Package 'PracTools'

November 5, 2026

Type Package

Title Designing and Weighting Survey Samples

Version 1.8

Date 2026-11-05

Author Richard Valliant [aut, cre],
Jill A. Dever [ctb],
Frauke Kreuter [ctb],
George Zipf [aut]

Maintainer Richard Valliant <valliant@umich.edu>

Description Functions and datasets to support Valliant, Dever, and Kreuter (2018), <doi:10.1007/978-3-319-93632-1>, ``Practical Tools for Designing and Weighting Survey Samples". Contains functions for sample size calculation for survey samples using stratified or clustered one-, two-, and three-stage sample designs, and single-stage audit sample designs. Functions are included that will group geographic units accounting for distances apart and measures of size. Other functions compute variance components for multistage designs, sample sizes in two-phase designs, and a stopping rule for ending data collection. A number of example data sets are included.

Suggests doBy, foreign, kableExtra, knitr, lpSolve, markdown, plyr, pps, Rcpp, reshape, rmarkdown, roxygen2, sampling, samplingbook, sp, survey

Depends R (>= 3.5.0)

Imports dplyr, geosphere, ggplot2, graphics, MASS, usmap

License GPL-3

LazyLoad yes

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2026-11-05 16:00:02 UTC

Contents

BW2stagePPS . . . 3
BW2stagePPSe . . . 5
BW2stageSRS . . . 7
BW3stagePPS . . . 9
BW3stagePPSe . . . 11
clusOpt2 . . . 14
clusOpt2fixedPSU . . . 16
clusOpt3 . . . 17
clusOpt3fixedPSU . . . 19
CompMOS . . . 21
CVcalc2 . . . 23
CVcalc3 . . . 24
deff . . . 26
deffCR . . . 29
deffH . . . 31
deffK . . . 33
deffS . . . 34
Domainy1y2 . . . 35
dub . . . 36
gamEst . . . 37
gammaFit . . . 38
GeoDistMOS . . . 39
GeoDistPSU . . . 41
GeoMinMOS . . . 43
HMT . . . 45
hospital . . . 46
labor . . . 47
MDarea.popA . . . 48
mibrfss . . . 49
nAuditAttr . . . 51
nAuditMUS . . . 53
nCont . . . 55
nContMoe . . . 56
nContOpt . . . 58
nDep2sam . . . 59
nDomain . . . 61
nEdge . . . 62
nhis . . . 65
nhis.large . . . 66
nhispart . . . 68
nLogOdds . . . 70
nPPS . . . 71
nProp . . . 72
nProp2sam . . . 73
nPropMoe . . . 75
NRadjClass . . . 76

NRFUopt 77
nWilson 79
pclass 80
quad_roots 82
SampStop 82
smho.N874 85
smho98 86
strAlloc 87
Test_Data_US 89
ThirdGrade 90
TPV 91
unitVar 92
wtd.moments 93
wtdvar 95

Index **96**

`BW2stagePPS` *Relvariance components for 2-stage sample*

Description

Compute components of relvariance for a sample design where primary sampling units (PSUs) are selected with probability proportional to size (*pps*) and elements are selected via simple random sampling (*srs*). The input is an entire sampling frame.

Usage

```
BW2stagePPS(X, pp, psuID, lonely.SSU = "mean")
```

Arguments

`X`	data vector; length is the number of elements in the population.
`pp`	vector of one-draw probabilities for the PSUs; length is number of PSUs in population.
`psuID`	vector of PSU identification numbers. This vector must be as long as `X`. Each element in a given PSU should have the same value in `psuID`. PSUs must be in the same order as in `X`.
`lonely.SSU`	indicator for how singleton SSUs should be handled when computing the within PSU unit relvariance. Allowable values are `"mean"` and `"zero"`.

Details

BW2stagePPS computes the between and within population relvariance components appropriate for a two-stage sample in which PSUs are selected with varying probabilities and with replacement. Elements within PSUs are selected by simple random sampling. The components are appropriate for approximating the relvariance of the probability-with-replacement (*pwr*)-estimator of a total when the same number of elements are selected within each sample PSU. The function requires that an entire frame of PSUs and elements be input.

If a PSU contains multiple SSUs, some of which have missing data, or contains only one SSU, a value is imputed. If `lonely.SSU = "mean"`, the mean of the non-missing PSU contributions is imputed. If `lonely.SSU = "zero"`, a 0 is imputed. The former would be appropriate if a PSU contains multiple SSUs but one or more of them has missing data in which case R will normally calculate an NA. The latter would be appropriate if the PSU contains only one SSU which would be selected with certainty in any sample.

If any PSUs have one-draw probabilities of 1 (pp=1), they will be excluded from all computations.

(Use BW2stagePPSe if only a sample of PSUs and elements is available.)

Value

List object with values:

B2	between PSU unit relvariance
W2	within PSU unit relvariance
unit relvar	unit relvariance for population
B2+W2	sum of between and within relvariance estimates
k	ratio of $B^2 + W^2$ to unit relvariance
delta	measure of homogeneity with PSUs estimated as $B^2/(B^2 + W^2)$

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran, W.G. (1977, pp.308-310). *Sampling Techniques*. New York: John Wiley & Sons.

Saerndal, C.E., Swensson, B., and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.2.3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

BW2stagePPSe, BW2stageSRS, BW3stagePPS, BW3stagePPSe

Examples

```
data(MDarea.popA)
MDsub <- MDarea.popA[1:100000,]
    # Use PSU and SSU variables to define psu's
pp.PSU <- table(MDsub$PSU) / nrow(MDsub)
pp.SSU <- table(MDsub$SSU) / nrow(MDsub)
    # components with psu's defined by the PSU variable
BW2stagePPS(MDsub$y1, pp=pp.PSU, psuID=MDsub$PSU, lonely.SSU="mean")
    # components with psu's defined by the SSU variable
BW2stagePPS(MDsub$y1, pp=pp.SSU, psuID=MDsub$SSU, lonely.SSU="mean")

    # Use census tracts and block groups to define psu's
trtBG <- 10*MDsub$TRACT + MDsub$BLKGROUP
pp.trt <- table(MDsub$TRACT) / nrow(MDsub)
pp.BG <- table(trtBG) / nrow(MDsub)
    # components with psu's defined by tracts
BW2stagePPS(MDsub$ins.cov, pp=pp.trt, psuID=MDsub$TRACT, lonely.SSU="mean")
    # components with psu's defined by block groups
BW2stagePPS(MDsub$ins.cov, pp=pp.BG, psuID=trtBG, lonely.SSU="mean")
```

`BW2stagePPSe` *Estimated relvariance components for 2-stage sample*

Description

Estimate components of relvariance for a sample design where primary sampling units (PSUs) are selected with *pps* and elements are selected via *srs*. The input is a sample selected in this way.

