Package 'tican'

April 2, 2025

Type Package

Title Plot and Analyse Time-Intensity Data

Version 1.0.2

Maintainer Samuel James Tingle <samjamestingle@gmail.com>

Description Plots and analyzes time-intensity curve data, such as data from (contrast-enhanced) ultrasound. Values such as peak intensity, time to peak and area under the curve are calculated.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

URL https://github.com/sjtingle/tican

BugReports https://github.com/sjtingle/tican/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Samuel James Tingle [aut, cre]

Repository CRAN

Date/Publication 2025-04-02 08:10:02 UTC

Contents

`tic_analyse` *Plot and analyse time-intensity data*

Description

This function plots and analyzes time-intensity data, such as data from (contrast-enhanced) ultrasound cineloops. Peak intensity and time to peak intensity are calculated from a smoothed curve through the data (loess smoother). Area under the curve is calculated from the raw data using the trapezium method for integration. Time to peak proportion (for example time to 90 percent of peak) can also be calculated.

Usage

```
tic_analyse(
  data,
  timevar,
  intensityvar,
  loess.span = 0.1,
  AUCmax = NULL,
  peakproportion = NULL,
  plotresult = TRUE,
  ...
)
```

Arguments

`data`	A dataframe with time and intensity values as columns.
`timevar`	A character string (in quotes) with the dataframe column name for the time variable.
`intensityvar`	A character string (in quotes) with dataframe column name for the intensity variable.
`loess.span`	A number between 0 and 1, with larger values resulting in a smoother curve.
`AUCmax`	A number - the maximum time that area under the curve is measured until.
`peakproportion`	A number between 0 and 1 which is used in the time to peak proportion calculations.
`plotresult`	TRUE or FALSE to determine whether a plot of the results is generated.
`...`	Additional arguments to be passed into the loess() function.

Details

A plot of the data is generated and a dataframe with the results is returned.

Value

A dataframe with the results. Depending on the plotresult argument can also return a plot of the smoothed curve.

Examples

```
# Example usage:

# Generating simulated data
set.seed(123)
example_data <- data.frame(time = seq(0, 82, by = 0.25))
random_vals <- sample(1:10, nrow(example_data), replace = TRUE)
example_data$regionA_intensity <- log(example_data$time + 1) * 50 -
  example_data$time * 2 + random_vals
example_data$regionB_intensity <- log(example_data$time + 7, base = 10) *
  80 - example_data$time * 1.5 + random_vals

# Example with defaults:

tic_analyse(data = example_data, timevar = "time", intensityvar = "regionA_intensity")

# Example with additional arguments:

tic_analyse(data = example_data, timevar = "time", intensityvar = "regionA_intensity",
                 loess.span = 0.1, AUCmax = 30, peakproportion = 0.9, plotresult = TRUE)
```

`tic_auc` *Return area under curve from time-intensity data*

Description

This function returns area under curve from raw time intensity curves using the trapezoid method. It is recommended that plotresult is set to TRUE in the first instance to visually confirm the analysis.

Usage

```
tic_auc(data, timevar, intensityvar, AUCmax = NULL, plotresult = FALSE)
```

Arguments

`data`	A dataframe with time and intensity values as columns.
`timevar`	A character string (in quotes) with the dataframe column name for the time variable.
`intensityvar`	A character string (in quotes) with dataframe column name for the intensity variable.
`AUCmax`	A number - the maximum time that area under the curve is measured until.
`plotresult`	TRUE or FALSE to determine whether a plot of the results is generated.

Details

A plot of the data is generated and a dataframe with the results is returned.

Value

Area under the curve. Depending on the plotresult argument can also return a plot of the raw data.

Examples

```
# Example usage:

# Generating simulated data
set.seed(123)
example_data <- data.frame(time = seq(0, 82, by = 0.25))
random_vals <- sample(1:10, nrow(example_data), replace = TRUE)
example_data$regionA_intensity <- log(example_data$time + 1) * 50 -
  example_data$time * 2 + random_vals
example_data$regionB_intensity <- log(example_data$time + 7, base = 10) *
  80 - example_data$time * 1.5 + random_vals

# Example with defaults:

tic_auc(data = example_data, timevar = "time", intensityvar = "regionA_intensity")

# Example with additional arguments:

tic_auc(data = example_data, timevar = "time", intensityvar = "regionA_intensity",
                  AUCmax = 30, plotresult = TRUE)
```

---

ttpeak *Return time to peak from time-intensity data*

Description

This function returns the time to peak from time-intensity curve data. Raw data is smoothed using a loess smoother, and the time of peak is returned. Time to a specified proportion of the peak (e.g. time to 90% of peak) can be calculated. It is recommended that plotresult is set to TRUE in the first instance to visually deconfirm the analysis.

Usage

```
ttpeak(
  data,
  timevar,
  intensityvar,
  loess.span = 0.1,
  peakproportion = NULL,
  plotresult = FALSE,
  ...
)
```

Arguments

data	A dataframe with time and intensity values as columns.
timevar	A character string (in quotes) with the dataframe column name for the time variable.
intensityvar	A character string (in quotes) with dataframe column name for the intensity variable.
loess.span	A number between 0 and 1, with larger values resulting in a smoother curve.
peakproportion	A number between 0 and 1 which is used in the time to peak proportion calculations. If a number is entered the function will return the time to peak proportion.
plotresult	TRUE or FALSE to determine whether a plot of the results is generated.
...	Additional arguments to be passed into the loess() function.

Value

The time at which the loess curve is at its peak. Depending on the plotresult argument can also return a plot of the smoothed curve.

Examples

```
# Example usage:

# Generating simulated data
set.seed(123)
example_data <- data.frame(time = seq(0, 82, by = 0.25))
random_vals <- sample(1:10, nrow(example_data), replace = TRUE)
example_data$regionA_intensity <- log(example_data$time + 1) * 50 -
  example_data$time * 2 + random_vals
example_data$regionB_intensity <- log(example_data$time + 7, base = 10) *
  80 - example_data$time * 1.5 + random_vals

# Example with defaults:

ttpeak(data = example_data, timevar = "time", intensityvar = "regionA_intensity")

# Example with additional arguments:

ttpeak(data = example_data, timevar = "time", intensityvar = "regionA_intensity",
                  loess.span = 0.1, peakproportion = 0.9, plotresult = TRUE)
```

Index