

Package ‘sievetest’

October 14, 2022

Type Package

Title Laboratory Sieve Test Reporting Functions

Version 1.2.3

Date 2018-06-07

Author Petr Matousu

Maintainer Petr Matousu <pmatousu@more-praha.cz>

Description Functions for making particle-size analysis. Sieve tests are widely used to obtain particle-size distribution of powders or granular materials.

Depends R (>= 3.1.0)

License GPL (>= 2)

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-07 20:03:49 UTC

R topics documented:

c.std	2
desc.std	2
drr	4
lignite	5
plot.std	6
read.std	7
rrx2x	8
std	9
summary.std	11
tweak.std	12

Index	13
--------------	-----------

c.std

Concatenate std object

Description

The function is used to concatenate more sievetest objects together.

Usage

```
## S3 method for class 'std'  
c(...)
```

Arguments

... one or more std objects

Value

Returns one std object concatenated of objects supplied in ... containing all the sieve test samples grouped together.

See Also

[read.std](#), [std](#), [summary.std](#)

Examples

```
data(lignite)  
# a and b simulate some new std objects  
a <- lignite[2]  
b <- lignite[1]  
#  
ab_group <- c(a,b)
```

desc.std*Sieve Test Data Object Description*

Description

The function for querying a std object for description meta data or creating new description meta data list.

Usage

```
desc.std(Description = NA,  
         Title = NA,  
         Sample = NA,  
         Unit = NA,  
         Date = NA,  
         x = NULL)
```

Arguments

Description	Terse description of a sample
Title	Title of particular measurement
Sample	Name of the sample
Unit	To which unit the sample belongs
Date	Day when the sample was taken
x	An existing std object

Details

To be used for creating input for desc argument in functions std and tweak.std.

Value

Returns a list of 5:

Description
Title
Sample
Unit
Date

See Also

[std](#)

Examples

```
# create the object by hand  
# sieve test data (std) object  
my_sample <- std(  
  a=c(500, 200, 90, 0),  
  r=c(1.01, 24, 42.8, 32.190),  
  desc=desc.std(Title="Coal powder, learning std")  
)
```

drr

*Rosin - Rammler Distribution Functions***Description**

Rosin - Rammler model of particle-size distribution and cumulative undersize and oversize distributions used to obtain approximation of of powders or granular materials originated by grinding.

Usage

```
drr(x, ex, xs)
orr(x, ex, xs)
urr(x, ex, xs)
```

Arguments

x	particle size, equivalent particle diameter
ex	Rosin - Rammler exponent, measure of the uniformity of grinding
xs	finesse of grinding, that width of mesh associated with a remainder equal to $\exp(-1) \sim 0.3679$

Details

Following functions are used, based on Rosin - Rammler mathematical model of particle-size distribution, for approximation of size distribution.

drr is Rosin - Rammler probability density function

urr is Rosin - Rammler cumulative distribution function (CDF) representing undersize mass fraction

orr is Rosin - Rammler complementary CDF representing oversize mass fraction ie. relative remainder on the sieve with the mesh size x

Rosin - Rammler model (1933) is the Weibull distribution which was proposed by Weibull in 1939, and Weibull distribution functions are part of R.

So the user can use `stats::dweibull(x, shape=ex, scale=xs)` the same way as drr, and use Weibull distribution functions provided by stats package for deeper analysis.

Similarly, `stats::pweibull(x, shape=ex, scale=xs)` can be used the same way as urr or `stats::pweibull(x, shape=ex, scale=xs, lower.tail=F)` the same way as orr.

Value

Both urr and orr returns value of distribution function.
Function drr returns density.

References

Rinne, H. (2008) *The Weibull Distribution: A Handbook*, chapter 1.1.2. Taylor & Francis.

See Also

[Weibull](#), [plot.std](#), [summary.std](#)

Examples

```
## The function drr is currently defined as
# function (x, ex, xs)
# {
#   (ex/xs) * (x/xs)^(ex - 1) * exp(-(x/xs)^ex)
# }

## The function urr is currently defined as
# function (x, ex, xs)
# {
#   1 - exp(-(x/xs)^ex)
# }

## The function orr is currently defined as
# function (x, ex, xs)
# {
#   exp(-(x/xs)^ex)
# }

x <- c(1,5,10,50,100)
ex <- 1.386
xs <- 178
stats::dweibull(x,shape=ex,scale=xs)
drr(x,ex,xs)
stats::pweibull(x,shape=ex,scale=xs)
urr(x,ex,xs)
stats::pweibull(x,shape=ex,scale=xs,lower.tail=FALSE)
orr(x,ex,xs)
```

lignite

Sieve Test Data Object Example

Description

This is example std object, which is used in examples and in demo code. It is based on sieve test results of powdered lignite.

