# Package 'scan'

June 21, 2016

**Type** Package

**Title** Single-Case Data Analyses for Single and Multiple AB Designs

**Version** 0.14.3

**Date** 2016-06-16

**Author** Juergen Wilbert

**Maintainer** Juergen Wilbert <jwilbert@uni-potsdam.de>

**Imports** stats, nlme, utils, graphics

**Depends** R (>= 1.8.0)

**Description** A collection of procedures for analysing single-case data of an AB-design. Some procedures support multiple-baseline designs.

**License** GPL

**URL** http://www.uni-potsdam.de/inklusion/inklusionspaedagogik/wilbert/projekte/single-case-data-analysis-with-r.html

**LazyData** true

**ENCODING** UTF-8

## R topics documented:

## `scan-package` *Single-Case Data Analyses*

### Description

A collection of procedures for analysing single-case data of an AB-design. Some procedures support multiple-baseline designs.

### Details

| | |
|---|---|
| Package: | scan |
| Type: | Package |
| Version: | 0.14.3 |
| Date: | 2016-06-16 |
| License: | GPL (version 2 or newer) |

### Author(s)

Juergen Wilbert

Department of Teacher Education

Faculty of Humanities

University of Potsdam

http://www.uni-potsdam.de/inklusion/inklusionspaedagogik/wilbert/projekte/single-case-data-analysis-with-r.html

Maintainer: Juergen Wilbert <jwilbert@uni-potsdam.de>

## `describeSC` *Descriptives*

### Description

The describeSC command gives some important descriptives of your single-case data including trends, standardized mean differences, measurements for data overlap, and autocorrelation.

### Usage

```
describeSC(data, decreasing = FALSE)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `decreasing` | If you expect the data to be lower in the B-phase, set decreasing = TRUE. |

### Author(s)

Juergen Wilbert

### Examples

```
set.seed(1234)
dat <- rSC(5, B.start = round(runif(5,5,10)), d.level = 1.0, d.slope = 0.1, d.trend = 0.05,
rtt = 0.75, round = 0, random.names = TRUE)
describeSC(dat)

set.seed(1233)
dat <- rSC(5, B.start = round(runif(5,5,10)), d.level = -0.4, d.slope = 0.0, d.trend = 0.0,
rtt = 0.70, random.names = TRUE)
describeSC(dat, decreasing = TRUE)
```

## `example-data` *Exampe single-case data*

### Description

This is a list of example single case data sets implemented in the package.

### Details

example1

Grosche2011

byHeart2011

GruenkeWilbert2014

Grosche2014

Huber2014

Beretvas2008

Huitema2000

Waddell2011

Borckardt2014

### Author(s)

Juergen Wilbert

## fillmissingSC *Fill in missing measurement times*

### Description

The command fillmissingSC() fills in missing measurement times and estimated values if a single-case or multiple single-cases by linear interpolation. Optionally, missing values (NA) can also be replaced. This procedure is recommended when you have larger gaps between measurement times or explicit missing values and want to calculate overlapping (PND, PAND, NAP ...) values or a randomization test.

### Usage

```
fillmissingSC(data, interpolation = "linear", na.rm = TRUE)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `interpolation` | Not in use yet. |
| `na.rm` | If set TRUE, NA values are also interpolated. |

### Author(s)

Juergen Wilbert

### Examples

```
#In his study, Grosche (2011)could not realize measurements each single week for
#all participants. During the course of 100 weeks, about 20 measurements per person
#at different times were administered. Here is a way to fill in the missing data:
new.data <- fillmissingSC(Grosche2011)
#Graphically compare the two datasets:
plotSC(c(Original = Grosche2011[2], Filled = new.data[2]))

set.seed(1234)
dat <- rSC(d.level = 1.0)
new.dat <- dat
replace.positions <- c(10,16,18)
new.dat[[1]][replace.positions,"values"] <- NA
filled.new.dat <- fillmissingSC(new.dat)
plotSC(c("original" = dat, "missing" = new.dat, "interpolated" = filled.new.dat),
      marks = list(positions = replace.positions))
```

hplm *Hierarchical Piecewise linear model/Piecewise Regression*

### Description

The hplm function computes a hierarchical piecewise regression model.

### Usage

```
hplm(data, model = "B&L-B", method = "ML", control = list(opt = "optim"),
     random.slopes = TRUE, ICC = TRUE)
```

### Arguments

| | |
|---|---|
| data | A data frame or a list of data frames. |
| model | Regression model used for computation (see Huitema & Mckean, 2000). Default is model = "B&L-B". Possible values are: B&L-B, H-M, Mohr#1, Mohr#2, Manly. |
| method | a character string. If "REML" the model is fit by maximizing the restricted log-likelihood. If "ML" the log-likelihood is maximized. Defaults to "ML". |
| control | a list of control values for the estimation algorithm to replace the default values returned by the function lmeControl of the nlme package. |
| random.slopes | If TRUE random slope effects of the level, trend, and treatment parameter are estimated. |
| ICC | If TRUE an intraclass-correlation is estimated. |

### Author(s)

Juergen Wilbert

### Examples

```
set.seed(1000)

dat <- rSC(3, MT = 30, B.start = 11, d.level = 1.0, d.slope = 0.05, d.trend = 0.05)
hplm(dat, method = "REML",random.slopes = FALSE)

set.seed(1000)

dat <- rSC(15, MT = 30, B.start = 11, d.level = 1.0, d.slope = runif(15,0,0.2), d.trend = 0.05)
hplm(dat)
```

## longSCDF *Create a single data frame of a list of single-case dataframes*

### Description

This functions helps you to create a single data frame from a list of multiple single cases.