Usage

```
BW2stagePPSe(Ni, ni, X, psuID, w, m, pp, lonely.SSU = "mean")
```

Arguments

`Ni`	vector of number of elements in the population of each sample PSU; length is the number of PSUs in the sample.
`ni`	vector of number of sample elements in each sample PSU; length is the number of PSUs in the sample. PSUs must be in the same order in `ni` and in `X`.
`X`	data vector for sample elements; length is the number of elements in the sample. These must be in PSU order. PSUs must be in the same order in `ni` and in `X`.
`psuID`	vector of PSU identification numbers. This vector must be as long as `X`. Each element in a given PSU should have the same value in `psuID`.
`w`	vector of full sample weights. This vector must be as long as `X`. Vector must be in the same order as `X`.
`m`	number of sample PSUs
`pp`	vector of 1-draw probabilities for the PSUs. The length of this vector is the number of PSUs in the sample. Vector must be in the same order as `Ni` and `ni`.

`lonely.SSU`	indicator for how singleton SSUs should be handled when computing the within PSU unit relvariance. Allowable values are `"mean"` and `"zero"`.

Details

`BW2stagePPSe` computes the between and within population variance and relvariance components appropriate for a two-stage sample in which PSUs are selected with varying probabilities and with replacement. Elements within PSUs are selected by simple random sampling. The number of elements selected within each sample PSU can vary but must be at least two. The estimated components are appropriate for approximating the relvariance of the *pwr*-estimator of a total when the same number of elements are selected within each sample PSU. This function can also be used if PSUs are selected by *srswr* by appropriate definition of `pp`.

If a PSU contains multiple SSUs, some of which have missing data, or contains only one SSU, a value is imputed. If `lonely.SSU = "mean"`, the mean of the non-missing PSU contributions is imputed. If `lonely.SSU = "zero"`, a 0 is imputed. The former would be appropriate if a PSU contains multiple SSUs but one or more of them has missing data in which case R will normally calculate an NA. The latter would be appropriate if the PSU contains only one SSU which would be selected with certainty in any sample.

If any PSUs have one-draw probabilities of 1 (pp=1), the function will be halted. Any such PSUs should be removed before calling the function.

Value

List with values:

`Vpsu`	estimated between PSU unit variance
`Vssu`	estimated within PSU unit variance
`B`	estimated between PSU unit relvariance
`W`	estimated within PSU unit relvariance
`k`	estimated ratio of `B+W` to estimated unit relvariance of the analysis variable
`delta`	intraclass correlation estimated as `B/(B+W)`

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran, W.G. (1977, pp.308-310). *Sampling Techniques*. New York: John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.4.1). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`BW2stagePPS`, `BW2stageSRS`, `BW3stagePPS`, `BW3stagePPSe`

Examples

```
require(sampling)
require(plyr)      # has function that allows renaming variables
data(MDarea.popA)
Ni <- table(MDarea.popA$TRACT)
m <- 20
probi <- m*Ni / sum(Ni)
    # select sample of clusters
sam <- cluster(data=MDarea.popA, clustername="TRACT", size=m, method="systematic",
               pik=probi, description=TRUE)
    # extract data for the sample clusters
samclus <- getdata(MDarea.popA, sam)
samclus <- rename(samclus, c("Prob" = "pi1"))


    # treat sample clusters as strata and select srswor from each
s <- strata(data = as.data.frame(samclus), stratanames = "TRACT",
            size = rep(50,m), method="srswor")
# extracts the observed data
samdat <- getdata(samclus,s)
samdat <- rename(samdat, c("Prob" = "pi2"))

    # extract pop counts for PSUs in sample
pick <- names(Ni) %in% sort(unique(samdat$TRACT))
Ni.sam <- Ni[pick]
pp <- Ni.sam / sum(Ni)
wt <- 1/samdat$pi1/samdat$pi2

BW2stagePPSe(Ni = Ni.sam, ni = rep(50,20), X = samdat$y1,
            psuID = samdat$TRACT, w = wt,
            m = 20, pp = pp, lonely.SSU="mean")
```

`BW2stageSRS` *Relvariance components for 2-stage sample*

Description

Compute components of relvariance for a sample design where primary sampling units (PSUs) and elements are selected via *srs*. The input is an entire sampling frame.

Usage

```
BW2stageSRS(X, psuID, lonely.SSU)
```

Arguments

`X`	data vector; length is the number of elements in the population.

psuID	vector of PSU identification numbers. This vector must be as long as X. Each element in a given PSU should have the same value in psuID. PSUs must be in the same order as in X.
lonely.SSU	indicator for how singleton SSUs should be handled when computing the within PSU unit relvariance. Allowable values are "mean" and "zero".

Details

BW2stageSRS computes the between and within population relvariance components appropriate for a two-stage sample in which PSUs are selected via *srs* (either with or without replacement). Elements within PSUs are assumed to be selected by *srswor*. The same number of elements is assumed to be selected within each sample PSU. The function requires that an entire frame of PSUs and elements be input.

If a PSU contains multiple SSUs, some of which have missing data, or contains only one SSU, a value is imputed. If lonely.SSU = "mean", the mean of the non-missing PSU contributions is imputed. If lonely.SSU = "zero", a 0 is imputed. The former would be appropriate if a PSU contains multiple SSUs but one or more of them has missing data in which case R will normally calculate an NA. The latter would be appropriate if the PSU contains only one SSU which would be selected with certainty in any sample.

(Use BW2stagePPSe if only a sample of PSUs and elements are available.)

Value

List with values:

B2	between PSU unit relvariance
W2	within PSU unit relvariance
unit relvar	unit relvariance for population
B2+W2	$B^2 + W^2$
k	ratio of $B^2 + W^2$ to unit relvariance
delta full	measure of homogeneity estimated as $B^2/(B^2 + W^2)$

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran, W.G. (1977, chap. 11). *Sampling Techniques*. New York: John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.2.1). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

BW2stagePPS, BW2stagePPSe, BW3stagePPS, BW3stagePPSe

Examples

```
data(MDarea.popA)
MDsub <- MDarea.popA[1:100000,]
    # psu's are defined by PSU variable
BW2stageSRS(abs(MDsub$Hispanic-2), psuID=MDsub$PSU, lonely.SSU="mean")
    # psu's are defined by SSU variable
BW2stageSRS(abs(MDsub$Hispanic-2), psuID=MDsub$SSU, lonely.SSU="mean")
```

`BW3stagePPS` *Relvariance components for 3-stage sample*

Description

Compute components of relvariance for a sample design where primary sampling units (PSUs) are selected with *ppswr* and secondary sampling units (SSUs) and elements within SSUs are selected via *srs*. The input is an entire sampling frame.

