

# Package: accrualPlot (via r-universe)

October 15, 2024

**Type** Package

**Title** Accrual Plots and Predictions for Clinical Trials

**Version** 1.0.9

**Maintainer** Lukas Bütikofer <lukas.buetikofer@ctu.unibe.ch>

**Description** 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.

**License** MIT + file LICENSE

**URL** <https://github.com/CTU-Bern/accrualPlot>,  
<https://ctu-bern.github.io/accrualPlot/>

**BugReports** <https://github.com/CTU-Bern/accrualPlot/issues>

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Depends** lubridate, R (>= 2.10)

**Imports** dplyr, ggplot2, grid, magrittr, purrr, rlang

**RoxygenNote** 7.2.1

**Suggests** knitr, markdown, patchwork, rmarkdown, testthat, vdiff

**VignetteBuilder** knitr

**LazyData** true

**Repository** <https://ctu-bern.r-universe.dev>

**RemoteUrl** <https://github.com/CTU-Bern/accrualPlot>

**RemoteRef** HEAD

**RemoteSha** 4f35a3007b820b8a95bcab220827c1bb846f3570

## Contents

accrualdemo . . . . .	2
accrual_create_df . . . . .	2
accrual_linear_model . . . . .	4
accrual_plot_abs . . . . .	5
accrual_plot_cum . . . . .	7
accrual_plot_predict . . . . .	9
accrual_predict . . . . .	14
accrual_table . . . . .	15
accrual_time_unit . . . . .	16
as.data.frame.accrual_list . . . . .	17
plot.accrual_df . . . . .	18
print.accrual_df . . . . .	19
summary.accrual_df . . . . .	19
<b>Index</b>	<b>21</b>

---

accrualdemo	<i>Demonstration data set</i>
-------------	-------------------------------

---

### 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	<i>accrual_create_df</i>
-------------------	--------------------------

---

### 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**

```
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" (last 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**

```

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)

```

---

```
accrual_linear_model  accrual_linear_model
```

---

**Description**

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

**Usage**

```

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

```

**Arguments**

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

**Value**

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

**Examples**

```

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)))

#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	<i>Absolute accrual plots</i>
------------------	-------------------------------

---

### 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,  
  xlabssel = NA,  
  xlabpos = NULL,  
  xlabprt = 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 <code>accrual_create_df</code> .
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.
xlabse1	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.
xlabprt	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 faceting

### Value

accrual\_plot\_abs returns a barplot of absolute accrual by time unit (stacked if accrual\_df is a list).

ggplot object

### 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",xlabprt=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")
gg_accrual_plot_abs(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)
gg_accrual_plot_abs(accrual_df = accrual_df, unit = "week")
gg_accrual_plot_abs(accrual_df = accrual_df, unit = "week") +
  ggplot2::scale_fill_discrete(type = c("black", "red", "blue", "green"))

```

---

accrual\_plot\_cum      *Cumulative accrual plots*

---

## Description

Plot of cumulative recruitment using an accrual data frame produced by `accrual_create_df`.

## Usage

```

accrual_plot_cum(
  accrual_df,
  ylim = NA,
  xlim = NA,
  ylab = "Recruited patients",
  xlabn = 5,
  xlabminn = xlabn%/2,
  xlabformat = "%d%b%Y",
  xlabpos = NA,
  xlabprt = 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 xlabel, default=5.
xlabminn	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,
  xlabprt = 45,
  xlabadj = c(1, 1),
  xlabcex = 1,
  mar = NA,
  legend.list = NULL,
  ...,
  center_start_dates = NULL
)

gg_accrual_plot_predict(
  accrual_df,
  target,
  overall = TRUE,
  name_overall = attr(accrual_df, "name_overall"),
```