### Usage

```
longSCDF(data)
```

### Arguments

| | |
|---|---|
| `data` | A list with single-case dataframes |

### Value

Returns a data frame with one or multiple single-cases divided by the variabele 'case'.

### Author(s)

Juergen Wilbert

## makeSCDF *Create a single case data frame*

### Description

This procedure is used to create the basic data structure for storing a single or multiple single-cases. This basic data structure is used in all scan functions

### Usage

```
makeSCDF(data, B.start = NULL, MT = NULL)
```

### Arguments

| | |
|---|---|
| `data` | A vector with measurements. |
| `B.start` | The start of the B phase. |
| `MT` | A vector defining measurement times. Defaut is MT = (1,2,3,...,n). |

### Value

Returns a data frame formated as a single-case dataset.

### Author(s)

Juergen Wilbert

## Examples

```
person1 <- makeSCDF(c(5, 7, 8, 5, 7, 12, 16, 18, 15, 14, 19), B.start = 6)
plotSC(person1)

charlotte <- makeSCDF(c(5, 7, 8, 5, 7, 12, 16, 18, 15, 14, 19), 6)
theresa <- makeSCDF(c(3, 4, 3, 5, 7, 8, 7, 9, 8, 10, 12),5)
antonia <- makeSCDF(c(9, 8, 8, 7, 5, 7, 13, 14, 15, 12, 16), 7)
study1 <- list(charlotte, theresa, antonia)
names(study1) <- c("Charlotte", "Theresa", "Antonia")
describeSC(study1)
```

---

# makesingleSC *Aggregate multiple cases into one case*

---

## Description

The makesingleSC command combines multiple single cases into one single case which can be used for further analysis.

## Usage

```
makesingleSC(data, scale = FALSE, type = "add")
```

## Arguments

| | |
|---|---|
| data | A vector with measurements, a data frame or a list of data frames. |
| scale | Unused |
| type | By default values with the same measurement-time are added. If type is set to mean or median, values of the same MT are replaced with their mean or median. Default is add. |

## Details

The algorithm works the following way: 1) All measurements of each single-case are mean-centred with respect to the mean of the A-phase of each single-case. 2) The values of the A-phases of all single-cases are combined and ordered ascending with respect to the measurement times. 3) The values of the B-phases are combined and ordered as well. 4) The measurements of the B-phases are attached behind the A-phase measurements. The measurement times of the B-phase are shifted to start one measurement time after the A-phase.

## Author(s)

Juergen Wilbert

## Examples

```
study1 <- rSC(3, d.level = 0.8, d.slope = 0.1, round = 0)
new.sc <- makesingleSC(study1)
plotSC(new.sc, type = "p", lines = "trend")
```

## nap *Non-overlap of all Pairs*

### Description

The index Nonoverlapping of all pairs was developed by Parker and colleagues (Parker & Vannest, 2009). It is recommended as an overlapping measurement.

### Usage

```
nap(data, decreasing = FALSE)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `decreasing` | If you expect the data to be lower in the B phase, set decreasing = TRUE (default is FALSE). |

### Value

| | |
|---|---|
| `NAP` | Nonoverlap of all pairs. |
| `N` | Number of cases. |

### Author(s)

Juergen Wilbert

### References

Parker, R. I., & Vannest, K. (2009). An improved effect size for single-case research: Nonoverlap of all pairs. Behavior Therapy, 40(4), 357-367.

### Examples

```
set.seed(1234)
dat <- rSC(1, d.level = -1.0)
nap(dat, decreasing = TRUE)
```

## outlierSC *Cleaning outliers*

### Description

Identifies and drops outliers for each case of your data file. The function returns a new data file without the outlier data. Criteria for outlier identification can be standard deviations, confidence intervals, and Cook's distance (based on a piecewise linear regression model).

### Usage

```
outlierSC(data, criteria = c("SD", "2"))
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `criteria` | Specifies criteria for dropping data. SD for standard deviations (e.g., criteria = c("SD", 2)). CI for Confidence interval (e.g., criteria = c("CI", 0.99) represents a 99 percent confidence interval. Cook for Cook's distance based on the piecewise-linear-regression model. To exclude cases where cook exceeds 4/n set: criteria = c("Cook", "4/n"). |

### Value

| | |
|---|---|
| `data` | Cleaned up data file. |
| `dropped.n` | A list with the number of dropped values for each case. |
| `dropped.mt` | A list with the measuremt-times of dropped cases for each case (values are based on the mt variable of each data frame). |
| `ci.matrix` | A list with a matrix for each case with values for the upper and lower boundaries based on the confidence interval. |
| `sd.matrix` | A list with a matrix for each case with values for the upper and lower boundaries based on the standard deviation. |
| `cook` | A list with the cook values for each measuremnt-time of each case. |
| `criteria` | Method and criteria used for outlier analysis. |
| `N` | Number of cases in the design. |
| `case.names` | Identifier of the cases. |

