Package 'MF'

January 16, 2026

Type Package

Title Mitigated Fraction

Version 4.4.11

Date 2024-07-25

Description Calculate MF (mitigated fraction) with clustering and bootstrap
options. See http://goo.gl/pcXYVr for definition of MF. No endorsement,
claim, or warranty is implied for this package. It is made available for
investigational or pedagogical use only.

License file LICENSE

URL https://github.com/ABS-dev/MF/, https://abs-dev.github.io/MF/

BugReports https://github.com/ABS-dev/MF/issues/

LazyLoad true

LazyData true

Depends R (>= 4.2)

Imports methods,
dplyr (>= 0.7.1),
plyr,
stringr,
tidyr (>= 1.3.0),
forcats,
purrr,
rlang (>= 1.1.1)

Suggests rmarkdown,
tidyverse,
kableExtra,
testthat,
covr,
knitr,
R.rsp

Collate 'aaa.r'
'classes.r'
'generic_methods.r'
'MF.r'
'MF-package.r'
'HLBoot.r'

'MFBoot.r'
'MFClus.r'
'MFClusBoot.r'
'MFClusHier.r'
'MFmp.r'
'MFnestBoot.r'
'MFr.r'
'MFSubj.r'
'MFHier-wrappers.r'

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder R.rsp

Contents

---

calflung	*calflung dataset*

Description

Post-mortem examination of the lungs of groups of calves.

Format

a data frame with 50 observations of the following 2 variables, no NAs

- group: Treatment group. One of con = control or vac = vaccinate
- lesion: Fraction of lungs with gross lesions.

HLBoot	*Bootstrap CI for MF, HL, and Qdif*

Description

Estimates bootstrap confidence intervals for MF, HL, and Qdif.

Usage

```
HLBoot(
  formula,
  data,
  compare = c("con", "vac"),
  b = 100,
  B = 100,
  alpha = 0.05,
  hpd = TRUE,
  bca = FALSE,
  return.boot = FALSE,
  trace.it = FALSE,
  seed = sample(1:1e+05, 1)
)
```

Arguments

formula	Formula of the form y ~ x + cluster(w), where y is a continuous response, x is a factor with two levels of treatment, and w is a factor indicating the clusters.
data	Data frame
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared
b	Number of bootstrap samples to take with each cycle
B	Number of cycles, giving the total number of samples = B * b
alpha	Complement of the confidence level
hpd	Boolean whether to estimate highest density intervals for MF and HL.

bca	Boolean whether to estimate BCa intervals for MF.
return.boot	Boolean whether to save the bootstrap samples of the statistics.
trace.it	Boolean whether to display verbose tracking of the cycles.
seed	to initialize random number generator for reproducibility. Passed to set.seed.

Details

Estimates bootstrap confidence intervals for the mitigated fraction (MF), Hodge-Lehmann estimator (HL), and the difference of medians and quartiles (Qdif). Equal tailed intervals are provided for all three, highest density intervals are optionally provided for MF and HL, and BCa intervals are optionally provided for MF. The Hodges-Lehmann estimator is the median difference; it assumes that the two distributions have the same shape and differ by a constant shift. Assumes data is single pool (no nesting).

Value

a mfhlboot data object

Author(s)

MF-package

References

Hodges JL, Lehmann EL, (1963). Estimates of location based on rank tests. *Annals of Mathematical Statistics.* **34:598–611**.

Siev D, (2005). An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* **4:500–508**.

Efron B, Tibshirani RJ. *An Introduction to the Bootstrap.* Chapman and Hall, New York, 1993.

See Also

mfhlboot

Examples

```
HLBoot(lesion ~ group, calflung, seed = 12345)

# Bootstrapping
# . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
# . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
# . . . . . . . . . . . . . . . . . . . . . . . .
#
# 10000 bootstrap samples
# 95% confidence intervals
# Comparing vac to con
#
#
# Mitigated Fraction
#
# observed median lower  upper
# Equal Tailed      0.44 0.4496 0.152 0.7088
# Highest Density   0.44 0.4496 0.152 0.7088
#
```

```
#
# Hodges-Lehmann
#
# observed    median    lower       upper
# Equal Tailed    -0.07335 -0.07615 -0.17220 -0.01565000
# Highest Density -0.07335 -0.07615 -0.15635 -0.00850065
#
#
# Quartile Differences (quartiles of vac - quartiles of con)
#
# observed     median    lower     upper
# Q25 -0.041500 -0.041500 -0.10340 -0.000905
# Q50 -0.112525 -0.111175 -0.28115  0.019350
# Q75 -0.168000 -0.170425 -0.38890  0.005300
#
#
# Quartiles of con
# observed    median    lower    upper
# Q25 0.054000 0.054000 0.021005 0.11275
# Q50 0.139275 0.139275 0.061400 0.31000
# Q75 0.315000 0.315000 0.173000 0.44625
#
#
# Quartiles of vac
# observed  median   lower     upper
# Q25  0.01250 0.01250 0.00125 0.026000
# Q50  0.02675 0.02675 0.01665 0.144575
# Q75  0.14700 0.14700 0.02810 0.219250
```

---

mf-class   *Class mf*

Description

Parent class for package MF data objects.

