# Package ‘accrualPlot’

October 12, 2022

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Accrual Plots and Predictions for Clinical Trials</td>
</tr>
<tr>
<td>Version</td>
<td>1.0.7</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Lukas Bütikofer <a href="mailto:lukas.buetikofer@ctu.unibe.ch">lukas.buetikofer@ctu.unibe.ch</a></td>
</tr>
<tr>
<td>Description</td>
<td>Tracking accrual in clinical trials is important for trial success. If accrual is too slow, the trial will take too long and be too expensive. If accrual is much faster than expected, time sensitive tasks such as the writing of statistical analysis plans might need to be rushed. 'accrualPlot' provides functions to aid the tracking of accrual and predict when a trial will reach it's intended sample size.</td>
</tr>
<tr>
<td>License</td>
<td>MIT + file LICENSE</td>
</tr>
<tr>
<td>BugReports</td>
<td><a href="https://github.com/CTU-Bern/accrualPlot/issues">https://github.com/CTU-Bern/accrualPlot/issues</a></td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>Depends</td>
<td>lubridate, R (&gt;= 2.10)</td>
</tr>
<tr>
<td>Imports</td>
<td>dplyr, ggplot2, grid, magrittr, purrr, rlang</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>7.1.1</td>
</tr>
<tr>
<td>Suggests</td>
<td>knitr, markdown, patchwork, rmarkdown, testthat, vdiffr</td>
</tr>
<tr>
<td>VignetteBuilder</td>
<td>knitr</td>
</tr>
<tr>
<td>LazyData</td>
<td>true</td>
</tr>
<tr>
<td>NeedsCompilation</td>
<td>no</td>
</tr>
<tr>
<td>Author</td>
<td>Lukas Bütikofer [cre, aut], Alan G. Haynes [aut]</td>
</tr>
<tr>
<td>Repository</td>
<td>CRAN</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2022-08-16 10:10:05 UTC</td>
</tr>
</tbody>
</table>
R topics documented:

- accrualdemo
- accrual_create_df
- accrual_linear_model
- accrual_plot_abs
- accrual_plot_cum
- accrual_plot_predict
- accrual_predict
- accrual_table
- accrual_time_unit
- as.data.frame.accrual_list
- plot.accrual_df
- print.accrual_df
- summary.accrual_df

Index

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>accrualdemo</td>
<td>2</td>
</tr>
<tr>
<td>accrual_create_df</td>
<td>2</td>
</tr>
<tr>
<td>accrual_linear_model</td>
<td>4</td>
</tr>
<tr>
<td>accrual_plot_abs</td>
<td>5</td>
</tr>
<tr>
<td>accrual_plot_cum</td>
<td>7</td>
</tr>
<tr>
<td>accrual_plot_predict</td>
<td>10</td>
</tr>
<tr>
<td>accrual_predict</td>
<td>14</td>
</tr>
<tr>
<td>accrual_table</td>
<td>15</td>
</tr>
<tr>
<td>accrual_time_unit</td>
<td>17</td>
</tr>
<tr>
<td>as.data.frame.accrual_list</td>
<td>17</td>
</tr>
<tr>
<td>plot.accrual_df</td>
<td>18</td>
</tr>
<tr>
<td>print.accrual_df</td>
<td>19</td>
</tr>
<tr>
<td>summary.accrual_df</td>
<td>20</td>
</tr>
</tbody>
</table>

accrualdemo

Demonstration data set

Description

Simulated recruitment data from three sites. Each row represents one participant. Sites one and two started on 2020-07-01, site three on 2020-09-01.

Usage

accrualdemo

Format

A data frame with two variables: date, and site.

accrual_create_df

accrual_create_df

Description

Creates a data frame or a list of data frames that contains the absolute and cumulative number of participants recruited at each date from a vector with enrollment dates. Used as input for accrual plot functions.
Usage

```r
accrual_create_df(
    enrollment_dates,
    by = NA,
    start_date = "site",
    current_date = "common",
    overall = TRUE,
    name_overall = "Overall",
    pos_overall = c("last", "first"),
    force_start0 = TRUE
)
```

