Package 'accrued'

August 24, 2016

Type Package

Title Data Quality Visualization Tools for Partially Accruing Data

Version 1.4.1

Date 2016-06-07

Author Julie Eaton and Ian Painter

Maintainer Julie Eaton <jreaton@uw.edu>

Depends R (>= 2.14.1), grid

Description Package for visualizing data quality of partially accruing data.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-08-24 22:19:26

R topics documented:

`accrued-package` *Data Quality Visualization Tools for Partially Accruing Data*

Description

Package for visualizing data quality of partially accruing data.

Details

Package:	accrued
Type:	Package
Version:	1.4.1
Date:	2016-06-07
License:	GPL-3

Author(s)

Julie Eaton (jreaton@uw.edu) and Ian Painter

References

[1] Painter I, Eaton J, Olson D, Revere D, Lober W. How good is your data. In conference abstracts for the International Society for Disease Surveillance Conference 2011: Building the Future of Public Health Surveillance. Emerging Health Threats Journal. 2011;4. (http://www.eht-journal.net/index.php/ehtj/article/view/11907)

[2] Painter I, Eaton J, Olson D, Lober W, Revere D. (2011). Visualizing data quality: tools and views. In conference abstracts for the International Society for Disease Surveillance Conference 2011: Building the Future of Public Health Surveillance. Emerging Health Threats Journal. 2011;4. (http://www.eht-journal.net/index.php/ehtj/article/view/11907)

[3] Lober W, Reeder B, Painter I, Revere D, Bugni P, McReynolds J, Goldov K, Webster E, Olson D. Technical Description of the Distribute Project: A Community-based Syndromic Surveillance System Implementation. Online Journal of Public Health Informatics. 2014;5(3). (http://dx.doi.org/10.5210/ojphi.v5i3.4938)

[4] J. Eaton, I. Painter, D. Olson, W. Lober. Visualizing the quality of partially accruing data for use in decision making. Online Journal of Public Health Informatics. 2015;7(3). (http://dx.doi.org/10.5210/ojphi.v7i3.6096)

Examples

```
data(accruedDataExample)
testData <- data.accrued(accruedDataExample)
```

```
plot(testData)
summary(testData)
plot(summary(testData))
uploadPattern(testData)
laggedTSarray(testData, lags=c(1,3,5,7) )
lagHistogram(testData)
summary(accruedErrors(testData))
plot(accruedErrors(testData))
currentValues = asOf(testData, currentDate=20)
# plot(currentValues)

data(accruedDataILIExample)
testData2 <- data.accrued(accruedDataILIExample)
plot(accruedErrors(testData, testData2))
```

`accruedDataExample` *Data set for illustrating accrued data methods*

Description

Simulated dataset of 12 months of counts accrued over 12 days. This data set is a `data.frame`. To use the dataset with any of the functions in the package, the user must first convert it to an `accrued` object.

Usage

```
data(accruedDataExample)
```

Format

A data frame of count data showing the cumulative count data received, with each row representing an encounter date and columns representing the number of days after the encounter. The "final" column represents the final count for each encounter date.

Details

A simulated dataset designed to represent typical observed behavior.

Examples

```
data(accruedDataExample)
plot(data.accrued(accruedDataExample))
summary(data.accrued(accruedDataExample))
uploadPattern( data.accrued(accruedDataExample), horizontal=TRUE )
```

accruedDataILIExample *Data set for illustrating accrued data methods*

Description

Simulated dataset of 12 months of counts accrued over 12 days. This data set is a `data.frame`. To use the dataset with any of the functions in the package, the user must first convert it to an `accrued` object.

Usage

```
data(accruedDataILIExample)
```

Format

A data frame of influenza-like-illness count data showing the cumulative count data received, with each row representing an encounter date and columns representing the number of days after the encounter. The "final" column represents the final count for each encounter date.

Details

A simulated dataset designed to represent typical observed behavior.

Examples

```
# denominator
data(accruedDataExample)
testData = data.accrued(accruedDataExample)

# numerator
data(accruedDataILIExample)
testData2 = data.accrued(accruedDataILIExample)

errors = accruedErrors(testData, testData2)
plot(errors)
summary(errors)
```

accruedErrors *Function to calculate error between partially accrued data and final data*

Description

Returns the error between final data values and partially accruing values, by accrual lag.

