

# Package ‘samon’

August 16, 2017

**Version** 4.0.0

**Type** Package

**Date** 2017-08-14

**Title** Sensitivity Analysis for Missing Data

**Author** Daniel O. Scharfstein [aut], Aidan McDermott [aut, cre]

**Maintainer** Aidan McDermott <amcderm1@jhu.edu>

**Description** In a clinical trial with repeated measures designs, outcomes are often taken from subjects at fixed time-points. The focus of the trial may be to compare the mean outcome in two or more groups at some pre-specified time after enrollment. In the presence of missing data auxiliary assumptions are necessary to perform such comparisons. One commonly employed assumption is the missing at random assumption (MAR). The 'samon' package allows the user to perform a (parameterized) sensitivity analysis of this assumption. In particular it can be used to examine the sensitivity of tests in the difference in outcomes to violations of the MAR assumption. The sensitivity analysis can be performed under two scenarios, a) where the data exhibit a monotone missing data pattern (see the samon() function), and, b) where in addition to a monotone missing data pattern the data exhibit intermittent missing values (see the samonIM() function).

**License** GPL-2

**Depends** R (>= 2.10)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2017-08-15 22:02:24 UTC

## R topics documented:

alpha0Tables . . . . .	2
alpha0TablesIM . . . . .	3
ManData . . . . .	4
samon . . . . .	5
samonCombine . . . . .	8
samonCombineIM . . . . .	9
samonCrossSummary . . . . .	10

samonCrossSummaryIM . . . . .	11
samonDataCheck . . . . .	12
samonDifferenceSummary . . . . .	13
samonDifferenceSummaryIM . . . . .	14
samonECompleterStatus . . . . .	15
samoneval . . . . .	16
samonevalIM . . . . .	17
samonGen . . . . .	19
samonGenIM . . . . .	21
samonIM . . . . .	24
samonSummary . . . . .	29
samonSummaryIM . . . . .	29
samonTable1 . . . . .	30
The PANSS trial. . . . .	31
The DepWork trial . . . . .	32
The VAS trial. . . . .	32

**Index** **34**

---

alpha0Tables	<i>Makes samon result tables at alpha = 0</i>
--------------	---

---

**Description**

alpha0TableResults collects results for tables at sensitivity parameter alpha = 0 and alpha0Tables prints the output from alpha0TableResults

**Usage**

```
alpha0TableResults( Y1, samonR1, Y2, samonR2, samonRD)
alpha0Tables( alpha0TableResultsObj, trt1lab = "Treatment 1",
trt2lab = "Treatment 2")
```

**Arguments**

Y1	treatment 1's raw data.
samonR1	samon results for treatment 1.
Y2	treatment 2's raw data.
samonR2	samon results for treatment 2.
samonRD	samon results for the difference of treatment 2 minus treatment 1.
alpha0TableResultsObj	a list produced by the function alpha0TableResults.
trt1lab	a label for treatment 1.
trt2lab	a label for treatment 2.

**Details**

alpha0TableResults computes material to build the table and alpha0Tables prints tables as simple text.

**Value**

alpha0TableResults returns table data. alpha0Tables returns nothing.

**Examples**

```
data("samonPANSS1")
data("samonPANSS2")

data("PSummary1")
data("PSummary2")
data("PSummaryD")

alpha0Results <- alpha0TableResults( samonPANSS1, PSummary1,
                                     samonPANSS2, PSummary2,
                                     PSummaryD )

alpha0Tables(alpha0Results)
```

---

alpha0TablesIM	<i>Makes samonIM result tables at sensitivity parameter alpha = 0.</i>
----------------	--

---

**Description**

alpha0TableResultsIM collects results for tables at sensitivity parameter alpha = 0 and alpha0TablesIM prints the output from alpha0TableResultsIM

**Usage**

```
alpha0TableResultsIM( Y1, samonR1, Y2, samonR2, samonRD)
alpha0TablesIM( alpha0TableResultsObj, trt1lab = "Treatment 1",
               trt2lab = "Treatment 2")
```

**Arguments**

Y1	treatment 1's raw data.
samonR1	samon results for treatment 1.
Y2	treatment 2's raw data.
samonR2	samon results for treatment 2.
samonRD	samon results for the difference of treatment 2 minus treatment 1.
alpha0TableResultsObj	a list produced by the function alpha0TableResultsIM.
trt1lab	a label for treatment 1.
trt2lab	a label for treatment 2.

**Details**

alpha0TableResultsIM computes material to build the table and alpha0TablesIM prints tables as simple text.

**Value**

alpha0TableResultsIM returns table data. alpha0TablesIM returns nothing.

**Examples**

```
data("VAS1")
data("VAS2")

data("VSummary1")
data("VSummary2")
data("VSummaryD")

alpha0Results <- alpha0TableResultsIM( VAS1, VSummary1,
                                       VAS2, VSummary2,
                                       VSummaryD )

alpha0TablesIM(alpha0Results)
```

---

ManData

*Data used in the man page examples.*

---

**Description**

Some functions within the 'samon' package take some time to run, particularly when the number of bootstraps requested is large. To illustrate some of the functions on the man pages pre-packaged results are sometime used to make the examples run faster. Here we list some of this data.

**Details**

The samonPANSS1 and samonPANSS2 datasets are documented on their own man pages.

**P1Res001 and P1Res002** are two small output objects from the samon function and are used to illustrate the functioning of the samonCombine function.

**P1Results and P2Results** are combined results for samonPANSS1 and samonPANSS2 respectively and are used to illustrate the functioning of the samonSummary function.

**PSummary1 and PSummary2** are summary objects from the samonSummary function and are used to illustrate the use of the samonDifferenceSummary and samonCrossSummary functions.

**PSummaryD** this is the results from the samonDifferenceSummary function and is used along with PSummary1 and Psummary2 to illustrate the alpha0Table and samonEYCompleterStatus functions.

The VAS1 and VAS2 datasets are documented on their own man pages.

**V1Res001 and V1Res002** are two small output objects returned from the samonIM function and are used to illustrate the functioning of the samonCombineIM function.

**V1Results and V2Results** are combined results for VAS1 and VAS2 respectively and are used to illustrate the functioning of the samonSummaryIM function.

**VSummary1 and VSummary2** are summary objects from the samonSummaryIM function and are used to illustrate the use of the samonDifferenceSummaryIM and samonCrossSummaryIM functions.

**VSummaryD** is the results from the samonDifferenceSummaryIM function and is used along with VSummary1 and VSummary2 to illustrate the alpha0Table and samonEYCompleterStatus functions.

### Source

Pre-calculated results from packaged data

---

samon

*Sensitivity Analysis for monotone missing data*

---

### Description

Given data from one arm of a repeated measures clinical trial, produces estimates of the expected value of the outcome at the final time-point for a range of sensitivity parameters.

