
Compare(e1, e2)

## S4 method for signature 'numeric,qw'
Compare(e1, e2)
```

Arguments

e1, e2	numeric, censored, or water-quality data. Missing values are permitted in either argument and result in a missing value in the output.

Details

For all comparisons, the censored data are converted to multiply censored data, class "mcens." The conversion from class "qw" preserves the strict interpretation of less-than values as greater than or equal to 0 and less than the reporting level. The distinction of being strictly less than is lost for the conversion of left-censored data, class "lcens."

Value

A logical vector the represents the result of the comparison between each element in e1 and e2.

Examples

```
as.lcens(c(1, 3), 2) > 1
as.lcens(c(1, 3), 2) < 4
```

---

convert2qw *Water-Quality Data*

Description

Converts information from discrete water-quality samples to data of class "qw." Attempts to construct the best-fit estimates for data of class "qw" from more limited data sources like just remark codes and values. But will also convert data created by as.data.frame(object of class qw, expand=TRUE).

Usage

```
convert2qw(data, scheme = "booker")
```

Arguments

data	a data frame that contains limited sample information.
scheme	a character string that describes the naming style of the variables in data that are to be conterted to class "qw." See **Details**.

Details

The options for scheme are "booker," "qw," and "partial." The "booker" naming scheme assumes that remark columns begin with "R" or "r" and the value columns begin with "P" or "p." That scheme is the most limited form because it forces missing values or blanks for almost all other of the meta data. The "qw" scheme assumes that the columns are named according to as.data.frame(object of class qw, expand=TRUE). The "partial" scheme assumes that the basic scheme from as.data.frame(object of class qw, expand=TRUE) is retained, but not all columns are required. Variations include no suffix for the value and .dl in lieu of .rlv for the detection limit. When .dl is used, the method detection limit is assumed to be "MDL" if not supplied. If the scheme is "partial," then at least the value and remark or detetion limit or reporting limit must be supplied.

Value

A data frame of the water-quality data of class "qw" organized by sample. Column names for the water-quality constituents are generated automatically. Other columns are retained unchanged.

References

Lorenz, D.L., 2014, smwrQW OFR.
See information about discrete samples at https://nwis.waterdata.usgs.gov/usa/nwis/qw.

See Also

readNWISqw, makeColNames

Examples

```
# Make a dummy partial dataset
tmp.dum <- data.frame(sta=c("A", "B"),
 Val=c(1, 1.2),
 Val.rmk=c("<", ""),
 Val.dl=c(1, 1), stringsAsFactors=FALSE)
convert2qw(tmp.dum, scheme="partial")
```

---

convertFqw *Water-Quality Data*

Description

Converts information from data of class "qw" to separate columns representing the data in the "qw" columns. This can facilitate transfer to other systems or editing the meta data in a "qw" column. Uses as.data.frame(object of class qw, expand=TRUE) to convert each "qw" column.

Usage

```
convertFqw(data, columns = "")
```

Arguments

data	a data frame that contains limited sample information.
columns	a character vector listing the names of the columns to convert. If a single empty character string, the default, then convert all "qw" columns.

Value

A data frame of the converted water-quality data and all other columns in the original dataset.

Note

This function can be used in conjunction with convert2qw to update or modify reporting levels or inspect the details of the meta data.

References

Lorenz, D.L., 2014, USGSwsQW OFR.
See information about discrete samples at https://nwis.waterdata.usgs.gov/usa/nwis/qw.

See Also

readNWISqw, convert2qw

Examples

```
# Make a dummy partial dataset
tmp.dum <- data.frame(sta=c("A", "B"),
 Val=c(1, 1.2),
 Val.rmk=c("<", ""),
 Val.dl=c(1, 1), stringsAsFactors=FALSE)
tmp.qw <- convert2qw(tmp.dum, scheme="partial")
# Expand the result
convertFqw(tmp.qw)
```

---

dlimit *Impute Detection Limits*

Description

Imputes values for missing detection limits for left-censored data.

Usage

```
dlimit(values, censor.codes, default = 1e-25)
```

Arguments

values	the numeric values.
censor.codes	logical or character, TRUE or "<" indicates left-censored at value and FALSE, " " or "" indicates a quantified value. Character codes "E" and "J" are censored at the current detection limit. Other characters are ignored and generate a warning.
default	the default detection limit value to assign to an uncensored value that is less than the currently imputed detection limit value.

Value

A vector of detection limits matched with each of the input values.

Warning

Tim Cohn, written communication, 5 Nov 2002 states that there are several problems with this approach:

1) for large data sets, it may be difficult for the user to group the observations into subsets as in the example above. If the observations are not grouped into subsets or the subsets do not have the censored observations as the first records, incorrect detection limits may be applied to the uncensored data.

2) for the case of multiple constituents, it may be impossible to place the censored observation within each subset at the top of the subset. Consider constituents Y and Z, each analyzed at Lab B, and each with one censored observation. The censored observations occur on different dates. In this case its not possible to have both observations as the first line in the subset from Lab B. (Note: the user could work around this problem by doing seperate runs for each constituent.)

3) When a subset of observations is completely uncensored, the detection limit from another subset will be applied. Consider the example above, but with the 10 observations from Lab B being completely uncensored. In this case, the detection limit from Lab A will be used for all 20 observations.

4) When all of the observations are uncensored (no '<' signs for a given constituent), there is no way for the user to specify the detection limit(s). In this case, a default value of 1.E-25 is used for all observations (consistent with ESTIMATOR).

Despite these problems, the approach is satisfactory for most applications: "the estimates are not very sensitive to the precise value of the censoring threshold for the above-threshold values."

Note

This function is based on the DLIMIT function in LOADEST (Runkel and others, 2004). The general assumption for imputing detection limits is that the data are sorted by time and the detection limits changes in a consistent pattern over time.

References

Runkel, R.L., Crawford, C.G., and Cohn, T.A., 2004, Load estimator (LOADEST) a FORTRAN program for estiamting constituent laods in streams and rivers: U.S. Gelogical Survey Techniques and Methods 4-A5, 69. Available online as https://pubs.er.usgs.gov/publication/tm4A5.

See Also

readNWQLdl

Examples

```
## The actual detection limits are 2,2,2,1,1,1.
dlimit(c(2,2,3,1,1,2), c(" ", "<", " ", " ", "<", " "))
```

`dotPlot-censored` *Censored Data Methods for Function* `dotPlot`

Description

Produces a dot plot: methods for "mcens" and "qw" data.

Usage

```
## S4 method for signature 'mcens'
dotPlot(x, y, Plot = list(name = "", what = "points", type =
  "solid", width = "standard", symbol = "circle", filled = TRUE, size = 0.09,
  color = "black"), yaxis.orient = "table", yaxis.order = "none",
  yaxis.grid = TRUE, xaxis.log = FALSE, xaxis.range = c(NA, NA),
  ylabels = "full", xlabels = 7, xtitle = deparse(substitute(x)),
  ytitle = "", caption = "", margin = c(NA, NA, NA, NA),
  jitter.y = TRUE, Censored = list(name = "", what = "points", symbol =
  "circle", filled = FALSE, size = 0.09, color = "black", bar = TRUE), ...)

## S4 method for signature 'qw'
dotPlot(x, y, Plot = list(name = "", what = "points", type =
  "solid", width = "standard", symbol = "circle", filled = TRUE, size = 0.09,
  color = "black"), yaxis.orient = "table", yaxis.order = "none",
  yaxis.grid = TRUE, xaxis.log = FALSE, xaxis.range = c(NA, NA),
  ylabels = "full", xlabels = 7, xtitle = deparse(substitute(x)),
  ytitle = "", caption = "", margin = c(NA, NA, NA, NA),
  jitter.y = TRUE, Censored = list(name = "", what = "points", symbol =
  "circle", filled = FALSE, size = 0.09, color = "black", bar = TRUE), ...)
```

Arguments

x	the x-axis data. Missing values are permitted and not plotted.
y	the y-axis data, expected to be be either character or factor. Missing values are permitted and removed before plotting.
Plot	control parameters for uncensored values of the plot, see link{setMultiPlot} and **Details** for details.
yaxis.orient	orientation of the y-axis values, must be either "table" or "grid." "Table" is sorted from top to bottom, "grid" is sorted from bottom to top.
yaxis.order	the order of the y-axis values, must be one of "none," "ascending," or "descending."
yaxis.grid	draw grid lines?
xaxis.log	log-transform the x axis?
xaxis.range	set the range of the x-axis
ylabels	set up y-axis labels.
xlabels	set up x-axis labels.
xtitle	x-axis title.
ytitle	y-axis title.
caption	the figure caption.
margin	the parameters of the margin.
jitter.y	logical, if TRUE, then adjust y values to reduce overlap for each group, or adjust randomly if no groups. If FALSE, then no adjustment is made.
Censored	control parameters for censored values of the plot.
...	arguments for specific methods.

Details

The what component of the Plot argument must be either "points" or "none."

Value

Information about the graph.

Note

A call should be made to setPage to set up the graphics environment before calling dotPlot.

See Also

setPage

`ecdfPlot.lcens` *Empirical Distribution Function*

Description

Produces a graph of the empirical distribution function: methods for "lcens" and "qw" data.

Usage

```
## S3 method for class 'lcens'
ecdfPlot(x, group = NULL, Plot = list(name = "Auto", what =
  "stairstep", type = "solid", width = "standard", symbol = "circle", filled =
  TRUE, size = 0.09, color = "Auto"), xaxis.log = TRUE, xaxis.range = c(NA,
  NA), xlabels = 11, ylabels = 5, ytitle = "Cumulative Probability",
  xtitle = deparse(substitute(x)), caption = "", margin = c(NA, NA, NA,
  NA), ...)

## S3 method for class 'qw'
ecdfPlot(x, group = NULL, Plot = list(name = "Auto", what =
  "stairstep", type = "solid", width = "standard", symbol = "circle", filled =
  TRUE, size = 0.09, color = "Auto"), xaxis.log = TRUE, xaxis.range = c(NA,
  NA), xlabels = 11, ylabels = 5, ytitle = "Cumulative Probability",
  xtitle = deparse(substitute(x)), caption = "", margin = c(NA, NA, NA,
  NA), ...)
```

Arguments

`x`	the data to plot. Missing values are removed before plotting.
`group`	create groups of the left-censored data vector.
`Plot`	control parameters of the uncensored data plot.
`xaxis.log`	log-transform the x axis?
`xaxis.range`	set the range of the x axis.
`xlabels`	set the x-axis labels. See `linearPretty` for details.
`ylabels`	set the y-axis labels. See `linearPretty` for details.
`ytitle`	the y-axis title.
`xtitle`	the x-axis title.
`caption`	the figure caption.
`margin`	the parameters of the margin of the plot area.
`...`	any additional arguments needed by specific methods.

Value

Information about the graph.

See Also

probPlot.lcens

Examples

```
## Not run:
set.seed(1932)
Xu <- rlnorm(32)
setPage("sq") # required page set up
ecdfPlot(as.lcens(Xu, 1.0))

## End(Not run)
```

`extractAIC.censReg` *Extract AIC from a Fitted Model*

Description

Computes the Akaike (1974) An Information Criterion for a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
extractAIC(fit, scale = 0, k = 2, correct = FALSE, ...)
```

Arguments

`fit`	an object of class "censReg"—output from censReg.
`scale`	set scale, ignored in current version.
`k`	the weight of the equivalent degrees of freedom.
`correct`	logical if TRUE and k is set to 2, then return corrected AIC, if FALSE or k is not set to 2,then return the uncorrected value, either AIC or BIC.
`...`	further arguments passed to or from other methods.

Value

A vector of length 2, giving the equivalent degrees of freedom and the AIC for `fit`.

References

Akaike, H., 1974, A new look at the statistical model identification: IEE transactions on Automatic Control, v. 19, no. 6, p. 716–723.

See Also

censReg, extractAIC

`fillIn` *Estimate Censored Values*

Description

Estimates values for censored data.

Usage

```
fillIn(x, method, alpha)

## S3 method for class 'numeric'
fillIn(x, method, alpha)

## S3 method for class 'lcens'
fillIn(x, method = "ROS", alpha = 0.4)

## S3 method for class 'mcens'
fillIn(x, method = "ROS", alpha = 0.4)
```

Arguments

`x`	an object of class "lcens." Missing values are ignored.
`method`	the method to use for estimating censored values: "ROS," "log ROS," "MLE," or "log MLE" are valid for any left- or multiply-censored data ; "triangular," "fill" and "log fill" are valid for left-censored data with a single detection limit; `method` is ignored for uncensored data.
`alpha`	the offset fraction to be used for plotting position; typically in [0,0.5].