Usage

```
BW3stagePPS(X, pp, psuID, ssuID, lonely.SSU = "mean", lonely.TSU = "mean")
```

Arguments

`X`	data vector; length is the number of elements in the population.
`pp`	vector of one-draw probabilities for the PSUs; length is number of PSUs in population.
`psuID`	vector of PSU identification numbers. This vector must be as long as `X`. Each element in a given PSU should have the same value in `psuID`. PSUs must be in the same order as in `X`.
`ssuID`	vector of SSU identification numbers. This vector must be as long as `X`. Each element in a given SSU should have the same value in `ssuID`. PSUs and SSUs must be in the same order as in `X`. `ssuID` should have the form `psuID`‖(`ssuID` within PSU).
`lonely.SSU`	indicator for how singleton SSUs should be handled when computing the within PSU unit relvariance. Allowable values are `"mean"` and `"zero"`.
`lonely.TSU`	indicator for how singleton third-stage units (TSUs) should be handled when computing the within SSU unit relvariance. Allowable values are `"mean"` and `"zero"`.

Details

`BW3stagePPS` computes the between and within population relvariance components appropriate for a three-stage sample in which PSUs are selected with varying probabilities and with replacement. SSUs and elements within SSUs are selected by simple random sampling. The components are appropriate for approximating the relvariance of the *pwr*-estimator of a total when the same number

of SSUs are selected within each PSU, and the same number of elements are selected within each sample SSU. The function requires that an entire sampling frame of PSUs and elements be input.

If a PSU contains multiple SSUs, some of which have missing data, or contains only one SSU, a value is imputed. If `lonely.SSU = "mean"`, the mean of the non-missing PSU contributions is imputed. If `lonely.SSU = "zero"`, a 0 is imputed. The former would be appropriate if a PSU contains multiple SSUs but one or more of them has missing data in which case R will normally calculate an NA. The latter would be appropriate if the PSU contains only one SSU which would be selected with certainty in any sample. `lonely.TSU` has a similar purpose for third-stage units.

(Use `BW2stagePPSe` if only a sample of PSUs, SSUs, and elements is available.)

Value

List with values:

`B`	between PSU unit relvariance
`W`	within PSU unit relvariance computed as if the sample were two-stage
`W2`	unit relvariance among SSU totals
`W3`	unit relvariance among elements within PSU/SSUs
`unit relvar`	unit relvariance for population
`k1`	ratio of $B^2 + W^2$ to unit relvariance
`k2`	ratio of $W_2^2 + W_3^2$ to unit relvariance
`delta1`	homogeneity measure among elements within PSUs estimated as $B^2/(B^2 + W^2)$
`delta2`	homogeneity measure among elements within SSUs estimated as $W_2^2/(W_2^2 + W_3^2)$

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Hansen,M.H., Hurwitz,W.N., and Madow,W.G. (1953, chap. 9, p.211). *Sample Survey Methods and Theory*, Vol.I. John Wiley & Sons.

Saerndal, C.E., Swensson, B., and Wretman, J. (1992, p.149). *Model Assisted Survey Sampling*. Springer.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.2.4). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`BW2stagePPS`, `BW2stagePPSe`, `BW2stageSRS`, `BW3stagePPSe`

Examples

```
data(MDarea.popA)
MDsub <- MDarea.popA[1:100000,]
M <- length(unique(MDsub$PSU))
    # srs/srs/srs design
pp.PSU <- rep(1/M,M)
BW3stagePPS(X=MDsub$y1, pp=pp.PSU, psuID=MDsub$PSU, ssuID=MDsub$SSU,
    lonely.SSU = "mean", lonely.TSU = "mean")
    # ppswr/srs/srs design
pp.PSU <- table(MDsub$PSU) / nrow(MDsub)
BW3stagePPS(X=MDsub$y1, pp=pp.PSU, psuID=MDsub$PSU, ssuID=MDsub$SSU,
    lonely.SSU = "mean", lonely.TSU = "mean")
```

BW3stagePPSe *Estimated relvariance components for 3-stage sample*

Description

Estimate components of relvariance for a sample design where primary sampling units (PSUs) are selected with probability proportional to size with replacement (*ppswr*) and secondary sampling units (SSUs) and elements within SSUs are selected via simple random sampling (*srs*). The input is a sample selected in this way.

Usage

```
BW3stagePPSe(dat, v, Ni, Qi, Qij, m, lonely.SSU = "mean", lonely.TSU = "mean")
```

Arguments

dat	data frame for sample elements with PSU and SSU identifiers, weights, and analysis variable(s). The data frame should be sorted in hierarchical order: by PSU and SSU within PSU. Required names for columns: psuID = PSU identifier; ssuID = SSU identifier. These must be unique, i.e., numbering should not restart within each PSU. Setting ssuID = psuID‖(ssuID within PSU) is a method of doing this. w1i = vector of weights for PSUs; w2ij = vector of weights for SSUs (PSU weight*SSU weight within PSU); w = full sample weight
v	Name or number of column in data frame dat with variable to be analyzed.
Ni	m-vector of number of SSUs in the population in the sample PSUs; m is number of sample PSUs.
Qi	m-vector of number of elements in the population in the sample PSUs
Qij	vector of numbers of elements in the population in the sample SSUs
m	number of sample PSUs
lonely.SSU	indicator for how singleton SSUs should be handled when computing the within PSU unit relvariance. Allowable values are "mean" and "zero".
lonely.TSU	indicator for how singleton third-stage units (TSUs) should be handled when computing the within SSU unit relvariance. Allowable values are "mean" and "zero".

Details

BW3stagePPSe computes the between and within population relvariance components appropriate for a three-stage sample in which PSUs are selected with varying probabilities and with replacement. SSUs and elements within SSUs are selected by simple random sampling. The estimated components are appropriate for approximating the relvariance of the *pwr*-estimator of a total when the same number of SSUs are selected within each PSU, and the same number of elements are selected within each sample SSU.

If a PSU contains multiple SSUs, some of which have missing data, or contains only one SSU, a value is imputed. If `lonely.SSU = "mean"`, the mean of the non-missing PSU contributions is imputed. If `lonely.SSU = "zero"`, a 0 is imputed. The former would be appropriate if a PSU contains multiple SSUs but one or more of them has missing data in which case R will normally calculate an NA. The latter would be appropriate if the PSU contains only one SSU which would be selected with certainty in any sample. `lonely.TSU` has a similar purpose for third-stage units.