### Usage

```
samon(mat, Npart = 10, InitialSigmaH = 1.0, HighSigmaH = 2.0,
InitialSigmaF = 1.0, HighSigmaF = 2.0, lb = 0, ub = 101, zeta1 = 1, zeta2 = 1,
NSamples = 0, seed0 = 1, MaxIter = 25, FAconv = 1E-7, FRconv = 1E-7,
SAconv = 1E-7, alphaList = c(0), MJackknife = FALSE, SJackknife = FALSE,
retIFiM = FALSE, retIFiS = FALSE, Tfun = NULL)
```

### Arguments

mat	a matrix with the (i,j) entry representing the outcome value for subject i at time-point j.
Npart	Number of partitions to use when estimating optimal smoothing parameters, sigma H and sigma F.
InitialSigmaH	Initial value when calculating optimal sigma H.
HighSigmaH	Upper bound of search region when calculating optimal sigma H.
InitialSigmaF	Initial value when calculation optimal sigma F.
HighSigmaF	Upper bound of search region when calculating optimal sigma F.
lb	Lower bound for Y.
ub	Upper bound for Y.
zeta1	parameter to cumulative beta.

zeta2	parameter to cumulative beta.
NSamples	Number of parametric bootstrap samples to generate.
seed0	Seed to use.
MaxIter	Maximum iterations to use in optimizer.
FAconvg	Absolute change in function convergence criterion.
FRconvg	Relative change in function convergence criterion.
SAconvg	Step size convergence criterion.
alphaList	a vector of sensitivity parameters
MJackknife	Jackknife main data (logical)
SJackknife	Jackknife bootstrap samples (logical)
retIFiM	return individual IF estimates from main data.
retIFiS	return individual IF estimates from samples.
Tfun	n by 2 matrix with tilting function values.

## Details

The matrix `mat` represents repeated measure outcome data from a single arm or treatment group of a trial. Each row represents the data from a single subject and each column data from a single time-point.

The values in the first column of `mat` are the baseline values and should not have missing values. Samon creates one-step bias corrected estimates of the mean value of  $Y$  at the last time-point for a number of specified sensitivity parameters,  $\alpha$ .

Samon determines two smoothing parameters,  $\sigma_H$ , which represents smoothing in the "missingness" model and,  $\sigma_F$ , which represents smoothing in the "outcome" model. These smoothing parameters are determined by minimizing loss functions. Minimization is performed using Newton's method. The parameter `InitialSigmaH` is used as the initial value in the optimization of the missingness model. In a similar vein `InitialSigmaF` is used as the initial value in the optimization of the outcome model.

A number of stopping criteria are available: `MaxIter`: the maximum number of iterations to perform `FAconvg`: stop when  $\text{abs}(f_{i+1} - f_i) < \text{FAconvg}$ , where  $f_i$  is the loss function value at iteration  $i$ . `FRconvg`: stop when  $\text{abs}((f_{i+1} - f_i)/(f_{i+1} + f_i)) < \text{FRconvg}$ . `SAconvg`: stop when the absolute step size falls below `SAconvg`, i.e.,  $\text{abs}(x_{i+1} - x_i) < \text{SAconvg}$ . `HighSigmaH`: should the value of  $\sigma_H$  go above this value, then the optimal value of  $\sigma_H$  is set to `HighSigmaH`. This is useful if larger values of  $\sigma_H$  do not change the missingness model substantially. `HighSigmaF`: should the value of  $\sigma_F$  go above this value, then the optimal value of  $\sigma_F$  is set to `HighSigmaF`. This is useful if larger values of  $\sigma_F$  do not change the missingness model substantially.

Using the optimal values for  $\sigma_H$  and  $\sigma_F$ , samon produces estimates of the mean value of the outcome at the last visit, for the set of sensitivity parameters given in the vector `alphaList`.

If the sensitivity parameters are applied to the cumulative beta function, `IB`. Four parameters are used to map the values of the outcome,  $y$  to the domain of the cumulative beta.  $r(y) = \text{IB}((y - \text{lb})/(\text{ub} - \text{lb}), \text{zeta1}, \text{zeta2})$  where `zeta1` and `zeta2` are the shape parameters.

Alternatively samon may be called with a user specified sensitivity function. This is given as an  $n$  by 2 matrix with the first column holding the unique values of the data in `mat` and the second column the value of the sensitivity function at that value.

**Value**

samon returns a list which includes the following:

HM	matrix of results from sigma H optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma H, and the loss function value at optimal sigma H. The "M" in the name "HM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.
FM	matrix of results from sigma F optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma F, and loss function value at optimal sigma F. The "M" in the name "FM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.

The return code takes the following values:

1	absolute function convergence was met
2	relative function convergence was met
3	second derivative has become too small
4	maximum iterations reached
5	value reset to HighSigmaH or HighSigmaF
6	loss function smaller at HighSigmaH or HighSigmaF
IFM	matrix with 7 columns containing one-step bias-corrected estimates (sometimes referred to here as IF estimates). Columns are sample, type, alpha, est0, se0, IFEst, IFVar. The columns sample and type take the value 0. IFEst and IFVar are the one-step bias corrected estimates and an estimate of their variance for the original data, mat, at each value of the sensitivity parameters, alpha.

In a similar manner, "HS", "FS", and "IFS" contain results from parametric bootstraps (should NSamples be greater than 0). The first columns in this case contain the sample number, 1, 2, ..., NSamples. In other respects these matrices have the same structure as the corresponding matrix "HM", "FM" and "IFM".

If MJackknife is set to TRUE, then, "HMjk", "FMjk", "IFMjk" are produced, holding the jackknife results for the main data mat. Again the first column "sample" is set to 0, and the second column is termed "Dropped" and gives the observation number dropped to form the jackknife sample.

Finally, if SJackknife is set to TRUE, then, "HSjk", "FSjk", "IFSjk" are produced. These hold the jackknife results for the bootstrap samples.

**See Also**

The samon\_userDoc.pdf file in the Examples subdirectory.

**Examples**

```
data("samonPANSS1")

Results1 <- samon(
  mat      = samonPANSS1,
  Npart    = 10, # number of partitions
```

```

InitialSigmaH =      15.0, # initial value
HighSigmaH    =      50.0, # high value for H

InitialSigmaF =       8.0, # initial value
HighSigmaF    =      50.0, # high value for F

lb            =       30, # parameters to
ub            =      210, # cumulative
zeta1         =       4.0, # beta distribution
zeta2         =       7.0,

alphaList     =      -1:1 )

```

---

samonCombine

*Combines results from separate runs of the samon function*


---

## Description

Combines lists returned from the samon function to form a single object.

## Usage

```
samonCombine( filenames, replaceSampleNo=TRUE, ForceAppend =
FALSE, objlist = NULL )
```

## Arguments

filenames	a vector of rds filenames each containing the output from samon
replaceSampleNo	A logical value determining if the sample numbers should be replaced in the output object. Usually the sample numbers are 1,2,...,NSamples in each file. When the results in filenames are from different runs of samon with different seeds, the results in the output are given unique sample numbers, usually 1, 2, ... NTot, where NTot is the total number of samples.
ForceAppend	put results together even if the files don't match on some criteria, such as the start values for sigma H or sigma F. Otherwise report the problem and return.
objlist	an alternative input of samon objects to be combined.