Details

The methods of Regression on Order Statistics (ROS) and MLE is explained in Helsel (2012). The "log ROS" first log-transforms the data and back-transforms the estimated values. The triangular method distributes the censored value assuming a triangular distribution between 0 and the single detection limit. Quinn and Keogh (2003) describe alternatives to simple substituion of a single value that sample from an alternate distribution. The triangular distribution is a reasonable distribution when the percentage of censored data is relatively small, say less than 30 percent. The "fill" and "log fill" methods implement the method described by Gleit (1985).

Value

A vector of sorted estimates and actual values with an attribute of the censoring levels.

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.
Quinn, G.P. and Keogh, M.J., 2003, Experimental design and data analysis for biologists, Cambridge University Press, Cambridge, UK, 539 p.

See Also

as.double.qw

Examples

```
set.seed(5420)
XR <- sort(rnorm(10))
XF <- fillIn(as.lcens(XR, -1)) # censors lowest 3 values
#How'd we do?
rbind(XR, XF)
#Note that this is unusual because all 10 random values were less than the mean!
```

---

fitted.censReg *Extract Model Fitted Values*

Description

Extracts the fitted values from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
fitted(object, suppress.na.action = FALSE,
  type = c("response", "link", "mean"), ...)
```

Arguments

object	an object of class "censReg"—output from censReg
suppress.na.action	logical, suppress the effects of the na.action in the call to censReg and return only the fitted values corresponding to the fitted data.
type	the type of fitted value. To get fitted values on the original scale of the response, use "response." For back-transformed log-normal fitted values, use "link." To get the unbiased estimates of the mean for log-normal data, use "mean." Note that "link" and "mean" do not work for the common-log distribution.
...	further arguments passed to or from other methods.

Value

The fitted values from the regression.

See Also

censReg

`format.lcens` *Encode in a Common Format*

Description

Formats data for pretty printing: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
format(x, digits = NULL, ...)

## S3 method for class 'mcens'
format(x, digits = NULL, ...)

## S3 method for class 'qw'
format(x, round = TRUE, units = FALSE, style = "asis", ...)
```

Arguments

`x`	the censored data object to format.
`digits`	how many significant digits are to be used for numbers?
`...`	further arguments passed to or from other methods.
`round`	round the data before formating? Can also be a numeric vector of length 2 containing the maximum number of significant digits and the maximum number of decimal digits.
`units`	logical, if `TRUE` then include the units of the data.
`style`	the style for objects of class "qw." See **Details**.

Details

The argument `style` must be one of "asis," which tries to format the data to best show the values; "simple," which is like "asis" except that interval censored data are the midrange values preceded by "I"; or "interpreted," which shows the values as they would be used in analysis and `round` is ignored. Only the first letter is required.

Value

A character representations of the elements of `x`.

Note

For `style` set to "asis," it is often useful to set `round` to a single value so that the interval data are not rounded to the same value.

Examples

```
format(as.lcens(1:3, 1))
format(as.mcens(1:3, 1:3))
```

`gehanScores` *Gehan Test Scores*

Description

Computes Gehan scores that compare the values in one group to the values in another group.

Usage

```
gehanScores(x, y, na.rm = FALSE)
```

Arguments

`x`	any object that can be converted to class "mcens." Missing values are permitted and result in corresponding missing values in the output.
`y`	any object that can be converted to class "mcens." Missing values are permitted but result in missing values in the output unless `na.rm` is `TRUE`. See **Details**.
`na.rm`	remove missing values from `y` before comparing to `x`?

Details

If `y` is missing, then `x` is compared to itself (with `NA`s removed) and the output is the u-score described in Helsel (2012).

Value

A numeric vector that is the result of the comparison of each value in `x` to all values in `y`. It is the equivalent of the "h" score in described by Helsel, but the signs are reversed so that larger values in `x` have larger values in the output.

Note

If the output is the u-score, `y` is missing, then the values can be converted to ranks by dividing by 2 and adding the mean rank, (length(`x`) + 1)/2.

References

Gehan, E.A., 1965, A generalized Wilcoxon test for comparing arbitraritly singly censored samples: Biometrika, v. 52, p. 203-223.

Helsel, D.R. 2012, Statistics for Censored Environmental Data Using Minitab and R: New York, Wiley, 324 p.

See Also

survdiff, survfit, lcens-class

---

genWilcox.test *Test for Difference in Censored Samples*

Description

This function tests for differences in censored samples from 2 groups. Two methods are available—the Peto-Prentice test is appropriate only for left-censored data. The Gehan test has been extended to multiply censored data as suggested in Gehan (1965) but uses a permutation test to compute the variance.

Usage

```
genWilcox.test(x, y, alternative = "two.sided", method = "best", data.names,
  gehan.seed = 0)
```

Arguments

x	the samples from each group. Forced to the appropriate class. Missing values are removed before the analysis.
y	either another set of samples or a group identifier with exactly two groups. Missing values are removed before the analysis. See **Details**.
alternative	character string describing the alternative hypothesis. Must be one of "two.sided, "greater," or "less."
method	a character string indicating which method to use for the analysis. See **Details**.
data.names	character string to be used to explain the data. Default names are derived from the data arguments.
gehan.seed	an integer value to set the seed to compute the variance of the Gehan statistic. If 0 (the default), then use equation 9.5 from Helsel (2012) to compute the variance.

Details

If y is either type character or factor, then it is assumed to be a group identifier. Anything else is treated as another set of sample and forced to the appropriate class of censored data.

The argument method must be one of "peto" or "gehan." It may also be "best," which selects "peto" for uncensored or left-censored data and "gehan" otherwise.

Value

An object of class "htest" that inherits "genWilcox."

Null Hypothesis

The null hypothesis is that the distributions are not different from one another.

Note

The `genWilcox.test` function uses the `survfit` function. Helsel (2012) describes flipping the left-censored data so that small values become large and left-censored values become right-censored values and adapt nonparametric techniques from survival analysis. The results from `survfit` are printed as the sample estimates when printing the output, the important columns are records, the number in each group; events, the number of uncensored values; and median, the group median. A `plot` method is supported for the returned object.

References

Gehan, E.A., 1965, A generalized Wilcoxon test for comparing arbitrarily singly censored samples: Biometrika, v. 52, p. 203-223.

Harrington, D.P., and Fleming, T.R., 1982, A class of rank test procedures for censored survival data: Biometrika, v. 69, p. 553-566.

Helsel, D.R. 2012, Statistics for Censored Environmental Data Using Minitab and R: New York, Wiley, 324 p.

Peto, R., and Peto, J., 1972, Asymptotically efficient rank invariant test procedures (with discussion): Journal of the Royal Statistical Society, Series A v. 135, p. 185-206.

Prentice, R.L. 1978, Linear rank tests with right-censored data: Biometika, v 65, p 167-179.

Prentice, R.L., and Marke, P., 1979, A qualitative discrepancy between censored data rank tests: Biometrika, v. 35, p. 861-867.

See Also

`survdiff`, `survfit`, `lcens-class`

Examples

```
# Compare uncensored results
# First for grouped data
set.seed(69)
Xu <- rlnorm(22, 0, 1)
Yu <- rlnorm(22, .6, 1)
genWilcox.test(Xu, Yu)
wilcox.test(Xu, Yu)
# Compare effect of censoring
genWilcox.test(as.lcens(Xu, 1), Yu)
genWilcox.test(as.lcens(Xu, 1), as.lcens(Yu, 1))
```

`importNWISqw` *Water-Quality Data*

Description

Imports discrete water-quality sample data from NWISweb.

Usage

```
importNWISqw(sites, params = "All", begin.date = "", end.date = "",
  keep = NULL, use.pnames = FALSE)
```

Arguments

`sites`	a vector of the USGS station identifiers.
`params`	A character string contains the name of a group of parameter codes, or a vector of parameter codes. See **Details**.
`begin.date`	the earliest date for data, must be a character with the format "YYYY-mm-dd."
`end.date`	the latest date for data, must be a character with the format "YYYY-mm-dd."
`keep`	a character vector for any additional columns to retain from the retrieved data.
`use.pnames`	create columnn names based on pcode (like P00000)? Default is to generate more English-like names.

Details

Valid parameter code groups are "All," or group codes:

Code	Description
INF	Information
PHY	Physical
INM	Inorganics, Major, Metals (major cations)
INN	Inorganics, Major, Non-metals (major anions)
NUT	Nutrient
MBI	Microbiological
BIO	Biological
IMN	Inorganics, Minor, Non-metals
IMM	Inorganics, Minor, Metals
TOX	Toxicity
OPE	Organics, pesticide
OPC	Organics, PCBs
OOT	Organics, other
RAD	Radiochemical
SED	Sediment

POP Population/community

Value

A data frame of the water-quality data of class "qw" organized by sample. Column names for the water-quality constituents are generated automatically, but can be set by the user, see `makeColNames`.

Note

The sample information columns that are automatically retained are the USGS station identifier, sample date, sample time, time zone code, and medium code. If composite samples are found in the data, then sample end date and sample end time are also included. Use the `keep` argument to retain more if needed.

NWIS has several remark codes that have special meaning in addition to "<," which indicates a left-censored value and ">," which indicates a right censored value. Most other codes are ignored in processing, but passed through and retained as the remark code in the data; a warning is printed when special remark codes are retained. NWIS also uses the remark code "M" to indicate whan a result value is rounded to 0. In that case, `importNWISqw` tries to recode the data as less than the reportiong level, but is the reporting level is the missing value `NA`, the remark code "M" is retained. Warnings are generated for both conversion and retention of the "M" remark codes.

References

Lorenz, D.L., 2014, smwrQW OFR.
See information about discrete samples at `https://nwis.waterdata.usgs.gov/usa/nwis/qw`.

See Also

`readNWISqw`

Examples

```
## Not run:
importNWISqw("05330000", "00608") # Ammonia samples from the Minnesota River at Jordan.
Empty <- importNWISqw('05330000',c('00400','00403'),begin.date='2008-01-01')

## End(Not run)
```

`importQW` *Water-Quality Data*

Description

Converts a dataset containing stacked discrete water-quality sample data to a dataset representing those data as class "qw."

Usage

```
importQW(data, keep = c("STAID", "DATES", "TIMES", "MEDIM"),
  values = "VALUE", remark.codes = "REMRK", value.codes = "NWIS",
  reporting.level = "RPLEV", reporting.method = "RLTYP",
  reporting.units = "UNITS", analyte.method = "METHD",
  analyte.name = "PSNAM", unique.code = "PCODE", ColNames = "Auto")
```

Arguments

data	the dataset.
keep	the names of the columns that represent a single sample and any other common information.
values	the name of the column containing the reported numeric values.
remark.codes	he name of the column containing the remark codes.
value.codes	the name of the column containing any value modifiers, or the character string to use. See **Details**.
reporting.level	the name of the column containing the reporting level, or the numeric value to use. See **Details**.
reporting.method	the name of the column containing the type of the reporting level, or the character string to use. See **Details**.
reporting.units	the name of the column containing the measurement units, or the character string to use. See **Details**.
analyte.method	the name of the column containing the analytic method, or the character string to use. See **Details**.
analyte.name	the name of the column containing the name of the analyte, or the character string to use. See **Details**.
unique.code	the name of the column containing any unique codes, or the character string to use. See **Details**.
ColNames	the name of the column containing the column name to use to make column names, the character string to use if only a single analyte, or "Auto" if the data in unique.code are USGS parameter codes. See **Details**.

Details

Only values and remark.codes are required. All others can be interpreted as constant values if the column name is not in data. For automatic generation of column names, see makeColNames.
For reporting.level, it is better to use NA than to use an arbitrary small value because the functions to convert to objects for analysis will create reasonable reporting level values if they are not known.
For other columns when the actual value is not known, the actual value is less important for analysis and more important for the user, so arbitrary values can be used.

Value

A data frame containing the columns in keep and those generated by converting the other columns into class "qw."

Note

Need some notes

References

Lorenz, D.L., 2014, USGSqw OFR.
See information about discrete samples at https://nwis.waterdata.usgs.gov/usa/nwis/qw.

