Package 'robustToxicities'

November 22, 2017

Type Package

Title Toxicity tables

Version 1.1.0

Date 2017-11-22

Author Peter Dutton

Maintainer Peter Dutton <peter.dutton@csm.ox.ac.uk>

Description Creates publication ready table and graphs from time based toxicity data. Performs some built in data cleaning actions.

License GPL-2

LazyData true

Imports methods, stringr, prettyTables

Recommends ReporteRs

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

`CreateTimeDividers` *Create period divides as a function of time from start of toxicity window*

Description

Takes a set of numeric time boundaries from baseline and creates the corresponding columns in patientData, periodDividerCols and periodDividerLabels. This returns the updated robustToxicitiesClass object.

Usage

```
CreateTimeDividers(rt, timeBoundaries, labelUnits = "days")
```

Arguments

`rt`	Object of class robustToxicitiesClass
`timeBoundaries`	a numeric vector of times from the start of toxicity window
`labelUnits`	days, weeks or months used to automatically generate labels in the form of from-to.

`DefaultToxicityOptions` *Default* `toxicityOptions`.

Description

Default `toxicityOptions`.

Usage

```
DefaultToxicityOptions()
```

FT_ToxTable *FlexTable wrapper for toxicity tables*

Description

These functions act as wrapper functions to create ConstructFlexTable objects.

Usage

```
FT_ToxTable_summary(rt, tble = NULL)

FT_ToxTable_cycle(rt, cycles = "all", tble = NULL)

FT_ToxTable_category(rt, cycles = "all", tble = NULL)
```

Arguments

rt	robustToxicity object.
tble	Optionally pass a pre computed table into the wrapper instead of computing it here. Note if columns don't match what is expected this is likely to fail. May be usefull for dropping some rows, or saving time rerunning the table generator function for large data sets. This functions return the ConstructFlexTable object from the prettyTables package. To create the FlexTable you can call object$GetTable() on the returned object. This can then be added to a word document using addFlexTable from the ReporteRs package.
cycles	The cycle column names, or index in rt@periodDividerCols of the cycles to tabulate. May also be "all" to use all cycles

Functions

- FT_ToxTable_summary: A wrapper for ToxTable_cycle
- FT_ToxTable_cycle: A wrapper for ToxTable_cycle
- FT_ToxTable_category: A wrapper for ToxTable_category

See Also

ToxTable_summary, ToxTable_cycle, ToxTable_category

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
```

```
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)

# Look for queries. Note: must be called before running any
# of the functions on this class.
rt = QueryRobustToxicities(rt)


############################################################
# flexTable summary
ft = FT_ToxTable_summary(rt)
# returns an object of class ConstructFlexTable for additional formatting and editing
ft$GetTable()


############################################################
# flexTable worst grade of each adverse event by patient
ft = FT_ToxTable_cycle(rt)
# returns an object of class ConstructFlexTable for additional formatting and editing
ft$GetTable()

############################################################
# flexTable worst grade of each category by patient
ft = FT_ToxTable_category(rt)
# returns an object of class ConstructFlexTable for additional formatting and editing
ft$GetTable()

############################################################
# The table generated by ToxTable_summary, ToxTable_cycle or ToxTable_category may also
# be passed as an argument. This provides a convinent oportunity to edit the data
tble = ToxTable_summary(rt)
# Edits to tble can go here
ft = FT_ToxTable_summary(rt, tble = tble)
# returns an object of class ConstructFlexTable for additional formatting and editing
ft$GetTable()
```

`QueryRobustToxicities` *Query RobustToxicities*

Description

A function which checks the provided data

Usage