Fields

- nboot: numeric value specifying number of samples
- alpha: numeric value specifying complement of confidence interval
- seed: vector of integers specifying seed for pseudo-random number generator used
- compare: vector of character strings naming groups compared
- rng: character string naming type of random number generator

Author(s)

[MF-package](#)

See Also

Other mf: [mfboot-class](#), [mfbootcluster-class](#), [mfhlboot-class](#)

MFBoot   *Bootstrap MF CI*

Description

Estimates bootstrap confidence intervals for the mitigated fraction.

Usage

```
MFBoot(
  formula,
  data,
  compare = c("con", "vac"),
  b = 100,
  B = 100,
  alpha = 0.05,
  hpd = TRUE,
  bca = FALSE,
  return.boot = FALSE,
  trace.it = FALSE,
  seed = sample(1:1e+05, 1)
)
```

Arguments

formula	Formula of the form y ~ x, where y is a continuous response and x is a factor with two levels.
data	Data frame
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared
b	Number of bootstrap samples to take with each cycle
B	Number of cycles, giving the total number of samples = B * b
alpha	Complement of the confidence level
hpd	Estimate highest density intervals?
bca	Estimate BCa intervals?
return.boot	Save the bootstrap sample of the MF statistic?
trace.it	Verbose tracking of the cycles?
seed	to initialize random number generator for reproducibility. Passed to set.seed.

Details

Resamples the data and produces bootstrap confidence intervals. Equal tailed intervals are estimated by the percentile method. Highest density intervals are estimated by selecting the shortest of all possible intervals. For BCa intervals, see Efron and Tibshirani section 14.3.

Value

a mfboot data object

Author(s)

MF-package

References

Siev D. (2005). An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* **4:500–508**

Efron B, Tibshirani RJ. *An Introduction to the Bootstrap.* Chapman and Hall, New York, 1993.

See Also

mfboot

Examples

```
MFBoot(lesion ~ group, calflung, seed = 12345)

# 10000 bootstrap samples
# 95% confidence interval
# Seed = 12345
#
# Comparing vac to con
# observed median lower  upper
# Equal Tailed        0.44 0.4496 0.152 0.7088
# Highest Density     0.44 0.4496 0.152 0.7088
```

---

mfboot-class   *Class mfboot*

Description

class for data objects produced by MFBoot, contains class mf with the two additional fields *stat* and *stuff*.

Fields

- nboot: numeric value specifying number of samples
- alpha: numeric value specifying complement of confidence interval
- seed: vector of integers specifying seed for pseudo-random number generator used
- compare: vector of character strings naming groups compared
- rng: character string naming type of random number generator
- sample: what is this?
- stat: matrix of estimates

Contains

mf

Author(s)

MF-package

See Also

[MFBoot]

Other mf: `mf-class`, `mfbootcluster-class`, `mfhlboot-class`

---

mfbootcluster-class *Class mfbootcluster*

Description

Class mfbootcluster is created from output of function MFClusBoot

Fields

- `nboot`: numeric value specifying number of samples
- `alpha`: numeric value specifying complement of confidence interval
- `seed`: vector of integers specifying seed for pseudo-random number generator used
- `compare`: vector of character strings naming groups compared
- `rng`: character string naming type of random number generator
- `stat`: matrix matrix with columns *observed*, *median*, *lower*, *upper* for estimates
- `what`: character vector naming what was resampled: *clusters*, *units*, *both*
- `excludedClusters`: character vector naming clusters excluded because of missing treatment(s)
- `call`: the call to `MFClusBoot`
- `sample`: what is this?
- `All`: Field "All" from MFClus call.

Contains

[mf]

Author(s)

[MF-package]

See Also

[MFClusBoot]

Other mf: `mf-class`, `mfboot-class`, `mfhlboot-class`

MFClus   *Clustered mitigated fraction*

Description

Estimates mitigated fraction from clustered or stratified data.

Usage

```
MFClus(formula, data, compare = c("con", "vac"), trace.it = FALSE)
```

Arguments

formula	Formula of the form y ~ x + cluster(w), where y is a continuous response, x is a factor with two levels of treatment, and w is a factor indicating the clusters.
data	Data frame. See Note for handling of input data with more than two levels.
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared
trace.it	Verbose tracking of the cycles? Default FALSE.

Details

Averages the U statistic over the clusters and computes MF from it. Clusters are excluded if they do not include both treatments.

Value

a mfcluster data object

Note

If input data contains more than two levels of treatment, rows associated with unused treatment levels will be removed.

Factor levels for treatments not present in the input data will be ignored.

Clusters with missing treatments will be excluded. See mfbootcluster or use trace.it to identify excluded clusters.

Author(s)

MF-package

References

Siev D. (2005). An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* **4:500–508**

See Also

mfcluster

Examples

```
## Not run:
MFClus(lesion ~ group + cluster(litter), piglung)

#  Comparing vac to con
#
#  MF = 0.3533835
#
#  By Cluster
#      w  u         r n1 n2         mf
#  U 25 10 0.4000000  5  5 -0.2000000
#  K 12  2 0.2500000  4  2 -0.5000000
#  Z 16 10 0.8333333  3  4  0.6666667
#  D  3  2 1.0000000  1  2  1.0000000
#  N  1  0 0.0000000  1  3 -1.0000000
#  T  8  5 0.8333333  2  3  0.6666667
#  P  4  1 0.5000000  2  1  0.0000000
#  L  3  2 0.6666667  1  3  0.3333333
#  G 15  9 0.7500000  3  4  0.5000000
#  J 15  9 1.0000000  3  3  1.0000000
#  W  6  3 0.7500000  2  2  0.5000000
#  A  9  3 0.3333333  3  3 -0.3333333
#  X 12  6 1.0000000  3  2  1.0000000
#  F 13  7 0.7777778  3  3  0.5555556
#  S 21 11 0.9166667  4  3  0.8333333
#  H 14  8 0.8888889  3  3  0.7777778
#  Y  2  1 1.0000000  1  1  1.0000000
#  E  2  1 1.0000000  1  1  1.0000000
#
#  All
#        w  u         r n1 n2         mf
#  All 181 90 0.6766917 50 52 0.3533835
#
#  Excluded Clusters
#  [1] M, Q, R, B, O, V, I, C

## End(Not run)
```