### Author(s)

Juergen Wilbert

### Examples

```
#To 'clean' the original data from Grosche (2011) without the outliers and using
#Cook's distance greater 4/n as a criteria, type:
(res <- outlierSC(Grosche2011, criteria = c("Cook", "4/n")))
plotSC(Grosche2011, marks = list(positions = res$dropped.mt))
```

## `pand` *Percent all non-overlapping data*

### Description

The percent non-overlapping data is a measurement for estimating a level increase (or decrease) in performance after an intervention. It can be computed with the pand command. PAND allows for a Chi-square statistical test (comparing real and estimated association a data points with phase A and B) and thereby allows an estimation of the effect size phi, which is identical to Pearsons r computed with dichotomous data. Phi-Square is thus the amount of explained variance. The original procedure for computing PAND as proposed by Parker and colleagues (Parker u. a., 2007) does not take ambivalent data points due to ties into account, as does the computation for the nonoverlap of all pairs index (Parker & Vannest, 2009). In the pand procedure, the frequency matrix is corrected for ties.

### Usage

```
pand(data, decreasing = FALSE, correction = TRUE)
```

### Arguments

`data` A data frame or a list of data frames.

`decreasing` If you expect the data to be lower in the B phase, set decreasing = TRUE (default is FALSE).

`correction` If set TRUE, the frequency matrix is corrected for ties. A tie is counted as a half-measuremnt in each phase.

### Author(s)

Juergen Wilbert

### References

Parker, R. I., Hagan-Burke, S., & Vannest, K. (2007). Percentage of All Non-Overlapping Data (PAND) An Alternative to PND. The Journal of Special Education, 40(4), 194 - 204. Parker, R. I., & Vannest, K. (2009). An improved effect size for single-case research: Nonoverlap of all pairs. Behavior Therapy, 40(4), 357 - 367.

### Examples

```
set.seed(1234)
dat <- rSC(1, d.level = 1.0)
pand(dat)

# PAND when expecting a decrease in performance
# after intervention in a multiple-baseline design:
set.seed(1234)
dat <- rSC(3, B.start = c(5, 7, 9), d.level = -1.0)
pand(dat, decreasing = TRUE)
```

## `pem` *Percent exceeding the median*

### Description

The function pem returns the percentage of the B-phase data exceeding the median of the A-phase data. Additionally, a chi square test against a 50/50 distribution is computed. Different measures of central tendency could be addressed for alternative analyses. If you expect a decrease in the B-phase, set the parameter decreasing = TRUE.

### Usage

```
pem(data, decreasing = FALSE, binom.test = TRUE, chi.test = FALSE, FUN = median, ...)
```

### Arguments

`data` A data frame or a list of data frames.

`decreasing` If you expect the data to be lower in the B phase, set decreasing = TRUE (default is FALSE).

`binom.test` Computes a binomial test for a 50/50 distribution. By default set TRUE.

`chi.test` By default set FALSE. If set FALSE the Chi-square Test is skipped.

`FUN` By default median. Alternatively, you could implement any other function for computing an average.

`...` Additional arguments for the FUN parameter (e.g. FUN = mean, trim = 0.1 will use a 10 percent trimmed mean instead of a median for analysing the data).

### Author(s)

Juergen Wilbert

### Examples

```
set.seed(1234)
dat <- rSC(5, d.level = 0.5)
pem(dat, chi.test = TRUE)
```

## `pet` *Percent exceeding the trend*

### Description

The function pet return the percentage of the B-phase data exceeding the predicted values on basis of the trend of the A-phase. A binomial test against a 50/50 distribution is computed. Furthermore, the percent of data points in the B-phase exceeding the upper (or lower) threshold of the 95 percent confidence interval of the predicted values is returned. If you expect a decrease in the B-phase, set the parameter decreasing = TRUE.

### Usage

```
pet(data, ci = 0.95, decreasing = FALSE)
```

### Arguments

`data` A data frame or a list of data frames.

`decreasing` If you expect the data to be lower in the B phase, set decreasing = TRUE (default is FALSE).

`ci` The size of the confidence interval (by default ci = .95).

### Author(s)

Juergen Wilbert

### Examples

```
set.seed(1234)
dat <- rSC(5, d.slope = 0.2)
pet(dat)
```

## plm *Piecewise linear model/Piecewise Regression*

### Description

The plm function computes a piecewise regression model (see Huitema & Mckean, 2000). A piecewise regression models a linear trend in the data, a level and a slope increase with the start of the intervention. Additionally, the plm function computes two effect-sizes by comparing restricted to full regression models (see Beretvas & Chung, 2008). Furthermore, an autoregression model can be used to take into account autocorrelated data (Beretvas & Chung, 2008). For the latter, your R system must comprise an installation of the nlme package.