Arguments

- `enrollment_dates`: date vector with one entry per participants.
- `by`: factor or character vector with sites, has to have the same length as enrollment dates. If not NA, a list with an accrual data frame for each site is generated.
- `start_date`: date when recruitment started. Single date (used for all sites in by), named date vector (with length and names corresponding to the levels of by), "common" (first date overall) or "site" (first date for each site, default).
- `current_date`: date of the data export or database freeze. Single date, named date vector (with length and names corresponding to the levels of by), "common" (last date overall, default) or "site" (first date for each site).
- `overall`: logical indicates that accrual_df contains a summary with all sites (only if by is not NA).
- `name_overall`: name of the summary with all sites (if by is not NA and overall==TRUE).
- `pos_overall`: overall as last or first element of the list (if by is not NA and overall==TRUE).
- `force_start0`: logical, adds an extra 0 line to the accrual data frame in cases where a start date is given and corresponds to the earliest enrollment date.

Value

Returns a data frame of class 'accrual_df' or a list of class 'accrual_list' with an 'accrual_df' for each level of by (if by is not NA). The 'accrual_df' contains a row per accrual day and the following three columns:

- **Date**: date of accrual
- **Freq**: absolute number accrued at Date
- **Cumulative**: cumulative number accrued up to Date

See Also

`accrual_plot_cum()`, `accrual_plot_abs()` and `accrual_plot_predict()` to generate cumulative, absolute and prediction plots, and `accrual_table()` to generate an accrual table.
Examples

```r
data(accrualdemo)
accrual_create_df(accrualdemo$date)
# different start and current date
accrual_create_df(accrualdemo$date, start_date=as.Date("2020-07-08"),
current_date=as.Date("2020-10-15"))

# by site
accrual_create_df(accrualdemo$date, by=accrualdemo$site)
```

Description

Creates a weighted linear regression model using an accrual data frame produced by `accrual_create_df`.

Usage

```r
accrual_linear_model(
  accrual_df,
  fill_up = TRUE,
  wfun = function(x) seq(1/nrow(x), 1, by = 1/nrow(x))
)
```

Arguments

- `accrual_df` object of class ‘accrual_df’ or ‘accrual_list’ produced by `accrual_create_df`.
- `fill_up` whether to fill up days where no recruitment was observed,
- `wfun` function to calculate the weights with accrual data frame as argument, default is `wfun<-function(x) seq(1 / nrow(x), 1, by = 1/nrow(x))`.

Value

Returns an object of class ‘lm’ with a weighted linear regression of cumulative accrual on dates.

Examples

```r
data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
accrual_linear_model(accrual_df)

# unweighted
accrual_linear_model(accrual_df, wfun=function(x) rep(1,nrow(x)))
```
accrual_plot_abs

#different start and current date
accrual_df<-accrual_create_df(accrualdemo$date, start_date=as.Date("2020-07-08"),
    current_date=as.Date("2020-07-15"))
accrual_linear_model(accrual_df)

#accrual_df with by option
accrual_df<-accrual_create_df(accrualdemo$date, by=accrualdemo$site)
accrual_linear_model(accrual_df)

accrual_plot_abs

Absolute accrual plots

Description

Plot of absolute recruitment by time unit using an accrual data frame produced by accrual_create_df.

Usage

accrual_plot_abs(
    accrual_df,
    unit = c("month", "year", "week", "day"),
    target = NULL,
    overall = TRUE,
    name_overall = attr(accrual_df, "name_overall"),
    ylim = NULL,
    xlim = NULL,
    ylab = "Recruited patients",
    xlabformat = NULL,
    xlabsel = NA,
    xlabpos = NULL,
    xlabsrt = 45,
    xlabadj = c(1, 1),
    xlabcex = 1,
    col = NULL,
    legend.list = NULL,
    ...
)

gg_accrual_plot_abs(
    accrual_df,
    unit = c("month", "year", "week", "day"),
    xlabformat = NULL
)
Arguments