MFClusBoot   *Bootstrap MF CI from clustered data*

Description

Estimates bootstrap confidence intervals for the mitigated fraction from clustered or stratified data.

Usage

```
MFClusBoot(
  formula,
  data,
  compare = c("con", "vac"),
  boot.cluster = TRUE,
  boot.unit = TRUE,
```

```
    b = 100,
    B = 100,
    alpha = 0.05,
    hpd = TRUE,
    return.boot = FALSE,
    trace.it = FALSE,
    seed = sample(1:1e+05, 1)
)
```

Arguments

formula	Formula of the form y ~ x + cluster(w), where y is a continuous response, x is a factor with two levels of treatment, and w is a factor indicating the clusters.
data	Data frame. See Note for handling of input data with more than two levels.
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared
boot.cluster	Boolean whether to resample the clusters.
boot.unit	Boolean whether to resample the units within cluster.
b	Number of bootstrap samples to take with each cycle
B	Number of cycles, giving the total number of samples = B * b
alpha	Complement of the confidence level
hpd	Boolean whether to estimate highest density intervals.
return.boot	Boolean whether to save the bootstrap sample of the MF statistic.
trace.it	Boolean whether to display verbose tracking of the cycles.
seed	to initialize random number generator for reproducibility. Passed to set.seed.

Details

Resamples the data and produces bootstrap confidence intervals. Equal tailed intervals are estimated by the percentile method. Highest density intervals are estimated by selecting the shortest of all possible intervals.

Value

a [mfbootcluster](#) data object

Note

If input data contains more than two levels of treatment, rows associated with unused treatment levels will be removed.

Factor levels for treatments not present in the input data will be ignored.

Clusters with missing treatments will be excluded. See [mfbootcluster](#) or use trace.it to identify excluded clusters.

Author(s)

[MF-package](#)

References

Siev D. (2005). An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* **4:500–508**

Efron B, Tibshirani RJ. *An Introduction to the Bootstrap.* Chapman and Hall, New York, 1993.

Examples

```
## Not run:
MFClusBoot(lesion ~ group + cluster(litter), piglung, seed = 12345)
# Bootstrapping clusters. . . . . . . . . . . . . . . . .
#
# Bootstrapping units. . . . . . . . . . . . . . . . .
#
# 10000 bootstrap samples of clusters and units in treatment in cluster
# Comparing vac to con
#
# 95% confidence interval
#
# observed    median        lower      upper
# Equal Tailed    0.3533835 0.3648649 -0.01409471 0.7109966
# Highest Density 0.3533835 0.3648649  0.00000000 0.7236842
#
# Excluded Clusters
# M, Q, R, B, O, V, I, C

## End(Not run)
```

MFClusBootHier   *MFClusBootHier*

Description

Combines MFhBoot and MFnestBoot into a single function.

Usage

```
MFClusBootHier(
  formula,
  data,
  compare = c("con", "vac"),
  nboot = 10000,
  boot.unit = TRUE,
  boot.cluster = TRUE,
  which.factor = "All",
  alpha = 0.05,
  seed = sample(1:1e+05, 1)
)
```

Arguments

formula	Formula of the form y ~ x + a/b/c, where y is a continuous response, x is a factor with two levels of treatment, and a/b/c are grouping variables corresponding to the clusters. Nesting is assumed to be in order, left to right, highest to lowest. So a single level of "a" will contain multiple levels of "b" and a single level of "b" will contain multiple levels of "c".
data	a data.frame or tibble with the variables specified in formula. Additional variables will be ignored.
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared.
nboot	number of bootstrapping events
boot.unit	Boolean whether to sample observations from within those of the same core.
boot.cluster	Boolean whether to sample which cores are present. If TRUE, some trees have all the cores while others only have a subset.
which.factor	Which variables to include in the mitigated fraction summation. Default is "All", to sum over entire tree.
alpha	Passed to emp_hpd to calculate high tailed upper and high tailed lower of mitigated fraction.
seed	Passed to [MFhBoot](#) to to initialize random number generator for reproducibility.

Value

A list with the following elements:

- MFhBoot: as output from [MFhBoot](#).
- MFnestBoot: as output from [MFnestBoot](#).

Note

Core variable is the variable corresponding to the lowest nodes of the hierarchical tree. Nest variables are those above the core. All refers to a summary of the entire tree.

Author(s)

[MF-package](#)

See Also

[MFhBoot](#), [MFnestBoot](#).