Usage

```
data(lignite)
```

Examples

```
data(lignite)
str(lignite)
summary(lignite)
plot(lignite)
```

plot.std

Sieve Test Data Object Plot

Description

The function plots graphical presentation of Rosin-Rammler particle size distribution of powder which is common in grinding and comminution.

Usage

```
## S3 method for class 'std'
plot(x,
     type=c("rr", "rrdist"),
     n=30,
     lgd=T,
     col=c("#1B9E77", "#D95F02", "#7570B3", "#E7298A",
           "#66A61E", "#E6AB02", "#A6761D", "#666666"),
     bx=1,
     by=5,
     lgd.x="topleft",
     ...)
```

Arguments

x	an std object representing sieve test data.
type	type of plot
n	a factor influencing density of ticks marks
lgd	whether to show legend
col	vector of colors
bx	a factor influencing rounding of x values for labels
by	a factor influencing rounding of y values for labels
lgd.x	legend position
...	additional graphical parameters

Value

Plots an std object to the current device.

See Also[std](#), [par](#)**Examples**

```
# load example data and make plot
data(lignite)
plot(lignite)
plot(lignite,type='rrdist',lwd=2,lgd.x="right")
```

read.std	<i>Data input from file</i>
----------	-----------------------------

Description

Reads std data from CSV file and creates std object.

Usage

```
read.std(file, sep = "\t", dec = ".")
```

Arguments

file	file path.
sep	field separator used in the file, defaults to "\t".
dec	the character used in the file for decimal points, defaults to ".".

Details

The file format allows to have more test data within one file.

# Meta data	—	—	—	—
D1 names	D1 values	DN names	DN values	
# Data	—	—	—	
aperture	retained	aperture	retained	
A1	R1 values	An	Rn values	

Section D1 or DN is 5 rows of meta data.

Commented lines must not be removed.

Multi row columns A and R corresponds to a and r arguments of std function, D corresponds to desc.

The file must have at least two columns, then it is just for 1 sample data, or it can have Nx2 columns for N samples.

Data section allows for different number of A1 and AN rows.

Value

Returns a std object.

See Also

[std](#), [c.std](#), [summary.std](#)

Examples

```
# provided example file
fi <- system.file("lignite.csv",package="sievetest")
# inspect the file
file.show(fi)
# read the file
# there will be warning, because the file is prepared the way
# to show them, they are OK
a <- read.std(fi)
plot(a)
```

rrx2x

Sieve Test Data Object Plot Coordinates Transformation Functions

Description

Function transform user coordinates of plot to axis coordinates or vise versa.

Usage

```
rrx2x(x)
rry2y(x)
x2rrx(x)
y2rry(x)
```

Arguments

x Value to transform.

Details

The default plot type "rr" uses transformed coordinates, so the user can read percents on y axis and micrometers on x axis, while `par('usr')` returns coordinates in $\log(x)$ and $\log(\log(100/y))$. Hence the user can use transformation functions to convert between the types. The "rr" keyword in the name of function, belongs to the log x log-log (usr) space.

Value

Functions return appropriate transformed value.

See Also[plot.std](#)**Examples**

```

data(lignite)
plot(lignite)
# query for exact x coordinates within the plot
if(interactive()) rrx2x(locator())$x)
# query for exact y coordinates within the plot
if(interactive()) rry2y(locator())$y)
# or
y2rry(36.78794)
# or
rrx2x(par('usr')[1])

```

std*Sieve Test Data Object*

Description

Functions to create, test and subset a std object.

Usage

```

std(a, r, desc = desc.std(), lmargs = list())
is.std(x)
## S3 method for class 'std'
x[i, ...]