accrual_df    object of class 'accrual_df' or 'accrual_list' produced by accrual_create_df.
unit          time unit for which the bars should be plotted, one of "month", "year", "week" or "day".
target        adds horizontal line for target recruitment per time unit.
overall        logical, indicates that accrual_df contains a summary with all sites that should be removed from stacked barplot (only if by is not NA).
name_overall   name of the summary with all sites (if by is not NA and overall==TRUE).
ylim           limits for y-axis.
xlim           limits for x-axis.
ylab           y-axis label.
xlabformat     format of date on x-axis.
xlabsel        selection of x-labels if not all should be shown, by default all are shown up to 15 bars, with more an automated selection is done, either NA (default), NULL (show all), or a numeric vector.
xlabpos        position of the x-label.
xlabrt         rotation of x-axis labels in degrees.
xlabadj        adjustment of x-label, numeric vector with length 1 or 2 for different adjustment in x- and y-direction.
xlabcex        size of x-axis label.
col            colors of bars in barplot, can be a vector if accrual_df is a list, default is grayscale.
legend.list    named list with options passed to legend().
...            further arguments passed to barplot() and axis().

Details

When the accrual_df includes multiple sites, the dataframe passed to ggplot includes a site variable which can be used for facetting

Value

accrual_plot_abs returns a barplot of absolute accrual by time unit (stacked if accrual_df is a list).

Examples

set.seed(2020)
enrollment_dates <- as.Date("2018-01-01") + sort(sample(1:100, 50, replace=TRUE))
accrual_df<-accrual_create_df(enrollment_dates)
accrual_plot_abs(accrual_df,unit="week")

# time unit
accrual_plot_abs(accrual_df,unit="day")
#include target
accrual_plot_abs(accrual_df,unit="week",target=5)

# further plot options
accrual_plot_abs(accrual_df,unit="week",ylab="No of recruited patients",
xlabformat="%Y-%m-%d",xlabrts=30,xlabpos=-0.8,xlabadj=c(1,0.5),
col="pink",tck=-0.03,mgp=c(3,1.2,0))

# accrual_df with by option
set.seed(2020)
centers<-sample(c("Site 1","Site 2","Site 3"),length(enrollment_dates),replace=TRUE)
centers<-factor(centers,levels=c("Site 1","Site 2","Site 3"))
accrual_df<-accrual_create_df(enrollment_dates,by=centers)
accrual_plot_abs(accrual_df=accrual_df,unit=c("week"))

### ggplot2 approach
data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
gg_accrual_plot_abs(accrual_df, unit = "week")
accrual_df<-accrual_create_df(accrualdemo$date, by = accrualdemo$site)
accrual_plot_abs(accrual_df = accrual_df, unit = "week") +
  ggplot2::theme_classic()

time unit
gg_accrual_plot_abs(accrual_df, unit = "day")

# accrual_df with by option
accrual_df <- accrual_create_df(accrualdemo$date, by = accrualdemo$site)
accrual_plot_abs(accrual_df = accrual_df, unit = "week")
accrual_plot_abs(accrual_df = accrual_df, unit = "week") +
  ggplot2::scale_fill_discrete(type = c("black", "red", "blue", "green"))
accrual_plot_cum

```r
xlabpos = NA,
xlabsrt = 45,
xlabadj = c(1, 1),
xlabcex = 1,
col = rep(1:8, 5),
lty = rep(1:5, each = 8),
legend.list = NULL,
... )

gg_accrual_plot_cum(accrual_df, xlabformat = "%d%b%Y")
```

**Arguments**

- `accrual_df` object of class 'accrual_df' or 'accrual_list' produced by `accrual_create_df`.
- `ylim` limits for y-axis.
- `xlim` limits for x-axis.
- `ylab` y-axis label.
- `xlabn` integer giving the desired number of intervals for the x-label, default=5.
- `xlabmnn` negative integer giving the minimal number of intervals.
- `xlabformat` format of date on x-axis.
- `xlabpos` position of the x-label.
- `xlabsrt` rotation of x-axis labels in degrees.
- `xlabadj` adjustment of x-label, numeric vector with length 1 or 2 for different adjustment in x- and y-direction.
- `xlabcex` size of x-axis label.
- `col` color for line(s) in plot
- `lty` line type(s) in plot
- `legend.list` named list with options passed to `legend()`.
- `...` further options passed to `plot()` and `axis()`.

**Details**

When the `accrual_df` includes multiple sites, the dataframe passed to `ggplot` includes a site variable which can be used for faceting.

**Value**

`accrual_plot_cum` returns a plot of the cumulative accrual (per site if `accrual_df` is a list).