```

col.pred = "red",
lty.pred = 2,
pch.pred = 8,
fill_up = TRUE,
wfun = function(x) seq(1/nrow(x), 1, by = 1/nrow(x)),
pos_prediction = c("out", "in", "none"),
label_prediction = NULL,
format_prediction = "%B %d, %Y",
xlabformat = "%d%b%Y"
)

```

### Arguments

accrual_df	object of class 'accrual_df' or 'accrual_list' produced by accrual_create_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. For the latter, center-specific predictions are shown.
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).
fill_up	whether to fill up days where no recruitment was observed, otherwise these points do not contribute to the regression.
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)).
col.obs	line color of cumulative recruitment, can be a vector with the same length as accrual_df.
lty.obs	line type of cumulative recruitment, can be a vector with the same length as accrual_df.
col.pred	line color of prediction, can be a vector with the same length as accrual_df.
lty.pred	line color of prediction, can be a vector with the same length as accrual_df.
pch.pred	point symbol for end of prediction, can be a vector with the same length as accrual_df.
pos_prediction	position of text with predicted end date or sample size, either "out", "in" or "none".
label_prediction	label for predicted end date or sample size.
cex_prediction	text size for predicted end date or sample size.
format_prediction	date format for predicted end date (only if target is a sample size)
show_center	logical, whether the center info should be shown (if accrual_df is a list or if center_start_dates are given).
design	design options for the center info 1 (default): below plot, 2: within plot, top, 3: within plot, bottom.
center_label	label for the center info.

<code>center_legend</code>	either "number" to plot numbers in the center strip or "strip" to add a legend strip, requires specification of <code>center_colors</code> .
<code>targetc</code>	target number of centers, to scale the legend if it is "strip".
<code>center_colors</code>	colors to be used for the strip with the centers, a vector of length <code>targetc</code> .
<code>center_legend_text_size</code>	size of the text of the center or legend strip, only has a function
<code>ylim</code>	limits for y-axis.
<code>xlim</code>	limits for x-axis.
<code>ylab</code>	y-axis label.
<code>xlabformat</code>	format of date on x-axis.
<code>xlabn</code>	integer giving the desired number of intervals for the xlabel, default=5.
<code>xlabminn</code>	integer giving the minimal number of intervals.
<code>xlabpos</code>	position of the x-label.
<code>xlabsrt</code>	rotation of x-axis labels in degrees.
<code>xlabadj</code>	adjustment of x-label, numeric vector with length 1 or 2 for different adjustment in x- and y-direction.
<code>xlabcex</code>	size of x-axis label.
<code>mar</code>	vector of length 4 (bottom, left, top, right margins), overwrite default margins.
<code>legend.list</code>	named list with options passed to <code>legend()</code> , only if accrual data frame is a list.
<code>...</code>	further options passed to <code>plot()</code> and <code>axis()</code> .
<code>center_start_dates</code>	alternative way to add center info, vector with dates on which centers are enrolled.

### 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.  
ggplot object

### Examples

```
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, xlabprt=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
data(accrualdemo)
accrual_df<-accrual_create_df(accrualdemo$date)
gg_accrual_plot_predict(accrual_df = accrual_df, target = 300)
gg_accrual_plot_predict(accrual_df = accrual_df, target = 300) +
  ggplot2::theme_classic()

#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
gg_accrual_plot_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")
```

---

accrual_predict	<i>accrual_predict</i>
-----------------	------------------------

---

### Description

accrual\_predict

### Usage

```
accrual_predict(accrual_df, accrual_fit, target)
```

### Arguments

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

*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 <code>accrual_create_df</code> .
overall	logical, indicates that <code>accrual_df</code> contains a summary with all sites (only if <code>by</code> is not NA).
name_overall	name of the summary with all sites (if <code>by</code> is not NA and <code>overall==TRUE</code> ).
pos_overall	overall in last or first row (if <code>by</code> is not NA and <code>overall==TRUE</code> ).
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      *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 accrual\_list objects*

---

**Description**

as.data.frame method for accrual\_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	<i>Plot method for accrual data frames produced by accrual_create_df</i>
-----------------	--

---

### 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

<code>x</code>	object of class 'accrual_df' or 'accrual_list' produced by <code>accrual_create_df</code> .
<code>which</code>	one of "cumulative", "absolute" or "predict". Abbreviations are allowed.
<code>engine</code>	string to indicate the plotting engine (base/graphics or ggplot2)
<code>...</code>	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\(\)](#)

### 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 <code>accrual_df</code> 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, 9
- accrual\_plot\_predict(), 3, 18
- accrual\_predict, 14
- accrual\_table, 15
- accrual\_table(), 3
- accrual\_time\_unit, 16
- 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), 9
  
- plot.accrual\_df, 18
- print.accrual\_df, 19
- print.accrual\_list (print.accrual\_df),  
19
  
- summary.accrual\_df, 19