See Also

readNWISqw, importNWISqw, makeColNames

Examples

```
## Not run:
# Convert the stacked qw data supplied in smwrData
library(smwrData)
data(QWstacked)
QWstacked$result_va <- as.numeric(QWstacked$result_va) # raw data are character
# The units are both mg/l, no analyte name, generate column names from parameter codes
head(importQW(QWstacked, c("site_no", "sample_dt", "sample_tm", "medium_cd"),
 "result_va", "remark_cd", "val_qual_tx", "rpt_lev_va", "rpt_lev_cd",
 "mg/l", "meth_cd", "", "parm_cd", "parm_cd"))
# A simple example having the minimum information
data(QW05078470)
importQW(QW05078470, c("DATES", "TIMES"), "P00665", "R00665", "", 0.005, "User",
"mg/l", "Unk", "Dissolved Phosporus", "00665", "DissP")

## End(Not run)
```

---

imputeLessThans *Impute Censored Values*

Description

Imputes substitute values for left-censored values using the expectation maximization method.

Usage

```
imputeLessThans(..., type = c("MLE", "robust"), initial = c("complete.obs",
  "multRepl"))

## Default S3 method:
```

```
imputeLessThans(..., type = c("MLE", "robust"),
  initial = c("complete.obs", "multRepl"))

## S3 method for class 'data.frame'
imputeLessThans(..., group = NULL, type = c("MLE",
  "robust"), initial = c("complete.obs", "multRepl"))
```

Arguments

...	either a data frame that contains columns of class "qw" or any combination of individual vectors of class "numeric," "lcens," or "qw." Missing values are removed before processing.
type	the type of estimate, "MLE" for maximum likelihood estimates, or "robust" for robust estimation methods. See lrEM for details.
initial	the method to use for the initial log-ratio covariance matrix, either "complete.obs" that uses only the rows with no censored data to construct the matrix, or "multRepl" that uses simple substitution of censored values to compute the matrix. See lrEM for details.
group	character string, the name of the column in the data frame to indicate a group for imputation. See **Details**

Details

Imputation of left-censored data requires the assumption of multivariate log-normality for a single population. If the data represent samples from multiple populations, then they should be identified by the group argument. The minimum size for any group is 3.

Value

A data frame containing the original data with imputed censored values.

See Also

lrEM, mImputeLessThans

---

is.na.lcens *Missing Values*

Description

Indicates which elements are missing: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
is.na(x)

## S3 method for class 'mcens'
is.na(x)

## S3 method for class 'qw'
is.na(x)
```

Arguments

x	the object to be tested.

Value

A logical vector of the same length as its argument x, containing TRUE for those elements marked NA and FALSE otherwise.

Examples

```
is.na(as.lcens(c(1,3, NA), 2))
```

---

kendallATS.test *Kendall' tau*

Description

Computes Kendall's tau for left-censored data and the Akritas and others (1995) Theil-Sen slope estimator.

Usage

```
kendallATS.test(x, y, na.rm = TRUE)
```

Arguments

x	any data that can be converted to a left-censored data object.
y	any data that can be converted to a left-censored data object.
na.rm	remove missing values before performing the test?

Value

an object of class "htest" having these components:

`statistic`	the value of the taub statistic.
`p.value`	the attained p-value for the test.
`data.name`	a character string describing the name of the data used in the test.
`method`	a description of the method.
`alternative`	a description alternate hypotheses, always "two-sided."
`null.value`	the value of taub for the null hypothesis.
`estimate`	the ATS slope, and medians of `x` and `y`.
`coef`	the coefficients of the monotic fit line. See **Note**.
`slope.se`	an estimate of the standard error of the slope.

Note

The intercept is computed as slope * -median.x + median.y. The coef vector is designed to graphically represent the relation between x and y. It is not intended for prediction proposes.

References

Akritas, M.G., Murphy, S.A., and LaValley, M.P., 1995, The Theil-Sen estimator with doubly-censored data and applications to astronomy: Journal of the American Statistical Association, v. 90, p. 170–177.
Helsel, D.R. 2005, Nondetects and data analysis: New York, Wiley, 266 p.

Examples

```
# Simple no censoring
set.seed(450)
tmp.X <- rnorm(25)
tmp.Y <- tmp.X/2 + rnorm(25)
cor.test(tmp.X, tmp.Y, method="k", exact=FALSE, continuity=TRUE)
kendallATS.test(tmp.X, tmp.Y)
# Some censoring
kendallATS.test(as.lcens(tmp.X, -1), as.lcens(tmp.Y, -1))
```

---

lcens-class *Left-censored Data*

Description

Class "lcens" describes left-censored data.

Slots

`.Data` a 2-column matrix that contains the value and the detection limit of the left-censored data. Missing values are permitted.

`censor.codes` a logcial value, `TRUE` indicates a censored value, `FALSE` indates an uncensored value. Missing values are permitted only for missing values in `.Data`.

`names` a character vector of unique names for each element in `.Data`.

Objects from the Class

Objects can be created by calls to `as.lcens`.

References

Lorenz, D.L., in preparation

Examples

```
showClass("lcens")
```

`length.lcens` *Length of an Object*

Description

Gets the length of a vector: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
length(x)

## S3 method for class 'mcens'
length(x)

## S3 method for class 'qw'
length(x)

## S3 method for class 'mcens'
length(x)

## S3 method for class 'qw'
length(x)
```

Arguments

`x`	a censored-data or water-quality object.

Value

An integer of length 1 indicating the number of elements in x.

Examples

```
length(as.lcens(c(1,3, NA), 2))
```

---

logLik.censReg *Extract Log-Likelihood*

Description

Extracts the log-likelihood statistic from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
logLik(object, ...)
```

Arguments

object	the output from censReg.
...	further arguments passed to or from other methods.

Value

An object of class "logLik" containing the log-likelihood and the attributes "df" (degrees of freedom) and "nobs" (number of observations).

See Also

censReg,

makeColNames *Column Names*

Description

Creates valid, unique column names for water-quality constituents.

Usage

```
makeColNames(params, name, short)
```

Arguments

params	the parameter codes.
name	the long-name of the parameter code.
short	the short name of the parameter code.

Details

The arguments name and short are expected to be columns retreived by those arguments in the function pcodeNWISqw.

Value

A data frame containing the columns parm_cd and col_name.

Note

This function is a support function designed to be called within other functions.

See Also

pcodeNWISqw, pcodeColData

Math-censored *Mathematical Transform Methods for* lcens *objects*

Description

Some limited mathematical methods are possible and well-defined for censored or water-quality data.

Usage

```
## S4 method for signature 'lcens'
Math(x)

## S4 method for signature 'mcens'
Math(x)

## S4 method for signature 'qw'
Math(x)
```

Arguments

x	the object to transform

Examples

```
log(as.lcens(c(1, 3), 2))
```

---

mcenKM *Estimate Statistics*

Description

Support function for computing statistics for multiply-censored data.

Usage

```
mcenKM(x, group)
```

Arguments

x	an object of "mcens" to compute
group	the group variable

Value

An object of class "survfit."

mcenMLE *Estimate Statistics*

Description

Support function for computing statistics for multiply-censored data extended from the maximum likelihood method (Helsel and Cohn, 1988).

Usage

```
mcenMLE(x, method = "MLE", alpha = 0.4)
```

Arguments

x	the data to estimate, Missing values permitted and ignored. Must be an object of class "mcens," a numeric vector, or the output from censpp.
method	the method to use, either "MLE" or "log MLE."
alpha	the offset for plotting position, used to compute the filled in values.

Value

A list containing the mean and standard deviation, filled in values for the censored values, and the censored levels. If method is "log MLE," then the list also contains the mean and standard deviation of the natural log-transformed values computed by maximum likelihood.

References

Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp.1997-2004

mcenROS *Estimate Statistics*

Description

Support function for computing statistics for multiply-censored data extended from the "fill-in" probability plotting procedure method (Helsel and Cohn, 1988), now known as regression on order statistics.

Usage

```
mcenROS(x, method = "ROS", alpha = 0.4)
```

Arguments

x	the data to estimate, Missing values permitted and ignored. Must be an object of class "mcens," a numeric vector, or the output from censpp.
method	the method to use, either "ROS" or "log ROS."
alpha	the offset for plotting postion.

Value

A list containing the mean and standard deviation, filled in values for the censored values, and the censored levels. If method is "log ROS," then the list also contains the mean and standard deviation of the natural log-transformed values computed by regression on order statistics.

References

Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp.1997-2004

---

mcens-class *Mutliply-censored Data*

Description

Class "mcens" describes right- or multiply-censored data.

Slots

.Data a 2-column matrix that contains the lower and upper value for the data. Missing values are permitted.

censor.codes an integer vector that indicates censoring; -1 indicates left-censored, 0 indicates uncensored, and 1 indicates rigth-censored. Missing values are permitted only for elements of .Data that are missing.

interval a logical value; TRUE indicates that the value is interval censored and FALSE indicates otherwise. Missing values are permitted only for elements of .Data that are missing.

names a character vector of unique names for each element in .Data.

Objects from the Class

Objects can be created by calls to as.mcens.

Note

Note that censor.codes and interval are not necessary, but included to make some manipulations easier.

References

Lorenz, D.L., in preparation

Examples

```
showClass("mcens")
```

`mdlAMLE` *Estimate Statistics*

Description

Support function for computing statistics for left-censored data using the adjusted maximum-likelihood method (Cohn, 1988)

Usage

```
mdlAMLE(x, method = "AMLE", alpha = 0.4)
```

Arguments

`x`	the data to estimate, Missing values permitted and ignored. Must be an object of class "lcens," a numeric vector, or the output from censpp.
`method`	the method to use, either "AMLE" or "log AMLE."
`alpha`	the offset for plotting postion, used to compute the filled in values.

Value

A list containing the mean and standard deviation, filled in values for the censored values, and the maximum censor level. If `method` is "log AMLE," then the list also contains the mean and standard deviation of the natural log-transformed values computed by maximum likelihood.

References

Cohn, T.A., 1988, Adjusted maximum likelihood estimation of the moments of lognormal populations from type 1 censored samples: U.S. Geological Survey Open-File Report 88-350, 34 p.

`mdlKM` *Estimate Statistics*

Description

Support function for computing statistics for left-censored data.

Usage

```
mdlKM(x, group, conf.int = 0.95)
```

Arguments

x	an object of "lcens" to compute.
group	the group variable.
conf.int	the confidence interval .

Value

An object of class "survfit."

---

mdlKMstats *Estimate Statistics*

Description

Support function for computing statistics for left- and multiply censored data.

Usage

```
mdlKMstats(x, probs = c(0.25, 0.5, 0.75))
```

Arguments

x	out from mdlKM or mcenKM
probs	numeric vector of desrired probabilities

Value

A list containing summary statistics.

---

mdlMLE *Estimate Statistics*

Description

Support function for computing statistics for left-censored data using the maximum likelihood method (Helsel and Cohn, 1988).

Usage

```
mdlMLE(x, method = "MLE", alpha = 0.4)
```

Arguments

x	the data to estimate, Missing values permitted and ignored. Must be an object of class "lcens," a numeric vector, or the output from censpp.
method	the method to use, either "MLE" or "log MLE."
alpha	the offset for plotting position, used to compute the filled in values.

Value

A list containing the mean and standard deviation, filled in values for the censored values, and the censored levels. If method is "log MLE," then the list also contains the mean and standard deviation of the natural log-transformed values computed by maximum likelihood.

References

Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, p.1997–2004

---

mdlROS *Estimate Statistics*

Description

Support function for computing statistics for left-censored data using the "fill-in" probability plotting procedure method (Helsel and Cohn, 1988), now known as regression on order statistics. This method is also known as the "robust" approach (Helsel, 2012).

Usage

```
mdlROS(x, method = "ROS", alpha = 0.4)
```

Arguments

x	the data to estimate, Missing values permitted and ignored. Must be an object of class "lcens," a numeric vector, or the output from censpp.
method	the method to use, either "ROS" or "log ROS."
alpha	the offset for plotting postion.

Value

A list containing the mean and standard deviation, filled in values for the censored values, and the censored levels. If method is "log ROS," then the list also contains the mean and standard deviation of the natural log-transformed values computed by regression on order statistics.

References

Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp.1997–2004

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

`mean.lcens` *Arithmetic Mean*

Description

Computes the arithemtic mean. These functions return `NA` and are designed to stop the user from accidentally using the function `mean` on censored data.

Usage

```
## S3 method for class 'lcens'
mean(x, ...)