## Details

When the samon function is used in parallel a number of results are returned. If these results are stored in rds files, the samonCombine function can extract the results and combine them to produce a single object.

This is usually done for a single treatment arm at a time.



**Value**

Returns a list similar to that returned by samon itself. This list may be passed to samonSummary and other samon functions.

**Examples**

```
# outputs from samon -- run on samonPANSS1 with different seeds
data("P1Res001")
data("P1Res002")

# combine them
trt1Results <- samonCombine( objlist = list(P1Res001,P1Res002) )
```

---

samonCombineIM	<i>Combines results from separate runs of the samonIM function</i>
----------------	--

---

**Description**

Combines lists returned from the samonIM function to form a single object.

**Usage**

```
samonCombineIM( filenames, replaceSampleNo=TRUE, ForceAppend =
FALSE, objlist = NULL )
```

**Arguments**

filenames	a list of rds filenames each containing the output from samonIM
replaceSampleNo	A logical value determining if the sample numbers should be replaced in the output object. Usually the sample numbers are 1,2,...,NSamples in each file. When the results in filenames are from different runs of samon with different seeds, the results in the output are given unique sample numbers, usually 1, 2, ... NTot, where NTot is the total number of samples.
ForceAppend	put results together even if the files don't match on some criteria, such as the start values for sigma H or sigma F. Otherwise report the problem and return.
objlist	an alternative input of samon objects to be combined.

**Details**

When the samonIM function is used in parallel a number of results are returned. If these results are stored in rds files, the samonCombine function can extract the results and combine them to produce a single object.

This is usually done for a single treatment arm at a time.

**Value**

Returns a list similar to that returned by samonIM itself. This list may be passed to samonSummaryIM and other samon functions.

**Examples**

```
# outputs from samonIM -- run on VAS1 with different seeds
data("V1Res001")
data("V1Res002")

# combine them
V1Results <- samonCombineIM( objlist = list(V1Res001,V1Res002) )
```

---

samonCrossSummary	<i>Summary function for the difference in effect of two treatments at different sensitivity parameters alpha.</i>
-------------------	---

---

**Description**

Produces a summary of the difference in treatment effect using two samonSummary objects. Differences and confidence intervals are produced for each pair of sensitivity parameters.

**Usage**

```
samonCrossSummary( trt1, trt2, CIlevel = 0.95 )
```

**Arguments**

trt1	the result from a call to samonSummary
trt2	the result from a call to samonSummary
CIlevel	the confidence level for confidence intervals

**Details**

Combines estimates, bootstrap estimates and jackknife estimates to produce summary estimates and confidence intervals for the difference in estimates between two treatment groups. The difference is computed for all combinations of the smoothing parameter alpha found in the Sum1 input object.

The difference in treatment estimates is taken to be those of treatment 2 minus those of treatment 1.

**Value**

Returns a list. Items include TM which contains treatment estimates, their difference, and, standard error estimates and CI which contains confidence intervals. Other items are returned for convenience are n10, the number of subjects in the first treatment group, and n20 the number of subjects in the second treatment group.

## Examples

```
# samon results for treatment 1 and 2 respectively:
data("P1Results")
data("P2Results")

Summary1 <- samonSummary( P1Results )
Summary2 <- samonSummary( P2Results )
SummaryD <- samonDifferenceSummary( Summary1, Summary2 )
SummaryX <- samonCrossSummary( Summary1, Summary2 )
```

---

samonCrossSummaryIM     *Summary function for the difference in effect of two treatments at different sensitivity parameters.*

---

## Description

Produces a summary of the difference in treatment effect using two samonSummaryIM objects, Differences and confidence intervals are produced for each pair of sensitivity parameters.

## Usage

```
samonCrossSummaryIM( trt1, trt2, CIlevel = 0.95 )
```

## Arguments

trt1	the result from a call to samonSummaryIM for treatment 1
trt2	the result from a call to samonSummaryIM for treatment 2
CIlevel	the confidence level for confidence intervals

## Details

Combines bootstrap estimates to produce summary estimates and confidence intervals for the difference in estimates from two treatment groups. The difference is computed for all combinations of the smoothing parameter, alpha, found in the Sum1 input object.

The difference in treatment estimates is taken to be those of treatment 2 minus those of treatment 1.

## Value

Returns a list. This list includes TM which contains treatment estimates, their difference, and, standard error estimates. The list also contains, CI, which contains confidence intervals. Other items are returned for convenience are n10, the number of subjects in the first treatment group, and n20 the number of subjects in the second treatment group.

**Examples**

```
# V1Results and V2Results are the returned objects from the
# samonIM function associated with VAS treatment 1 and 2 respectively.
data("V1Results")
data("V2Results")

# summarize each arm, their difference and their cross difference.
VSummary1 <- samonSummaryIM(V1Results)
VSummary2 <- samonSummaryIM(V2Results)
VSummaryD <- samonDifferenceSummaryIM(VSummary1,VSummary2)
VSummaryX <- samonCrossSummaryIM(VSummary1,VSummary2)
```

---

samonDataCheck

*Samon Data Check*


---

**Description**

Checks data for intermittent missing or monotone missing pattern

**Usage**

```
samonDataCheck( data )
```

**Arguments**

data                    matrix with (i,j) entry representing value for subject i at time-point j.

**Details**

Returns a basic summary of samon input data. This includes a check that there is not missing data at baseline.

**Value**

Returns a list with items: N the number or rows in data, NT the number of columns in data, missingBaseline which is set to 1 if any baseline data is missing, and, interMissing, indicating if there are any intermittent data.

desc a matrix of dimension N by 4 with each row representing the same row in the original data. Columns are: col 1: 1 if non-missing baseline value col 2: last available time col 3: last available observation col 4: number of non missing values

Note that col2 - col4 gives the number of intermittent missing values.

missingPatterns, a list of missing patterns found in the data NmissingTables, frequencies of missing patterns, and PmissingTable the proportion of each missing pattern in the data.

## Examples

```
# monotone missing
data("samonPANSS1")
chkm <- samonDataCheck( samonPANSS1 )

# non monotone missing
data("VAS1")
chknm <- samonDataCheck( VAS1 )
```

---

samonDifferenceSummary

*Summary function for the difference in treatment effect*

---

## Description

Produces a summary of the difference in treatment effect using two samon objects. It combines bootstrap and jackknife results to produce confidence intervals

## Usage

```
samonDifferenceSummary( trt1, trt2, Clevel = 0.95 )
```

## Arguments

trt1	the result from a call to samonSummary
trt2	the result from a call to samonSummary
Clevel	the confidence level for confidence intervals

## Details

Combines estimates, bootstrap estimates and jackknife standard error estimates to produce summary estimates and confidence intervals for an estimate of the difference in treatment effect from two samon objects.