```

Arguments

a	A vector of sieve net aperture sizes in micrometers, zero size means the receiver.
i	An integer or vector of integers
r	Mass percent of remaining material on the sieve, vector with the same length as vector a, sum should be equal 100, if not the function will give warning and recalculate the values. If the value is 0, then will be excluded with warning.
desc	A description meta data prepared by desc.std
lmargs	A list of additional arguments to lm
x	An object of std class
...	Not used

Details

Basically the `std` object is list of lists, in the first level (trunk) are stored particular sieve test samples, in the second level (branches) are stored details of each single sample. If there is only one sample in the `std` object, then the length will be 1. The object can be created by hand using `std` function or by reading the basic information from CSV file using `read.std` function. Once the `std` object exists, it can be sub-selected using `x[i]` or concatenated by `c(...)`.

Value

Both `std(a,r)` and `x[i]` functions return `std` object, `x[[i]]` returns list holding following informations:

Description	Terse description of a sample
Title	Title of particular measurement
Sample	Name of the sample
Unit	To which unit the sample belongs
Date	Day when the sample was taken
stdata	Data frame with given and calculated values
lmfit	Result of linear model fit, <code>lm</code> object
RRcoefficients	List of coefficients for Rosin - Rammler distribution
size	List of characteristic sizes of Rosin - Rammler distribution

The `is.std(x)` function returns logical value.

See Also

[read.std](#), [c.std](#), [summary.std](#), [tweak.std](#)

Examples

```
## create std
sieve_aperture_size <- c(500, 200, 90, 0)
mass_ppc_retained <- c(1.01, 24, 42.8, 32.190)
# some meta data:
md <- desc.std("Coal powder, learning std")
# sieve test data (std) object
my_sample <- std(a=sieve_aperture_size, r=mass_ppc_retained, desc=md)
is.std(my_sample)
```

summary.std	<i>Summary of std object</i>
-------------	------------------------------

Description

The function makes summary table of std object.

Usage

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

Arguments

object	std object
...	not used

Value

Returns data frame with following columns:

Description	Terse description of a sample
Title	Title of particular measurement
Sample	Name of the sample
Unit	To which unit the sample belongs
Date	Day when the sample was taken
RRxs	Rosin - Rammler distribution xs coefficient
RRex	Rosin - Rammler distribution ex coefficient
RRms	modus calculated from Rosin - Rammler approximation
RRmn	mean calculated from Rosin - Rammler approximation
RRo90umPPC	CCDF (oversize) value in % calculated from Rosin - Rammler approximation for equivalent particle size of 90 micrometers

See Also

[drr](#), [std](#)

Examples

```
data(lignite)
summary(lignite)
```

tweak.std	<i>Modify std object</i>
-----------	--------------------------

Description

The function is used for modifying of existing std object.

Usage

```
tweak.std(
  x,
  desc = desc.std(x = x),
  lmargs = as.list(x[[1]]$lmfit$call[-c(1:3)]))
```

Arguments

x	std object to be modified
desc	list of description meta data, default is original meta data of object x
lmargs	list of lm additional arguments, default is original lmargs taken from lm call (all lm arguments except formula and data)

Details

The function modifies the std object or just refresh the x object while recalculating the lm fit, Rosin - Rammler coefficients and characteristic sizes. The refreshing of an old saved object can be useful in case that lm fit structure changes across R versions as the std object encapsulates lm fit result.

Value

Returns std object with length = 1.

See Also

[desc.std](#), [std](#)

Examples

```
# Load the data, modify linear model and plot.
data(lignite)
std1 <- lignite[2]
std2 <- tweak.std(lignite[2],lmargs = list(weights=c(1,1,0,1)),
  desc = desc.std(x=std1,Title="Lignite w 1 1 0 1"))
plot(c(std1,std2))
```

Index

- * **IO**
 - read.std, 7
- * **datasets**
 - lignite, 5
- * **distribution**
 - drr, 4
- * **dplot**
 - rrx2x, 8
- * **hplot**
 - plot.std, 6
- * **utilities**
 - c.std, 2
 - desc.std, 2
 - std, 9
 - summary.std, 11
 - tweak.std, 12
- [.std (std), 9
- c.std, 2, 8, 10
- desc.std, 2, 12
- drr, 4, 11
- is.std (std), 9
- lignite, 5
- orr (drr), 4
- par, 7
- plot.std, 5, 6, 9
- read.std, 2, 7, 10
- rrx2x, 8
- rry2y (rrx2x), 8
- std, 2, 3, 7, 8, 9, 11, 12
- summary.std, 2, 5, 8, 10, 11
- tweak.std, 10, 12
- urr (drr), 4
- Weibull, 5
- x2rrx (rrx2x), 8
- y2rry (rrx2x), 8