Usage

```
accruedErrors(x, y=NULL, func=NULL)
```

Arguments

x	Object of the accrued class.
y	Optional second object of the accrued class.
func	Error function to calcuate errors. Default behavior depends on whether one or two objects are included in the call. The first argument is the final value of the indicator and the second argument is the lagged value of the indicator.

Details

If only x is specified, calculates the error between each lag value and the final value for each encounter date. If both x and y are specified, then the error is calculated for the ratio y/x (except in that $X/0$ is set to NA for all X). The deault error function in only x is specified is the difference between the lag value and the corresponding final value. The default error function if both are specified is the difference in logs of the two values.

Value

An object of the accruedErrors class which consists of a matrix with columns "EncounterDate", "Lag" and "Error".

Author(s)

Julie Eaton and Ian Painter

See Also

data.accrued, plot.accruedErrors, summary.accruedErrors

Examples

```
data(accruedDataExample) # simulated accrued data
testData <- data.accrued(accruedDataExample)
errors <- accruedErrors(testData)
plot(errors)
errors <- accruedErrors(testData, func = function(a,b) {sqrt(a) - sqrt(b)} )
plot(errors)

data(accruedDataILIExample) # simulated accrued data - ILI counts
testData2 = data.accrued(accruedDataILIExample)
errors <- accruedErrors(testData, testData2)
plot(errors)
```

asOf *Function to recover the current data.*

Description

Creates a vector of current data as of a specified date.

Usage

```
asOf(x, currentDate)
```

Arguments

x	Object of the accrued class.
currentDate	The current date of interest.

Details

A column matrix of data current as of the specified date is returned. The row names are the dates. For dates prior to the currentDateNum minus the maximal accrual lag, the final value is used.

Value

None.

Author(s)

Julie Eaton and Ian Painter

See Also

data.accrued, stackedUploadData, uploadPattern, plot.accrued, plot.summary.accrued, lagHistogram, barcode

Examples

```
data(accruedDataExample) # simulated accrued data
testData <- data.accrued(accruedDataExample)
currentData <- asOf(testData)
plot(currentData)
currentData <- asOf(testData, currentDate=20) # If dates are indexed from 1.
plot(currentData)

testData <- data.accrued(accruedDataExample, start="2010-04-20")
currentData <- asOf(testData, currentDate="2010-07-12")
L = dim(currentData)[1]
yMax = max(currentData[,1], na.rm=TRUE) + 100
num_labels = round(L/4,0)
y_tick_places = seq(0, yMax, by=round(yMax/num_labels,0))
```

```
x_indices = 1:L
x_tick_places = x_indices[(0:num_labels)*3]
x_labels = dimnames(currentData)[[1]][(0:num_labels)*3]
plot(currentData[,1], axes=FALSE, xlab="", ylab="Current counts", ylim=c(0,yMax))
abline(h=0)
abline(v=0)
axis(1, at=x_tick_places, labels=x_labels, las=2 )
axis(2, at=y_tick_places, labels=y_tick_places, las=2 )

## Not run:
library(animation)
# Requires a separate application called "ImageMagik"

NROW = nrow(testData[["data"]])
saveGIF({
for( R in 1:NROW ) {
plot(  c(0, NROW+1),
c(0, max(dat[["data"]], na.rm=TRUE)),
type='n', xlab='day', ylab='count' )
currentValues = asOf(testData, currentDate=R)
points(1:R, currentValues)
lines(1:R, currentValues, col='lightblue')
}
}, cmd.fun=system, interval=0.1)

## End(Not run)
```

---

barcode *Function to create a barcode plot of partially accruing data*

Description

Creates a sparkline graph or "barcode" for partially accruing data showing dates on which data are received. If data are received containing 0 or negative counts, the bars are gray. Otherwise they are dark blue.

Usage

```
barcode(x, ...)
```

Arguments

x	Object of the accrued class containing data to be plotted.
...	Parameters to pass to plot.

Details

A plot is produced showing days on which data were received. Note that this function calls the `stackedUploadData` which should only be used on count data.

Value

None.