The difference in treatment estimates is taken to be those of treatment 2 minus those of treatment 1 at the same value of sensitivity parameter alpha.

## Value

Returns a list. Items include: TM which contains treatment estimates, standard errors for these estimates, the estimate of their difference and estimates of the standard error of the difference. Also contains CI which contains confidence intervals. Other items are returned for convenience are n10, the number of subjects in the first treatment group, and n20 the number of subjects in the second treatment group.

## Examples

```
# samon results for treatment 1 and 2 respectively:
data("P1Results")
data("P2Results")

Summary1 <- samonSummary( P1Results )
Summary2 <- samonSummary( P2Results )
SummaryD <- samonDifferenceSummary( Summary1, Summary2 )
SummaryX <- samonCrossSummary( Summary1, Summary2 )
```

---

samonDifferenceSummaryIM

*Summary function for the difference in treatment*

---

## Description

Produces a summary of the difference in treatment effect using two samonIM objects. It combines bootstrap imputed results to produce confidence intervals

## Usage

```
samonDifferenceSummaryIM( trt1, trt2, CIlevel = 0.95 )
```

## Arguments

trt1	the result from a call to samonSummaryIM
trt2	the result from a call to samonSummaryIM
CIlevel	the confidence level for confidence intervals

## Details

Combines estimates (imputed) bootstrap estimates to produce summary estimates and confidence intervals for an estimate of the difference in treatment effect from two samon objects.

The difference in treatment estimates is taken to be those of treatment 2 minus those of treatment 1 at the same value of sensitivity parameter alpha.

## Value

Returns a list. Items include: TM which contains treatment estimates and the difference, together with standard error estimates based on multiple imputation of bootstrap samples. It also contains CI which contains confidence intervals. Other items are returned for convenience are n10, the number of subjects in the first treatment group, and n20 the number of subjects in the second treatment group.

**Examples**

```
# V1Results and V2Results are the returned objects from the
# samonIM function associated with VAS treatment 1 and 2 respectively.
data("V1Results")
data("V2Results")

# summarize each arm, their difference and their cross difference.
VSummary1 <- samonSummaryIM(V1Results)
VSummary2 <- samonSummaryIM(V2Results)
VSummaryD <- samonDifferenceSummaryIM(VSummary1,VSummary2)
VSummaryX <- samonCrossSummaryIM(VSummary1,VSummary2)
```

---

samonECompleterStatus *Computes the difference in the expected value of non-completers and completers*

---

**Description**

Computes the difference in the expected value of non-completers and completers

**Usage**

```
samonECompleterStatus( Y, BC )
```

**Arguments**

Y	vector of outcomes
BC	a vector of estimates for the mean of Y

**Details**

Computes the expected value for completers and non-completers and hence the difference between the two.

**Value**

Returns a vector of expected differences.

**Examples**

```
data("samonPANSS1")
data( "PSummary1" )
PSummary1 <- (PSummary1$TM)[,c("alpha","IFEst")]
Pdiffer1 <- samonECompleterStatus( samonPANSS1[,ncol(samonPANSS1)], PSummary1[,2] )

data("VAS1")
data("VSummary1" )
VSummary1 <- (VSummary1$TM)[,c("alpha","IFEst")]
Vdiffer1 <- samonECompleterStatus( VAS1[,ncol(VAS1)], VSummary1[,2] )
```

---

`samoneval`*Evaluates the loss function on a range of sigmas*

---

### Description

Computes the loss function for a range of sigma

### Usage

```
samoneval( mat, Npart = 10, sigmaList = c(1), type="both")
```

### Arguments

<code>mat</code>	matrix with (i,j) entry representing value for subject i at time-point j.
<code>Npart</code>	Data is partitioned into Npart parts when estimating lossH and lossF.
<code>sigmaList</code>	vector of sigmas at which to evaluate the loss function.
<code>type</code>	one of "H", "F" or "both" to evaluate lossH, lossF or both for sigmaH or sigmaF in indicated range.

### Details

Stepping through the values in `sigmaList` this functions computes the associated loss function. If `type` is set to "H" or "h", the loss function associated with dropout is computed. If `type` is set to "F" or "f" the loss function associated with outcome is computed. If `type` is set any other value then both types of loss function are evaluated.

### Value

Returns an N by 2 matrix of results if `type` is "H" or "F" or an N by 4 matrix if `type` is "both". There is one row for each value of sigma. If `type` is "H" then the columns are `sigmaH` and `lossH`, if `type` is "F" then the columns are `sigmaF` and `lossF`, and, if `type` = "both" then the columns are `sigmaH`, `lossH`, `sigmaF`, and `lossF`.

### Examples

```
data("samonPANSS1")
## dropout loss function
H1 <- samoneval( mat = samonPANSS1, Npart = 5,
  sigmaList      = seq(1,10,by=1),
  type           = "H" )
```



---

samonevalIM	<i>Evaluates the loss function at a range of smoothing parameters</i>
-------------	---

---

**Description**

Computes the loss function for a range of smoothing parameters

**Usage**

```
samonevalIM( mat, Npart = 10, sigmaList = c(1), inmodel =
inmodel, seed = 1, type="both")
```

**Arguments**

mat	matrix with (i,j) entry representing value for subject i at time-point j.
Npart	number of parts to partition the data when evaluating the loss functions lossH and lossF.
sigmaList	vector of sigmas at which to evaluate the loss function.
inmodel	NT by 6 matrix indicating which variables to include in logistic model.
seed	seed to use when imputing intermittent data.
type	one of "H", "F" or "both" to evaluate lossH, lossF or both for sigmaH or sigmaF in indicated range.

**Details**

Stepping through the values in sigmaList this functions computes the values of a loss function. If type is set to "H" or "h" then the loss function associated with dropout is computed. If type is set to "F" or "f" then the loss function associated with outcome is computed. If type is set any other value then both types of loss function are evaluated.

Since the input matrix, mat, can possess intermittent missing data, samonevalIM imputes these values using a logistic model. This is the model specified by inmodel.

inmodel is an NT by 6 matrix and contains a row for each time-point represented in the mat matrix. Since there are no intermittent missing data at time 1 and at time NT these rows should be set to zero.

A separate logistic model is fit for each timepoint t0, with t0 ranging from 2 to NT-1. The dependent variable in these models is an indicator indicating that an individual provided data at time t0. Only individuals who are on study at time t0+1 are considered in these models. Setting the ith column in inmodel to 1 indicates that the following terms should be included in the model:

- 1 the intercept.
- 2 the previous value of Y, that is the value of Y at t0-1.
- 3 the next available value of Y after time t0.
- 4 if the next available value of Y is at time t1, then this is (t1 - t0 - 1).
- 5 an interaction between the variable in column 3 and column 4.

**6** 0 if an individual is last observed at time NT, 1 if they are last observed at time NT - 1, 2 if they are last observed at time NT - 2, etc.

For the model that runs at the penultimate time, NT-1, including all six terms in the model will lead to over specification.