### Usage

```
plm(data, AR = NULL, model = "B&L-B", count.data = FALSE,
    family = ifelse(count.data, "poisson", "gaussian"), ...)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `AR` | Autoregression modelled up to the given lag based on the ARMA function. |
| `model` | Regression model used for computation (see Huitema & Mckean, 2000). Default is model = "B&L-B". Possible values are: B&L-B, H-M, Mohr#1, Mohr#2, Manly. |
| `count.data` | |
| `family` | |
| `...` | |

### Author(s)

Juergen Wilbert

### References

Beretvas, S., & Chung, H. (2008). An evaluation of modified R2-change effect size indices for single-subject experimental designs. Evidence-Based Communication Assessment and Intervention, 2, 120-128.

Huitema, B. E., & Mckean, J. W. (2000). Design specification issues in time-series intervention models. Educational and Psychological Measurement, 60(1), 38-58.

### Examples

```
set.seed(1000)
dat <- rSC(1, MT = 30, B.start = 11, d.level = 1.0, d.slope = 0.05, d.trend = 0.05)
plm(dat, AR = 3)
```

## plotSC *Plot AB-designs*

### Description

The plotSC command draws a nice plot of one or multiple single-cases. By default, plotSC draws a multiple single-case plot if the given data contains more than one case.

### Usage

```
plotSC(data, ylim = NULL, xlim = NULL, fill = "", lines = "", marks = NULL,
       annotations = NULL, phase.names = c("A", "B"), FUN.AB = NULL,
       xlab = "Measurement time", ylab = "Score", text.ABlag = NULL, lwd = 2,
       pch = 17, type = "b", mai = c(0.6, 0.82, 0.2, 0.42), ...)
```

### Arguments

`data` A data frame or a list of data frames.

`ylim` Lower and upper limit for the y-axis of the plot (e.g., ylim = c(0, 20) sets the y-axis to a scale from 0 to 20). In multiple single cases plots you can use ylim = c(0, NA) to scales the y-axis from 0 to the maximum of each case. If the ylim parameter is not set, scan automatically sets a proper scale.

`xlim` Lower and upper limit for the x-axis of the plot. If the xlim parameter is not set, scan automatically computers a proper scale.

`fill` If set, the area beneath the plot line is coloured in the given colour (e.g., fill = "grey"). Use the standard R command colours() to get a list of possible colours.

`lines` A character or list defining one or more lines or curves to be plotted for a better estimation of statistical parameters. The argument is either given as a single character string lines = "median" or a list of character strings lines = list("median", "trend"). Additionally, some of the procedures can be refined by a single argument e.g., lines = list("mean" = 0.10) adds a 10 percent trimmed mean line. See details.

`marks` A list of parameters defining the marking of certain data points.

"positions" = A vector or a list of vectors indicating measurement-times to be marked. In case of a vector, the marked measurement-times are the same for each plotted case. In case of a list of vectors, marks are set differently for each case. The list must have the same length as there are cases in the datafile.

"col" = A character string. The colour of the marks.

"cex" = The size of the marks.

`annotations` A list defnining several parameters for plotting annotations to each data point. col = colour of the annotations, cex = size of the annotations, pos = position of plotting annotations (1=below, 2=left, 3=above, 4=right), round = rounds the values of the annotations to the specified number of decimal places.

`phase.names` scan uses the default names 'A' and 'B' for the two experimental phases. You can change these with the phase.names parameter. Example: phase.names = c("Baseline", "Treatment").

`FUN.AB` Unused.

`xlab` The label of the x-axis.

`ylab` The labels of the y-axis.

`text.ABlag` By default, scan draws a vertical line to separate the A and B-phase. Alternatively, you could print a character string between the two phases using the textAB.lag parameter (e.g., textAB.lag = "Start").

`lwd` Width of the plot lines. Default is 2.

`pch` Point type. Default is 17. (e.g., pch = 16 draws filled circles; pch = "A" draws As).

`type` Sets the plot type. "l" draws lines, "p" points, "b" draws lines and points, and "n" draws nothing onto the plot (default of plotSC is "b"). The "n" argument can be useful in combination with the lines argument (e.g., type = "n", lines = "loreg").

`mai` Sets the margins of the plot.

`...` Further arguments for passing to the plot command.

## Details

Paramters for the argument "lines".

"median" A median line for the A and B-phase.

"mean" A mean line for the A and B-phase. Default is a 10 percent trimming. Other trimming values can be set by a second parameter e.g., lines = list(mean = 0.2) draws a 20 percent trimmed mean line.

"trend" A trend line for the A and B-phase.

"trendA" A trend line for the A-phase.

"maxA" A line for the maximum of the A-phase.

"medianA" A line for the median of the A-phase.

"meanA" A line for the 10 percent trimmed mean of the A-phase. Trimming can be changed in the following way lines = list(meanA = 0.15).

"plm" A regression line for a piecewise linear regression model.

"plm.ar" A regression line for a piecewise autoregression model. The lag is specified by plm.ar = 2. Where the number is the maximum order of the autoregression analysis.

"movingMean" Draws a moving mean curve. The lag can be specified by lines = list(movingMean = 2). Default is a lag 1 curve.

"movingMedian" Draws a moving median curve. The lag can be specified by lines = list(movingMedian = 3). Default is a lag 1 curve.

"loreg" A non-parametric local regression line. The proportion of data influencing each data point can be specified by lines = list("loreg" = 0.66). The default is 0.5.

"lty", "lwd", "col" Specifies line type ltw, line width lwd, and line colour col. Default is a black dashed line with the width 2. E.g., to draw a solid red thick mean line type lines = list("mean", lty = "solid", lwd = 4, col = "red"). Some of the possible line types are "solid", "dashed", and "dotted".

## Author(s)

Juergen Wilbert

### Examples