Author(s)

Julie Eaton and Ian Painter

See Also

`data.accrued`, `stackedUploadData`, `uploadPattern`, `plot.accrued`, `plot.summary.accrued`, `lagHistogram`, `asOf`

Examples

```
data(accruedDataExample) # simulated accrued data
testData <- data.accrued(accruedDataExample)
barcode(testData)
```

---

data.accrued *Data structure for partially accruing data*

Description

Creates an object of the `accrued` class for use in the plot, summary, and error functions in the accrued package.

Usage

```
data.accrued(data, start = NULL, final = NULL)
```

Arguments

`data`	A matrix of time series data, with rows as consecutive encounter dates and columns as accrual lag.
`start`	The start date. If none is specified it defaults to 1.
`final`	Vector of final counts for each encounter date. If omitted, the final column of `data` is used for the final counts.

Details

The function converts a matrix of patrically accrued counts into an `accrued` object. The value at the ith row and jth column of `data` represent the count for encounter date i as known $j-1$ days after the encounter date. If `final` is not specified, the last column of `data` is used as the final count.

Summary and plot methods exist for objects of this class. Several additional graph types are available, including `barcode`, `uploadPattern`, `laggedTSarray` and `lagHistogram`.

Value

`data.accrued` returns an object of the `accrued` class.

An object of the `accrued` class is a list containing the following components:

`final`	Vector of final counts
`data`	Matrix of counts. The value at the ith row and jth column represent the count for encounter date i as known $j-1$ days after the encounter date.
`start`	List of boolean start variable and start date

Print, summary and plot methods exist for objects of the `accrued` class.

Author(s)

Julie Eaton and Ian Painter

See Also

`print.accrued`, `plot.accrued`, `summary.accrued`, `plot.summary.accrued`

Examples

```
# simulated accrued data
data(accruedDataExample)

# Convert to data.accrued object
dat <- data.accrued(accruedDataExample)

dat$start
# This is 1 because no start date was specified.

dat # uses print.accrued
summary(dat) # uses summary.accrued

plot(summary(dat))  # produces a plot of the summary object
plot(dat) # plots time series of differences between lags


# Convert to data.accrued object, start date now specified.
dat <- data.accrued(accruedDataExample, start="2012-09-15")
```

`errorQuantileSummary` *Summarizes accrual errors*

Description

Summarizes accrual errors by accrual lag using quantiles where the user can specify quantiles.

Usage

```
errorQuantileSummary(x, quantiles = c(0.1, 0.5, 0.9), ...)
```

Arguments

`x`	Object of the `accruedErrors` class.
`quantiles`	A vector of quantiles to use as a summary.
`...`	Parameters to pass to `summary`.

Value

A matrix of error quantiles by lag.

Author(s)

Julie Eaton and Ian Painter

See Also

`data.accrued`, `accruedErrors`, `plot.accruedErrors`

Examples

```
data(accruedDataExample) # simulated accrued data
dat <- data.accrued(accruedDataExample)
errors <- accruedErrors(dat)
errorQuantileSummary(errors, quantiles=seq(0.1,0.9,by=0.1) )
```

`laggedTSarray` *Function to create an array of time series across accrual lags using the running median*

Description

Creates an array of time series graphs–one for each accrual lag–and displays a running median and MAD-based bounds on each graph.

Usage

```
laggedTSarray(x, daysOfHistory, lags, ...)
```

Arguments

`x`	Object of the `accrued` class containing data to be plotted.
`daysOfHistory`	An integer greater or equal to 1. If `NULL`, then defaults to 30. The number of days of previous data used to calculate the running median.
`lags`	A vector of nonnegative integers specifying which lagged time series will be plotted. If NULL, all lags are used.
`...`	Parameters to pass to `plot`.

Details

An array of graphs is produced, with the ith graph showing for each encounter date the cumulative counts received $i-1$ days after the encounter date, with the final graph showing the final counts. A running median and error bars are disiplayed by default. Error bars are calculated as $\pm(2)(MAD)$ (median absolute deviations) from the running median, calculated using the same running window.

Value

None.

Author(s)

Julie Eaton and Ian Painter

See Also

`data.accrued`, `stackedUploadData`, `uploadPattern`, `plot.accrued`, `plot.summary.accrued`, `lagHistogram`, `asOf`

Examples

```
data(accruedDataExample) # simulated accrued data
dat <- data.accrued(accruedDataExample) # Convert to data.accrued object
laggedTSarray(dat)
```

`lagHistogram` *Function to create a a summary histogram of completeness for each lag.*

Description

Creates a column of histograms, the ith histogram showing the distribution of completeness of data $i - 1$ days after the encounter date.