Value

List with values:

`Vpsu`	estimated between PSU unit variance
`Vssu`	estimated second-stage unit variance among SSU totals
`Vtsu`	estimated third-stage unit variance
`B`	estimated between PSU unit relvariance
`W`	estimated within PSU unit relvariance computed as if the sample were two-stage
`k1`	estimated ratio of `B+W` to estimated unit relvariance of the analysis variable
`W2`	estimated unit relvariance among SSU totals
`W3`	estimated third-stage unit relvariance among elements within PSU/SSUs
`k2`	estimated ratio of `W2+W3` to estimated unit relvariance of the analysis variable
`delta1`	homogeneity measure among elements within PSUs estimated as $B^2/(B^2 + W^2)$
`delta2`	homogeneity measure among elements within SSUs estimated as $W_2^2/(W_2^2 + W_3^2)$

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Hansen, M.H., Hurwitz, W.N., and Madow, W.G. (1953, chap. 9, sect. 10). *Sample Survey Methods and Theory*, Vol.II. New York: John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.4.2). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

BW2stagePPS, BW2stagePPSe, BW2stageSRS, BW3stagePPS

Examples

```
    # select 3-stage sample from Maryland population
set.seed(-780087528)
data(MDarea.popA)
MDpop <- MDarea.popA
require(sampling)
require(plyr)      # has function that allows renaming variables
    # make counts of SSUs and elements per PSU
xx <- do.call("rbind",list(by(1:nrow(MDpop),MDpop$SSU,head,1)))
pop.tmp <- MDpop[xx,]
Ni <- table(pop.tmp$PSU)
Qi <- table(MDarea.popA$PSU)
Qij <- table(MDpop$SSU)
m <- 30         # no. of PSUs to select
probi <- m*Qi / sum(Qi)
    # select sample of clusters
sam <- cluster(data=MDpop, clustername="PSU", size=m, method="systematic",
               pik=probi, description=TRUE)
    # extract data for the sample clusters
samclus <- getdata(MDpop, sam)
samclus <- rename(samclus, c("Prob" = "p1i"))
samclus <- samclus[order(samclus$PSU),]
    # treat sample clusters as strata and select srswor of block groups from each
    # identify psu IDs for 1st instance of each ssuID
xx <- do.call("rbind",list(by(1:nrow(samclus),samclus$SSU,head,1)))
SSUs <- cbind(PSU=samclus$PSU[xx], SSU=samclus$SSU[xx])
    # select 2 SSUs per tract
n <- 2
s <- strata(data = as.data.frame(SSUs), stratanames = "PSU",
            size = rep(n,m), method="srswor")
s <- rename(s, c("Prob" = "p2i"))
    # extract the SSU data
    # s contains selection probs of SSUs, need to get those onto data file
SSUsam <- SSUs[s$ID_unit, ]
SSUsam <- cbind(SSUsam, s[, 2:3])
    # identify rows in PSU sample that correspond to sample SSUs
tmp <- samclus$SSU %in% SSUsam$SSU
SSUdat <- samclus[tmp,]
SSUdat <- merge(SSUdat, SSUsam[, c("p2i","SSU")], by="SSU")
    # select srswor from each sample SSU
n.SSU <- m*n
s <- strata(data = as.data.frame(SSUdat), stratanames = "SSU",
            size = rep(50,n.SSU), method="srswor")
s <- rename(s, c("Prob" = "p3i"))
samclus <- getdata(SSUdat, s)
del <- (1:ncol(samclus))[dimnames(samclus)[[2]] %in% c("ID_unit","Stratum")]
samclus <- samclus[, -del]
    # extract pop counts for PSUs in sample
pick <- names(Qi) %in% sort(unique(samclus$PSU))
Qi.sam <- Qi[pick]
    # extract pop counts of SSUs for PSUs in sample
pick <- names(Ni) %in% sort(unique(samclus$PSU))
```

```
Ni.sam <- Ni[pick]
    # extract pop counts for SSUs in sample
pick <- names(Qij) %in% sort(unique(samclus$SSU))
Qij.sam <- Qij[pick]
    # compute full sample weight and wts for PSUs and SSUs
wt <- 1 / samclus$p1i / samclus$p2i / samclus$p3i
w1i <- 1 / samclus$p1i
w2ij <- 1 / samclus$p1i / samclus$p2i
samdat <- data.frame(psuID = samclus$PSU, ssuID = samclus$SSU,
                     w1i = w1i, w2ij = w2ij, w = wt,
                     samclus[, c("y1","y2","y3","ins.cov", "hosp.stay")])
BW3stagePPSe(dat=samdat, v="y1", Ni=Ni.sam, Qi=Qi.sam, Qij=Qij.sam, m,
    lonely.SSU = "mean", lonely.TSU = "mean")
```

`clusOpt2` *Compute optimal sample sizes for a two-stage sample*

Description

Compute the sample sizes that minimize the variance of the *pwr*-estimator of a total in a two-stage sample.

Usage

```
clusOpt2(C1, C2, delta, unit.rv, k=1, CV0=NULL, tot.cost=NULL, cal.sw)
```

Arguments

`C1`	unit cost per primary sampling unit (PSU)
`C2`	unit cost per element
`delta`	homogeneity measure δ
`unit.rv`	unit relvariance
`k`	ratio of $B^2 + W^2$ to unit relvariance
`CV0`	target CV
`tot.cost`	total budget for variable costs
`cal.sw`	specify type of optimum: 1 = find optimal `m.opt` for fixed total budget; 2 = find optimal `m.opt` for target `CV0`

Details

`clusOpt2` will compute m_{opt} and $\bar{n}_{opt}$ for a two-stage sample which uses simple random sampling at each stage or *ppswr* at the first stage and *srs* at the second.

Value

List with values:

`C1`	unit cost per PSU
`C2`	unit cost per element
`delta`	homogeneity measure
`unit relvar`	unit relvariance
`k`	ratio of $B^2 + W^2$ to unit relvariance
`cost`	total budget for variable costs, $C - C_0$ if `cal.sw`=1; or computed cost if `cal.sw`=2
`m.opt`	optimum number of sample PSUs
`n.opt`	optimum number of sample elements per PSU
`CV`	computed CV if `cal.sw`=1; or target CV if `cal.sw`=2

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Hansen,M.H., Hurwitz,W.N., and Madow,W.G. (1953, chap. 6, sect. 16). *Sample Survey Methods and Theory*, Vol.I. John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.3.1). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`clusOpt2fixedPSU`, `clusOpt3`, `clusOpt3fixedPSU`