```
QueryRobustToxicities(rt)
```

Arguments

rt	The robustToxicitiesClass object

Value

An S4 object of class robustToxicitiesClass. The queries are stored as a data.frame in the slot queries (rt@queires)

---

robustToxicitiesClass *The robustToxicitiesClass*

Description

This is core object of this package. This object stores the original dataset as well as the automatically cleaned dataset and a list of notes and queries generated when cleaning the dataset. A list of options is also provided to store plot and tabulation options and provide additional metadata. Finally treatment and cycle labels are also required.

Slots

`toxData` The toxicitydataset

`patientData` Patient level data

`patidCol` Column name for the participant identifier

`treatmentCol` Column name for the treatment

`toxCategoryCol` Column name for aderse event category

`toxNameCol` Column name for adverse event name

`toxGradeCol` Column name for the adverse event grade

`dateOfStartTox` Column name for date of adverse event start or change in grade

`dateOfEndTox` Column name for date of adverse event end or change in grade

`dateOfStartOfToxWindow` Column name for date of study entry (eg registration)

`dateOfEndOfToxWindow` Column name for the end of the time window for the particitant to be observed for toxicities

`periodDividerCols` Column names for date dividing times into periods or cycles (optional)

`periodDividerLabels` Display names for data periodDividerCols

`treatmentCodes` Codes which match the values in treatmentCol

`treatmentLabels` Labels to used instea of the treatment codes

`queries` A data.frame containing all the queries and note generated when loading the data

`options` An s4 object of class `toxicityOptions-class` containing options and metadata for the files.

`wasQueried` Logical detailing if queries were run on this object.

rt_patientData *Toxicity data*

Description

An example dataset of dummy data used to create examples for this package.

Usage

```
data("rt_patientData")
data("rt_toxicityData")
```

Format

Two data.frames which are linked by patientNo. rt_patientData contains data for 7 patients with one row per patient. rt_toxicityData contains 29 toxicities for the patients in rt_patientData.

SetupRobustToxicities *robustToxicitiesClass generator*

Description

The robustToxicities package aims to make creating publication ready table and graphs from time based toxicity data easy. The package also performs some built in data cleaning actions.

Usage

```
SetupRobustToxicities(toxData, patientData, patidCol, treatmentCol = NULL,
  toxCategoryCol, toxNameCol, toxGradeCol, dateOfStartOfToxWindow,
  dateOfStartTox, dateOfEndTox, dateOfEndOfToxWindow,
  periodDividerCols = character(0), periodDividerLabels = character(0),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)
```

Arguments

toxData	The toxicity level data set
patientData	The patient level data
patidCol	Column name for the participant identifier
treatmentCol	Column name for the treatment. Will be created if not provided
toxCategoryCol	Column name for aderse event category
toxNameCol	Column name for adverse event name
toxGradeCol	Column name for the adverse event grade
dateOfStartOfToxWindow	Column name for date of study entry (eg registration)
dateOfStartTox	Column name for date of adverse event start or change in grade
dateOfEndTox	Column name for date of adverse event end or change in grade

`dateOfEndOfToxWindow`
: Column name for the end of the time window for the particitant to be observed for toxicities (optional)

`periodDividerCols`
: Column names for date dividing times into periods or cycles (optional)

`periodDividerLabels`
: Display names for data periodDividerCols param treatmentLabels A vector of treatment labels

`treatmentCodes`
: Levels of treatment in the treatmentCol

`treatmentLabels`
: What to name each treatment in output tables

`options`
: Optional. An object of class toxicityOptions. The easiest place to start is with `DefaultToxicityOptions()`. See `DefaultToxicityOptions` for more details on options.

Details

Run this to create an object of class `robustToxicitiesClass`. Then run it through `QueryRobustToxicities` to check for errors before creating tables and graphs of the data.

This function takes two linked data.frames. A one row per patient, patient level data.frame (`patientData`) and a one row per toxicity data.frame of toxicities (`toxData`). The remaining parameters tell the package where the columns which are required to create the plots and graphs are. There is also an options class (`toxicityOptions-class`) which can be edited from the default later. Since the `robustToxicitiesClass` object is an s4 class slots are accessed using the @ symbol.

Value

An object of class `robustToxicitiesClass`

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)

# Look for queries. Note: must be called before running any
# of the functions on this class.
```

```
rt = QueryRobustToxicities(rt)

############################################################
# Table Examples.
############################################################
# Summary, worst grade by cycle
ToxTable_summary(rt)

# ReporteRs flextable version
ft = FT_ToxTable_summary(rt)
ft$GetTable()


# Worst grade by patient for each toxicity type
ToxTable_cycle(rt)

# ReporteRs flextable version
ft = FT_ToxTable_cycle(rt)
ft$GetTable()

# Worst grade by category
ToxTable_category(rt)

# ReporteRs flextable version
ft = FT_ToxTable_category(rt)
ft$GetTable()

# Alternative style for worst grade by category
ToxTable_categories(rt)

############################################################
# Plot Examples
############################################################
ToxPlot_byToxicity(rt)

# With causality
# Not provided so generate some
rt@toxData$causality1 = sample(1:5,28, replace = TRUE)
rt@toxData$causality2 = sample(1:5,28, replace = TRUE)

causality = ToxPlot_causalityInfo(
  columns = c("causality1","causality2"),
  names = c("A","BA"),
  width = 1.5,
  pch = c(NA,NA,4,8,16),
  cex = 1.2)


ToxPlot_byToxicity(rt,
                   causality = causality)



ToxPlot_byPatient(rt)

ToxPlot_byCycle(rt)
```

```
############################################################
# Alternative specification to cycles
############################################################
# wrapper for toxPlot_byCycle adding alternative boundaries
timeBoundaries = c(0,21,42,63,84,105,126)
rt2 = CreateTimeDividers(rt, timeBoundaries)
ToxTable_summary(rt2)


ToxPlot_byTime(rt, timeBoundaries = timeBoundaries)
```