## S3 method for class 'mcens'
mean(x, ...)
```

Arguments

`x`	the censored data object.
`...`	not used, require for other methods.

Value

An error message.

Note

The mean and standard deviation of censored data should be computed using `censStats`.

See Also

`censStats`

`mean.qw` *Arithmetic Mean*

Description

Computes the arithmetic mean of water-quality data. This function is intended primarily to compute a useful mean of a small set of data, for example to compute the mean of daily samples. See **Note**.

Usage

```
## S3 method for class 'qw'
mean(x, lt.tol = 0.5, ...)
```

Arguments

`x`	the water-quality data object.
`lt.tol`	an arbitrary tolerance metric to drop interval-censoring. See **Details**.
`...`	further arguments passed to or from other methods.

Details

The `lt.tol` argument can be used to collapse interval censored data, creating an uncensored value, when the range of interval censored data is less than `lt.tol` time the detection limit. This generally results in reasonable estimates of values for the intended purpose.

Value

The mean of thw water-quality data as class "qw."

Note

The mean and standard deviation of censored water-quality data should be computed using `censStats`.

See Also

`censStats`

`median.lcens` *Median Value*

Description

Computes the sample median: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
median(x, na.rm = FALSE, ...)

## S3 method for class 'mcens'
median(x, na.rm = FALSE, ...)

## S3 method for class 'qw'
median(x, na.rm = FALSE, ...)
```

Arguments

`x`	the censored data object.
`na.rm`	remove missing values before computation?
`...`	potentially further arguments for methods; not used in the default or lcens method.

Value

A vector of length one representing the sample median.

Note

The median is computed using the flipped Kaplan-Meier method described in Helsel (2012).

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.

See Also

`censQuantile`

Examples

```
median(as.lcens(1:4, 2))
```

`mImputeLessThans` *Impute Censored Values*

Description

Imputes substitute values left-censored values multiple times using the data augmentation method.

Usage

```
mImputeLessThans(..., m = 10, initial = c("lrEM", "complete.obs",
  "multRepl"))

## Default S3 method:
mImputeLessThans(..., m = 10, initial = c("lrEM",
  "complete.obs", "multRepl"))

## S3 method for class 'data.frame'
mImputeLessThans(..., group = NULL, m = 10,
  initial = c("lrEM", "complete.obs", "multRepl"))
```

Arguments

`...`	either a data frame that contains columns of class "qw" or any combination of individual vectors of class "numeric," "lcens," or "qw." Missing values are removed before processing.
`m`	the number of multiple imputations desired.
`initial`	the method to use for the initial log-ratio covariance matrix, either "lrEM" that uses expectation maximization to construct the matrix, "complete.obs" that uses only the rows with no censored data to construct the matrix, or "multRepl" that uses simple substitution of censored values to compute the matrix. See `lrDA` for details.
`group`	character string, the name of the column in the data frame to indicate a group for imputation. See **Details**

Details

Imputation of left-censored data requires the assumption of multivariate log-normality for a single population. If the data represent samples from multiple populations, then they should be identified by the `group` argument. The minimum size for any group is 3.

Value

A list of length `m` containing data frames with imputed censored values. If `group` is used, then the returned object is a tagged list with names corresponding to the values in the grouping column. Each group contains the `m` repititions of imputed values.

See Also

lrDA, imputeLessThans

---

multiply *Multiply Water-Quality Data by a Number*

Description

Multiplies water-quality data by a number while maintaining the integrity of the data.

Usage

```
multiply(x, factor, units, analyte, analyte.method = "ALGOR", pcode = "")
```

Arguments

x	an object of water-quality (class "qw"). Missing value are permitted and result in corresponding missing values in the output.
factor	a single numeric value
units	character reporting units of the resulting operation, if not supplied, retained from x.
analyte	character the name of the resulting analyte.
analyte.method	character the name of the analytical method for the result, by default "ALGOR."
pcode	the parameter code of the result, if not supplied, retained from x.

Details

Multiplication is not defined for objects of class "qw" because it can change some of the metadata, in particular the units, the analytical method, or the unique identifier (pcode).

Value

An object of class "qw" that is the result of the requested operation.

nobs.censReg *Extract the Number of Observations from a Fit*

Description

Extracts the number of observations from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
nobs(object, ...)
```

Arguments

object	an object of class "censReg"—output from censReg.
...	further arguments passed to or from other methods.

Value

The number of observations used to construct the model.

See Also

censReg

pairedPratt.test *Test for difference in left-censored samples*

Description

Tests for differences in paired left-censored samples from two groups using Pratt's (1959) adjustment for ties.

Usage

```
pairedPratt.test(x, y, alternative = "two.sided", mu = 0, data.names)
```

Arguments

x	the paired samples to y. Forced to class "lcens." Missing values are removed before the analysis.
y	the paired values to x. Missing values are removed before the analysis.
alternative	character string describing the alternative hypothesis for (x - y compared to mu. Must be one of "two.sided," "greater," or "less."
mu	a numeric value specifying the test difference between x and y.
data.names	character string to be used to explain the data. Default names are derived from the data arguments.

Value

An object of class "htest" that inherits "ppw."

Null Hypothesis

The null hypothesis is that the distributions are not different from one another.

Note

The Pratt (1959) adjustment to the Wilcoxon signed-rank test was used by Lindsey and Rupert (2012) to evaluate decadal changes in groundwater quality. Lindsey and Rupert (2012) used a fixed reporting level (0.06, the largest among all of the data) and used a simple substitute value 0.0599 for the censored values. The approach in pairedPratt.test is to recensor values at the largest reporting limit, use one-half that value as the simple substitute and round differences greater than the simple substitute value to multiples of the largest reporting limit; nonzero differences less than or equal to the simple substitute value are rounded to that value. That rounding scheme eliminates the ambiguity of comparing interval values created by the difference between a censored value and an uncensored value.

A plot method is supported for the returned object.

References

Lindsey, B.D., and Rupert, M.G., 2012, Methods for evaluating temporal groundwater quality data and results of decadal-scale changes in chloride, dissolved solids, and nitrate concentrations in groundwater in the United States, 1988–2010: U.S. Geological Survey Scientific Investigations Report 2012–5049, 46 p.

Pratt, J.W., 1959, Remarks on zeros and ties in the Wilcoxon signed rank procedures: Journal of the American Statistical Association, v. 54, no. 287 p. 655–667.

See Also

ppw.test, lcens-class

Examples

```
# Compare uncensored results
set.seed(699)
Xu <- sort(rlnorm(22, 0, 1))
Yu <- sort(rlnorm(22, .425, 1))
# Treat as paired samples
pairedPratt.test(Xu, Yu)
wilcox.test(Xu, Yu, paired=TRUE)
# The differences in the p-values are due to use of approximate p-value for
# the Pratt test
```

---

pcodeColData *Selected USGS parameter Codes*

Description

Selected U.S. Gelogical Survey (USGS) parameter codes to create column names and data type conversion when importing data from NWISweb

Usage

```
pcodeColData
```

Format

Data frame with 1178 rows and 3 columns

Name	Type	Description
parm_cd	character	The 5-digit parameter code
col_name	character	The column name to use when importing data from NWISweb
data_type	character	The data type to convert the imported data

Note

This data set can be copied and modifed by the user to add or change column names. The `data_type` must be either "qw" or "numeric" any other value will be ignored.

References

Lorenz, D.L., 2016, smwrQW—An R Package for Managing and Analyzing Water-Quality Data, Version 0.7.3

Examples

```
data(pcodeColData)
# just print some rows
head(pcodeColData)
tail(pcodeColData)
```

pcodeNWISqw *Parameter Code Information*

Description

Returns information about parameter codes. Support function for `importNWISqw`.

Usage

```
pcodeNWISqw(params = "all", group = TRUE, name = TRUE, CASRN = FALSE,
  short = TRUE, units = TRUE, col.name = FALSE)
```

Arguments

`params`	A character string contains the name of a group of parameter codes, or a vector of pamater codes. See **Details**.
`group`	include the parameter group in the output?
`name`	include the parameter name in the output?
`CASRN`	include the parameter Chemical Abstracts Service Registry Number (CASRN) in the output?
`short`	include the parameter short name in the output?
`units`	include the parameter units in the output?
`col.name`	include a column name for the parameter in the output?

Details

Valid groups for `param` are "All," "INF," "PHY," "INM," "INN," "NUT," "MBI," "BIO," "IMN," "IMM," "TOX," "OPE," "OPC," "OOT," "RAD," "SED," or "POP."

Value

A data frame with the parameter codes and the selected columns.

References

Lorenz, D.L., 2014, smwrQW OFR.
See information about discrete samples at https://nwis.waterdata.usgs.gov/usa/nwis/qw.

See Also

`readNWISpCode`

Examples

```
## Not run:
pcodeNWISqw("00925")

## End(Not run)
```

---

pctCens *Percent Censored*

Description

Computes the percentage of censoring ("left," "right," or "multiple") for an object.

Usage

```
pctCens(x, type = "multiple")
```

Arguments

x	the object to compute the percentage of censoring. Missing values are permitted and removed before computing the percentage,
type	the type of censoring desired. For "left," only the percent of left-censored data is computed. For "right," only the percent of right-censored data is computed. For "multiple," the percentage of left-, right-, and interval-censored data is returned.

Value

The percentage of the kind of censoring specified by type is returned. Zero is returned if all data are missing!

Note

This function will work for any type of data that has a method for censoring and as.mcens.

See Also

censoring

Examples

```
pctCens(as.lcens(c(1:5, NA), 2)) # left-censored
```

`percentile.lcens` *Empirical Cumulative Percent*

Description

Computes the empirical cumulative percent or percent exceedance of observed data for specific values: methods for "lcens" and "qw" data.

Usage

```
## S3 method for class 'lcens'
percentile(x, q, test = ">=", na.rm = TRUE,
  percent = TRUE, exact = TRUE, ...)

## S3 method for class 'qw'
percentile(x, q, test = ">=", na.rm = TRUE, percent = TRUE,
  exact = TRUE, ...)
```

Arguments

`x`	a vector of class "lcens" or "qw" representing the observed data.
`q`	a vector of numeric quantiles for which the cumulative percent or percent exceedence is desired.
`test`	a character string indicating the test. The default value, '>=,' is the percent equalling or exceeding the quantile and '<' would return the cumulative percent below the quantile.
`na.rm`	a logical value indication whether missing values (NAs) should be removed or not. If na.rm is `FALSE` and there are missing values in `x`, then the result will be NA. The default value is `TRUE`.
`percent`	a logical value indicating whether the result should be expressed as a percent or proportion. The default value, `TRUE`, will express the result as a percent.
`exact`	logical, if `TRUE`, then force strict comparisons. Any other value is ignored in the current version.
`...`	not used, required for method function

Value

A named vector as long as `q` corresponding to the requested value.

Note

The stats package contains the `ecdf` function that performs a similar function when `test` is "<=."

See Also

`ecdf`

Examples

```
set.seed(2342)
Xr <- rlnorm(24)
# The percentage of the observarions greater than or equal to 2
percentile(as.lcens(Xr, 2), 1:5)
# Compare to the raw data:
percentile(Xr, 1:5)
```

`plot.censReg` *Diagnostic Plot*

Description

Plot simple AMLE/MLE model diagnostics.

Usage

```
## S3 method for class 'censReg'
plot(x, which = "All", set.up = TRUE, ...)
```

Arguments

`x`	an object of class "censReg"—output from `censReg`
`which`	either "All," 1, or 2 indicating which plot, see **Details**.
`set.up`	set up the graphics page?
`...`	further arguments passed to or from other methods.

Details

Two graphs can be produced by this function. Number 1 is a plot of fitted versus actual and number 2 is a Q–normal plot. More complete diagnostic plots are available from the `summary.censReg` output.

Value

The object `x` is returned invisibly.

See Also

`censReg`

plot.ppw *Plot*

Description

Create diagnostic plots for selected hypothesis tests.

Usage

```
## S3 method for class 'ppw'
plot(x, which = "All", set.up = TRUE, labels, ...)