`ggplot2` object
Examples

set.seed(2020)
enrollment_dates <- as.Date("2018-01-01") + sort(sample(1:30, 50, replace=TRUE))
accrual_df<-accrual_create_df(enrollment_dates)
accrual_plot_cum(accrual_df)
accrual_plot_cum(accrual_df,cex.lab=1.2,cex.axis=1.1,xlabcex=1.1)

#several sites
set.seed(1)
centers<-sample(c("Site 1","Site 2","Site 3"),length(enrollment_dates),replace=TRUE)
accrual_df<-accrual_create_df(enrollment_dates,by=centers)
accrual_plot_cum(accrual_df)

#assuming a common start and current date
accrual_df<-accrual_create_df(enrollment_dates,by=centers,start_date="common",current_date="common")
accrual_plot_cum(accrual_df)

#plot and legend options
accrual_plot_cum(accrual_df,col=c("red",rep(1,3)),lty=c(1,1:3),cex.lab=1.2,cex.axis=1.1,xlabcex=1.1)
accrual_plot_cum(accrual_df,legend.list=list(ncol=2,bty=TRUE,cex=0.8))

#without overall
accrual_df<-accrual_create_df(enrollment_dates,by=centers,overall=FALSE)
accrual_plot_cum(accrual_df)

### ggplot2 approach
data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
gg_accrual_plot_cum(accrual_df)
gg_accrual_plot_cum(accrual_df) +
  ggplot2::theme_classic()

#several sites
accrual_df <- accrual_create_df(accrualdemo$date, by = accrualdemo$site)
gg_accrual_plot_cum(accrual_df)

#assuming a common start and current date
accrual_df <-
  accrual_create_df(  
    accrualdemo$date,  
    by = accrualdemo$site,  
    start_date = "common",  
    current_date = "common"  
  )
gg_accrual_plot_cum(accrual_df)

#without overall
accrual_df <-
  accrual_create_df(accrualdemo$date, by = accrualdemo$site, overall = FALSE)
gg_accrual_plot_cum(accrual_df)
accrual_plot_predict  Accrual prediction plots

Description

Generates an accrual prediction plot using an accrual data frame produced by accrual_create_df and a target sample size. Prediction is based on a weighted linear regression. If the accrual data frame is a list (i.e. using the by option in accrual_create_df), or if center start dates are given, the number of enrolled and targeted sites is included.

Usage

accrual_plot_predict(
  accrual_df,
  target,
  overall = TRUE,
  name_overall = attr(accrual_df, "name_overall"),
  fill_up = TRUE,
  wfun = function(x) seq(1/nrow(x), 1, by = 1/nrow(x)),
  col.obs = NULL,
  lty.obs = 1,
  col.pred = "red",
  lty.pred = 2,
  pch.pred = 8,
  pos_prediction = c("out", "in", "none"),
  label_prediction = NULL,
  cex_prediction = 1,
  format_prediction = "%B %d, %Y",
  show_center = TRUE,
  design = 1,
  center_label = "Centers",
  center_legend = c("number", "strip"),
  targetc = NA,
  center_colors = NULL,
  center_legend_text_size = 0.7,
  ylim = NA,
  xlim = NA,
  ylab = "Recruited patients",
  xlabformat = "%d%b%Y",
  xlabn = 5,
  xlabminn = xlabn%/2,
  xlabpos = NA,
  xlabsr = 45,
  xlabadj = c(1, 1),
  xlabcex = 1,
  mar = NA,
  legend.list = NULL,


\[\ldots,\]

\[
\text{center_start_dates} = \text{NULL}\]

\)

\texttt{gg_accrual_plot_predict(}

\texttt{accrual_df,}

\texttt{target,}

\texttt{overall = TRUE,}

\texttt{name_overall = attr(accrual_df, "name_overall"),}

\texttt{col.pred = "red",}

\texttt{lty.pred = 2,}

\texttt{pch.pred = 8,}

\texttt{fill_up = TRUE,}

\texttt{wfun = function(x) seq(1/nrow(x), 1, by = 1/nrow(x)),}

\texttt{pos_prediction = c("out", "in", "none"),}

\texttt{label_prediction = NULL,}

\texttt{format_prediction = "%B %d, %Y",}

\texttt{xlabformat = "%d%b%Y"}

\texttt{)}

\textbf{Arguments}

\texttt{accrual_df} \hspace{1cm} \text{object of class 'accrual_df' or 'accrual_list' produced by accrual_create_df.}