Examples

```
a <- data.frame(
  room = paste("Room", rep(c("W", "Z"), each = 24)),
  pen = paste("Pen", rep(LETTERS[1:6], each = 8)),
  litter = paste("Litter", rep(11:22, each = 4)),
  tx = rep(rep(c("vac", "con"), each = 2), 12))
set.seed(76153)
a$lung[a$tx == "vac"] <- rnorm(24, 5, 1.3)
a$lung[a$tx == "con"] <- rnorm(24, 7, 1.3)
thismf1 <- MFClusBootHier(lung ~ tx + room / pen / litter, a, nboot = 10000,
                   boot.cluster = TRUE, boot.unit = TRUE, seed = 12345)
thismf1
```

---

mfclusboothier-class   *Class mfclusboothier*

Description

Class mfclusboothier is created from output of function MFClusBootHier.

Fields

- `MFhBoot` output from MFhBoot. A list.
- `MFnestBoot` output from MFnestBoot. A list.

Author(s)

MF-package

See Also

MFhBoot, MFnestBoot

MFClusHier   *MFClusHier*

Description

Calculate mitigated fraction directly from hierarchial nested data. Combines [MFh](MFh) and MFnest into a single function.

Usage

```
MFClusHier(formula, data, compare = c("con", "vac"), which.factor = "All")
```

Arguments

formula	Formula of the form y ~ x + a/b/c, where y is a continuous response, x is a factor with two levels of treatment, and a/b/c are grouping variables corresponding to the clusters. Nesting is assumed to be in order, left to right, highest to lowest. So a single level of "a" will contain multiple levels of "b" and a single level of "b" will contain multiple levels of "c".
data	a data.frame or tibble with the variables specified in formula. Additional variables will be ignored.
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared.
which.factor	one or more variable(s) of interest. This can be any of the core or nest variables from the data set. If none or NULL is specified, MF will be calculated for the whole tree.

Value

A list with the following elements:

- `MFh`: as output from MFh.
- `MFnest`: as output from MFnest.

Note

`Core` variable is the variable corresponding to the lowest nodes of the hierarchical tree. `Nest` variables are those above the core. `All` refers to a summary of the entire tree.

Author(s)

MF-package

See Also

MFh, MFnest

Examples

```
a <- data.frame(
room = paste("Room", rep(c("W", "Z"), each = 24)),
pen = paste("Pen", rep(LETTERS[1:6], each = 8)),
litter = paste("Litter", rep(11:22, each = 4)),
tx = rep(rep(c("vac", "con"), each = 2), 12))
set.seed(76153)
a$lung[a$tx == "vac"] <- rnorm(24, 5, 1.3)
a$lung[a$tx == "con"] <- rnorm(24, 7, 1.3)
thismf <- MFClusHier(lung ~ tx + room / pen / litter, a)
thismf$MFnest
aCore <- thismf$MFh
aCore
aCore$data
aCore$formula
aCore$compare
```

---

mfclushier-class   *Class mfclushier*

Description

Class mfclushier is created from output of function MFClusHier

Fields

- `MFh`: output from MFh. A mfhierdata object.
- `MFnest`: output from MFnest. A tibble.

Author(s)

MF-package

See Also

[MFh,](#) [MFnest](#)

---

mfcluster-class   *Class mfcluster*

Description

Class mfcluster is created from output of function MFClus

Fields

- `All`: vector with elements
    - `w`: Wilcoxon statistic
    - `u`: Mann-Whitney statistic
    - `r`: mean ridit
    - `n1`: size of group 1
    - `n2`: size of group 2
    - `mf`: mitigated fraction
- `byCluster`: As for All, by clusters
- `excludedClusters`: character vector naming clusters excluded because of missing treatment
- `call`: the call to `MFClus`
- `compare`: character vector naming groups compared

Author(s)

[MF-package](#)

See Also

[MFClus](#)

---

mfcomponents-class   *Class mfcomponents*

Description

Class mfcomponents is created from output of function MFSubj

Fields

- `mf`: numeric estimator for mitigated fraction
- `x`: numeric vector containing responses of group 1
- `y`: numeric vector containing responses of group 2
- `subj`: matrix where mf_jre the subject components
- `compare`: character vector naming groups being compared

Author(s)

MF-package

See Also

MFSubj

MFh *Identify ranks for use when evaluating MF for nested hierarchy.*

Description

Identify ranks for use when evaluating MF for nested hierarchy.

Usage

```
MFh(formula, data, compare = c("con", "vac"))
```

Arguments

formula	Formula of the form y ~ x + a/b/c, where y is a continuous response, x is a factor with two levels of treatment, and a/b/c are vgrouping variables corresponding to the clusters. Nesting is assumed to be in order, left to right, highest to lowest. So a single level of "a" will contain multiple levels of "b" and a single level of "b" will contain multiple levels of "c".
data	a data.frame or tibble with the variables specified in formula. Additional variables will be ignored.
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared.

Value

A mfhierdata object, which is a list of three items.

- coreTbl A dplyr::tibble with one row for each unique core level showing values for:
  - con_n & vac_n: counts of observations for each treatment level in the core level.
  - con_medResp & vac_medResp: median of the y continuous response for each treatment level.
  - n1n2: product of the counts, con_n × vac_n.
  - w: Wilcoxon statistic
  - u: Mann-Whitney statistic
- data: A dplyr::tibble of the restructured input data used for calculations.
- compare: The compare variables as input by user.
- formula: The formula as input by user.

Note

Core variable is the variable corresponding to the lowest nodes of the hierarchial tree. Nest variables are those above the core.

Author(s)

MF-package

See Also

MFnest for calculation of MF for nest, core and all variables. mfhierdata for returned object. MF-ClusHier for a wrapper.