`toxicityOptions` *Toxicity Options class*

Description

An object containing all the key options for creating the toxicity tables. This is a slot in the `robustToxicitiesClass`. This class can be generated using `DefaultToxicityOptions()` and updated manually.

Slots

`displayNotes` A logical value used by `robustToxicities` to determine whether or not to print notes

`toxTable_cycle_tabulationMethod` One of "worst" or "all" determining if all toxicity changes are counted or only the worst reported grade in a time period

`toxTable_tabulationPercent` A logical value used to determine if toxicity tables should report counts (FALSE, default) or percentages (TRUE)

`toxTable_tabulationZeros` A logical value used to determine if zeros should be included, default TRUE

`toxTable_cumulativeGrades` A logical value used to determine whether toxicity grades should be reported cumulatively or not, default TRUE

`toxTable_discardToxAtStudyEntry` A logical value used to determine if toxicities reported at baseline should be reported or not, default FALSE

`toxTable_mergeGrades` Grades to merge in the tables. Columns are seperated by "|" and merged values are seperated by ",". "n"

`toxTable_cycle_toxicityOrder` What order should the data be returned in. "c" ordered by categories and then adverse events. "a" ordered by adverse events. "n" ordered by number of adverse events. The n option can be followed by a number to denote the minimum grade to use for sorting. e.g. "n3" will order by grades 3-5 and then 1-5 for ties within grades 3-5.

ToxPlot_byCycle *Summary plot of toxicities by cycle*

Description

This plot summarises the proportion of patients having an adverse event in each time period as defined by periodDividerCols in the robustToxicitiesClass.

Usage

```
ToxPlot_byCycle(rt, gradeRequired = 1, col = c("blue", "red"),
  tableSpace = 0.1, las = 1, legendPosition = "right", add = FALSE)
```

Arguments

rt	An object of class robustToxicities
gradeRequired	Only include adverse events with at least this grade
col	A vector of colours to plot each arm with
tableSpace	A parameter to assist in vertical row spacing the table appropriately
las	numeric in 0,1,2,3; the style of axis labels. 0: always parallel to the axis, 1: always horizontal [default], 2: always perpendicular to the axis, 3: always vertical
legendPosition	The location to place the legend see legend for details
add	TRUE/FALSE whether to add to an existing plot or start a new one

See Also

ToxPlot_byPatient, ToxPlot_byToxicity, ToxPlot_byTime

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)
```

```
# Look for queries. Note: must be called before running any
# of the functions on this class.
rt = QueryRobustToxicities(rt)