Examples

```
    # optimum for a fixed total budget
clusOpt2(C1=750, C2=100, delta=0.05, unit.rv=1, k=1, tot.cost=100000, cal.sw=1)
clusOpt2(C1=750, C2=100, delta=seq(0.05,0.25,0.05), unit.rv=1, k=1, tot.cost=100000, cal.sw=1)
    # optimum for a target CV
clusOpt2(C1=750, C2=100, delta=0.01, unit.rv=1, k=1, CV0=0.05, cal.sw=2)
```

`clusOpt2fixedPSU` *Optimal number of sample elements per PSU in a two-stage sample when the sample of PSUs is fixed*

Description

Compute the optimum number of sample elements per primary sampling unit (PSU) for a fixed set of PSUs

Usage

```
clusOpt2fixedPSU(C1, C2, m, delta, unit.rv, k=1, CV0=NULL, tot.cost, cal.sw)
```

Arguments

`C1`	unit cost per PSU
`C2`	unit cost per element
`m`	number of sample PSU's (fixed)
`delta`	homogeneity measure
`unit.rv`	unit relvariance
`k`	ratio of $B^2 + W^2$ to unit relvariance
`CV0`	target CV
`tot.cost`	total budget for variable costs
`cal.sw`	specify type of optimum: 1 = find optimal $\bar{n}$ for fixed total budget; 2 = find optimal $\bar{n}$ for target CV0

Details

`clusOpt2fixedPSU` will compute $\bar{n}_{opt}$ for a two-stage sample which uses simple random sampling at each stage or *ppswr* at the first stage and *srs* at the second. The PSU sample is fixed.

Value

List with values:

`C1`	unit cost per PSU
`C2`	unit cost per element
`m`	number of (fixed) sample PSUs
`delta`	homogeneity measure
`unit relvar`	unit relvariance
`k`	ratio of $B^2 + W^2$ to unit relvariance
`cost`	total budget for variable costs, $C - C_0$ if `cal.sw`=1; or computed cost if `cal.sw`=2
`n`	optimum number of sample elements per PSU
`CV`	computed CV if `cal.sw`=1; or target CV if `cal.sw`=2

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.3.3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

clusOpt2, clusOpt3, clusOpt3fixedPSU

Examples

```
    # optima for a vector of budgets
clusOpt2fixedPSU(C1=500, C2=100, m=100, delta=0.05, unit.rv=2, k=1, CV0=NULL,
       tot.cost=c(100000, 500000, 10^6), cal.sw=1)
    # optima for a target CV and vector of PSU costs
clusOpt2fixedPSU(C1=c(500,1000,5000), C2=100, m=100, delta=0.05, unit.rv=2, k=1,
       CV0=0.05, tot.cost=NULL, cal.sw=2)
```

---

clusOpt3 — *Compute optimal sample sizes for a three-stage sample*

Description

Compute the sample sizes that minimize the variance of the *pwr*-estimator of a total in a three-stage sample.

Usage

```
clusOpt3(unit.cost, delta1, delta2, unit.rv, k1=1, k2=1, CV0=NULL, tot.cost=NULL, cal.sw)
```

Arguments

unit.cost	vector with three components for unit costs: C1 = unit cost per primary sampling unit (PSU); C2 = unit cost per secondary sampling unit (SSU); C3 = unit cost per element
delta1	homogeneity measure among elements within PSUs
delta2	homogeneity measure among elements within SSUs
unit.rv	population unit relvariance
k1	ratio of $B^2 + W^2$ to the population unit relvariance
k2	ratio of $W_2^2 + W_3^2$ to the population unit relvariance
CV0	target CV
tot.cost	total budget for variable costs
cal.sw	specify type of optimum: 1 = find optimal m.opt for fixed total budget; 2 = find optimal m.opt for target CV0

Details

clusOpt3 will compute m_{opt}, $\bar{n}_{opt}$, and $\bar{\bar{q}}_{opt}$ for a three-stage sample which uses simple random sampling at each stage or *ppswr* at the first stage and *srs* at the second and third stages.

Value

List with values:

C1	unit cost per PSU
C2	unit cost per SSU
C3	unit cost per element
delta1	homogeneity measure among elements within PSUs
delta2	homogeneity measure among elements within SSUs
unit relvar	unit relvariance
k1	ratio of $B^2 + W^2$ to the population unit relvariance
k2	ratio of $W_2^2 + W_3^2$ to the population unit relvariance
cost	total budget for variable costs if cal.sw=1; or computed cost if cal.sw=2
m.opt	optimum number of sample PSUs
n.opt	optimum number of sample SSUs per PSU
q.opt	optimum number of sample elements per SSU
CV	achieved CV if cal.sw=1 or target CV if cal.sw=2

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Hansen,M.H., Hurwitz,W.N., and Madow,W.G. (1953, p. 225). *Sample Survey Methods and Theory*, Vol.II. John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.3.2). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

clusOpt2, clusOpt2fixedPSU, clusOpt3fixedPSU

Examples

```
    # optima for a fixed total budget
clusOpt3(unit.cost=c(500, 100, 120), delta1=0.01, delta2=0.10, unit.rv=1,
       k1=1, k2=1, tot.cost=100000,cal.sw=1)
    # optima for a target CV
clusOpt3(unit.cost=c(500, 100, 120), delta1=0.01, delta2=0.10, unit.rv=1,
       k1=1, k2=1, CV0=0.01,cal.sw=2)
```

`clusOpt3fixedPSU` *Compute optimal number of sample secondary sampling units (SSUs) and elements per SSU for a fixed set of primary sampling units (PSUs) in a three-stage sample*

Description

Compute the sample sizes that minimize the variance of the *pwr*-estimator of a total in a three-stage sample when the PSU sample is fixed.

Usage

```
clusOpt3fixedPSU(unit.cost, m, delta1, delta2, unit.rv, k1=1, k2=1, CV0=NULL,
         tot.cost=NULL, cal.sw)
```

Arguments

`unit.cost`	3-vector of unit costs: `C1` = unit cost per PSU; `C2` = unit cost per SSU; `C3` = unit cost per element
`m`	number of sample PSUs (fixed)
`delta1`	homogeneity measure among elements within PSUs
`delta2`	homogeneity measure among elements within SSUs
`unit.rv`	unit relvariance
`k1`	ratio of $B^2 + W^2$ to unit relvariance
`k2`	ratio of $W_2^2 + W_3^2$ to unit relvariance
`CV0`	target CV
`tot.cost`	total budget for variable costs, including PSU costs
`cal.sw`	specify type of optimum: 1 = find optimal `m.opt` for fixed total budget; 2 = find optimal `m.opt` for target CV0

Details

`clusOpt3` will compute $\bar{n}_{opt}$ and $\bar{\bar{q}}_{opt}$ for a three-stage sample which uses simple random sampling at each stage or *ppswr* at the first stage and *srs* at the second and third stages. The set of sample PSUs is assumed to be fixed. "Variable costs" in `tot.cost` includes the budget for all costs that vary with the number of sample PSUs, SSUs, and elements, i.e., $C_1 m + C_2 m\bar{n} + C_3 m\bar{n}\bar{\bar{q}}$.