Examples

```
a <- data.frame(
 room  = paste("Room", rep(c("W", "Z"), each = 24)),
 pen   = paste("Pen", rep(LETTERS[1:6], each = 8)),
 litter = paste("Litter", rep(11:22, each = 4)),
 tx    = rep(rep(c("vac", "con"), each = 2), 12))
set.seed(76153)
a$lung[a$tx == "vac"] <- rnorm(24, 5, 1.3)
a$lung[a$tx == "con"] <- rnorm(24, 7, 1.3)

aCore <- MFh(lung ~ tx + room / pen / litter, a)
aCore
#  A tibble: 12 x 10
#      room    pen    litter    con_medResp con_n      w vac_medResp vac_n  n1n2
#      <chr>  <chr> <chr>           <dbl> <dbl> <dbl>       <dbl> <dbl> <dbl>
#   1 Room W Pen A Litter 11        8.24     2     7        5.13     2     4
#   2 Room W Pen A Litter 12        4.91     2     5        3.81     2     4
#   3 Room W Pen B Litter 13        8.10     2     7        5.23     2     4
#   4 Room W Pen B Litter 14        8.11     2     7        5.59     2     4
#   5 Room W Pen C Litter 15        8.09     2     7        5.26     2     4
#   6 Room W Pen C Litter 16        6.77     2     7        4.50     2     4
#   7 Room Z Pen D Litter 17        5.58     2     7        4.26     2     4
#   8 Room Z Pen D Litter 18        7.44     2     6        6.33     2     4
#   9 Room Z Pen E Litter 19        7.98     2     7        4.58     2     4
#  10 Room Z Pen E Litter 20        6.78     2     7        4.86     2     4
#  11 Room Z Pen F Litter 21        6.82     2     7        5.36     2     4
#  12 Room Z Pen F Litter 22        7.27     2     7        5.13     2     4
```

MFhBoot   *MFhBoot*

Description

Calculate rank tables for MF using bootstrapping.

Usage

```
MFhBoot(
  formula,
  data,
  compare = c("con", "vac"),
  nboot = 10000,
  boot.unit = TRUE,
  boot.cluster = TRUE,
  seed = sample(1:1e+05, 1)
)
```

Arguments

formula	Formula of the form y ~ x + a/b/c, where y is a continuous response, x is a factor with two levels of treatment, and a/b/c are grouping variables corresponding to the clusters. Nesting is assumed to be in order, left to right, highest to lowest. So a single level of "a" will contain multiple levels of "b" and a single level of "b" will contain multiple levels of "c".
data	a data.frame or tibble with the variables specified in formula. Additional variables will be ignored.
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared.
nboot	number of bootstrapping events
boot.unit	Boolean whether to sample observations from within those of the same core.
boot.cluster	Boolean whether to sample which cores are present. If TRUE, some trees have all the cores while others only have a subset.
seed	to initialize random number generator for reproducibility. Passed to set.seed.

Value

A list with the following elements:

- bootmfh: Rank table for the bootstrapped values as output from [MFh](). Includes a new bootID variable to distinguish each bootstrapped incidence.
- clusters: Table of unique nodes with an ID.
- compare: Compare vector as specified by user.
- mfh: MFh run on original data input.

Author(s)

[MF-package]()

See Also

[MFClusBootHier](), [MFnestBoot]()

Examples

```
set.seed(76153)
a <- data.frame(room = paste("Room", rep(c("W", "Z"), each = 24)),
                pen = paste("Pen", rep(LETTERS[1:6], each = 8)),
                litter = paste("Litter", rep(11:22, each = 4)),
                tx = rep(rep(c("vac", "con"), each = 2), 12))
a[a$tx == "vac", "lung"] <-  rnorm(24, 5, 1.3)
a[a$tx == "con", "lung"] <- rnorm(24, 7, 1.3)
a

formula <- lung ~ tx + room / pen / litter
nboot <- 10000
boot.cluster <- TRUE
boot.unit <- TRUE
which.factors <- c("All", "room", "pen", "litter")

system.time(test1 <- MFhBoot(formula, a,
```

```
                                nboot = 10000,
                                boot.cluster = TRUE,
                                boot.unit = TRUE,
                                seed = 12345))
  test1$bootmfh
```

---

mfhierdata-class   *Class mfhierdata*

Description

Class mfhierdata is created from output of function MFh

Fields

- coreTbl: data.frame with one row for each unique core level showing values for nx, ny, N, w, u, and median observed response.
- data: data.frame is the restructured input data used for calculations in MFh and MFnest.
- compare: character vector naming groups being compared.
- formula: formula that was called by user.

Author(s)

MF-package

See Also

MFh

---

mfhlboot-class   *Class mfhlboot*

Description

class for data objects produced by HLBoot, contains class mf with additional fields *MFstat*, *HLstat*, *QDIFstat*, *QXstat*, *QYstat*

Fields

- nboot: Numeric value specifying number of samples.
- alpha: Numeric value specifying complement of confidence interval.
- seed: Vector of integers specifying seed for pseudo-random number generator used.
- compare: Vector of character strings naming groups compared.
- rng: Character string naming type of random number generator.
- sample: The bootstrapped values.
- MFstat: Matrix with columns *observed*, *median*, *lower*, *upper* for Equal Tailed and Highest Density estimates of mitigated fraction (MF).

- HLstat: Matrix with columns *observed*, *median*, *lower*, *upper* for Equal Tailed and Highest Density estimates of Hodge-Lehmann estimator (HL).