```
plotSC(Grosche2011, fill = "grey", type = "n", ylim = c(0,NA),
       xlab = "Training session", ylab = "Words per minute",
       phase.names = c("Baseline", "Intervention"),
       lines = list("plm", lty = "solid", col = "black", lwd = 3))

set.seed(1234)
dat <- rSC(3)
plotSC(dat, marks = list(positions = list(c(2,4,5),c(1,2,3),c(7,8,9)), col = "blue",
       cex = 1.4),annotations = list(label = "values","col" = "red", cex = 0.75,
       offset = 1, round = 0))
```

## pnd *Percent non-overlapping data*

### Description

The function pnd return the percentage non-overlapping data. If you expect a decrease in the B-phase, set the parameter decreasing = TRUE. Due to its error-proneness you should not use PND but NAP or PAND instead (see Parker & Vannest, 2009).

### Usage

```
pnd(data, decreasing = FALSE)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `decreasing` | If you expect the data to be lower in the B phase, set decreasing = TRUE (default is FALSE). |

### Author(s)

Juergen Wilbert

### Examples

```
#
```

## power.testSC *Empirical Power Analysis for Single Cases*

### Description

The power.testSC command conducts a Monte-Carlo study on the test-power and alpha-error of a randomization-test and a piecewise-regression model. The distribution values of the Monte-Carlo sample is either defined or automatically estimated based on the data of an actual study.

## Usage

```
power.testSC(data = NULL, stat = c("rand.test", "plm"),
          test.parameter = c("level", "slope"), rand.test.stat = c("Mean B-A", "B"),
            cases = NULL, rtt = 0.8, d.level = NULL,
            d.slope = NULL, MT = NULL, B.start = NULL, d.trend = NULL, n = 100,
            limit = 5, m = 50, s = 10, startpoints = NA, extreme.p = 0,
            extreme.d = c(-4, -3), exclude.equal = "auto", alpha = 0.05,
            distribution = "normal", concise = TRUE, silent = FALSE)
```

## Arguments

`data` A data frame or a list of data frames. If data are provided, the power-analysis will be based on the design of these data and on the estimation of the effect sizes within the data.

`stat` Defines the statistics the power analysis is computed for. The default stat = c("rand.test","plm") computes a power-analysis for the randomization and the plm analysis.Further possibilities are "hplm" for a hierarchiacal linear regression model and "plm.poisson" for a generalized piecewise-regression model with the assumption of Poisson distributed error terms.

`test.parameter` Indicated whether the power and alpha error for a level effect, a slope effect, or both effects should be estimated.

`rand.test.stat` Defines the statistic a randomization test is based on. The first values stipulates the statistic for the level-effect computation and the second value for the slope-effect computation. Deafult is rand.test.stat = c("Mean B-A","B").

`cases` Number of cases per study.

`rtt` Reliability of the underlying simulated measurements.

`d.level` A single level increase with the beginning of the B-Phase in standard deviations.

`d.slope` Amount of increase in standard deviations per measurement time starting with the Beginning of the B-phase (e.g., d.slope = 0.1 generates an incremental increase of 0.1 standard deviations for each measurement of the B-phase).

`MT` Measurement times of single cases' random study.

`B.start` Starting values of the B-phase. A single value (e.g., B.start = 6) defines the B.start for all studies and cases. A vector of starting values is given with the chain command (e.g., B.start = c(6, 7, 8)). A value between 0 and 1 is interpreted as a proportion (e.g., B.start = c(0.3, 0.5, 0.8) are starting point at 30, 50, and 80 percent of the length defined in MT).

`d.trend` Defines a trend in d per measurement across all measurements. E.g., trend = 0.1 is an increase of 0.1 standard deviations per measurement.

`n` Number of sample studies created in the the Monte-Carlo study.

`limit` Minimal length of the A- und B-phase in the randomization sample.

`startpoints` Alternative to the limit parameter startpoints exactly defines the possible start points of the B-phase. E.g., startpoints = 4:9 restricts possible start points of the B-phase in the randomization design to measurements 4 to 9. startpoints arguments overwrite the limit parameter.

`m` Mean of the sample distribution the data are drawn from.

`s` Standard deviation of the sample distribution the data are drawn from.

`extreme.p` Probability of extreme values (e.g., extreme.p = 0.05 gives a 5 percent probability of an extreme value.

`extreme.d` Range of an extreme value in d (e.g., extreme.d = c(-7,-6) results in extreme values within a range of -7 and -6 standard deviations). Default is c(-4,-3). Caution: the first value must be smaller than the second, otherwise the procedure will fail.

`exclude.equal` If set FALSE, random distribution value that are equal to the observed distribution are counted as values of the distribution of the null-hypothesis. That is, they decrease the probability of rejecting the null-hypothesis (increase the p-value). Default is exclude.equal = "auto" (FALSE for multiple-baseline designs and TRUE for single baseline designs).

`alpha` Alpha level used to calculate the proportion of significant tests. Default is 0.05.

`distribution` Indicated whether the random sample is based on a "normal" (e.g., gaussian) distribution or on a "poisson" distribution.

`concise` If set TRUE, no effects on the generated random samples are given during the computation.

`silent` If set TRUE, the results are not printed afte computation.