Value

List with values:

`C1`	unit cost per PSU
`C2`	unit cost per SSU
`C3`	unit cost per element

m	number of sample PSUs (fixed)
delta1	homogeneity measure among elements within PSUs
delta2	homogeneity measure among elements within SSUs
unit relvar	unit relvariance
k1	ratio of $B^2 + W^2$ to unit relvariance
k2	ratio of $W_2^2 + W_3^2$ to unit relvariance
cost	budget constraint, tot.cost if cal.sw=1; computed cost if cal.sw=2
n	optimum number of sample SSUs per PSU
q	optimum number of sample elements per SSU
CV	achieved CV, used if cal.sw=1; or target CV, used if cal.sw=2
CV check	computed CV based on optimal sample sizes; used only if cal.sw=2

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Hansen,M.H., Hurwitz,W.N., and Madow,W.G. (1953, p. 225). *Sample Survey Methods and Theory*, Vol.II. John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.3.2). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

clusOpt2, clusOpt2fixedPSU, clusOpt3

Examples

```
    # optima for a fixed total budget
clusOpt3fixedPSU(unit.cost=c(500, 100, 120), m=100, delta1=0.01, delta2=0.05, unit.rv=1,
     k1=1, k2=1, tot.cost=500000,cal.sw=1)
    # optima for a target CV
clusOpt3fixedPSU(unit.cost=c(500, 100, 120), m=100, delta1=0.01, delta2=0.05, unit.rv=1,
     k1=1, k2=1, CV0=0.05,cal.sw=2)
```

CompMOS *Compute a composite measure of size for domain-based two-stage sampling*

Description

Compute a composite measure of size variable for domain-based sampling that accounts for desired sampling rates of domain units.

Usage

```
CompMOS(dsn = NULL, psuID = NULL, n.PSU = NULL, domain = NULL, domain.req.n = NULL,
 exp.domain.rr = NULL)
```

Arguments

dsn	Data (sampling) frame used for Composite MOS calculations
psuID	PSU Cluster ID
n.PSU	PSU sample size
domain	Vector of domain variable names
domain.req.n	Vector of required sample size from each domain
exp.domain.rr	Vector of expected response rate for each domain as a percentage between 0 and 1

Details

Two-stage samples are often selected from populations for which separate estimates are required for domains, i.e., subpopulations. Composite measures of size for selecting PSU samples with probability proportional to that size can accomplish three things:

1. Self-weighting samples from each of several domains
2. Equal workload in each PSU, i.e., same total sample size in each PSU (across all domains)
3. PSU selection probabilities that give "credit" for containing domains that are relatively rare in the population

CompMOS computes a single composite measure of size, probability of inclusion for each PSU in the sampling frame, and within-PSU sampling rates for each domain. Additional variables regarding survey operations at the PSU domain level are also provided (see Value section below).

Value

A list with four components:

warning	If domain sampling at the desired rate is not feasible in one of more PSUs (i.e. the domain population in the PSU is too small to meet the domain sampling requirements), a warning is included. Review CompMOS.psuID to see where the sampling is not feasible. If all PSUs pass the feasibility test, warning = "None".

`CompMOS.psuID`	A data frame containing the input psuID and domain variables, the composite measure of size, the probability of inclusion, and the PSU/domain sampling fractions, PSU/domain sample sizes, and a feasibility check on each PSU/domain to ensure that the PSU/domain population size is sufficient for sampling.
`CompMOS.design`	A data frame containing domain level survey design and sample information from the input data frame and input domain requirements.
`CompMOS.Ops`	A data frame containing the number of PSUs, the sample workload, and the PSU workload.

Author(s)

George Zipf, Richard Valliant

References

Aldworth J., Hirsch E. L., Martin P. C., Shook-Sa B. E. (2015). 2014 National Survey on Drug Use and Health sample design report. Tech. Rep. Prepared under contract no. HHSS283201300001C by RTI International, Substance Abuse and Mental Health Services Administration, `https://www.samhsa.gov/data/sites/default/files/NSDUHmrbSampleDesign2014v1.pdf`

Singh, A.C. and Harter, R. (2015). Domain sample allocation within primary sampling units in designing domain-level equal probability selection methods. *Survey Methodology*, 41(2), 297-314.

Valliant, R., Dever, J., Kreuter, F. (2018, sec. 10.5). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Examples

```
psuID <- c(1:10)
D1 <- c(50, 50, 50, 50, 50, 70, 50, 50, 50, 50)
D2 <- c(50, 30, 90, 40, 25, 40, 80, 65, 30, 50)
dsn <- cbind.data.frame(psuID, D1, D2)
n.PSU <- 4
domain <- c("D1", "D2")
domain.req.n <- c(130, 50)
exp.domain.rr <- c(1, 1)

CompMOS(dsn = dsn, psuID  = psuID, n.PSU  = n.PSU, domain = domain,
        domain.req.n = domain.req.n, exp.domain.rr = exp.domain.rr)

  # MDarea.popA has multiple rows for each TRACT; need to summarize TRACT/Age totals
  # for input to CompMOS
data(MDarea.popA)
MDpop <- MDarea.popA[,1:8]
MDpop$AgeGrp <- cut(MDpop$Age, breaks = c(0, 12, 17, 23),
                    labels = c("Age.44.or.under", "Age.45-64", "Age.65+"))

xx <- by(MDpop$TRACT, INDICES=MDpop$AgeGrp, table)
 # All tracts do not contain every age group; merge tract/domain count tables, retaining all tracts
xx1 <- cbind(tract=rownames(xx$Age.44.or.under), as.data.frame(unname(xx$Age.44.or.under)))
colnames(xx1)[3] <- 'Age.44.or.under'
xx2 <- cbind(tract=rownames(xx$`Age.45-64`), as.data.frame(unname(xx$`Age.45-64`)))
```

```
colnames(xx2)[3] <- 'Age.45-64'
xx3 <- cbind(tract=rownames(xx$`Age.65+`), as.data.frame(unname(xx$`Age.65+`)))
colnames(xx3)[3] <- 'Age.65+'
pop <- merge(xx1,xx2,by='tract', all=TRUE)
pop <- merge(pop,xx3,by='tract', all=TRUE)
pop <- pop[, -c(2,4,6)]
  # recode counts for missing tract/age-groups to 0
pop[is.na(pop)] <- 0

   # Note that one tract cannot be sampled at the desired rate for the 'Age.65+' domain
MDmos <- CompMOS(dsn  = pop, psuID  = pop$tract, n.PSU  = 15,
              domain = c("Age.44.or.under", "Age.45-64", "Age.65+"),
              exp.domain.rr = c(0.60, 0.70, 0.85),
              domain.req.n  = c(100, 100, 100))
```

`CVcalc2` *Coefficient of variation of an estimated total in a 2-stage sample*

Description

Compute the coefficient of variation of an estimated total in a two-stage design. Primary sampling units (PSUs) can be selected either with probability proportional to size (*pps*) or with equal probability. Elements are selected via simple random sampling (*srs*).