Usage

```
lagHistogram(x, maxLagPlot=NULL, ...)
```

Arguments

`x`	Object of the `accrued` class containing data to be graphed.
`maxLagPlot`	Optional argument specifying the maximum accrual lag reprented in the plot.
`...`	Parameters to pass to `plot`.

Details

This function shows a complete summary of the completeness of the data by lag. The ith row of the column shows a histogram of the completion percent at $i - 1$ days after the encounter date (so that the x-axis represents the completeness proportion in blocks of 0.1). Note that this function should only be used on count data.

Value

None.

Author(s)

Julie Eaton and Ian Painter

See Also

`plot.summary.accrued` displays the mean completion rate for each lag.

Examples

```
library(grid)
data(accruedDataExample)
dat <- data.accrued(accruedDataExample)
lagHistogram(dat, maxLagPlot=7)
```

`plot.accrued` *Plot function for accrued data.*

Description

Creates a stacklag difference plots of partially accruing data.

Usage

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

Arguments

`x`	Object of the `accrued` class containing data to be plotted.
`...`	Parameters to pass to `plot`.

Details

This is only appropriate for count data. This plots differences in counts between successive upload dates. It is the default plot for data of the `accrued` class.

Value

None.

Author(s)

Ian Painter and Julie Eaton

See Also

`data.accrued`, `summary.accrued`, `plot.summary.accrued`, `asOf` Many other plot methods are available for partially accruing data: `laggedTSarray`, `lagHistogram`,

Examples

```
data(accruedDataExample)
plot(data.accrued(accruedDataExample))
```

`plot.accruedErrors` *Function to plot errors in accrued data*

Description

Plots errors in partially accruing data, optionally with quantiles of those errors superimposed.

Usage

```
## S3 method for class 'accruedErrors'
plot(x, withSmoothing = FALSE, quantiles = c(0.1, 0.5, 0.9),
  quantileColors = switch(1+is.null(quantiles), rainbow(length(quantiles)), NULL), ...)
```

Arguments

`x`	Object of the `accruedErrors` class.
`withSmoothing`	Logical. If `TRUE` the the quantiles are smoothed prior to plotting.
`quantiles`	A vector of quantiles to be plotted.
`quantileColors`	Colors to use for quantiles.
`...`	Parameters to pass to `plot`.

Details

Produces a scatter plot of errors for each lag value, with the x-axis representing lag and the y-axis the error values. The x-value are jittered prior to plotting. If quantiles is specified, the corresponding quantiles of the errors will be superimposed on the plot.

Value

None.

Author(s)

Julie Eaton and Ian Painter

See Also

`accruedErrors`, `data.accrued`, `summary.accruedErrors`

Examples

```
data(accruedDataExample) # simulated accrued data
dat <- data.accrued(accruedDataExample)
errors <- accruedErrors(dat)
plot(errors)
```

`plot.summary.accrued` *Plot function for objects of the* `summary.accrued` *class.*

Description

Plots summarized partially accruing data objects of the `summary.accrued` class.

Usage

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

Arguments

`x`	Object of class `summary.accrued`.
`...`	Parameters to pass to `plot`.

Details

`plot.summary.accrued` generates a plot of the an object of the `summary.accrued` class.

Value

None.

Author(s)

Ian Painter and Julie Eaton

Examples

```
data(accruedDataExample)
testData <- data.accrued(accruedDataExample)
summ_dat = summary(testData) # summary.accrued object
plot(summ_dat)  # plot.summary.accrued
```

`print.accrued` *Print function for objects of the* `accrued` *class.*

Description

Prints partially accruing data objects of the `accrued` class.

Usage

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

Arguments

`x`	Object of the `accrued` class.
`...`	Parameters to pass to `print.default` or `plot`.

Details

`print.accrued` displays a matrix of counts with columns representing the lag and rows the encounter date.

Value

None.

Author(s)

Ian Painter and Julie Eaton

Examples

```
data(accruedDataExample)
testData <- data.accrued(accruedDataExample)

testData # print.accrued
```

print.summary.accrued *Print function for objects of the* summary.accrued *class.*

Description

Displays as text summarized partially accruing data objects of the summary.accrued class.