- QDIFstat: Matrix with columns *observed*, *median*, *lower*, *upper* for estimates of Quartile Differences.

- QXstat: Matrix with columns *observed*, *median*, *lower*, *upper* for quartiles of treatments, equal tailed.

- QYstat: Matrix with columns *observed*, *median*, *lower*, *upper* for quartiles of response, equal tailed.

Contains

[mf](mf)

Author(s)

MF-package

See Also

HLBoot

Other mf: mf-class, mfboot-class, mfbootcluster-class

MFmp	*Mitigated fraction from matched pairs*

Description

Estimates mitigated fraction from matched pairs.

Usage

```
MFmp(
  formula = NULL,
  data = NULL,
  compare = c("con", "vac"),
  x = NULL,
  alpha = 0.05,
  df = NA,
  tdist = TRUE
)
```

Arguments

formula	Formula of the form y ~ x + cluster(w), where y is a continuous response, x is a factor with two levels of treatment, and w is a factor indicating the clusters.
data	Data frame
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared
x	Trinomial vector $\{\Sigma I(x < y), \Sigma I(x = y), \Sigma I(x > y)\}$

alpha	Complement of the confidence level.
df	Degrees of freedom. Default N-2
tdist	Use quantiles of t or Gaussian distribution for confidence interval? Default t distribution.

Details

Estimates *MF* from matched pairs by the difference of multinomial fractions $(\Sigma I(x < y) - \Sigma I(x > y))/N$. The trinomial vector is $\{\Sigma I(x < y), \Sigma I(x = y), \Sigma I(x > y)\}$

Value

a [mfmp](#) data object

Note

upper confidence interval is truncated to 1; lower confidence interval is truncated to -1. Point estimate of 1.0 indicates complete separation.

Author(s)

[MF-package](#)

References

Siev D. (2005). An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* **4:500–508**

See Also

[mfmp](#)

Examples

```
MFmp(les ~ tx + cluster(cage), mlesions, compare = c("con", "vac"))
MFmp(x = c(12, 12, 2))
```

---

mfmp-class   *Class mfmp*

Description

Class mfmp is created from output of function MFmp

Fields

- ci: numeric vector of point and interval estimates
- x: numeric vector of length three holding data
- what: text string describing interval type
- alpha: numeric value specifying complement of confidence interval
- tdist: Logical indicating if t distribution(TRUE) or Gaussian (FALSE)
- df: numeric value indicating degrees freedom

Author(s)

MF-package

See Also

MFmp

MFnest   *Summations to calculate the MF for nested data from a rank table.*

Description

Summations to calculate the MF for nested data from a rank table.

Usage

```
MFnest(Y, which.factor = "All")
```

Arguments

Y   rank table (tibble or data.frame), structured as $coreTbl output from MFh or returned object from MFh.

which.factor   one or more grouping variable(s) of interest. This can be any of the core or nest variables from the data set. If none or All is specified, a summary MF will be calculated for the whole tree.

Value

A tibble with each unique level of a variable as a row. Other values include:

- MF: Mitigated fraction for the particular level of the variable in this row.
- N1N2: Sum of the n1n2 variable in $coreTbl field of mfhierdata object output by MFh for this particular variable-level combination.
- U: Sum of u variable in $coreTbl field of mfhierdata object output by MFh for this particular variable-level combination.
- _N: Sum of the _n variable in $coreTbl field of mfhierdata object output by MFh for this particular variable-level combination.
- _medResp: Median of observed responses for each comparison group for this particular variable-level combination.

Note

Core variable is the variable corresponding to the lowest nodes of the hierarchial tree. Nest variables are those above the core. All refers to a summary of the entire tree.

Author(s)

MF-package

See Also

  [MFh](MFh)