Usage

```
CVcalc2(V=NULL, m=NULL , nbar=NULL, k=1, delta=NULL, Bsq=NULL, Wsq=NULL)
```

Arguments

`V`	unit relvariance of analysis variable in the population
`m`	number of sample PSUs
`nbar`	number of sample elements per PSU
`k`	ratio of $B^2 + W^2$ to V. Default value is 1.
`delta`	measure of homogeneity equal to $B^2/(B^2 + W^2)$
`Bsq`	unit relvariance of PSU totals
`Wsq`	within PSU relvariance

Details

`CVcalc2` computes the coefficient of variation of an estimated total for a two-stage sample. PSUs can be selected either with varying probabilities and with replacement or with equal probabilities and with replacement. Elements within PSUs are selected by simple random sampling. The CV formula is appropriate for approximating the relvariance of the probability-with-replacement (*pwr*)-estimator of a total when the same number of elements is selected within each sample PSU. See Sections 9.2.1–9.2.3 of Valliant, Dever, and Kreuter (2013) for details of formulas.

Value

Value of the coefficient of variation of an estimated total

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran, W.G. (1977, pp.308-310). *Sampling Techniques*. New York: John Wiley & Sons.

Saerndal, C.E., Swensson, B., and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.2.1). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

CVcalc3

Examples

```
CVcalc2(V=1, m=20 , nbar=5, k=1, delta=0.05)
CVcalc2(V=10, m=20 , nbar=5, k=1, delta=0.5)
CVcalc2(V=2.5, m=20 , nbar=5, k=2, Bsq=1, Wsq=4)
```

CVcalc3 *Coefficient of variation of an estimated total in a 3-stage sample*

Description

Compute the coefficient of variation of an estimated total in a three-stage design. Primary sampling units (PSUs) can be selected either with probability proportional to size (*pps*) or with equal probability. Secondary units and elements within SSUs are selected via simple random sampling (*srs*).

Usage

```
CVcalc3(V=NULL, m=NULL , nbar=NULL, qbar=NULL, k1=1, k2=1, delta1=NULL, delta2=NULL,
            Bsq=NULL, Wsq=NULL, W2sq=NULL, W3sq=NULL)
```

Arguments

V	unit relvariance of analysis variable in the population
m	number of sample PSUs
nbar	number of sample secondary units per PSU
qbar	number of sample elements per SSU

k1	ratio of $B^2 + W^2$ to V. Default value is 1.
k2	ratio of $W_2^2 + W_3^2$ to V. Default value is 1.
delta1	measure of homogeneity between PSUs equal to $B^2/(B^2 + W^2)$
delta2	measure of homogeneity between SSUs within PSUs, equal to $W_2^2/(W_2^2 + W_3^2)$
Bsq	unit relvariance of PSU totals, equal to population variance of totals divided by $\bar{t}_U^2$
Wsq	within PSU relvariance of elements
W2sq	unit SSU relvariance
W3sq	unit element relvariance

Details

CVcalc3 computes the coefficient of variation of an estimated total for a three-stage sample. PSUs can be selected either with varying probabilities and with replacement or with equal probabilities and with replacement. SSUs and elements within SSUs are selected by simple random sampling. The CV formula is appropriate for approximating the relvariance of the probability-with-replacement (*pwr*)-estimator of a total when the same number of SSUs is selected in each PSU and the same number of elements is selected within each sample SSU. See Sect. 9.2.4 of Valliant, Dever, and Kreuter (2018) for details of formulas.

Value

Value of the coefficient of variation of an estimated total

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran, W.G. (1977, pp.308-310). *Sampling Techniques*. New York: John Wiley & Sons.

Saerndal, C.E., Swensson, B., and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 9.2.4). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

CVcalc3

Examples

```
CVcalc3(V=1, m=20 , nbar=5, qbar=10, delta1=0.02, delta2=0.10)
CVcalc3(V=1, m=20 , nbar=5, qbar=10, delta1=0.02, delta2=0.09)
CVcalc3(V=2, m=20 , nbar=5, qbar=10, k1=5, k2=10, Bsq=1, Wsq=9, W2sq=2 , W3sq=18 )
```

`deff` *Design effects of various types*

Description

Compute the Kish, Henry, Spencer, or Chen-Rust design effects.

Usage

```
deff(w, x=NULL, y=NULL, p=NULL, strvar=NULL, clvar=NULL, Wh=NULL, nest=FALSE,
      stages=NULL, type)
```

Arguments

`w`	vector of weights for a sample
`x`	matrix of covariates used to construct a GREG estimator of the total of y. This matrix does not include the intercept. Used only for Henry *deff*.
`y`	vector of the sample values of an analysis variable
`p`	vector of 1-draw selection probabilities, i.e., the probability that each unit would be selected in a sample of size 1. Used only for Spencer *deff*.
`strvar`	vector of stratum identifiers; equal in length to that of `w`. Used only for Chen-Rust *deff*.
`clvar`	vector of cluster identifiers; equal in length to that of `w`. Used only for Chen-Rust *deff*.
`Wh`	vector of the proportions of elements that are in each stratum; length is number of strata. Used only for Chen-Rust *deff*.
`nest`	Are cluster IDs numbered within strata (`TRUE` or `FALSE`)? If `TRUE`, cluster IDs can be restarted within strata, e.g., 1,2,3,1,2,3,...
`stages`	Number of stages used in sampling in each stratum; must be a combination of 1s and 2s, e.g. c(1,2,2,1) if there are four strata. Used only for type `"cr"`.
`type`	type of allocation; must be one of `"kish"`, `"henry"`, `"spencer"`, `"cr"`.

Details

`deff` calls one of `deffK`, `deffH`, `deffS`, or `deffCR` depending on the value of the `type` parameter. The Kish design effect is the ratio of the variance of an estimated mean in stratified simple random sampling without replacement (*stsrswor*) to the variance of the estimated mean in *srswor*, assuming that all stratum unit variances are equal. In that case, proportional allocation with equal weighting is optimal. deffK equals 1 + relvar(w) where relvar is relvariance of the vector of survey weights. This measure is not appropriate in samples where unequal weighting is more efficient than equal weighting.