Usage

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

Arguments

x	Object of class summary.data.accrued.
...	Parameters to pass to print.

Details

print.summary.accrued prints to the screen the contents of a summary.accrued object, namely, the counts consisting of the percentage of days on which an upload was recevied for each lag, the mean proportion of counts received by each lag, and the mean counts received by each lag.

Value

None.

Author(s)

Ian Painter and Julie Eaton

Examples

```
data(accruedDataExample)
testData <- data.accrued(accruedDataExample)

summ_dat = summary(testData) # summary.accrued object
summ_dat # prints the object
```

stackedUploadData *Subfunction used by the upload pattern and barcode plot functions*

Description

This subfunction organizes the partially accruing data into a matrix that includes accrual lag as a variable.

Usage

```
stackedUploadData(x)
```

Arguments

x	Object of the accrued class.

Details

This should only be used on accrued data that are counts. No error will be thrown in the case of non-count data, but the results will not make sense.

Value

A data frame with the columns consisting of event date (encounter date), record date (upload date), accrual lag (lag), counts (time series value), number added.

Author(s)

Julie Eaton and Ian Painter

See Also

data.accrued, uploadPattern, barcode

summary.accrued *Print and summary functions for objects of the* accrued *class.*

Description

Functions to display and summarize (as text or as a plot) partially accruing data objects of the accrued class.

Usage

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

Arguments

object	Object of the accrued class.
...	Parameters to pass to print.default or plot.

Details

summary.accrued creates a summary object (of class summary.accrued) of the counts consisting of the percentage of days on which an upload was recevied for each lag, the mean proportion of counts received by each lag, and the mean counts received by each lag.

Value

summary.accrued invisibly returns an object of the summary.accrued class, consisting of a list with the following components:

upload.prop	Vector containing the percentage of days on which an upload was recevied for each lag
mean.prop	Vector containing the mean proportion of counts received by each lag
mean.total	Vector containing the mean counts received by each lag
q1.prop	Vector containing the 1st quartile of the proportion of counts received by each lag
q2.prop	Vector containing the 2nd quartile of the proportion of counts received by each lag
q3.prop	Vector containing the 3rd quartile of the proportion of counts received by each lag

Author(s)

Ian Painter and Julie Eaton

Examples

```
data(accruedDataExample)
testData <- data.accrued(accruedDataExample)
summ_dat = summary(testData) # summary.accrued object
```

---

summary.accruedErrors *Summarizes accrual errors*

Description

Default function to summarize errors by accrual lag using quantiles.

Usage

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

Arguments

object	Object of the accruedErrors class.
...	Parameters to pass to summary.

Value

A matrix of error quantiles by lag.

Author(s)

Julie Eaton and Ian Painter

See Also

data.accrued, accruedErrors, plot.accruedErrors, The following function allows the user to specify a vector of quantiles: errorQuantileSummary

Examples

```
data(accruedDataExample) # simulated accrued data
testData <- data.accrued(accruedDataExample)
errors <- accruedErrors(testData)
summary(errors)
```

---

uploadPattern *Function to create an upload pattern plot*

Description

Creates an upload pattern plot for partially accruing data showing dates at which new data were received for each encounter date.

Usage

```
uploadPattern(x, horizontal = FALSE, ...)
```

Arguments

x	An object of the accrued class.
horizontal	logical value; if TRUE a compact horizontal plot is generated, if FALSE a diagonal upload plot is generated.
...	Parameters to pass to plot.

Details

A diagonal upload pattern plot displays for each encounter date (x-axis) whether or not new data was received at each upload date (y-axis). A horizontal upload pattern plot displays for each encounter date (x-axis) whether or not new data were received y days after the encounter date. Note that this function calls the "stackedUploadData" which should only be used on count data. If data were received containing 0 or negative counts, pixels are colored in gray. Otherwise they are dark blue.

Value

None.

Author(s)

Julie Eaton and Ian Painter

See Also

stackedUploadData, barcode, plot.accrued, plot.summary.accrued, lagHistogram

Examples

```
data(accruedDataExample)
testData <- data.accrued(accruedDataExample)
uploadPattern(testData) # Diagonal plot (default)
dev.new(height = 1.5, width = 10) # Horizontal plot - best short and wide
uploadPattern(testData, horizontal = TRUE)
```

Index