Examples

```
a <- data.frame(
 room = paste("Room", rep(c("W", "Z"), each = 24)),
 pen = paste("Pen", rep(LETTERS[1:6], each = 8)),
 litter = paste("Litter", rep(11:22, each = 4)),
 tx = rep(rep(c("vac", "con"), each = 2), 12))
set.seed(76153)
a$lung[a$tx == "vac"] <- rnorm(24, 5, 1.3)
a$lung[a$tx == "con"] <- rnorm(24, 7, 1.3)

aCore <- MFh(lung ~ tx + room / pen / litter, a)
MFnest(aCore)
# # A tibble: 1 x 9
#   variable level    MF   N1N2     U con_N vac_N con_medResp vac_medResp
#   <fct>    <chr> <dbl>  <dbl> <dbl> <dbl> <dbl>       <dbl>       <dbl>
# 1 All      All   0.875     48    45    24    24        7.24        4.91

MFnest(aCore$coreTbl)
# Skipping median summary, no response data provided.
# # A tibble: 1 x 7
#   variable level    MF   N1N2     U con_N vac_N
#   <fct>    <chr> <dbl>  <dbl> <dbl> <dbl> <dbl>
# 1 All      All   0.875     48    45    24    24

MFnest(aCore, "room")
# # A tibble: 2 x 9
#   variable level     MF   N1N2     U con_N vac_N con_medResp vac_medResp
#   <fct>    <chr>  <dbl>  <dbl> <dbl> <dbl> <dbl>       <dbl>       <dbl>
# 1 room     Room W 0.833     24    22    12    12        7.79        4.85
# 2 room     Room Z 0.917     24    23    12    12        6.71        4.98

MFnest(aCore, "pen")
# Complete separation observed for variable(s): pen
# # A tibble: 6 x 9
#   variable level    MF   N1N2     U con_N vac_N con_medResp vac_medResp
#   <fct>    <chr> <dbl>  <dbl> <dbl> <dbl> <dbl>       <dbl>       <dbl>
# 1 pen      Pen A  0.5      8     6     4     4        6.79        4.24
# 2 pen      Pen B  1        8     8     4     4        8.11        5.59
# 3 pen      Pen C  1        8     8     4     4        7.69        4.85
# 4 pen      Pen D  0.75     8     7     4     4        6.10        4.98
# 5 pen      Pen E  1        8     8     4     4        6.86        4.86
# 6 pen      Pen F  1        8     8     4     4        6.88        5.13

MFnest(aCore, c("All", "litter"))
# Complete separation observed for variable(s): litter
# # A tibble: 13 x 9
#    variable level        MF   N1N2     U con_N vac_N con_medResp vac_medResp
#    <fct>    <chr>     <dbl>  <dbl> <dbl> <dbl> <dbl>       <dbl>       <dbl>
# 1  All      All       0.875     48    45    24    24        7.24        4.91
# 2  litter   Litter 11 1          4     4     2     2        8.24        5.13
# 3  litter   Litter 12 0          4     2     2     2        4.91        3.81
# 4  litter   Litter 13 1          4     4     2     2        8.10        5.23
```

```
#  5 litter   Litter 14 1       4    4    2     2       8.11        5.59
#  6 litter   Litter 15 1       4    4    2     2       8.09        5.26
#  7 litter   Litter 16 1       4    4    2     2       6.77        4.50
#  8 litter   Litter 17 1       4    4    2     2       5.58        4.26
#  9 litter   Litter 18 0.5     4    3    2     2       7.44        6.33
# 10 litter   Litter 19 1       4    4    2     2       7.98        4.58
# 11 litter   Litter 20 1       4    4    2     2       6.78        4.86
# 12 litter   Litter 21 1       4    4    2     2       6.82        5.36
# 13 litter   Litter 22 1       4    4    2     2       7.27        5.13

MFnest(aCore, "litter")
# Complete separation observed for variable(s): litter
# # A tibble: 12 x 9
#    variable level       MF  N1N2      U con_N vac_N con_medResp vac_medResp
#    <fct>    <chr>    <dbl> <dbl>  <dbl> <dbl> <dbl>       <dbl>       <dbl>
#  1 litter   Litter 11  1       4    4    2     2       8.24        5.13
#  2 litter   Litter 12  0       4    2    2     2       4.91        3.81
#  3 litter   Litter 13  1       4    4    2     2       8.10        5.23
#  4 litter   Litter 14  1       4    4    2     2       8.11        5.59
#  5 litter   Litter 15  1       4    4    2     2       8.09        5.26
#  6 litter   Litter 16  1       4    4    2     2       6.77        4.50
#  7 litter   Litter 17  1       4    4    2     2       5.58        4.26
#  8 litter   Litter 18  0.5     4    3    2     2       7.44        6.33
#  9 litter   Litter 19  1       4    4    2     2       7.98        4.58
# 10 litter   Litter 20  1       4    4    2     2       6.78        4.86
# 11 litter   Litter 21  1       4    4    2     2       6.82        5.36
# 12 litter   Litter 22  1       4    4    2     2       7.27        5.13

MFnest(aCore, c("room", "pen", "litter"))
# # A tibble: 20 x 9
#    variable level       MF  N1N2      U con_N vac_N con_medResp vac_medResp
#    <fct>    <chr>    <dbl> <dbl>  <dbl> <dbl> <dbl>       <dbl>       <dbl>
#  1 room     Room W   0.833   24   22   12    12       7.79        4.85
#  2 room     Room Z   0.917   24   23   12    12       6.71        4.98
#  3 pen      Pen A    0.5      8    6    4     4       6.79        4.24
#  4 pen      Pen B    1        8    8    4     4       8.11        5.59
#  5 pen      Pen C    1        8    8    4     4       7.69        4.85
#  6 pen      Pen D    0.75     8    7    4     4       6.10        4.98
#  7 pen      Pen E    1        8    8    4     4       6.86        4.86
#  8 pen      Pen F    1        8    8    4     4       6.88        5.13
#  9 litter   Litter 11 1       4    4    2     2       8.24        5.13
# 10 litter   Litter 12 0       4    2    2     2       4.91        3.81
# 11 litter   Litter 13 1       4    4    2     2       8.10        5.23
# 12 litter   Litter 14 1       4    4    2     2       8.11        5.59
# 13 litter   Litter 15 1       4    4    2     2       8.09        5.26
# 14 litter   Litter 16 1       4    4    2     2       6.77        4.50
# 15 litter   Litter 17 1       4    4    2     2       5.58        4.26
# 16 litter   Litter 18 0.5     4    3    2     2       7.44        6.33
# 17 litter   Litter 19 1       4    4    2     2       7.98        4.58
# 18 litter   Litter 20 1       4    4    2     2       6.78        4.86
# 19 litter   Litter 21 1       4    4    2     2       6.82        5.36
# 20 litter   Litter 22 1       4    4    2     2       7.27        5.13
```