## S3 method for class 'genWilcox'
plot(x, which = "All", set.up = TRUE,
  xaxis.log = TRUE, ...)
```

Arguments

x	an object having classes "htest" and some other class for which a plot method exists.
which	either "All" or an number indicating which plot, see **Details**.
set.up	set up the graphics page?
labels	use supplied values for y-axis labels instead of sequence number.
...	not used, arguments passed to or from other methods.
xaxis.log	draw the x-axis on a logarithimic scale?

Details

The ppw method for plot creates 2 graphs on a single page. The first graph is the scaled differences from the PPW test with the mean difference shown by a blue line. The second graph is the actual differences in the paired values, assuming that the true minimum for left-censored data is 0. The median difference is shown by a blue line.

The genWilcox method creates a single graph that shows the emprical cumulative distributions for each group.

Value

The object x is returned invisibly.

plot.qw *Plot Water-Quality Data*

Description

Create a diagnostics plot of the reporting levels and observations

Usage

```
## S3 method for class 'qw'
plot(x, which = "All", set.up = TRUE, ...)
```

Arguments

x	an object of class "qw."
which	which plot, ignored in current version.
set.up	set up the graphics page?
...	additional arguments passed to xyPlot.

Value

A plot is created and the data are invisibly returned.

See Also

timePlot-censored

Examples

```
## see vignettes
```

plot.summary.censReg *Diagnostic Plot*

Description

Plot summary AMLE/MLE model diagnostics.

Usage

```
## S3 method for class 'summary.censReg'
plot(x, which = "All", set.up = TRUE, span = 1,
  ...)
```

Arguments

x	an object of class "censReg"—output from censReg
which	either "All" or any of a sequence from 1 to 7 indicating which plot, see **Details**.
set.up	set up the graphics page?
span	the span to use for the loess smooth. Set to 0 to suppress.
...	further arguments passed to or from other methods.

Details

Seven graphs can be produced by this function. If which is "All," then all plots are produced. The argument which can also be the name of an explanatory variable so that a partial residual plot is created for a single variable. Or which can be any of a sequence of numbers from 1 thorugh 7. Numeric values for which:

1. Fitted vs. Observed
2. Fitted vs. Residual
3. S-L plot
4. A correlogram if dates are available in the model or in the data set
5. Q-normal plot
6. Influence plot
7. Partial residual plots for each explanatory variable

Value

The object x is returned invisibly.

Note

The q-normal plot (number 5) plots the working residuals and portrays fits for relatively small percentages of censoring very well. For larger percentages of censoring, the q-normal plot in plot.censReg (number 2) can portray the normality of the residuals more accurately.

See Also

censReg, plot.censReg

pow *Power Transforms*

Description

Raises censored data to a positive power.

Usage

```
pow(x, lambda, out = "Auto")
```

Arguments

x	any object that can be converted to class "mcens" or "lcens." Must be non negative (zero data are permitted). Missing values are permitted and missing vlues in the returnned vector.
lambda	the power to raise x. Must be strictly positive.
out	the output format, must be one of "numeric," "lcens," "mcens," or "Auto." If "Auto," then if x is class "numeric," then the output format is "numeric" otherwise the output format is "mcens."

Value

An object of class "mcens" that is the result of the requested operations. Each of the range of the lower and upper values are raised to the power of lambda and the result divided by lambda to approximately maintain scale. But the reciprocal value of lambda cannot be used to restore the original values.

Examples

```
pow(as.mcens(c(0,2), c(1,2)), 2)
pow(as.mcens(c(0,2), c(1,2)), .5)
# Numeric values
pow(c(1,3), 2)
```

ppw.test *Test for difference in left-censored samples*

Description

Tests for differences in paired left-censored samples from two groups using the PPW test (O'Brien and Fleming, 1987).

Usage

```
ppw.test(x, y, alternative = "two.sided", OBrienFleming = TRUE, data.names)
```

Arguments

x	the paired samples to y. Forced to class "lcens." Missing values are removed before the analysis.
y	the paired values to x. Missing values are removed before the analysis.
alternative	character string describing the alternative hypothesis. Must be one of "two.sided," "greater," or "less."
OBrienFleming	logical, if TRUE, then apply the OBrienFleming otherwise it is not applied. See **Details**.
data.names	character string to be used to explain the data. Default names are derived from the data arguments.

Details

The O'Brien-Fleming adjustment forces the score for all paired censored values to be equal to each other. It also forces equality between observed values that are less than the detection limit in the paried sample by setting the value to less than the detection limit.

Value

An object of class "htest" that inherits "ppw."

Null Hypothesis

The null hypothesis is that the distributions are not different from one another.

Note

The ppw.test function uses the survfit function. Helsel (2012) describes flipping the left-censored data so that small values become large and left-censored values become right-censored values and adapt nonparametric techniques from survival analysis.
A plot method is supported for the returned object.

References

Helsel, D.R. 2012, Statistics for Censored Environmental Data Using Minitab and R: New York, Wiley, 324 p.

O'Brien, P.C. and Fleming, T.R., 1987, A paired Prentice-Wilcoxon test for censored paried data: Biometrics, v. 43, p. 451–455.

See Also

survdiff, survfit, lcens-class

Examples

```
# Compare uncensored results
set.seed(699)
Xu <- sort(rlnorm(22, 0, 1))
Yu <- sort(rlnorm(22, .425, 1))
# Treat as paired samples
ppw.test(Xu, Yu)
wilcox.test(Xu, Yu, paired=TRUE)
```

`predict.censReg` *Predict Values*

Description

Predicts values from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
predict(object, newdata, type = c("response", "link",
  "mean"), se.fit = c("none", "estimate", "prediction"),
  interval = c("none", "confidence", "prediction"), level = 0.95,
  na.action = na.pass, ...)
```

Arguments

`object`	the output from `censReg`.
`newdata`	an optional data frame in for predictions. If omitted, the observations from `object` are used.
`type`	the type of predicted value. To get predictions on the original scale of the response, use "response." To get predictions on the transformed scale of the response, use "link." To get the mean predictions for log-normal data, use "mean."
`se.fit`	the type of standard errors of the predicted value to include in the result.
`interval`	type type of interval calculation. See `Details`.
`level`	the confidence or prediction interval.
`na.action`	function determining what should be done with missing values in newdata. The default is to predict NA.
`...`	further arguments passed to or from other methods.

Details

Some combinations of `type` and `se.fit` and `interval` are not possible, depending on the distribution of the model. If `type` is "link" or the distribution is normal, then all combinations are possible. the value for type may be "mean" only if the distribution is log-normal, and the only additional option is "estimate" for `se.fit`. For either log-normal or common log distributions, if the `type` is "response," then the options for `se.fit` and `interval` ae "none" or "prediction."

Value

A vector of predictions or a matrix of predictions and the other selected statistics. In contrast with other regression predict functions, a list of fitted values and standard errors is never returned.

Note

The predicted value for `type = "mean"` uses the adjusted maximum likelihood method for bias correction described by Cohn (1988) and Cohn and others (1992). that method will give sligthly different predicted values than the minimum variance unbiased estimate (MVUE) described by Bradu and Mundlak (1970).

References

Bradu, D., and Mundlak, Y. 1979, Estimation in lognormal linear models: Journal of the American Statistical Association, v. 65, no. 39, p. 198–211.
Cohn, T.A., 1988, Adjusted maximum likelihood estimation of the moments of lognormal populations form type I censored samples: U.S. Geological Survey Open-File Report 88-350, 34 p.
Cohn, T.A., Gilroy, E.J., and Baier, W.G., 1992, Estimating fluvial transport of trace constituents using a regression model with data subject to censoring: Proceeding of the Joint Statistical Meeting, Boston, August 9–13, 1992, p. 142–151.

See Also

`censReg`, `predictMVUE`

`print.censMCT` *Print Results*

Description

Print the results of mutliple nonparametric comparison tests among groups of left-censored data.

Usage

```
## S3 method for class 'censMCT'
print(x, digits = 4, ...)
```

Arguments

`x`	an object of class "censMCT"—output from `censMulticomp.test`
`digits`	the number of significant digits to print
`...`	further arguments passed to or from other methods.

Value

The object `x` is returned invisibly.

Note

The printed output includes the name of the test, the error rate, the method for adjusting the p-values for the individual tests, the names of the response and group variables, and a table of the paired comparisons. The table includes the names of the groups comapred, the z-score from the `genWilcox.test`, the adjusted p-value, and a column to indicate if the adjusted p-value is less than the `alpha` level set in `censMulticomp.test`.

See Also

`censKSample.test`, `censMulticomp.test`, `genWilcox.test`

`print.censQuantile` *Print Results*

Description

Print the output of `censQuantile`.

Usage

```
## S3 method for class 'censQuantile'
print(x, digits = 4, ...)
```

Arguments

`x`	an object of class "censQuantile."
`digits`	the number of significant digits to print
`...`	further arguments passed to or from other methods.

Value

The object `x` is returned invisibly.

Note

The printed output appears as 2 columns, the first column is the percentile and the second column is the value.

print.censReg *Print Results*

Description

Print the results of an AMLE/MLE regression.

Usage

```
## S3 method for class 'censReg'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "censReg"—output from censReg
digits	the number of significant digits to print
...	further arguments passed to or from other methods.

Value

The object x is returned invisibly.

Note

The printed output includes the call, the coefficent table, the estimated residual standard error, the log-likelihood of the model and null model with the attained p-value, and the computational method.

See Also

censReg, coef.censReg

print.censStats *Print Results*

Description

Print the output of censStats.

Usage

```
## S3 method for class 'censStats'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "censStats."
digits	the number of significant digits to print
...	further arguments passed to or from other methods.

Value

The object x is returned invisibly.

Note

The printed includes the mean and standard deviation of the data and if a log-transform was used in the computation, then the mean and standard deviation of the log-transformed values.

---

print.lcens *Print Values*

Description

Print representations of censored or water-quality data values.

Usage

```
## S3 method for class 'lcens'
print(x, digits = 4, ...)

## S3 method for class 'mcens'
print(x, digits = 4, ...)

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

Arguments

x	the object to print.
digits	the number of significant digits for numbers.
...	further arguments passed to or from other methods.

Value

The object x is returned invisibly.

print.summary.censReg *Print Results*

Description

Print the summary output of an AMLE/MLE regression.

Usage

```
## S3 method for class 'summary.censReg'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "censReg"—output from censReg
digits	the number of significant digits to print
...	further arguments passed to or from other methods.

Value

The object x is returned invisibly.

Note

The printed output includes the call, the coefficent table, the estimated residual standard error, the log-likelihood of the model and null model with the attained p-value, and the computational method.

See Also

censReg, coef.censReg

probPlot.lcens *Probability Plot*

Description

Produces a probability plot: methods for "lcens" and "qw" data.

Usage

```
## S3 method for class 'lcens'
probPlot(x, truncate, FLIP = FALSE, distribution = "normal",
  alpha = 0.4, Plot = list(name = "Uncensored", what = "points", type =
  "solid", width = "standard", symbol = "circle", filled = TRUE, size = 0.09,
  color = "black"), yaxis.log = TRUE, yaxis.range = c(NA, NA),
  ylabels = 11, xlabels = 11, CDF = TRUE,
  xtitle = "Cumulative Probability", RI, RItitle,
  ytitle = deparse(substitute(x)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "Left censored", what = "points", symbol =
  "circle", filled = FALSE, size = 0.09, color = "black"), ...)

## S3 method for class 'qw'
probPlot(x, truncate, FLIP = FALSE, distribution = "normal",
  alpha = 0.4, Plot = list(name = "Uncensored", what = "points", type =
  "solid", width = "standard", symbol = "circle", filled = TRUE, size = 0.09,
  color = "black"), yaxis.log = TRUE, yaxis.range = c(NA, NA),
  ylabels = 11, xlabels = 11, CDF = TRUE,
  xtitle = "Cumulative Probability", RI, RItitle,
  ytitle = deparse(substitute(x)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "Left censored", what = "points", symbol =
  "circle", filled = FALSE, size = 0.09, color = "black"), ...)
```

Arguments

x	the data to plot. Missing values are removed before plotting.
truncate	not used for left-censored data.
FLIP	if TRUE, the plot the cumumlative distribution. Otherwise, plot as flipped data (largest values on left). This is not related to computing statistics by the flipped Kaplan-Meier method.
distribution	the name of the desired function converting from probabilities to coordinates.
alpha	the alpha value of the function for computing plotting positions.
Plot	control parameters of the uncensored data plot.
yaxis.log	log-transform the y axis?
yaxis.range	set the range of the y axis.
ylabels	set the y-axis labels. See linearPretty for details.
xlabels	set the x-axis labels. See probPretty for details.
CDF	if TRUE, then label with increasing probabilities. Otherwise label with decreasing probabilities.
xtitle	the x-axis title.
RI	not used for left-censored data.
RItitle	not used for left-censored data.
ytitle	the y-axis title.
caption	the figure caption.

margin	the parameters of the margin of the plot area.
Censored	control parameters of the left-censored data plot.
...	parameters for the distribution function or other method functions.

Details

The control parameters for left-censored data, Censored, are the same as those for uncensored data, Plot. See setPlot for details.

Value

Information about the graph.

See Also

qqPlot.lcens

Examples

```
## Not run:
set.seed(1932)
Xu <- rlnorm(32)
setPage("sq") # required page set up
probPlot(as.lcens(Xu, 1.0))

## End(Not run)
```

---

qqPlot.lcens *Q-Q Plot*

Description

Produces a quantile-quantile (q-q) or a q-normal plot: methods for "lcens" and "qw" data.