To include as many terms as possible in each model set inmodel to be

```

0 0 0 0 0 0
1 1 1 1 1 1
1 1 1 1 1 1
. . .
. . .
1 1 1 1 1 1
1 1 1 0 0 0
0 0 0 0 0 0

```

At a given timepoint t0, there may not be enough intermittent missing data to support all six terms. In such cases variables are removed from the model until the regression converges. If less than 4 observations have intermittent missing data at a given timepoint a model fitting the mean is used.

samonevalIM sequentially imputes intermittent missing data using these models. Once the intermittent data is imputed a loss function is evaluated on the data.

### Value

Returns an N by 2 matrix of results if type is "H" or "F" or an N by 4 matrix if type is "both". There is one row for each value of sigma. If type is "P" then the columns are sigmaH and lossH, if type is "F" then the columns are sigmaF and lossF, and, if type = "both" then the columns are sigmaH, lossH, sigmaF, and lossF.

### Examples

```

data("VAS1")

## imputation model.
NT          <- ncol(VAS1)
inmodel     <- matrix(1,NT,6)
inmodel[1,] <- 0
inmodel[NT,] <- 0
inmodel[NT-1,4:6] <- 0

## outcome loss function
F1 <- samonevalIM( mat = VAS1, Npart = 10,
                  sigmaList = seq(1,10,by=1),
                  inmodel = inmodel,
                  seed = 26,
                  type = "F" )

```

---

 samonGen

*Sensitivity Analysis for monotone missing data*


---

### Description

Given data from one arm of a repeated measures clinical trial, produces a sample based on the optimal smoothing parameters.

### Usage

```
samonGen(mat, Npart = 10, InitialSigmaH = 1.0, HighSigmaH = 2.0,
InitialSigmaF = 1.0, HighSigmaF = 2.0, NSamples = 0, seed0 = 1,
MaxIter = 25, FAconvg = 1E-7, FRconvg = 1E-7, SAconvg = 1E-7)
```

### Arguments

mat	a matrix with the (i,j) entry representing the outcome value for subject i at time-point j.
Npart	Number of partitions to use when estimating optimal smoothing parameters, sigma H and sigma F.
InitialSigmaH	Initial value when calculating optimal sigma H.
HighSigmaH	Upper bound of search region when calculating optimal sigma H.
InitialSigmaF	Initial value when calculating optimal sigma F.
HighSigmaF	Upper bound of search region when calculating optimal sigma F.
NSamples	Number of parametric bootstrap samples to generate.
seed0	Seed to use.
MaxIter	Maximum iterations to use in optimizer.
FAconvg	Absolute change in function convergence criterion.
FRconvg	Relative change in function convergence criterion.
SAconvg	Step size convergence criterion.

### Details

The matrix `mat` represents repeated measure outcome data from a single arm or treatment group of a trial. Each row represents the data from a single subject and each column data from a single time-point.

The values in the first column of `mat` are the baseline values and should not be missing.

Samon determines two smoothing parameters, sigma H, which represents smoothing in the "missingness" model and, sigma F, which represents smoothing in the "outcome" model. These smoothing parameters are determined by minimizing loss functions. Minimization is performed using Newton's method. The parameter `InitialSigmaH` is used as the initial value in the optimization of the missingness model, and, `InitialSigmaF` is used as the initial value in the optimization of the outcome model.

A number of stopping criteria are available: MaxIter: the maximum number of iterations to perform FAconv: stop when  $\text{abs}(f_{i+1} - f_i) < \text{FAconv}$ , where  $f_i$  is the loss function value at iteration  $i$ . FRconv: stop when  $\text{abs}((f_{i+1} - f_i)/(f_{i+1} + f_i)) < \text{FRconv}$ . SAconv: stop when the absolute step size falls below SAconv, i.e.,  $\text{abs}(x_{i+1} - x_i) < \text{SAconv}$ . HighSigmaH: should the value of sigma H go above this value, then the optimal value of sigma H is set to HighSigmaH. This is useful if larger values of sigma H do not change the missingness model substantially. HighSigmaF: should the value of sigma F go above this value, then the optimal value of sigma F is set to HighSigmaF. This is useful if larger values of sigma F do not change the missingness model substantially.

## Value

samonGen returns a list which includes the following:

HM	matrix of results from sigma H optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma H, and the loss function value at optimal sigma H. The "M" in the name "HM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.
FM	matrix of results from sigma F optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma F, and loss function value at optimal sigma F. The "M" in the name "FM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.

The return code takes the following values:

1	absolute function convergence was met
2	relative function convergence was met
3	second derivative has become too small
4	maximum iterations reached
5	value reset to HighSigmaH or HighSigmaF
6	loss function smaller at HighSigmaH or HighSigmaF
Sample	a matrix of size Nsamples by NT with the sample

## See Also

The samon\_userDoc.pdf file in the Examples subdirectory.

## Examples

```
data(samonPANSS1)

Sample1 <- samonGen(
  mat      = samonPANSS1,
  Npart    = 5, # number of partitions

  InitialSigmaH = 15.6, # initial value
  HighSigmaH    = 25.0, # high value for H

  InitialSigmaF = 8.6, # initial value
  HighSigmaF    = 15.0, # high value for F
```

```

seed      =      211,
NSamples  =      30 )

```

samonGenIM

*Sensitivity Analysis for monotone missing data***Description**

Given data from one arm of a repeated measures clinical trial with intermittent missing data, produces a sample based on the optimal smoothing parameters.

**Usage**

```

samonGenIM(mat, Npart = 10, InitialSigmaH = 1.0, HighSigmaH = 2.0,
InitialSigmaF = 1.0, HighSigmaF = 2.0, inmodel = inmodel, NSamples = 0,
seed0 = 1, seed1 = 1, MaxIter = 25, FAconvg = 1E-7, FRconvg = 1E-7,
SAconvg = 1E-7)

```

**Arguments**

mat	a matrix with the (i,j) entry representing the outcome value for subject i at time-point j.
Npart	Number of partitions to use when estimating optimal smoothing parameters, sigma H and sigma F.
InitialSigmaH	Initial value when calculating optimal sigma H.
HighSigmaH	Upper bound of search region when calculating optimal sigma H.
InitialSigmaF	Initial value when calculating optimal sigma F.
HighSigmaF	Upper bound of search region when calculating optimal sigmaF.
inmodel	NT by 6 matrix indicating which variables to include in logistic model.
NSamples	Number of parametric bootstrap samples to generate.
seed0	Seed to use when generating bootstrap samples.
seed1	Seed to use when imputing intermittent missing data.
MaxIter	Maximum iterations to use in optimizer.
FAconvg	Absolute change in function convergence criterion.
FRconvg	Relative change in function convergence criterion.
SAconvg	Step size convergence criterion.

## Details

The matrix `mat` represents repeated measure outcome data from a single arm or treatment group of a trial. Each row represents the data from a single subject and each column data from a single time-point.

The values in the first column of `mat` are the baseline values and should not be missing.