\texttt{target} \hspace{1cm} \text{target sample size or date to predict end date or expected sample size, respectively. A single number or date, or a named vector with the same length as accrual_df. For the latter, center-specific predictions are shown.}

\texttt{overall} \hspace{1cm} \text{logical, indicates that accrual_df contains a summary with all sites (only if by is not NA).}

\texttt{name_overall} \hspace{1cm} \text{name of the summary with all sites (if by is not NA and overall==TRUE).}

\texttt{fill_up} \hspace{1cm} \text{whether to fill up days where no recruitment was observed, otherwise these points do not contribute to the regression.}

\texttt{wfun} \hspace{1cm} \text{function to calculate the weights with accrual data frame as argument, default is wfun<-function(x) seq(1/nrow(x), 1, by = 1/nrow(x)).}

\texttt{col.obs} \hspace{1cm} \text{line color of cumulative recruitment, can be a vector with the same length as accrual_df.}

\texttt{lty.obs} \hspace{1cm} \text{line type of cumulative recruitment, can be a vector with the same length as accrual_df.}

\texttt{col.pred} \hspace{1cm} \text{line color of prediction, can be a vector with the same length as accrual_df.}

\texttt{lty.pred} \hspace{1cm} \text{line color of prediction, can be a vector with the same length as accrual_df.}

\texttt{pch.pred} \hspace{1cm} \text{point symbol for end of prediction, can be a vector with the same length as accrual_df.}

\texttt{pos_prediction} \hspace{1cm} \text{position of text with predicted end date or sample size, either "out", "in" or "none".}

\texttt{label_prediction} \hspace{1cm} \text{label for predicted end date or sample size.}
Details

When the accrual_df includes multiple sites, the dataframe passed to `ggplot` includes a site variable which can be used for facetting.

Value

`accrual_plot_predict` returns a plot with the accrual prediction.
### Examples

```r
data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
## Predict end date
accrual_plot_predict(accrual_df=accrual_df,target=300)
## Predict sample size
accrual_plot_predict(accrual_df=accrual_df,as.Date("2020-11-01"))

## Include site
accrual_df<-accrual_create_df(accrualdemo$date,by=accrualdemo$site)
accrual_plot_predict(accrual_df=accrual_df,target=300,center_label="Site")
## with strip and target
accrual_plot_predict(accrual_df=accrual_df,target=300,center_label="Site",
targetc=5,center_colors=heat.colors(5),center_legend="strip")

## Design for site
accrual_plot_predict(accrual_df=accrual_df,target=300,design=2)

## Format prediction end date
accrual_plot_predict(accrual_df=accrual_df,target=300,
pos_prediction="in",label_prediction="End of accrual: ",cex_prediction=1.2,
format_prediction="%Y-%m-%d",ylim=c(0,150))

## Format plot
accrual_plot_predict(accrual_df=accrual_df,target=300,
ylab="No of recruited patients",ylim=c(0,150),
xlabcex=1.2,xlabsrt=30,xlabn=5,xlabmin=5,
mgp=c(3,0.5,0),cex.lab=1.2,cex.axis=1.2)

## predictions for all sites
accrual_plot_predict(accrual_df=accrual_df,
target=c("Site 1"=160,"Site 2"=100,"Site 3"=40,"Overall"=300))
## different colors
accrual_plot_predict(accrual_df=accrual_df,
target=c("Site 1"=160,"Site 2"=100,"Site 3"=40,"Overall"=300),
col.obs=topo.colors(length(accrual_df)))
##not showing center info
accrual_plot_predict(accrual_df=accrual_df,
target=c("Site 1"=160,"Site 2"=100,"Site 3"=40,"Overall"=300),
show_center=FALSE)

##predictions of sample size for all sites
target<-rep(as.Date("2020-11-01"),4)
names(target)<-c("Site 1","Site 2","Site 3","Overall")
accrual_plot_predict(accrual_df=accrual_df,target=target,col.obs=topo.colors(length(accrual_df)))
### ggplot2 approach