Usage

```
## S3 method for class 'lcens'
qqPlot(x, y, alpha = 0.4, Plot = list(name = "", what =
  "points", type = "solid", width = "standard", symbol = "circle", filled =
  TRUE, size = 0.09, color = "black"), LineRef = list(name = "", what =
  "lines", color = "black"), Line1.1 = list(name = "Line of equality", what =
  "lines", color = "gray"), yaxis.log = FALSE, yaxis.range = c(NA, NA),
  xaxis.log = FALSE, xaxis.range = c(NA, NA), ylabels = 7, xlabels = 7,
  xtitle, ytitle, caption = "", margin = c(NA, NA, NA, NA),
  Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black"), Projected = list(name = "", what =
  "points", symbol = "plus", filled = FALSE, size = 0.09, color = "black"), ...)
```

```
## S3 method for class 'qw'
qqPlot(x, y, alpha = 0.4, Plot = list(name = "", what =
  "points", type = "solid", width = "standard", symbol = "circle", filled =
  TRUE, size = 0.09, color = "black"), LineRef = list(name = "", what =
  "lines", color = "black"), Line1.1 = list(name = "Line of equality", what =
  "lines", color = "gray"), yaxis.log = FALSE, yaxis.range = c(NA, NA),
  xaxis.log = FALSE, xaxis.range = c(NA, NA), ylabels = 7, xlabels = 7,
  xtitle, ytitle, caption = "", margin = c(NA, NA, NA, NA),
  Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black"), Projected = list(name = "", what =
  "points", symbol = "plus", filled = FALSE, size = 0.09, color = "black"), ...)
```

Arguments

x	the x-axis data, or data to plot if y is missing. Must be of class "lcens," "mcens," or "qw." Missing values are permitted and removed bedfore plotting.
y	the y-axis data. Missing values are permitted and removed bedfore plotting. If missing, then produce a quantile-normal plot from the data in x.
alpha	the alpha value of the function for computing plotting positions.
Plot	control parameters of the uncensored data plot.
LineRef	control parameters of the reference line (best fit between x and y. See **Details**.
Line1.1	control parameters for the 1:1 line. Drawn only for q-q plot. See **Details**.
yaxis.log	log-transform the y axis?
yaxis.range	set the range of the y axis.
xaxis.log	log-transform the x axis?
xaxis.range	set the range of the x axis.
ylabels	set the y-axis labels. See linearPretty for details.
xlabels	set the x-axis labels. See linearPretty for details.
xtitle	the x-axis title.
ytitle	the y-axis title.
caption	the figure caption.
margin	the parameters of the margin of the plot area.
Censored	control parameters of the censored values q-normal plot.
Projected	control parameters of the projected (filled in) left-censored data q-normal plot.
...	any additional arguments required for specific methods.

Details

The argument what for either LineRef or Line1.1 may be set to "none" to suppress drawing of either line.

Value

Information about the graph.

See Also

probPlot.lcens

Examples

```
## Not run:
set.seed(1932)
Xu <- rlnorm(32)
setPage("sq") # required page set up
qqPlot(as.lcens(Xu, 1.0), yaxis.log=TRUE)

## End(Not run)
```

---

quantile.lcens *Sample Quantiles*

Description

Computes sample or estimated quantiles corresponding to the given probabilities: methods for "lcens," "mcens," and "qw" data. The smallest observation (censored or uncensored) corresponds to a probability of 0 and the largest to a probability of 1.

Usage

```
## S3 method for class 'lcens'
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  names = TRUE, method = "flipped K-M", type = 2, alpha = 0.4, ...)

## S3 method for class 'mcens'
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  names = TRUE, method = "flipped K-M", type = 2, alpha = 0.4, ...)

## S3 method for class 'qw'
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  names = TRUE, method = "flipped K-M", type = 2, alpha = 0.4, ...)
```

Arguments

x	an object of a censored-data class whose sample quantiles are wanted. NA and NaN values are not allowed unless na.rm is TRUE.
probs	numeric vector of probabilities with values in [0,1].
na.rm	logical; if TRUE, any NA and NaNs are removed from x before the quantiles are computed.
names	logical; if true, the result has a names attribute.
method	the method to use for computing quantiles. See **Details**.

type	an integer between 1 and 9 selecting one of the nine quantile algorithms described in quantile.
alpha	the offset fraction to be used, depending on method; typically in [0, 0.5].
...	not used, required for other methods.

Details

The methods available in the current version are "flipped K-M," "log ROS," "ROS," "log MLE," and "MLE." The method "flipped K-M" produces quantiles using the Kaplan-Meier method on flipped data described by Helsel (2012). The methods "log ROS" and "log MLE" are described by Helsel, 2012 and Helsel and Cohn (1988). The methods "ROS" and "MLE" are similar to "log ROS" and "log MLE" except that no log- and back-transforms are made on the data.

Value

An optionally named vector of the requested probabilities. The names of values that would be left-censored are marked with "*."

References

Helsel, D.R. 2012, Statistics for censored environmental data using Minitab and R: New York, Wiley, 324 p.
Helsel, D.R. and Cohn, T.A., 1988, Estimation of descriptive statistics for multiply censored water quality data: Water Resources Research v. 24, n. 12, pp. 1997-2004

See Also

quantile, censQuantile

Examples

```
set.seed(28)
Xu <- rnorm(23)
quantile(as.lcens(Xu, 0))
```

---

qw-class *Water-quality Data*

Description

Class "qw" describes water-quality data.

Slots

`.Data` a 2-column matrix that contains the lower and upper limit of the value of the water-quality data. Missing values are permitted.

`remark.codes` a character vector of remark codes. Only "<" and ">" are meaningful for the class; other values will be ignored, but may generate warnings. Missing values are permitted only for corresponding missing values in `.Data`.

`value.codes` a character vector for user-defined code that may help characterize the value of the water-quality data. Missing values are permitted only for corresponding missing values in `.Data`.

`reporting.level` a numeric vector describing the reporting level of the water-quality data. Missing values are permitted if the reporting level is not known.

`reporting.method` a character vector for user-defined code or description of the `reporting.level`. Generally used to distinguish between detection limits and quantitation levels.

`reporting.units` a character vector describing the units. Some values, like "mg/L" are recognized by some applications.

`analyte.method` a character vector to describe the analytical method.

`analyte.name` a character vector of the analyte name. Some values may be recognized by some applications.

`rounding` a vector of length 2 that describes how to round the data for printing. The first value is the number of significant digits, the second is the maximum number of decimal places, default is c(2,3).

`unique.code` a character vector that defines a distinct collection of data. Typically used to define columns is datasets organized by sample.

`names` a character vector of unique names for each element in `.Data`.

Objects from the Class

Objects can be created by calls to `as.qw`.

References

Lorenz, D.L., in preparation

Examples

```
showClass("qw")
```

qw2mcens *Conversion to class "mcens"*

Description

Converts objects of class "qw" to "mcens" retaining interval censoring for "less-than" data.

Usage

```
qw2mcens(x)
```

Arguments

x	an object of class "qw" to convert.

Value

A vector of the data converted to class "mcens" retaining interval censoring for "less-than" data.

See Also

as.mcens

qwCoalesce *Replace missing values*

Description

Constructs a vector with as few missing values as possible from a selected sequence of vectors.

Usage

```
qwCoalesce(..., means = FALSE, lt.tol = 0.1)
```

Arguments

...	any number of objects of class "qw." Missing values are permitted.
means	logical, if TRUE, then compute the mean of all nonmissing values in the data sepcified in .... if FALSE, then select the first nonmissing value the sequence spcified in ....
lt.tol	an arbitrary tolerance metric to drop interval-censoring. See mean.qw for details.

Value

For means set to FALSE, an object of class "qw" in which each element is determined by selecting the first non-missing value in the order in which they are specified in the argument list. The first step is to construct a matrix from all arguments. The output is initially set to column 1; for any missing values in the column, the data from column 2 are used and so on until all columns have been searched or all missing values replaced. The metadata are set from the corresponding column.

For means set to FALSE, an object of class "qw" computed from the mean of the nonmissing values in each row of the data specified in .... The metadata, except for reporting limit information, are set from the first entry in ....

Note

This function is most useful for creating a column in a dataset from related columns that represent different methods. For example, a single column of alkalinity may be desired when there are multiple columns of alkalinity determined by various methods.

---

qwRPD *Relative Differences*

Description

Computes the relative percent differences (RDP) between samples and replicates for data of class "qw."

Usage

```
qwRPD(x, y, incomparables = NA, plotit = FALSE)
```

Arguments

x	sample data of class "qw."
y	replicate data of class "qw."
incomparables	logical, NA means the RPD for any pair that includes left-censored values is NA (default), FALSE means the RPD for any pair that includes left-censored values is the minimum possible value, and TRUE means the RPD for any pair that includes includes left-censored values returns the range of possible values. See Notes.
plotit	logical, if TRUE, then create a Bland-Altman mean-difference plot (Bland and Altman, 1986); otherwise no plot is created. If incomparables is set to TRUE, then the values corresponding to the respective maximums are plotted.

Value

A vector of the relative percent differences for each paired sample and duplicate. If incomparables is TRUE, then the vector is class "mcens," otherwise the vector id numeric.

Note

This function works differently from other functions that process data of class "qw"—it ignores the reporting level information and uses the data as they are recorded. The treatment of interval-censored data varies depending on `incomparables`. If `incomparables` is set to `NA` or `FALSE`, then interval-censored data are set to the midrange, otherwise the data are retained as interval-censored data. Right-censored data always return NA regardless of `incomparables`.

Inverval-censored data are not common and require much additional processing to get accurate ranges for the RPD. Approximate values are returned in this version.

References

Bland J.M. and Altman D.G., 1986 Statistical methods for assessing agreement between two methods of clinical measurement: Lancet, i, p. 307–310.

See Also

`rpd`

`ratio` *Compute Ratios*

Description

Computes ratios from water-quality or other data and maintain the integrity of the data, as opposed to methods that use simple substitution

Usage

```
ratio(numerator, denominator)
```

Arguments

`numerator`	any valid vector that can be converted to class "mcens." Missing values are permitted but result in missing values in the output.
`denominator`	any valid vector that can be converted to class "mcens." Missing values are permitted but result in missing values in the output.

Details

Objects of class "qw" are converted to class "mcens" using the `qw2mcens` function, which preserves the minimum value of 0 for "less than" data rather than `as.mcens`, which converts the to left-censored values.

Value

An object of class "mcens" that is the result of the requested division.

rbindQW *Combine Data by Rows*

Description

Combines a sequence of data frame arguments and combine by rows. This is a specialized version of rbind that works for data frames that contain columns of class "qw."

Usage

```
rbindQW(...)
```

Arguments

...	any number of data frames with identical columns. The missing value NA is permitted as a special case to allow the addition of missing values.

Value

A data frame with all columns combined in the order specified in ....

See Also

rbind

readNWQLdl *Historical Reporting Limits*

Description

Imports the historical reporting limits for all methods for a parameter code.

Usage

```
readNWQLdl(parm_cd)
```

Arguments

parm_cd	the parameter code.

Value

A data frame of the analyte, methods, begin and end date for each reporting level change, the reporting level type used, the long-term detection limit and the reporting level.

Note

This function works only within the internal USGS network.

References

Lorenz, D.L., 2014, USGSqw OFR.

See Also

qw-class

Examples

```
## Not run:
readNWQLdl("00608")

## End(Not run)
```

`residuals.censReg` *Extract Model Residuals*

Description

Extracts the residuals from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
residuals(object, type = "working",
  suppress.na.action = FALSE, ...)
```

Arguments

`object`	an object of class "censReg"—output from `censReg`
`type`	The type of residuals, see **Details**.
`suppress.na.action`	logical, suppress the effects of the `na.action` in the call to `censReg` and return only the fitted values corresponding to the fitted data.
`...`	further arguments passed to or from other methods.

Details

The value for type can be any one of the following:

Value	Description
"working"	Residuals with censored residuals replaced by their expected values
"response"	Residuals from the linear predictor
"influence"	An estimate of Cook's D values based on "working" residuals
"leverage"	The hat diagonals
"S-L"	The square-root of the absolute value of the residuals with censored residuals replaced by their expected value

Also, any other value of type for residuals.survreg can be used to obtain those residuals. Note that "working" and "response" are defined in the table above, in keeping with older versions of censReg.

Value

The residuals from the regression as specified by type.

See Also

censReg

`rmse.censReg` *Root-Mean-Squared Error*

Description

Computes the root-mean-squared error (RMSE) of a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
rmse(x, ...)
```

Arguments

x	the output from censReg.
...	further arguments passed to or from other methods.