Samon determines two smoothing parameters,  $\sigma_H$ , which represents smoothing in the "missingness" model and,  $\sigma_F$ , which represents smoothing in the "outcome" model. These smoothing parameters are determined by minimizing loss functions. Minimization is performed using Newton's method. The parameter `InitialSigmaH` is used as the initial value in the optimization of the missingness model, and, `InitialSigmaF` is used as the initial value in the optimization of the outcome model.

A number of stopping criteria are available: `MaxIter`: the maximum number of iterations to perform `FAconvg`: stop when  $\text{abs}(f_{i+1} - f_i) < \text{FAconvg}$ , where  $f_i$  is the loss function value at iteration  $i$ . `FRconvg`: stop when  $\text{abs}((f_{i+1} - f_i)/(f_{i+1} + f_i)) < \text{FRconvg}$ . `SAconvg`: stop when the absolute step size falls below `SAconvg`, i.e.,  $\text{abs}(x_{i+1} - x_i) < \text{SAconvg}$ . `HighSigmaH`: should the value of  $\sigma_H$  go above this value, then the optimal value of  $\sigma_H$  is set to `HighSigmaH`. This is useful if larger values of  $\sigma_H$  do not change the missingness model substantially. `HighSigmaF`: should the value of  $\sigma_F$  go above this value, then the optimal value of  $\sigma_F$  is set to `HighSigmaF`. This is useful if larger values of  $\sigma_F$  do not change the missingness model substantially.

Since the input matrix, `mat`, can possess intermittent missing data, `samonevalIM` imputes these values using a logistic model. This is the model specified by `inmodel`.

`inmodel` is an  $NT$  by 6 matrix and contains a row for each time-point represented in the `mat` matrix. Since there are no intermittent missing data at time 1 and at time  $NT$  these rows should be set to zero.

A separate logistic model is fit for each timepoint  $t_0$ , with  $t_0$  ranging from 2 to  $NT-1$ . The dependent variable in these models is an indicator indicating that an individual provided data at time  $t_0$ . Only individuals who are on study at time  $t_0+1$  are considered in these models. Setting the  $i$ th column in `inmodel` to 1 indicates that the following terms should be included in the model:

- 1 the intercept.
- 2 the previous value of  $Y$ , that is the value of  $Y$  at  $t_0-1$ .
- 3 the next available value of  $Y$  after time  $t_0$ .
- 4 if the next available value of  $Y$  is at time  $t_1$ , then this is  $(t_1 - t_0 - 1)$ .
- 5 an interaction between the variable in column 3 and column 4.
- 6 0 if an individual is last observed at time  $NT$ , 1 if they are last observed at time  $NT - 1$ , 2 if they are last observed at time  $NT - 2$ , etc.

For the model that runs at the penultimate time,  $NT-1$ , including all six terms in the model will lead to over specification.

To include as many terms as possible in each model set `inmodel` to be

```

0 0 0 0 0 0
1 1 1 1 1 1
1 1 1 1 1 1
. . .

```

```

      .   .   .
      1   1   1   1   1   1
      1   1   1   0   0   0
      0   0   0   0   0   0

```

At a given timepoint  $t_0$ , there may not be enough intermittent missing data to support all six terms. In such cases variables are removed from the model until the regression converges. If less than 4 observations have intermittent missing data at a given timepoint a model fitting the mean is used.

samonGenIM sequentially imputes intermittent missing data using these models. Once the intermittent data is imputed a loss function is evaluated on the data.

### Value

samonGen returns a list which includes the following:

HM	matrix of results from sigma H optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma H, and the loss function value at optimal sigma H. The "M" in the name "HM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.
FM	matrix of results from sigma F optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma F, and loss function value at optimal sigma F. The "M" in the name "FM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.

The return code takes the following values:

1	absolute function convergence was met
2	relative function convergence was met
3	second derivative has become too small
4	maximum iterations reached
5	value reset to HighSigmaH or HighSigmaF
6	loss function smaller at HighSigmaH or HighSigmaF
Sample	a matrix of size Nsamples by NT with the sample

### See Also

The samon\_userDoc.pdf file in the Examples subdirectory.

### Examples

```

data("VAS1")

# define the imputation models
NT      <- ncol(VAS1)
inmodel <- matrix(1,NT,6)
inmodel[1,] <- 0
inmodel[NT,] <- 0

```

```

inmodel[NT-1,4:6] <- 0

Sample1 <- samonGenIM(
  mat          = VAS1, # imput matrix
  Npart        = 5, # number of partitions

  InitialSigmaH = 27.9, # initial value
  HighSigmaH   = 100.0, # high value for H

  InitialSigmaF = 7.3, # initial value
  HighSigmaF    = 100.0, # high value for F

  NSamples     = 12, # number of samples

  seed0        = 441, # sample seed
  seed1        = 511, # impute seed
  inmodel      = inmodel) # input model

```

---

samonIM	<i>samonIM: Sensitivity analysis for monotone missing and intermittent missing data</i>
---------	---

---

## Description

Given data from one arm of a repeated measures clinical trial, produces estimates of the expected value of the outcome at the final time-point for a range of sensitivity parameters.

## Usage

```

samonIM(mat, Npart = 10, InitialSigmaH = 1.0, HighSigmaH = 2.0,
InitialSigmaF = 1.0, HighSigmaF = 2.0, inmodel = inmodel, NSamples = 0,
NIMimpute = 1, lb = 0, ub = 101, zeta1 = 1, zeta2 = 1, seed0 = 1, seed1 =
1, MaxIter = 25, FAconvg = 1E-7, FRconvg = 1E-7, SAconvg = 1E-7,
alphaList = c(0), retIFiM = FALSE, retIFiS = FALSE, retSample = FALSE,
retFMatM = FALSE, retFMatS = FALSE, Tfun=NULL)

```

## Arguments

mat	a matrix with the (i,j) entry representing the outcome value for subject i at time-point j.
Npart	Number of partitions to use when estimating optimal smoothing parameters, sigma H and sigma F.
InitialSigmaH	Initial value when calculating optimal sigma H.
HighSigmaH	Upper bound of search region when calculating optimal sigma H.
InitialSigmaF	Initial value when calculation optimal sigma F.
HighSigmaF	Upper bound of search region when calculation optimal sigma F.
NSamples	Number of parametric bootstrap samples to generate.



inmodel	NT by 6 matrix indicating which variables to include in logistic model.
NIMimpute	How many times to fill in intermittent missing data.
lb	Lower bound for Y.
ub	Upper bound for Y.
zeta1	parameter to cumulative beta.
zeta2	parameter to cumulative beta.
seed0	Seed to use for parametric bootstrap sampling
seed1	Seed to use for filling intermittent missing data
MaxIter	Maximum iterations to use in optimizer.
FAconvg	Absolute change in function convergence criterion.
FRconvg	Relative change in function convergence criterion.
SAconvg	Step size convergence criterion.
alphaList	a vector of sensitivity parameters
retIFiM	return individual IF estimates from main data.
retIFiS	return individual IF estimates from sample data.
retSample	return the Sample generated
retFMatM	return the main data with intermittent missing data filled in NIMimpute + 1 times
retFMatS	retrun the sample data with intermittent missing data filled in NIMimpute + 1 times
Tfun	n by 2 matrix with tilting function values.