#########################################################
par(mar = c(4,6,3,2))
ToxPlot_byCycle(rt)

#########################################################
# subset to a specific set of adverse events
rt@toxData$ass_TRUE = rt@toxData$toxicity == "Sore Throat"
par(mar = c(4,6,3,2))
ToxPlot_byCycle(rt)
```

`ToxPlot_byPatient` *Plot patients worst grade over time*

Description

This function plots the worst grade adverse event for each patient over time.

Usage

```
ToxPlot_byPatient(rt, rowID_range = NULL, plot = TRUE,
  plotLeftSideOption = "treatment", xlim = c(-7, 60), xlab = character(0),
  plotCycleLength = 21, plotXLegendScale = "days", permitMarSet = TRUE,
  events = list(), offsetEvent = NULL)
```

Arguments

`rt`	an object of class robustToxicities
`rowID_range`	optional, a length 2 vector detailing the minimum and maximum row to plot
`plot`	whether to plot the graph or return the number of rows to plot
`plotLeftSideOption`	What to display on right axis. Options are: "treatment", "patid" or "both". Default is "treatment"
`xlim`	Range to plot on xaxis. Default is c(-7,60)
`xlab`	xaxis title / label
`plotCycleLength`	Cycle length is used to add greater highlights to vertical lines. Default is 21
`plotXLegendScale`	What scale to use on xaxis. Options are "days","weeks","months". Default is "days"
`permitMarSet`	Allow the function to set the mar for the plot
`events`	a list of Objects of type eventInfo.
`offsetEvent`	the name of a column in patientData to use as time 0. If not provided the start of assessment date is used

Value

This plot function return the number of row of unique toxicities * patients. This assists in computing optimal size for saved graphs.

#' @seealso ToxPlot_byToxicity, ToxPlot_byTime, ToxPlot_byCycle

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)

# Look for queries. Note: must be called before running any
# of the functions on this class.
rt = QueryRobustToxicities(rt)

############################################################
ToxPlot_byPatient(rt)


############################################################
# Subset to a range. Usefull for plotting over a number of figures if there
# are lots of adverse events
ToxPlot_byPatient(rt, rowID_range = c(1,3), plotLeftSideOption = "both")

############################################################
# subset to a specific set of adverse events
rt@toxData$ass_TRUE = rt@toxData$toxicity == "Headache"
ToxPlot_byPatient(rt)


############################################################
# Add event data
rt@toxData$ass_TRUE = TRUE

event_EOT = ToxPlot_eventInfo(
  columns = c("end_of_treatment_date"),
  label = c("End Of Treatment"),
  lwd = 4,
  col = c("blue")
```

```
)

event_EOA = ToxPlot_eventInfo(
  columns = c("end_of_assessment_date"),
  label = c("End Of Assessment"),
  lwd = 4,
  col = c("green")
)
ToxPlot_byPatient(rt)
ToxPlot_byPatient(rt,
                  xlim = c(-7, 100),
                  events = list(event_EOT, event_EOA))

#########################################################
# Change offset event
event_SOT = ToxPlot_eventInfo(
  columns = c("Registration_date"),
  label = c("Registration Date"),
  lwd = 4,
  col = c("orange")
)

ToxPlot_byPatient(rt,
                  xlim = c(-67, 40),
                  xlab = "Days from end of treatment",
                  events = list(event_SOT, event_EOT, event_EOA),
                  offsetEvent = "end_of_treatment_date")
```

`ToxPlot_byTime` *Summary plot of toxicities over time*

Description

This is a wrapper function for `ToxPlot_byCycle` which takes timeBoundaries.