### Author(s)

Juergen Wilbert

### Examples

```
#Assume you want to conduct a study with 15 measurements.
#The test is highly reliable and you expect an intervention effect of d.level = 1.4.
#The starting point of the intevention phase is randomly set between the
#5th and the 12th measurement Can you expect to identify the effect with a
#plm or a randomization test (for a more precise estimation, set n = 1000)


power.testSC(MT = 15, B.start = round(runif(100,5,12)), test.parameter = "level",
             d.level = 1.4, rtt = 0.8, n = 10)

### would the testpower be higher if you would set up a multiple baseline design?
#(for a more precise estimation, set n = 1000)

power.testSC(cases = 3, MT = 15, stat = c("rand.test","hplm"),
             B.start = round(runif(300,5,12)), test.parameter = "level",
             d.level = 1.4, rtt = 0.8, n = 10, startpoints = 5:12)
```

---

## `randSC` *Randomization tests*

---

### Description

This function computes a randomization test for single or multiple baseline single-case datasets. The function is based on an algorithm of the R-package SCRT (Bulte & Onghena, 2009, 2012) but rewritten and extended for the use in AB designs.

### Usage

```
randSC(data, statistic = "Mean B-A", number = 500, complete = FALSE, limit = 5,
       startpoints = NA, exclude.equal = FALSE, graph = FALSE, output = "c")
```

## Arguments

`data` A data frame or a list of data frames.

`statistic` Statistic that is computed for randomization. Default is statistic = "Mean B-A").

Arguments for the stat parameter of the rand.test function:

"Mean A-B":Computes the difference between the mean of A and the mean of B. This is used if a level decrease in the B-phase is expected.

"Mean B-A": Computes the difference between the mean of B and the mean of A. This is used if a level increase in the B-phase is expected.

"Mean |A-B|": Computes the absolute value between the difference of the mean of the A and B-Phase.

"Median A-B": The same as Mean A-B but based on the median.

"Median B-A": The same as Mean B-A but based on the median.

"B": Computes the difference between the slope parameter of a linear regression model for the B and A-phase. The procedure has not yet been tested for practicality, so it is experimental.

"Bdecrease": Complementary to the "B" value. Computes the difference between the slopes of the A and the B-phase. This is useful when a decrease in the slope due to an intervention is expected.

"t plm level": Computes piecewise regression models and takes the t value for the level predictor as the critical statistic.

"t plm slope": Computes piecewise regression models and takes the t value for the slope predictor as the critical statistic.

`number` Size of the randomization distribution. The exactness of the p-value could not exceed 1/number (i.e., number = 100 results in p-values with an exactness of one percent). Default is number = 500. For a faster processing set number = 100. For a more precise p-value set number = 1000.

`complete` If TRUE, the distribution is based on a complete permutation of all possible starting combinations. This setting overwrites the number Argument. The default setting is FALSE.

`limit` Minimal length of the A- und B-phase in the randomization sample. Default is limit = 5.

`startpoints` Alternative to the limit parameter startpoints exactly defines the possible start points of the B-phase. E.g., startpoints = 4:9 restricts possible start points of the B-phase in the randomization design to measurements 4 to 9. startpoints arguments overwrite the limit parameter.

`exclude.equal` If set FALSE, random distribution value that are equal to the observed distribution are counted as values of the distribution of the null-hypothesis. That is, they decrease the probability of rejecting the null-hypothesis (increase the p-value). exclude.equal should be set to TRUE if you analyse one single-case design (not a multiple baseline data set) to get a sufficient test-power but be aware it increases the chance of an alpha-error.

`graph` If TRUE, a histogram of the resulting distribution is plotted.

`output` If set to "C", detailed information are given and if set to "p", only the resulting p value is returned.

## Author(s)

Juergen Wilbert

### References

Bulte, I., & Onghena, P. (2009). Randomization tests for multiple-baseline designs: An extension of the SCRT-R package. Behavior Research Methods, 41(2), 477-485.

Bulte, I., & Onghena, P. (2012). SCRT: Single-Case Randomization Tests. Abgerufen von http://cran.r-project.org/web/packages/SCRT/index.html

### Examples

```
set.seed(1234)
dat <- rSC(3, d.level = 0.6)
randSC(dat, statistic = "Median B-A", startpoints = 5:17, graph = TRUE)
```

## rciSC *Reliable Change Index*

### Description

Computes different measures for a reliable change index (see Wise, 2004).

### Usage

```
rciSC(data, rel = 0.8, ci = 0.95, graph = FALSE)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `rel` | Reliability necessary for computing standard errors. Default is rel = 0.8. |
| `ci` | Span of the confidence interval (default is ci = 0.95 indicating a 95 percent confidence interval). |
| `graph` | If set TRUE, a boxplot with the mean of the A and B-phase and the confidence intervals is plotted (default is FALSE). |

### Author(s)

Juergen Wilbert

### References

Wise, E. A. (2004). Methods for analyzing psychotherapy outcomes: A review of clinical significance, reliable change, and recommendations for future directions. Journal of Personality Assessment, 82(1), 50 - 59.

### Examples

```
rciSC(byHeart2011[1], graph = TRUE)
```

## readSC *Read single case datafiles*

### Description

Used for importing your data from a csv files.

### Usage