## Details

The matrix `mat` represents repeated measure outcome data from a single arm or treatment group of a trial. Each row represents the data from a single subject and each column data from a single time-point.

The values in the first column of `mat` are the baseline values and should not be missing. Samon creates bias-corrected estimates of the mean value of Y at the last time-point for a number of specified sensitivity parameters, `alpha`.

Samon determines two smoothing parameters, `sigma H`, which represents smoothing in the "missingness" model and, `sigma F`, which represents smoothing in the "outcome" model. These smoothing parameters are determined by minimizing loss functions. Minimization is performed using Newton's method. The parameter `InitialSigmaH` is used as the initial value in the optimization of the missingness model and `InitialSigmaF` is used as the initial value in the optimization of the outcome model.

A number of stopping criteria are available: `MaxIter`: the maximum number of iterations to perform `FAconvg`: stop when  $\text{abs}(f_{i+1} - f_i) < \text{FAconvg}$ , where  $f_i$  is the loss function value at iteration  $i$ . `FRconvg`: stop when  $\text{abs}((f_{i+1} - f_i)/(f_{i+1} + f_i)) < \text{FRconvg}$ . `SAconvg`: stop when the absolute step size falls below `SAconvg`, i.e.,  $\text{abs}(x_{i+1} - x_i) < \text{SAconvg}$ . `HighSigmaH`: should the value of `sigma H` go above this value, then the optimal value of `sigma H` is set to `HighSigmaH`. This is useful if larger values of `sigma H` do not change the missingness model substantially. `HighSigmaF`: should

the value of sigma F go above this value, then the optimal value of sigma F is set to HighSigmaF. This is useful if larger values of sigma F do not change the missingness model substantially.

The inmodel matrix specifies a set of logistic models used to impute intermittent missing data. This is an NT by 6 matrix of 0s and 1s. Each row represents a time-point. Since intermittent missing data cannot occur at baseline or at the final timepoint, the first and NT'th row of inmodel are 0. The six values on a row indicate the following terms should be included in the model:

- 1** a value of 1 in this position indicates that the logistic model should include an intercept
- 2** a value of 1 in this position indicates that the model should contain a term for the previous available outcome
- 3** a value of 1 in this position indicates that the model should contain a term for the next available outcome
- 4** a value of 1 in this position indicates that the model should contain a term for the location of the next available outcome
- 5** a value of 1 in this position indicates that the model should contain a term for the interaction between the next available outcome and the position of the next available outcome
- 6** a value of 1 in this position indicates that the model should contain a term for the last available outcome before dropout.

For example suppose the data consists of the 8 columns v1-v8 and suppose we wish to model the intermittent missingness at the third time-point, that is, we wish to model when v3 has an intermittent missing value. We might model this as  $\text{is.na}(v3) \sim 1 + v2 + n3 + i3 + i3 * n3 + l3$  where n3 refers to the next non-missing value of v after the third time-point. i3 refers to the location of that non-missing value, i3 \* n3 and interaction between these and l3 the last available value of v for an individual. Specifically we might calculate n3, i3 and l3 as shown:

v1	v2	v3	v4	v5	v6	v7	v8	v2	n3	i3	l3
10	8	NA	9	32	NA	NA	NA	8	9	1	32
11	4	3	5	4	7	7	NA	4	5	1	7
22	15	NA	NA	NA	6	2	9	15	6	3	9

Note that when considering missing values at the NT - 1 time-point the last three terms in the model are not identifiable.

Fitting as many terms in the intermittent missing model might use an inmodel parameter with a matrix like this:

0	0	0	0	0	0
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	0	0	0
0	0	0	0	0	0

A simpler model using only the intercept and the previous and next available non-missing value would be specified by

```

0 0 0 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0

```

This simpler model but restricted to an intercept only model at the second and fourth timepoints can be specified by using:

```

0 0 0 0 0 0
1 0 0 0 0 0
1 1 1 0 0 0
1 0 0 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0

```

In practice the amount of intermittent missing data at a timepoint may not be large enough to support a logistic model with all six parameters. At each step samonIM will attempt to fit the models indicated by the inmodel matrix and reduce the number of model parameters if collinearity is found among the independent variables.

## Value

samon returns a list which includes the following:

HM	matrix of results from sigma H optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma H, and the loss function value at optimal sigma H. The "M" in the name "HM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.
FM	matrix of results from sigma F optimization for the input data mat. Columns are sample, type, return code, iterations, optimal sigma F, and loss function value at optimal sigma F. The "M" in the name "FM" refers to the main or input matrix mat. In this case the sample and type columns are set to 0.

The return code takes the following values:

1. absolute function convergence was met
2. relative function convergence was met

3. second derivative has become too small
4. maximum iterations reached
5. value reset to HighSigmaH or HighSigmaF
6. loss function smaller at HighSigmaH or HighSigmaF

Sample                    The generated sample of dimension NSamples by ncol(Mat)

### See Also

The samon\_userDoc.pdf file in the Examples subdirectory.

### Examples

```
data("VAS1")

# inputation model
NT      <- ncol(VAS1)
inmodel <- matrix(1,NT,6)
inmodel[1,] <- 0
inmodel[NT,] <- 0
inmodel[,4:6] <- 0

Results <- samonIM(
  mat      = VAS1, # input matrix
  Npart    = 2, # number of partitions

  InitialSigmaH = 25.0, # initial value
  HighSigmaH    = 100.0, # high value for H

  InitialSigmaF = 8.0, # initial value
  HighSigmaF    = 100.0, # high value for F

  lb      = 0, # parameters for
  ub      = 102, # cumulative
  zeta1   = 1.2, # beta distribution
  zeta2   = 1.6,

  NSamples = 0, # no of bootstraps
  NIMimpute = 2, # no of imputations

  seed0    = 441, # seed for bootstraps
  seed1    = 511, # seed for imputations
  inmodel  = inmodel, # input model

  alphaList = -1:1 )
```

---

`samonSummary`*Summary function for an object returned by the samon function*

---

**Description**

Produces summaries of samon objects, combining bootstrap and jackknife results to produce confidence intervals

**Usage**

```
samonSummary( trt, CIlevel = 0.95 )
```

**Arguments**

`trt`                    the result from a call to `samon` or `samonCombine`  
`CIlevel`                the confidence level for confidence intervals

**Details**

Combines estimates, bootstrap estimates and jackknife estimates to produce summary estimates and confidence intervals for a samon object.

**Value**

Returns a list. Items include `TM` which contains treatment estimates and standard errors for the main data; `TS` contains estimates for the bootstrap samples and includes jackknifes standard error estimates; `CI` contains confidence intervals.