```
#Include site
accrual_df<-accrual_create_df(accrualdemo$date, by=accrualdemo$site)
gg_accrual_plot_predict(accrual_df=accrual_df, target=300)

#Format prediction end date
gg_accrual_plot_predict(accrual_df = accrual_df, 
target=300, 
  pos_prediction="in", 
  format_prediction="%Y-%m-%d")

#predictions for all sites
accrual_predict(accrual_df = accrual_df, 
target=c("Site 1"=160,"Site 2"=100,"Site 3"=40,"Overall"=300))
gg_accrual_plot_predict(accrual_df = accrual_df, 
  target=c("Site 1"=160,"Site 2"=100,"Site 3"=40,"Overall"=300)) + 
ggplot2::theme(legend.position = c(0.15,.9)) + 
ggplot2::labs(col = "Site")

---

Description
accrual_predict

Usage
accrual_predict(accrual_df, accrual_fit, target)

Arguments
accrual_df accrual data frame produced by accrual_create_df (optionally with by option as a list)
accrual_fit linear model produced by accrual_linear_model, can be a list with the same length as accrual_df

target target sample size or date to predict end date or expected sample size, respectively. A single number or date, or a named vector with the same length as accrual_df (to add site-specific targets).

Details
Prediction of end date based on an accrual data frame produced by accrual_create_df, a fitted regression model produced by accrual_linear_model and a target sample size.

Value
Returns the predicted end date(s) or the predicted sample size(s).
Examples

data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
accrual_model<-accrual_linear_model(accrual_df)
# predict date for a specific n
accrual_predict(accrual_df, accrual_model, target=300)
# predict n at a specific date
accrual_predict(accrual_df, accrual_model, target=as.Date("2020-11-01"))

# different start and current date
accrual_df<-accrual_create_df(accrualdemo$date, start_date=as.Date("2020-07-09"),
current_date=as.Date("2020-10-15"))
accrual_model<-accrual_linear_model(accrual_df)
accrual_predict(accrual_df, accrual_model, target=300)

# accrual_df with by option
accrual_df<-accrual_create_df(accrualdemo$date, by=accrualdemo$site)
accrual_model<-accrual_linear_model(accrual_df)
accrual_predict(accrual_df, accrual_model, 
   target=c("Site 1"=160,"Site 2"=100,"Site 3"=40,"Overall"=300))
accrual_predict(accrual_df, accrual_model, target=as.Date("2020-11-01"))

accrual_table

Description

Table of recruitment overview by site, rate of recruitment

Usage

accrual_table(
   accrual_df,
   overall = TRUE,
   name_overall = "Overall",
   pos_overall = c("last", "first"),
   unit = c("month", "year", "week", "day"),
   format_table_date = "%d%b%Y",
   format_time = "%1.0f",
   format_rrate = "%1.2f",
   header = TRUE
)

Arguments

accrual_df object of class 'accrual_df' or 'accrual_list' produced by accrual_create_df.
overall logical, indicates that accrual_df contains a summary with all sites (only if by is not NA).
name_overall name of the summary with all sites (if by is not NA and overall==TRUE).
pos_overall overall in last or first row (if by is not NA and overall==TRUE).
unit time unit for time recruiting and the rate, one of "month", "year", "week" or "day".
format_table_date format of start date in table.
format_time format of time recruiting in table.
format_rrate format of recruitment rate in table.
header include header, logical or character vector of length 4 or 5 (if accrual_df is a list).

Value

Returns data frame with a header, a row per site and overall and the following columns:

name name of the site (if accrual_df is a list)
start_date accrual start date
time time accruing
n number of patients accrued
rate accrual rate per time unit

Examples

data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date,by=accrualdemo$site)
accrual_table(accrual_df)

# format
accrual_table(accrual_df, format_time="%1.1f", format_rrate="%1.1f")

# unit
accrual_table(accrual_df, unit="day")

# common start and current dates
accrual_df<-accrual_create_df(accrualdemo$date, by=accrualdemo$site, start_date="common", current_date="common")
accrual_table(accrual_df)
accrual_df<-accrual_create_df(accrualdemo$date, by=accrualdemo$site, start_date=as.Date("2020-07-09"),
  current_date=as.Date("2020-10-15"))
accrual_table(accrual_df)
**accrual_time_unit**

Description

Generates summary of recruitment per time unit

Usage

```
accrual_time_unit(accrual_df, unit = c("month", "year", "week", "day"))
```

Arguments

- `accrual_df`: accrual data frame produced by `accrual_create_df` with `by=NA`.
- `unit`: time unit for which the bars should be plotted, one of "month", "year", "week" or "day".