```
readSC(filename, sep = ",", dec = ".", sort.labels = FALSE,
       phase.names = c("A", "B"), ...)
```

### Arguments

| | |
|---|---|
| filename | Name of the file to be imported. |
| sep | Decimal sign. |
| dec | Variable separator. |
| sort.labels | If set TRUE, the resulting list is sorted by label names (alphabetically, increasing). |
| phase.names | Name of the phases. Usually they should be "A" and "B". IF the datafile containes different names, these are converted to "A" and "B". |
| ... | Further arguments passed to the read.table command. |

### Value

Returns a scan Data structure

### Author(s)

Juergen Wilbert

### Examples

```
#
```

## rSC *Random sample generator*

### Description

Generate random single- and multiple-baseline samples.

### Usage

```
rSC(n = 1, MT = 20, B.start = 6, m = 50, s = 10, prob = 0.5, d.trend = 0, d.level = 0,
    d.slope = 0, rtt = 0.8, concise = FALSE, cases = 1, round = NA, extreme.p = 0,
     extreme.d = c(-4, -3), missing.p = 0, distribution = "normal",
     random.names = FALSE, output.long = FALSE)
```

### Arguments

| | |
|---|---|
| `n` | Number of studies to be created (default is n = 1). |
| `MT` | Measurement times of single cases in each study. |
| `B.start` | Starting values of the B-phase. A single value (e.g., B.start = 6) defines the B.start for all studies and cases. A vector of starting values is given with the chain command (e.g., B.start = c(6, 7, 8)). If the number of n*cases surpasses the length of the vector, values are repeated. |
| `m` | Mean of the sample distribution the data are drawn from. A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. |
| `s` | Standard deviation of the sample distribution the data are drawn from. A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. |
| `prob` | When the paramter distribution is set to "binomial", prob indicates the probability of the occurence of an event. |
| `d.trend` | Defines a trend in d per measurement across all measurements. A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. E.g., d.trend = 0.1 is an increase of 0.1 standard deviations per measurement. In case of a binomial or poisson distribution, d.trend indicates the increase in points/counts per measurement. |
| `d.level` | A single level increase with the beginning of the B-Phase in standard deviations. A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. In case of a binomial or poisson distribution, d.level indicates the increase in points/counts. |
| `d.slope` | Amount of increase in standard deviations per measurement time starting with the Beginning of the B-phase (e.g., d.slope = 0.1 generates an incremental increase of 0.1 standard deviations for each measurement of the B-phase). A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. In case of a binomial or poisson distribution, d.slope indicates the increase in points/counts. |
| `rtt` | Reliability of the underlying simulated measurements (default is rtt = 0.8). A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. In case of a binomial or poisson distribution rtt does have no effect onthe generated data. |
| `concise` | If set TRUE, no information about the generated sample is given. |
| `cases` | Number of cases per study. E.g., n = 10, cases = 3, B.start = c(7,9,11) creates 10 multiple-baseline designs with 3 persons (Starting point of the B-phase at 7, 9, 11, respectively) each. |
| `round` | Rounds the measurement values to a specific decimal position (e.g., round = 2). |
| `extreme.p` | Probability of extreme values (e.g., extreme.p = 0.05 gives a 5 percent probability of an extreme value. A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated. |
| `extreme.d` | Range of an extreme value in d (e.g., extreme.d = c(-7,-6) results in extreme values within a range of -7 and -6 standard deviations). In case of a binomial or poisson distribution, extreme.d indicates points/counts. Caution: the first value must be smaller than the second, otherwise the procedure will fail. |

`missing.p` Proportion of missing values (e.g., missing.p = 0.1 renders 10 percent of all values as missing). A vector of values defines alterating values for each case. If the number of n*cases surpasses the length of the vector, values are repeated.

`distribution` Possible values are "normal", "binomial", or "poisson". If set to normal, the sample is created based on the assumption of a normal distribution of values (with the parameters m and s as the mean and the standard deviation), including a measurement error defined in the parameter rtt. If set to "binomial", data are drawn from a binomial distribution with the expectation value defined by the m parameter. This setting is useful for generating criterial data like correct answers in a test. If set to "poisson", data are drawn from a poisson distribution representing count-data as are often collected in behavioural observations. A measurement error is not included and the m parameter is used as the expectation value of the poisson distribution lambda.

`random.names` Just a gimmick. If set TRUE cases have random first names. The selection of names is drawn from a list of the 2000 most popular names for births in the U.S.A. 2012 (1000 names for boys and 1000 names for girls).

`output.long` If set TRUE, true, trend, level, slope, and error values are included for each case.

### Author(s)

Juergen Wibert

### Examples

```
set.seed(1234)
dat <- rSC(n = 3, rtt = 0.75, round = 1, d.slope = 0.1, extreme.p = 0.1,
           missing.p = 0.1, random.names = TRUE)
describeSC(dat)

set.seed(1234)
dat <- rSC(n = 50, rtt = 0.6, round = 1, d.trend = 0.05,
           d.level = 0.5, d.slope = 0.1)
hplm(dat)
```

## smoothSC *Smoothing*

### Description

This command provides three different procedures to smooth the data (i.e., to eliminate noise). A moving average function (either mean based or median based) replaces step-by-step each data point by the average of the surrounding data. With a local regression function, each data point is regressed by its surrounding values.

### Usage