**Examples**

```
data("P1Results")  
Summary1 <- samonSummary( P1Results )
```

---

`samonSummaryIM`*Summary function for an object returned by the samonIM function*

---

**Description**

Produces summaries of samonIM objects, combining multiple imputed results to produce confidence intervals

**Usage**

```
samonSummaryIM( trt, CIlevel = 0.95 )
```

**Arguments**

trt                    the result from a call to samonIM or samonCombineIM  
 CIlevel                the confidence level for confidence intervals

**Details**

Combines multiple imputation estimates and multiple imputation bootstrap estimates to produce summary estimates and confidence intervals for a samonIM object.

**Value**

Returns a list. Items include TM which contains treatment estimates and multiple imputation standard errors for the main data; TS contains bootstrap estimates and multiple imputation standard error bootstrap estimates. CI contains confidence intervals.

**Examples**

```
# V1Results and V2Results are the returned objects from the
# samonIM function associated with VAS treatment 1 and 2 respectively.
data("V1Results")
data("V2Results")

# summarize each arm, their difference and their cross difference.
VSummary1 <- samonSummaryIM(V1Results)
VSummary2 <- samonSummaryIM(V2Results)
VSummaryD <- samonDifferenceSummaryIM(VSummary1,VSummary2)
VSummaryX <- samonCrossSummaryIM(VSummary1,VSummary2)
```

---

samonTable1

*Summarizes two treatment groups.*

---

**Description**

samonTabmat1 takes a dataset Y and produces simple summary statistics which can be printed with the samonTable1 function.

**Usage**

```
samonTabmat1( Y )
samonTable1( Tabdata, trtlab = "Treatment 1")
```

**Arguments**

Y                      Data frame or matrix of data. Rows are individual measurement and columns represent time-points.  
 Tabdata                Matrix of summary stats produced by the samonTabmat1 function.  
 trtlab                 Label to place above table.

**Value**

samonTabmat1 returns an NT by 10 matrix where NT is the number of columns in the input matrix Y. The output matrix contains the columns 1. t the time-point. 2. Number On Study 3. Number Observed 4. Number last seen at time t 5. Proportion last seen (of number on-study) 6. Proportion last seen (of number observed) 7. Number intermittent missing data 8. proportion intermittent missing (of number on-study) 9. mean observed value at time t 10. std observed value at time t

samonTable1 does not return any object.

**Examples**

```
# monotone missing
data("samonPANSS1")
samonTable1( samonTabmat1( samonPANSS1 ), trtlab = "PANSS Placebo" )

# non monotone missing
data("VAS1")
samonTable1( samonTabmat1( VAS1 ), trtlab = "VAS Placebo" )
```

---

The PANSS trial.

*The PANSS randomized trial.*

---

**Description**

These dataframes store data from a repeated measures clinical trial. samonPANSS1 contains data from the placebo arm and samonPANSS2 contains data from the active arm.

**Usage**

```
data(samonPANSS1)
data(samonPANSS2)
```

**Format**

samonPANSS1 is a 88 by 6 dataframe samonPANSS2 is a 86 by 6 dataframe

**Details**

In each of samonPANSS1 and samonPANSS2 a row represents the outcome measure (PANSS) for one subject at each of 6 timepoints.

- V1 PANSS score at baseline
- V2 PANSS score at follow-up time-point 1
- V3 PANSS score at follow-up time-point 2
- ...
- V6 PANSS score at follow-up time-point 5 the last time-point.

**Source**

Data are simulated for the samon package

---

The DepWork trial      *The depression and working status trial.*

---

**Description**

These dataframes store data from a placebo controlled repeated measures clinical trial. DepWork1 contains data from the control arm and DepWork2 contains data from the active arm.

**Usage**

```
data(DepWork1)
data(DepWork2)
```

**Format**

DepWork1 is a 552 by 4 dataframe DepWork2 is a 1249 by 4 dataframe

**Details**

In each of DepWork1 and DepWork2 a row represents the outcome measure (working status) for one subject at each of 4 timepoints. At a given timepoint, working status is coded as 1 if an individual is not employed at that timepoint, and, as 2 if the individual is employed. No individual was working at baseline. Both treatment arms contain intermittent missing values.

- V1 work status at baseline
- V2 work status at follow-up time-point 1
- V3 work status at follow-up time-point 2
- V4 work status at follow-up time-point 3

---

The VAS trial.      *The VAS randomized trial.*

---

**Description**

These dataframes store data from a repeated measures clinical trial. VAS1 contains data from the control arm and VAS2 contains data from the active arm.

**Usage**

```
data(VAS1)
data(VAS2)
```



**Format**

VAS1 is a 255 by 9 dataframe VAS2 is a 256 by 9 dataframe

**Details**

In each of VAS1 and VAS2 a row represents the outcome measure (VAS) for one subject at each of 9 timepoints. Both treatment arms contain intermittent missing values.

- V1 VAS score at baseline
- V2 VAS score at follow-up time-point 1
- V3 VAS score at follow-up time-point 2
- ...
- V9 VAS score at follow-up time-point 8 the last time-point.

# Index

alpha0TableResults (alpha0Tables), [2](#)  
alpha0TableResultsIM (alpha0TablesIM), [3](#)  
alpha0Tables, [2](#)  
alpha0TablesIM, [3](#)

DepWork1 (The DepWork trial), [32](#)  
DepWork2 (The DepWork trial), [32](#)

ManData, [4](#)

P1Res001 (ManData), [4](#)  
P1Res002 (ManData), [4](#)  
P1Results (ManData), [4](#)  
P2Results (ManData), [4](#)  
PSummary1 (ManData), [4](#)  
PSummary2 (ManData), [4](#)  
PSummaryD (ManData), [4](#)

samon, [5](#)  
samonCombine, [8](#)  
samonCombineIM, [9](#)  
samonCrossSummary, [10](#)  
samonCrossSummaryIM, [11](#)  
samonDataCheck, [12](#)  
samonDifferenceSummary, [13](#)  
samonDifferenceSummaryIM, [14](#)  
samonECompleterStatus, [15](#)  
samoneval, [16](#)  
samonevalIM, [17](#)  
samonGen, [19](#)  
samonGenIM, [21](#)  
samonIM, [24](#)  
samonPANSS1 (The PANSS trial.), [31](#)  
samonPANSS2 (The PANSS trial.), [31](#)  
samonSummary, [29](#)  
samonSummaryIM, [29](#)  
samonTable1, [30](#)  
samonTabmat1 (samonTable1), [30](#)

The DepWork trial, [32](#)  
The PANSS trial., [31](#)  
The VAS trial., [32](#)

V1Res001 (ManData), [4](#)  
V1Res002 (ManData), [4](#)  
V1Results (ManData), [4](#)  
V2Results (ManData), [4](#)  
VAS1 (The VAS trial.), [32](#)  
VAS2 (The VAS trial.), [32](#)  
VSummary1 (ManData), [4](#)  
VSummary2 (ManData), [4](#)  
VSummaryD (ManData), [4](#)