MFnestBoot   *MFnestBoot*

Description

MFnest using bootstrapping

Usage

```
MFnestBoot(x, which.factor = "All", alpha = 0.05)
```

Arguments

x	output from MFhBoot
which.factor	one or more grouping variable(s) of interest. This can be any of the core or nest variables from the data set. A MF value will be calculated for each level of the variable(s) specified. Default is "All", to sum over entire tree.
alpha	Passed to emp_hpd to calculate eq tailed upper and high lower of mitigated fraction

Value

A list with the following elements:

- mfnest_details: The MF and summary statistics as calculated for each bootstrap event. Variables as in MFnest output.
- mfnest_summary: Statistical summary of bootstrapped MF with each unique level of a core or nest variable passed to which.factor as a row. Other variables include:
  - median: Median of MFs from all of the bootstrap events.
  - etlower: Lower value of equal tailed range.
  - etupper: Upper value of equal tailed range.
  - hdlower: Lower value of the highest posterior density range.
  - hdupper: Upper value of the highest posterior density range.
  - mf.obs: MF calculated from data using MFh.

Author(s)

MF-package

See Also

MFClusBootHier, MFhBoot

Examples

```
set.seed(76153)
a <- data.frame(room = paste("Room", rep(c("W", "Z"), each = 24)),
                pen = paste("Pen", rep(LETTERS[1:6], each = 8)),
                litter = paste("Litter", rep(11:22, each = 4)),
                tx = rep(rep(c("vac", "con"), each = 2), 12))
a[a$tx == "vac", "lung"] <- rnorm(24, 5, 1.3)
a[a$tx == "con", "lung"] <- rnorm(24, 7, 1.3)
a

formula <- lung ~ tx + room / pen / litter
nboot <- 10000
boot.cluster <- TRUE
```

```
boot.unit <- TRUE
which.factors <- c("All", "room", "pen", "litter")


#################

test1 <- MFhBoot(formula, a,
                 nboot = 10000,
                 boot.cluster = TRUE, boot.unit = TRUE, seed = 12345)
MFnestBoot(test1, c("All", "litter"))

## Not run:
system.time(test2 <- MFnestBoot(test1, which.factors))
test2
system.time(test3 <- MFnestBoot(test1, which.factors[1]))
test3
system.time(test4 <- MFnestBoot(test1, which.factors[2]))
test4
system.time(test5 <- MFnestBoot(test1, which.factors[2:3]))
test5
system.time(test6 <- MFnestBoot(test1, which.factors[2:4]))
test6

## End(Not run)
```

MFr   *Mitigated fraction*

Description

Mitigated fraction comparing treatment to control.

Usage

```
MFr(formula, data, compare = c("con", "vac"))
```

Arguments

formula	Formula of the form y ~ x, where y is a continuous response and x is a factor with two levels
data	Data frame
compare	Text vector stating the factor levels – compare[1] is the control or reference group to which compare[2] is compared

Details

The mitigated fraction is an estimator that quantifies an intervention's effect on reducing the severity of a condition. Since its units are on the probability scale, it is often a good idea to accompany it with an estimator on the original scale of measurement.

Value

The estimated mitigated fraction.

Author(s)

MF-package

References

Siev D, 2005. An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* 4:500-508

Examples

```
MFr(lesion ~ group, calflung)
# [1] 0.44
```

MFSubj   *Subject components of mitigated fraction*

Description

Estimates the subject components of the mitigated fraction.

Usage

```
MFSubj(formula, data, compare = c("con", "vac"))
```

Arguments

formula	Formula of the form y ~ x, where y is a continuous response and x is a factor with two levels
data	Data frame
compare	Text vector stating the factor levels - compare[1] is the control or reference group to which compare[2] is compared

Details

The mitigated fraction is an estimator that quantifies an intervention's effect on reducing the severity of a condition. Since its units are on the probability scale, it is often a good idea to accompany it with an estimator on the original scale of measurement.

The subject components are the individual contributions of the treated subjects to *MF*, which is the average of the subject components.

Value

a mfcomponents data object

Author(s)

MF-package

References

Siev D. (2005). An estimator of intervention effect on disease severity. *Journal of Modern Applied Statistical Methods.* **4:500–508**

Examples

```
x <- MFSubj(lesion ~ group, calflung)
x

#  MF = 0.44 comparing vac to con
#
#  MF Subject Components
#
#    mf_j freq     min_y    max_y
#    1.00    6 0.000030 0.00970
#    0.84    1 0.012500 0.01250
#    0.76    3 0.016650 0.02030
#    0.68    6 0.023250 0.03190
#    0.04    1 0.132100 0.13210
#   -0.04    3 0.144575 0.16325
#   -0.20    2 0.210000 0.21925
#   -0.36    1 0.292000 0.29200
#   -0.52    1 0.356500 0.35650
#   -0.84    1 0.461500 0.46150


mean(x$subj[, "mf_j"])

# [1] 0.44
```

---

mlesions	*mlesions dataset*

Description

Post-mortem examination of the lungs of dogs housed in cages by pairs.

Format

A data frame with 52 observations of the following 3 variables, no NAs.

- cage: Cage ID. 1 - 26.
- tx: Treatment. One of "con" or "vac".
- les: Percent gross lung lesions.

---

piglung	*piglung dataset*

Description

Post-mortem examination of the lungs of pigs in litters.

Format

A data frame with 102 observations of the following 3 variables, no NAs.

- lesion: Percent gross lung lesions.
- group: Treatment group. One of "con" or "vac".
- litter: Litter ID.

Index