Usage

```
ToxPlot_byTime(rt, gradeRequired = 1, timeBoundaries, xlab = "days",
  col = c("blue", "red"), tableSpace = 0.1, las = 1,
  legendPosition = "right", add = FALSE)
```

Arguments

`rt`	An object of class robustToxicities
`gradeRequired`	Only include adverse events with at least this grade
`timeBoundaries`	A vector of times from the dateOfStartOfToxWindow
`xlab`	The xaxis label. "days","weeks" and "months" are converted to "Time from registration (days)" etc.
`col`	A vector of colours to plot each arm with
`tableSpace`	A parameter to assist in vertical row spacing the table appropriately
`las`	numeric in 0,1,2,3; the style of axis labels. 0: always parallel to the axis, 1: always horizontal [default], 2: always perpendicular to the axis, 3: always vertical

legendPosition	The location to place the legend see legend for details
add	TRUE/FALSE whether to add to an existing plot or start a new one

See Also

ToxPlot_byPatient, ToxPlot_byToxicity, ToxPlot_byCycle

ToxPlot_byToxicity *Plot toxicities over time*

Description

This function plots the worst grade of each toxicity over time. There should be no overlap between toxicities but in the case that there is the wors grade is given priority.

Usage

```
ToxPlot_byToxicity(rt, rowID_range = NULL, plotNow = TRUE,
  plotLeftSideOption = "treatment", xlim = c(-7, 60), xlab = character(0),
  plotCycleLength = 21, plotXLegendScale = "days", permitMarSet = TRUE,
  causality = NULL, events = list(), offsetEvent = NULL)
```

Arguments

rt	an object of class robustToxicities
rowID_range	optional, a length 2 vector detailing the minimum and maximum row to plot
plotNow	whether to plot the graph or return the number of rows to plot
plotLeftSideOption	What to display on right axis. Options are: "treatment", "patid" or "both". Default is "treatment"
xlim	Range to plot on xaxis. Default is c(-7,60)
xlab	xaxis title / label
plotCycleLength	Cycle length is used to add greater highlights to vertical lines. Default is 21
plotXLegendScale	What scale to use on xaxis. Options are "days","weeks","months". Default is "days"
permitMarSet	Allow the function to set the mar for the plot
causality	Adds causality columns to the plot on the righthand side. This must be an object of type causalityInfo-class
events	a list of Objects of type eventInfo.
offsetEvent	the name of a column in patientData to use as time 0. If not provided the start of assessment date is used

Value

This plot function return the number of row of unique toxicities * patients. This assists in computing optimal size for saved graphs.

See Also

ToxPlot_byPatient, ToxPlot_byTime, ToxPlot_byCycle

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)

# Look for queries. Note: must be called before running any
# of the functions on this class.
rt = QueryRobustToxicities(rt)

############################################################
ToxPlot_byToxicity(rt)

############################################################
# Subset to a range. Usefull for plotting over a number of figures if there
# are lots of adverse events
ToxPlot_byToxicity(rt, rowID_range = c(1,7))

############################################################
# subset to a specific set of adverse events
rt@toxData$ass_TRUE = rt@toxData$Treatment == "Placebo"
ToxPlot_byToxicity(rt)

############################################################
# Add causality data
rt@toxData$ass_TRUE = TRUE

# With causality
# Not provided so generate some for illustrative purposes
rt@toxData$causality1 = sample(1:5,28, replace = TRUE)
rt@toxData$causality2 = sample(1:5,28, replace = TRUE)

causality = ToxPlot_causalityInfo(
  columns = c("causality1","causality2"),
  names = c("A","B"),
  width = 1.5,
```

```
  cex = 1.2)

ToxPlot_byToxicity(rt,
                   causality = causality)


#########################################################
# Add event data

event_EOT = ToxPlot_eventInfo(
  columns = c("end_of_treatment_date"),
  label = c("End Of Treatment"),
  lwd = 4,
  col = c("blue")
)

event_EOA = ToxPlot_eventInfo(
  columns = c("end_of_assessment_date"),
  label = c("End Of Assessment"),
  lwd = 4,
  col = c("green")
)

ToxPlot_byToxicity(rt,
                   causality = causality,
                   xlim = c(-7, 100),
                   events = list(event_EOT, event_EOA))


#########################################################
# Change offset

event_SOT = ToxPlot_eventInfo(
  columns = c("Registration_date"),
  label = c("Registration Date"),
  lwd = 4,
  col = c("orange")
)