Value

The estimated root-mean-squared error, also know as the residual standard error.

See Also

censReg

sdlFill *Estimate Statistics*

Description

Support function for computing statistics for left-censored data.

Usage

```
sdlFill(x, method = "fill", alpha = 0.4)
```

Arguments

x	the data to estimate, Missing values permitted and ignored. Must be an object of class "lcens," a numeric vector, or the output from censpp.
method	the method to use, either "fill" or "log fill"
alpha	the offset for plotting postion (not used, required for other methods).

Value

A list containing the mean and standard deviation, filled in values for the censored values, and the maximum censor level.

Note

This function uses the method of Gleit (1985). That method computes the mean and standard deviation of the data, estimates synthetic values for the censored values using a method similar to ROS, and repeats the process until the mean and standard deviation stabilize.

References

Gleit, A., 1985, Estimation for small normal data sets with detection limits: Environmental Science and Technology, v. 19, no. 12, p. 1201–1206.

show-censored *Show Methods for* WSqw *objects*

Description

Prints the object.

Usage

```
## S4 method for signature 'lcens'
show(object)

## S4 method for signature 'mcens'
show(object)

## S4 method for signature 'qw'
show(object)
```

Arguments

object	the object to show.

Value

The object is retruned invisibly.

---

sort.lcens *Sorting Vectors*

Description

Sorts a vector into ascending or descending order: methods for "lcens" and "mcens" data. This function sorts left-censored values immediately below the equivalent uncensored values and right-censored values immediately above the equivalent uncensored values.

Usage

```
## S3 method for class 'lcens'
sort(x, decreasing = FALSE, na.last = NA, ...)

## S3 method for class 'mcens'
sort(x, decreasing = FALSE, na.last = TRUE, ...)
```

Arguments

x	the data to be sorted.
decreasing	sort the data in descending order?
na.last	for controlling the treatment of NAs. If TRUE, then missing values in the data are put last; if FALSE, then they are put first; if NA, then they are removed.
...	arguments to be passed to or from methods.

Value

An object like x, but with the data sorted. Left-censored values are placed between smaller numeric values and the corresponding uncensored values.

Examples

```
set.seed(5422)
XR <- rnorm(10)
sort(as.lcens(XR, rep(c(0, -1), each=5))) # censors at 0 and -1
```

---

splitQual *Split Leading Qualifiers*

Description

Splits a vector of strings of combined data into numeric and remark columns.

Usage

```
splitQual(x, name = deparse(substitute(x)))
```

Arguments

x	a character vector, missing values are permitted and result in missing values in the output
name	a character string giving the names of the output columns.

Value

A data frame with 2 columns, containing the numeric values and the qualification codes.

Examples

```
splitQual(c("<1", "1", "<1", "1", "2"), name="Dummy")
```

---

str.lcens *Display Structure*

Description

Displays the basic information about an object: methods for "lcens," "mcens," and "qw" data.

Usage

```
## S3 method for class 'lcens'
str(object, ...)

## S3 method for class 'mcens'
str(object, ...)

## S3 method for class 'qw'
str(object, ...)
```

Arguments

object	an object of class "lcens,", "mcens," or "qw."
...	any additional valid arguments ot the default method for str and give.censoring, a logical value that includes the type of censoring in the output if TRUE.

Value

Nothing is returned, the side effect is to print a short summary of the object.

See Also

str

Examples

```
str(as.lcens(c(1,3), 2))
```

---

summary.censReg *Summarizing Linear Model Fits*

Description

Summarizes the output from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
summary(object, correlation = FALSE, ...)
```

Arguments

object	an object of class "censReg"—output from censReg.
correlation	include the correlation matrix of the estimated parameters?
...	further arguments passed to or from other methods.

Value

An object of class "summary.censReg" containing the object, the pseudo R-squared, variance inflation factors, a table of diagnostic statistics, critical values for selected diagnostic statistics, an indication of which observations exceed any of the selected diagnostic statistics, and optionally the parameter correlation matrix.

Note

The pseudo R-squared is computed using the McKelvey-Zavoina (1975) method, which tries to describe the proportion of variance explained by the fit and tries to capture the square of the correlation between the fitted and actual values. For uncensored data, it is equal to the usual R-squared for ordinary least squares.

References

McKelvey, R.D., and Zavoina, W., 1975, A statistical model for the analysis of ordinal dependent variables: The Journal of Mathematical Sociology, v. 4, no. 1, p. 103–120.

See Also

censReg

---

summary.lcens *Summarize Data*

Description

Provide a summary of censored or water-quality data.

Usage

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

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

## S3 method for class 'qw'
summary(object, details = FALSE, ...)
```

Arguments

object	the object to summarize.
...	further arguments passed to other methods.
details	provide details on multiple entries?

Value

For details = FALSE, a named character vector describe some characteristics of object. For details = TRUE, a named list containing vectors with all unique values.

Note

This function is used when summary is called on a data frame. Setting details to TRUE can be useful for individual columns in a data frame.

Examples

```
## Not run:
# Summrize these data
summary(importNWISqw("05330000", "00608"))

## End(Not run)
```

`timePlot-censored` *Censored Data Methods for Function* `timePlot`

Description

Creates a plot of time-series data: selected methods for "lcens" and "qw" data.

Usage

```
## S4 method for signature 'Date,lcens'
timePlot(x, y, Plot = list(name = "", what = "points",
  type = "solid", width = "standard", symbol = "circle", filled = TRUE, size =
  0.09, color = "black"), yaxis.log = FALSE, yaxis.rev = FALSE,
  yaxis.range = c(NA, NA), xaxis.range = range(x, na.rm = TRUE),
  ylabels = 7, xlabels = "Auto", xtitle = "",
  ytitle = deparse(substitute(y)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black", bar = FALSE), ...)

## S4 method for signature 'Date,qw'
timePlot(x, y, Plot = list(name = "", what = "points",
  type = "solid", width = "standard", symbol = "circle", filled = TRUE, size =
  0.09, color = "black"), yaxis.log = FALSE, yaxis.rev = FALSE,
  yaxis.range = c(NA, NA), xaxis.range = range(x, na.rm = TRUE),
  ylabels = 7, xlabels = "Auto", xtitle = "",
  ytitle = deparse(substitute(y)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black", bar = FALSE), ...)
```

Arguments

`x`	the time/date data.
`y`	the y-axis data.
`Plot`	control parameters of the plot for uncensored data.
`yaxis.log`	log-transform the y axis?
`yaxis.rev`	reverse the y axis?
`yaxis.range`	set the range of the y-axis.

xaxis.range	set the range of the x-axis.
ylabels	set up y-axis labels. See linearPretty for details.
xlabels	set up x-axis labels. See **Details**.
xtitle	the x-axis title.
ytitle	the y-axis title.
caption	the figure caption.
margin	set up the plot area margins.
Censored	control parameters of the plot for censored data.
...	arguments for specific methods.

Details

The value for xlabels must be one of "full," the full month names; "abbrev," abbreviations; or "letter," the first letter of the month. The default is "Auto," which selects one of the rpevious options based on the number of labels.

Value

Information about the graph.

---

vcov.censReg *Variance-Covariance Matrix*

Description

Extracts the variance-covariance matrix of the main parameters from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
vcov(object, ...)
```

Arguments

object	an object of class "censReg"—output from censReg.
...	not used, further arguments passed to or from other methods.

Value

A matrix of the estimated covariances between the parameter estimates for each main parameter of the model. The matrix has row and column names corresponding to the parameter names.

See Also

censReg

View *Invoke a Data Viewer*

Description

Invoke a spreadsheet-style data viewer on a matrix-like R object.

Usage

```
View(x, title)
```

Arguments

x	an R object that can be forecd to a character representation of a data frame.
title	the title for the viewer window. Defaults to the name of x.

Value

Invisibly returns NULL. The functions opens a widnow showing the formatted contents of x and returns immediately.

vif.censReg *Variance Inflation Factors*

Description

Extracts the variance inflation factors (Helsel and Hirsch, 2002) from a censored regression object: method for "censReg" object.

Usage

```
## S3 method for class 'censReg'
vif(model, ...)
```

Arguments

model	an object of class "censReg"—output from censReg.
...	not used, further arguments passed to or from other methods.

Value

A named numeric vector containing the variance inflation factor for each variable.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

See Also

censReg

---

xyPlot-censored *Censored Data Methods for Function* xyPlot

Description

Creates a line/scatter plot: selected methods for "lcens," "mcens," and "qw" data.

Setting xlabels to 0 or negative values will suppress ticks and labels. If negative, then try to create that absolute value number of labels. That can be useful for relative axes or specialized labeling, but is only valid for numeric x.

Usage

```
## S4 method for signature 'numeric,lcens'
xyPlot(x, y, Plot = list(name = "", what = "points",
  type = "solid", width = "standard", symbol = "circle", filled = TRUE, size =
  0.09, color = "black"), yaxis.log = FALSE, yaxis.rev = FALSE,
  yaxis.range = c(NA, NA), xaxis.log = FALSE, xaxis.range = c(NA, NA),
  ylabels = 7, xlabels = 7, xtitle = deparse(substitute(x)),
  ytitle = deparse(substitute(y)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black", width = "hairline", bar = FALSE), ...)

## S4 method for signature 'numeric,qw'
xyPlot(x, y, Plot = list(name = "", what = "points",
  type = "solid", width = "standard", symbol = "circle", filled = TRUE, size =
  0.09, color = "black"), yaxis.log = FALSE, yaxis.rev = FALSE,
  yaxis.range = c(NA, NA), xaxis.log = FALSE, xaxis.range = c(NA, NA),
  ylabels = 7, xlabels = 7, xtitle = "",
  ytitle = deparse(substitute(y)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black", width = "hairline", bar = FALSE), ...)

## S4 method for signature 'lcens,lcens'
xyPlot(x, y, Plot = list(name = "", what = "points",
  type = "solid", width = "standard", symbol = "circle", filled = TRUE, size =
  0.09, color = "black"), yaxis.log = FALSE, yaxis.rev = FALSE,
  yaxis.range = c(NA, NA), xaxis.log = FALSE, xaxis.range = c(NA, NA),
  ylabels = 7, xlabels = 7, xtitle = deparse(substitute(x)),
  ytitle = deparse(substitute(y)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black", width = "hairline", bar = TRUE), ...)