Value

Returns a data frame with the number of patients accrued for each time unit.

Examples

```
data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
accrual_time_unit(accrual_df,"week")
accrual_time_unit(accrual_df,"day")
```

**as.data.frame.accrual_list**

as.data.frame method for accural_list objects

Description

as.data.frame method for accural_list objects

Usage

```
## S3 method for class 'accrual_list'
as.data.frame(x, ...)
```
Arguments

x accrual_list
...
for consistency with other as.data.frame methods (not used)

Note

methods from within the package will not work on the output from this function.

Examples

data(accrualdemo)
x <- accrual_create_df(accrualdemo$date, accrualdemo$site)
as.data.frame(x)

plot.accrual_df

Plot method for accrual data frames produced by accrual_create_df

Description

Plot method for accrual data frames produced by accrual_create_df

Usage

## S3 method for class 'accrual_df'
plot(x, which = "cum", engine = c("base", "ggplot2"), ...)

Arguments

x object of class 'accrual_df' or 'accrual_list' produced by accrual_create_df.
which one of "cumulative", "absolute" or "predict". Abbreviations are allowed.
engine string to indicate the plotting engine (base/graphics or ggplot2)
...
options passed to other functions

Value

A plot with cumulative or absolute accrual, or accrual prediction.

See Also

accrual_plot_abs(), accrual_plot_cum() and accrual_plot_predict()
print.accrual_df

Examples

data(accrualdemo)
accrual_df <- accrual_create_df(accrualdemo$date)
plot(accrual_df)
plot(accrual_df, "abs", unit="week")
plot(accrual_df, "pred", target = 300)
plot(accrual_df, "pred", target = 300, engine = "ggplot")

print.accrual_df

Print methods for accrual objects

Description

Print methods for accrual objects

Usage

## S3 method for class 'accrual_df'
print(x, head = TRUE, ...)

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

Arguments

x object of class ‘accrual_df’ or ‘accrual_list’ produced by accrual_create_df.
head show header of the accrual data?
... arguments passed to head

Value

No return value

Examples

data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
print(accrual_df)
# only show text
print(accrual_df, head = FALSE)
# show first 15 days
print(accrual_df, n = 15)
summary.accrual_df  Summary method for accrual_dfs (as created by accrual_create_df)

Description
Summary method for accrual_dfs (as created by accrual_create_df)

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

Arguments
object  object of class ‘accrual_df’ or ‘accrual_list’ produced by accrual_create_df.
...    options passed to other functions

Value
Returns data frame with a header, a row per site and overall and the following columns:

name  name of the site (if accrual_df is a list)
start_date  accrual start date
time  time accruing
n  number of patients accrued
rate  accrual rate per time unit

Examples

data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date, accrualdemo$site)
summary(accrual_df)
Index

* datasets
  * accrualdemo, 2
  * accrual_create_df, 2
  * accrual_linear_model, 4
  * accrual_plot_abs, 5
  * accrual_plot_abs(), 3, 18
  * accrual_plot_cum, 7
  * accrual_plot_cum(), 3, 18
  * accrual_plot_predict, 10
  * accrual_plot_predict(), 3, 18
  * accrual_predict, 14
  * accrual_table, 15
  * accrual_table(), 3
  * accrual_time_unit, 17
  * accrualdemo, 2
  * as.data.frame.accrual_list, 17
  * gg_accrual_plot_abs (accrual_plot_abs), 5
  * gg_accrual_plot_cum (accrual_plot_cum), 7
  * gg_accrual_plot_predict (accrual_plot_predict), 10
  * plot.accrual_df, 18
  * print.accrual_df, 19
  * print.accrual_list (print.accrual_df), 19
  * summary.accrual_df, 20

21