ToxPlot_byToxicity(rt,
                   causality = causality,
                   xlim = c(-67, 40),
                   xlab = "Days from end of treatment",
                   events = list(event_SOT, event_EOT, event_EOA),
                   offsetEvent = "end_of_treatment_date")
```

`ToxPlot_causalityInfo` *causalityInfo class*

Description

Stores causality data to pass to toxPlot_byToxicity in the causality parameter. The columns containing the data in toxData should hold numeric data.

Usage

```
ToxPlot_causalityInfo(columns, names = character(0), width = 1.5,
  pch = c(NA, NA, 4, 8, 16), cex = 1, col = 1,
  labels = c("Possibly related", "Probably related", "Definitely related"))
```

Arguments

columns	A vector of column names for toxData
names	A short identifier to place at the top of each column
width	The width to provide on the plot for each causality. This is on the scale of days on the plot
pch	The pch symbol to use for each level of causality
cex	The size of each symbol
col	The colour to use for each level of causality
labels	Labels for the non NA pch levels, used in the legend

Details

The default values expect a number between 1 and 5 and plot a symbol for values 3,4 and 5. This can be changed using pch.

Value

An object of class causalityInfo containing the same slots as paramters taken by ToxPlot_causalityInfo

---

ToxPlot_eventInfo *eventInfo class*

Description

Stores event data to pass to toxPlot_byToxicity in the causality parameter. The columns containing the data in patientData should hold date of class "Date".

Usage

```
ToxPlot_eventInfo(columns, label = columns[1], lwd = 4, col = "grey")
```

Arguments

columns	A vector of columns contain the event
label	A short name for the event
lwd	Line width
col	The colour to use for each event

Value

An object of class eventInfo containing the same slots as paramters taken by ToxPlot_causalityInfo

ToxTable_categories *Generate a prettyTable of worst toxicity by category.*

Description

This is a wrapper for the table_values function in prettyTables for worst grade by category. Note that there is an alternative version of this table `ToxTable_category` which formats similarly to all the other objects in this package.

Usage

```
ToxTable_categories(rt, categoryList = NULL, strata.count = TRUE,
  overall = TRUE, count = "n", round = 0)
```

Arguments

`rt`	RobustToxicitiesClass object
`categoryList`	A list of categories. Default is all. A subset is selected by changing this value.
`strata.count`	TRUE/FALSE for displaying strata counts at the top of each column
`overall`	TRUE/FALSE for including an overall column
`count`	"n","miss" or "none" providing the counts, missing values or omitting for each column for numeric variables
`round`	A value or vector for the number of significant figures to report the data to categoryList can be replaced by other column names if the standard categorisation is not being used. For more flexibility you can view the code by typing table_tox_categories. Available methods and values for **Type**:

"miqr"	median (Q25,Q75)
"miqrr"	median (Q25,Q75)[min,max]
"mrng"	median (Q0,Q100)
"avsd"	mean (sd)
"avci"	mean (confidence interval)
"st"	count
"str"	count/total
"stp"	count (percent)
"strp"	count/total (percent)

Value

Returns a data.frame

ToxTable_category *Tabulation of toxicity categories in a cycle*

Description

Returns a toxicity table with the requested data according to the ass_TRUE column, for the cycles requested. Note this is a wrapper function which essentially replaces the toxicity names with the categories, updates the toxID's and then calls `ToxTable_cycle` on the categories. This function could be used as a template for summarising other data.

Usage

```
ToxTable_category(rt, cycles = "all")
```

Arguments

`rt`	an object of class robustToxicities
`cycles`	The cycle column names, or index in rt@rt@periodDividerCols of the cycles to tabulate. May also be "all" to use all cycles

Details

This function acts as a wrapper for `ToxTable_cycle` to get category data instead of toxicity level data.

`ToxTable_cycle` *Tabulation of toxicities in a cycle*

Description

Returns a toxicity table with the requested data according to the ass_TRUE column and for the cycles requested.

Usage