## S4 method for signature 'numeric,mcens'
```

```
xyPlot(x, y, Plot = list(name = "", what = "points",
  type = "solid", width = "standard", symbol = "circle", filled = TRUE, size =
  0.09, color = "black"), yaxis.log = FALSE, yaxis.rev = FALSE,
  yaxis.range = c(NA, NA), xaxis.log = FALSE, xaxis.range = c(NA, NA),
  ylabels = 7, xlabels = 7, xtitle = deparse(substitute(x)),
  ytitle = deparse(substitute(y)), caption = "", margin = c(NA, NA, NA,
  NA), Censored = list(name = "", what = "points", symbol = "circle", filled =
  FALSE, size = 0.09, color = "black", width = "hairline", bar = TRUE), ...)
```

Arguments

x	the x-axis data.
y	the y-axis data.
Plot	control parameters of the plot for uncensored data.
yaxis.log	log-transform the y axis?
yaxis.rev	reverse the y axis?
yaxis.range	set the range of the y-axis.
xaxis.log	logical, if TRUE, then log-transform the x axis.
xaxis.range	set the range of the x-axis.
ylabels	set up y-axis labels. See linearPretty for details.
xlabels	set up x-axis labels. See **Details**.
xtitle	the x-axis title.
ytitle	the y-axis title.
caption	the figure caption.
margin	set up the plot area margins.
Censored	control parameters of the plot for censored data. Same arguments as for Plot, but adds bar which if set to TRUE will draw lines from left-censored values to the minimum and draw lines from right-censored values to the maximum. No bars are drawn for interval censored values.
...	additional arguments for specific methods.

Value

Information about the graph.

Note

A call should be made to setPage to set up the graphics environment before calling xyPlot.

Index

∗ **IO**
convert2qw, 48
convertFqw, 49
importNWISqw, 62
importQW, 63
pcodeNWISqw, 86
readNWQLdl, 110
∗ **attribute**
censoring, 32
censorLevels, 33
length.lcens, 69
pctCens, 87
∗ **censored**
censCor, 24
censKSample.test, 26
censMove.1, 29
censMulticomp.test, 30
censoring, 32
censorLevels, 33
censReg, 40
censReg_AMLE.fit, 42
genWilcox.test, 60
kendallATS.test, 67
mean.lcens, 78
mean.qw, 79
median.lcens, 80
pairedPratt.test, 83
pctCens, 87
plot.qw, 91
ppw.test, 93
∗ **classes**
lcens-class, 68
mcens-class, 74
qw-class, 105
∗ **datasets**
atrazine.df, 18
atrazine.dl, 19
atrazine.spk, 20
convert2qw, 48
convertFqw, 49
importNWISqw, 62
importQW, 63
pcodeColData, 85
pcodeNWISqw, 86
∗ **data**
rbindQW, 110
∗ **hplot**
boxPlot.lcens, 21
dotPlot-censored, 52
ecdfPlot.lcens, 54
plot.censReg, 89
plot.ppw, 90
plot.qw, 91
plot.summary.censReg, 91
probPlot.lcens, 100
qqPlot.lcens, 102
timePlot-censored, 118
xyPlot-censored, 121
∗ **htest**
censKSample.test, 26
censMulticomp.test, 30
censPPCC.test, 36
genWilcox.test, 60
kendallATS.test, 67
pairedPratt.test, 83
ppw.test, 93
∗ **manip**
[.lcens, 5
%~=%, 7
Arith-censored, 8
as.character.lcens, 10
as.data.frame.lcens, 10
as.double.qw, 11
as.lcens, 12
as.mcens, 14
as.qw, 16
censor, 31
censpp, 34

Compare-censored, 47
dlimit, 50
fillIn, 56
format.lcens, 58
is.na.lcens, 66
Math-censored, 71
percentile.lcens, 88
qwCoalesce, 107
show-censored, 113
sort.lcens, 114
* **math**
percentile.lcens, 88
* **methods**
Arith-censored, 8
as.lcens, 12
as.mcens, 14
as.qw, 16
censor, 31
Compare-censored, 47
dotPlot-censored, 52
Math-censored, 71
show-censored, 113
timePlot-censored, 118
xyPlot-censored, 121
* **misc**
mcenKM, 72
mcenMLE, 73
mcenROS, 73
mdlAMLE, 75
mdlKM, 75
mdlKMstats, 76
mdlMLE, 76
mdlROS, 77
sdlFill, 113
summary.lcens, 117
* **nonparametric**
censMulticomp.test, 30
* **package**
smwrQW-package, 5
* **print**
print.lcens, 99
* **regression**
censMove.1, 29
censReg, 40
extractAIC.censReg, 55
fitted.censReg, 57
nobs.censReg, 83
plot.censReg, 89
plot.summary.censReg, 91
residuals.censReg, 111
summary.censReg, 116
vcov.censReg, 119
vif.censReg, 120
* **summary**
censCor, 24
* **univariate**
censStats, 43
mean.lcens, 78
mean.qw, 79
median.lcens, 80
* **univar**
censMean.CI, 27
censQtiles.CI, 37
censQuantile, 38
censSumStats, 44
percentile.lcens, 88
quantile.lcens, 104
* **utilities**
censReg_AMLE.fit, 42
makeColNames, 71
print.censMCT, 96
print.censQuantile, 97
print.censReg, 98
print.censStats, 98
print.summary.censReg, 100
[.lcens, 5
[.mcens *([.lcens)*, 5
[.qw, *6*
[.qw *([.lcens)*, 5
%~=%, 7

add, 8, *9*
Arith,lcens,lcens-method
*(*Arith-censored*)*, 8
Arith,lcens,numeric-method
*(*Arith-censored*)*, 8
Arith,mcens,mcens-method
*(*Arith-censored*)*, 8
Arith,mcens,numeric-method
*(*Arith-censored*)*, 8
Arith,numeric,lcens-method
*(*Arith-censored*)*, 8
Arith,numeric,mcens-method
*(*Arith-censored*)*, 8
Arith,numeric,qw-method
*(*Arith-censored*)*, 8

Arith,qw,numeric-method (Arith-censored), 8
Arith,qw,qw-method (Arith-censored), 8
Arith-censored, 8
as.character.lcens, 10
as.character.mcens (as.character.lcens), 10
as.character.qw (as.character.lcens), 10
as.data.frame.lcens, 10
as.data.frame.mcens (as.data.frame.lcens), 10
as.data.frame.qw (as.data.frame.lcens), 10
as.double.qw, 11, *57*
as.lcens, 12, *32*
as.lcens,character,missing,missing-method (as.lcens), 12
as.lcens,lcens,missing,missing-method (as.lcens), 12
as.lcens,numeric,missing,character-method (as.lcens), 12
as.lcens,numeric,missing,logical-method (as.lcens), 12
as.lcens,numeric,missing,missing-method (as.lcens), 12
as.lcens,numeric,numeric,character-method (as.lcens), 12
as.lcens,numeric,numeric,logical-method (as.lcens), 12
as.lcens,numeric,numeric,missing-method (as.lcens), 12
as.lcens,qw,missing,missing-method (as.lcens), 12
as.lcens,qw,numeric,missing-method (as.lcens), 12
as.mcens, *13*, 14, *32*, *107*
as.mcens,character,missing,missing-method (as.mcens), 14
as.mcens,lcens,missing,missing-method (as.mcens), 14
as.mcens,mcens,missing,missing-method (as.mcens), 14
as.mcens,numeric,missing,character-method (as.mcens), 14
as.mcens,numeric,missing,integer-method (as.mcens), 14
as.mcens,numeric,missing,missing-method (as.mcens), 14
as.mcens,numeric,missing,numeric-method (as.mcens), 14
as.mcens,numeric,numeric,missing-method (as.mcens), 14
as.mcens,qw,missing,missing-method (as.mcens), 14
as.qw, 16
as.qw,numeric,character,character,numeric,character,charac (as.qw), 16
as.qw,numeric,character,character,numeric,character,charac (as.qw), 16
atrazine.df, 18
atrazine.dl, 19
atrazine.spk, 20

boxPlot, *22*
boxPlot.data.frame (boxPlot.lcens), 21
boxPlot.lcens, 21
boxPlot.qw (boxPlot.lcens), 21
boxPlotCensStats, 23

c.lcens, 24
c.mcens (c.lcens), 24
c.qw (c.lcens), 24
censCor, 24
censKSample.test, 26, *31*, *97*
censMean.CI, 27
censMove.1, 29
censMulticomp.test, 30, *97*
censor, 31
censoring, 32, *34*, *87*
censorLevels, 33, *33*
censpp, 34, *37*
censPPCC.test, 36
censQtiles.CI, 37
censQuantile, 38, *38*, *80*, *105*
censReg, 40, *42*, *43*, *47*, *55*, *58*, *70*, *83*, *89*, *92*, *96*, *98*, *100*, *112*, *117*, *119*, *121*
censReg_AMLE.fit, 42
censReg_AMLE.pred, 42
censReg_MLE.fit (censReg_AMLE.fit), 42
censStats, *28*, 43, *78*, *79*
censSumStats, 44
code01, 45
codeU, 46
coef.censReg, 46, *98*, *100*
Compare,lcens,numeric-method (Compare-censored), 47

Compare,mcens,numeric-method
(Compare-censored), 47
Compare,numeric,lcens-method
(Compare-censored), 47
Compare,numeric,mcens-method
(Compare-censored), 47
Compare,numeric,qw-method
(Compare-censored), 47
Compare,qw,numeric-method
(Compare-censored), 47
Compare-censored, 47
convert2qw, 48, *50*
convertFqw, 49

data.frame, *11*
dlimit, 50
dotPlot,mcens-method
(dotPlot-censored), 52
dotPlot,qw-method (dotPlot-censored), 52
dotPlot-censored, 52

ecdf, *88*
ecdfPlot.lcens, 54
ecdfPlot.qw (ecdfPlot.lcens), 54
extractAIC, *55*
extractAIC.censReg, 55

fillIn, *12*, *23*, 56
fitted.censReg, 57
format.lcens, 58
format.mcens (format.lcens), 58
format.qw (format.lcens), 58

gehanScores, 59
genWilcox.test, 60, *97*

importNWISqw, *18*, 62, *65*
importQW, 63
imputeLessThans, 65, *82*
is.na.lcens, 66
is.na.mcens (is.na.lcens), 66
is.na.qw (is.na.lcens), 66

kendallATS.test, 67

lcens-class, 68
length.lcens, 69
length.mcens (length.lcens), 69
length.qw (length.lcens), 69
linearPretty, *22*, *54*, *101*, *103*, *119*, *122*
lm, *41*
logLik.censReg, 70
lrDA, *81*, *82*
lrEM, *66*

makeColNames, *49*, *63–65*, 71
Math,lcens-method (Math-censored), 71
Math,mcens-method (Math-censored), 71
Math,qw-method (Math-censored), 71
Math-censored, 71
mcenKM, 72
mcenMLE, 73
mcenROS, 73
mcens-class, 74
mdlAMLE, *44*, 75
mdlKM, *44*, 75
mdlKMstats, *44*, 76
mdlMLE, *44*, 76
mdlROS, *44*, 77
mean.lcens, 78
mean.mcens (mean.lcens), 78
mean.qw, 79, *107*
median.lcens, 80
median.mcens (median.lcens), 80
median.qw (median.lcens), 80
mImputeLessThans, *66*, 81
move.1, *30*
multiply, 82

nobs.censReg, 83

p.adjust, *31*
pairedPratt.test, 83
pcodeColData, *71*, 85
pcodeNWISqw, *71*, 86
pctCens, 87
percentile.lcens, 88
percentile.qw (percentile.lcens), 88
plot.censReg, 89, *92*
plot.genWilcox (plot.ppw), 90
plot.ppw, 90
plot.qw, 91
plot.summary.censReg, 91
pow, 93
ppcc.test, *37*
ppoints, *35*
ppw.test, *84*, 93
predict.censReg, 95
predictMVUE, *96*

print.censMCT, 96
print.censQuantile, 97
print.censReg, 98
print.censStats, 98
print.lcens, 99
print.mcens *(*print.lcens*)*, 99
print.qw *(*print.lcens*)*, 99
print.summary.censReg, 100
probPlot.lcens, *55*, 100, *104*
probPlot.qw *(*probPlot.lcens*)*, 100
probPretty, *101*

qqPlot.lcens, 102, *102*
qqPlot.qw *(*qqPlot.lcens*)*, 102
quantile, *39*, *105*
quantile.lcens, *39*, 104
quantile.mcens *(*quantile.lcens*)*, 104
quantile.qw *(*quantile.lcens*)*, 104
qw-class, 105
qw2mcens, 107
qwCoalesce, 107
qwRPD, 108

ratio, *9*, 109
rbind, *110*
rbindQW, 110
readNWISpCode, *86*
readNWISqw, *49, 50*, *63*, *65*
readNWQLdl, *52*, 110
residuals.censReg, 111
residuals.survreg, *112*
rmse.censReg, 112
rpd, *109*

sdlFill, 113
setPage, *53*
setPlot, *102*
show,lcens-method *(*show-censored*)*, 113
show,mcens-method *(*show-censored*)*, 113
show,qw-method *(*show-censored*)*, 113
show-censored, 113
smwrQW-package, 5
sort.lcens, 114
sort.mcens *(*sort.lcens*)*, 114
splitQual, 115
str, *116*
str.lcens, 115
str.mcens *(*str.lcens*)*, 115
str.qw *(*str.lcens*)*, 115
subset.qw *(*[.lcens*)*, 5
summary.censReg, *89*, 116
summary.lcens, 117
summary.mcens *(*summary.lcens*)*, 117
summary.qw *(*summary.lcens*)*, 117
Surv, *15*
survdiff, *27*, *60*, *61*, *94*
survfit, *60*, *61*, *94*
survreg, *41*, *42*

timePlot,Date,lcens-method
*(*timePlot-censored*)*, 118
timePlot,Date,qw-method
*(*timePlot-censored*)*, 118
timePlot-censored, 118

vcov.censReg, 119
View, 120
vif.censReg, 120

xyPlot,Date,qw-method
*(*xyPlot-censored*)*, 121
xyPlot,lcens,lcens-method
*(*xyPlot-censored*)*, 121
xyPlot,numeric,lcens-method
*(*xyPlot-censored*)*, 121
xyPlot,numeric,mcens-method
*(*xyPlot-censored*)*, 121
xyPlot,numeric,qw-method
*(*xyPlot-censored*)*, 121
xyPlot-censored, 121