```
smoothSC(data, FUN = "movingMedian", intensity = NULL)
```

## Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `FUN` | Function to compute the smoothing. Default is a lag 1 moving Median functioon (see details). |
| `intensity` | In the case of movingMean or movingMedian it is the lag (i.e., the range of values around the target value) that is used for computing the average (default is 1). In case of localRegression it is the proportion of data influencing each data point (default is 0.2). |

## Details

Values for the FUN arguments.

"movingMean" Computes the moving mean.

"movingMedian" Computes the moving median.

"localRegression" Computes non-parametric local regressions.

## Author(s)

Juergen Wilbert

## Examples

```
new.dat <- smoothSC(Huber2014$Berta)
new.dat2 <- smoothSC(Huber2014$Berta, FUN = "movingMean")
new.dat3 <- smoothSC(Huber2014$Berta, FUN = "localRegression")
plotSC(list("Original" = Huber2014$Berta,"Moving Median" = new.dat[[1]],
            "Moving Mean" = new.dat2[[1]],"Local Regression" = new.dat3[[1]]))
```

---

# `tauUSC` *Tau-U*

---

## Description

The tau-U index was proposed by Parker and colleagues.

## Usage

```
tauUSC(data, ties.method = "omit")
```

## Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `ties.method` | Defines how to handle ties. 'omit' does not include ties into the calculatioin of tau-u, 'positive' and 'negative' are two further possibilities. |

## Details

#

### Value

| | |
|---|---|
| `table` | The differnt comparisons for estimating the tau-U effect. Tau-U can be derived as the tau value of the A vs. B + trend B - trend A comparison. |
| `matrix` | The matrix for calculating tau-U |

### Author(s)

Juergen Wilbert

### References

Parker, R. I., Vannest, K. J., Davis, J. L., & Sauber, S. B. (2011). Combining Nonoverlap and Trend for Single-Case Research: Tau-U. Behavior Therapy, 42, 284-299.

### Examples

```
#example of Parker et al. 2011
exa <- makeSCDF(c(2, 3, 5, 3, 4, 5, 5, 7, 6),5)
tauUSC(exa)
```

## `trendSC` *Trends in single Cases*

### Description

If you want to get an overview of linear trends in the different phases of a single-case you can use the trendSC command. By default, it gives you the intercept and slope of a linear and a squared regression model of the measurement-time on the values. Models are computed for the A phase, the B-phase (with correcting the measurement-time to start at one), and the combined A and B-Phase. For a more advanced application, you can add further regression models using the R specific formula class, which will be applied again to the A-phase, the B-phase, and the combined A and B-phase.

### Usage

```
trendSC(data, B.offset = -1, model = NA)
```

### Arguments

| | |
|---|---|
| `data` | A data frame or a list of data frames. |
| `B.offset` | An offset for the first measurement-time (MT) of the B-phase. If set 0, the first MT of the B-phase is set to 1. The default is -1, so the MT of the B-phase begin with 0. |
| `model` | A string or a chain of (named) strings each depicting one regression model. This is a formula expression of the standard R class. The parameters of the model are values, mt and phase. |

### Author(s)

Juergen Wilbert

### Examples

```
set.seed(1000)
dat <- rSC(1, d.slope = 0.5)
trendSC(dat)

#In addition to the linear and squared regression model, two further models are computed:
#a) a cubic model, and b) the values predicted by the natural
#logarithm of the measurement time. Additionally, the data of eight
#single-cases are aggregated.
set.seed(1000)
dat <- rSC(1, d.slope = 0.3)
trendSC(dat, B.offset = 0, model = c("Cubic" = "values ~ I(mt^3)",
        "Log Time" = "values ~ log(mt)"))
```

## truncateSC *Truncating data*

### Description

This function is used to skip some measuremnts at the beginning and/or end of each phase.

### Usage

```
truncateSC(data, A = c(0,0), B = c(0,0))
```

### Arguments

| | |
|---|---|
| data | A data frame or a list of data frames. |
| A | A vector of two values. The first sets the number of measurements at the beginning of the A phase that are truncated and the second the umber of mesearments at the end of the A phase. In the case of multiple baseline datasets, the truncation is adapted to the length of the A phase for each single case. |
| B | A vector of two values. The first sets the number of measurements at the beginning of the B phase that are truncated and the second the umber of mesearments at the end of the B phase. In the case of multiple baseline datasets, the truncation is adapted to the length of the B phase for each single case. |

### Author(s)

Juergen Wilbert

### Examples

```
# truncate the first two measurements of the A and B phase
newData <- truncateSC(byHeart2011[1], A = c(2,0), B = c(2,0))
plotSC(c(original = byHeart2011[1], selected = newData))
```

## writeSC *Export data into a csv file.*

### Description

Data are restructured and exported into a csv file for later use with other software.

### Usage

```
writeSC(dat, filename, sep = ",", dec = ".", ...)
```

### Arguments

| | |
|---|---|
| `dat` | Data file to be exported. |
| `filename` | Target file name. |
| `sep` | Variable separator. |
| `dec` | Decimal sign. |
| `...` | Further arguments passed to write.table. |

### Author(s)

Juergen Wilbert

### Examples

```
#
```

# Index