```
ToxTable_cycle(rt, cycles = "all")
```

Arguments

`rt`	an object of class robustToxicities
`cycles`	The cycle column names, or index in rt@periodDividerCols of the cycles to tabulate. May also be "all" to use all cycles. "all" is the default.

Value

data.frame

See Also

`toxicityOptions-class`

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)

# Look for queries. Note: must be called before running any
# of the functions on this class.
rt = QueryRobustToxicities(rt)

############################################################
# default table
ToxTable_cycle(rt)

############################################################
# We can change some of the options which are used to generate this table
# or subset the data using the ass_TRUE column in rt@toxData
# with percentages
rt@options@toxTable_tabulationPercent = TRUE
ToxTable_cycle(rt)
# A data.frame is returned if further manipulation is required

# without zeros
rt@options@toxTable_tabulationZeros = FALSE
ToxTable_cycle(rt)

# only toxicities with category "Nervous systen disorders"
rt@toxData$ass_TRUE = rt@toxData$category == "Nervous system disorders"
ToxTable_cycle(rt)
############################################################
# flexTable default formatting
ft = FT_ToxTable_cycle(rt)
# returns an object of class ConstructFlexTable for additional formatting and editing
ft$GetTable()
```

ToxTable_summary *Summary table of toxicities*

Description

Returns a summary toxicity table with the requested data according to the ass_TRUE column.

Usage

```
ToxTable_summary(rt)
```

Arguments

rt	an object of class robustToxicities

Value

data.frame

See Also

toxicityOptions-class

Examples

```
# Patient Level Data
data("rt_patientData")
# Toxicity Level Data
data("rt_toxicityData")


# Run the setup command passing in all the column names.
rt = SetupRobustToxicities(
  toxData = rt_toxicityData,
  patientData = rt_patientData,
  patidCol = "patientNo", treatmentCol = "Treatment",
  toxCategoryCol = "category", toxNameCol = "toxicity",
  toxGradeCol = "grade", dateOfStartOfToxWindow = "Registration_date",
  dateOfStartTox = "ae_onset_date", dateOfEndTox = "ae_resolve_date",
  dateOfEndOfToxWindow = "end_of_assessment_date",
  periodDividerCols = c("Registration_date", "Cycle_1_date","Cycle_2_date",
                        "Cycle_3_date", "Cycle_4_date", "Cycle_5_date", "Cycle_6_date"),
  periodDividerLabels = c("Pre treatment", "Cycle 1","Cycle 2",
                          "Cycle 3","Cycle 4","Cycle 5", "Cycle 6"),
  treatmentCodes = NULL, treatmentLabels = NULL, options = NULL)

# Look for queries. Note: must be called before running any
# of the functions on this class.
rt = QueryRobustToxicities(rt)

############################################################
# default table
ToxTable_summary(rt)

############################################################
# We can change some of the options which are used to generate this table
# or subset the data using the ass_TRUE column in rt@toxData
# with percentages
rt@options@toxTable_tabulationPercent = TRUE
ToxTable_summary(rt)
```

```
# A data.frame is returned if further manipulation is required

# without zeros
rt@options@toxTable_tabulationZeros = FALSE
ToxTable_summary(rt)

# only toxicities with category "Nervous systen disorders"
rt@toxData$ass_TRUE = rt@toxData$category == "Nervous system disorders"
ToxTable_summary(rt)
############################################################
# flexTable default formatting
ft = FT_ToxTable_summary(rt)
# returns an object of class ConstructFlexTable for additional formatting and editing
ft$GetTable()
```

`worstGradeByPatient` *Generate worst grade by patient*

Description

Returns a patient level data.frame containing the worst grade for each patient. The ass_TRUE is used as a filter. Optionally a subset of rt@toxData can be passed in as the second variable. In this case ass_TRUE is still used as a filter.

Usage

```
worstGradeByPatient(rt, toxData = NULL)

worstGradeByPatientCategory(rt, categoryList = NULL)
```

Arguments

`rt`	an object of class robustToxicities
`toxData`	A data.frame subset of rt@toxData if not all data should be used
`categoryList`	A vector of categories

Details

worstGradeByPatientCategory does the same thing but wraps over all categories.

Value

Returns a data.frame

Index