The Henry design effect is the ratio of the variance of the general regression (GREG) estimator of a total of y to the variance of the estimated total in *srswr*. Calculations for the Henry *deff* are done as if the sample is selected in a single-stage and with replacement. Varying selection probabilities

can be used. The model for the GREG is assumed to be $y = \alpha + \beta x + \epsilon$, i.e., the model has an intercept.

The Spencer design effect is the ratio of the variance of the *pwr*-estimator of the total of *y*, assuming that a single-stage sample is selected with replacement, to the variance of the total estimated in *srswr*. Varying selection probabilities can be used.

The Chen-Rust *deff* accounts for stratification, clustering, and unequal weights, but does not account for the use of any auxiliary data in the estimator of a mean. The Chen-Rust *deff* returned here is appropriate for stratified, two-stage sampling.

Value

Numeric design effect for types `kish`, `henry`, `spencer`. For type `cr` a list with components:

`strata components`
: Matrix with *deff*'s due to weighting, clustering, and stratification for each stratum

`overall deff`
: Design effect for full sample accounting for weighting, clustering, and stratification

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Chen, S. and Rust, K. (2017). An Extension of Kish's Formula for Design Effects to Two- and Three-Stage Designs with Stratification. *Journal of Survey Statistics and Methodology*, 5(2), 111-130.

Henry, K.A., and Valliant, R. (2015). A Design Effect Measure for Calibration Weighting in Single-stage Samples. *Survey Methodology*, 41, 315-331.

Kish, L. (1965). *Survey Sampling*. New York: John Wiley & Sons.

Kish, L. (1992). Weighting for unequal Pi. *Journal of Official Statistics*, 8, 183-200.

Park, I., and Lee, H. (2004). Design Effects for the Weighted Mean and Total Estimators under Complex Survey Sampling. *Survey Methodology*, 30, 183-193.

Spencer, B. D. (2000). An Approximate Design Effect for Unequal Weighting When Measurements May Correlate With Selection Probabilities. Survey Methodology, 26, 137-138.

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 14). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`deffK`, `deffH`, `deffS`, `deffCR`

Examples

```
require(reshape)      # has function that allows renaming variables
require(sampling)

set.seed(-500398777)
    # generate population using HMT function
pop.dat <- as.data.frame(HMT())
mos <- pop.dat$x
pop.dat$prbs.1d <- mos / sum(mos)
    # select pps sample
n <- 80
pk <- n*pop.dat$prbs.1d
sam <- UPrandomsystematic(pk)
sam <- sam==1

sam.dat <- pop.dat[sam, ]
dsgn.wts <- 1/pk[sam]
deff(w=dsgn.wts, type="kish")
deff(w=dsgn.wts, y=sam.dat$y, p=sam.dat$prbs.1d, type="spencer")
deff(w=dsgn.wts, x=sam.dat$x, y=sam.dat$y, type="henry")

data(MDarea.popA)

Ni <- table(MDarea.popA$TRACT)
m <- 20
probi <- m*Ni / sum(Ni)
# select sample of clusters
set.seed(-780087528)
sam <- sampling::cluster(data=MDarea.popA, clustername="TRACT", size=m,
                         method="systematic",
                         pik=probi, description=TRUE)
# extract data for the sample clusters
samclus <- getdata(MDarea.popA, sam)
samclus <- rename(samclus, c("Prob" = "pi1"))
# treat sample clusters as strata and select srswor from each
nbar <- 8
s <- sampling::strata(data = as.data.frame(samclus), stratanames = "TRACT",
                      size = rep(nbar,m), method="srswor")
# extracts the observed data
samdat <- getdata(samclus,s)
samdat <- rename(samdat, c("Prob" = "pi2"))
  # add artificial stratum ID
H <- 4
nh <- m * nbar / H
strat1 <- NULL
for (h in 1:H){
  strat1 <- c(strat1, rep(h,nh))
}
wt <- 1/(samdat$pi1*samdat$pi2) * runif(m*nbar)
samdat <- cbind(subset(samdat, select = -c(Stratum)), strat1, wt)

  # 4 strata + clusters; 2 stages in 4 strata
```

```
deffCR(w = samdat$wt, strvar = samdat$strat1, clvar = samdat$TRACT,
       Wh=NULL, y=samdat$y2, clnest = FALSE, stages = c(2,2,2,2))
```

---

deffCR *Chen-Rust design effect*

Description

Chen-Rust design effect for an estimated mean from a stratified, clustered, two-stage sample

Usage

```
deffCR(w, strvar=NULL, clvar=NULL, clnest=NULL, Wh=NULL, y, stages)
```

Arguments

w	vector of weights for a sample; weights should be scaled for estimating population totals
strvar	vector of stratum identifiers; equal in length to that of w.
clvar	vector of cluster identifiers; equal in length to that of w.
clnest	Are cluster IDs numbered within strata (TRUE or FALSE)? If TRUE, cluster IDs can be restarted within strata, e.g., 1,2,3,1,2,3,...
Wh	vector of the proportions of elements that are in each stratum; length is number of strata.
y	vector of the sample values of an analysis variable
stages	Number of stages used in sampling in each stratum. stages must be in the same order as the **sorted**, unique values of strvar and must be a combination of 1s and 2s, e.g. c(1,2,2,1) if there are four strata.

Details

The Chen-Rust *deff* for an estimated mean accounts for stratification, clustering, and unequal weights, but does not account for the use of any auxiliary data in the estimator of a mean. The formula is based on a superpopulation model in which units have a common mean within each stratum; units are correlated within clusters but units in different clusters are uncorrelated. The Chen-Rust *deff* returned here is appropriate for stratified, one-stage or two-stage sampling. Designs may have a combination of one-stage sampling in some strata and two-stage sampling in others. Separate *deff*'s are produced for weighting, clustering, and stratification within each stratum. These cannot be added across strata unless the stratum values of the coefficient of variation of the weights, the sample size of clusters, and the intracluster correlation of y are equal across all strata (see Chen and Rust 2017, p.117).

The weights in w should be scaled for estimating population totals. If the weights sum to the sample size or less in any stratum or overall, an error will result. This can occur if normalized weights are used. All sampling within a stratum must have the same number of stages. In particular, all sampling must be either 2-stage cluster sampling or single-stage sampling. If the vector Wh is not supplied for a stratified sample, it will be estimated based on w.

Value

A list with components:

`strata components`
: Matrix with number of sample first-stage units, intracluster correlation (`rhoh`), coefficient of variation of the weights (`cv2wh`), and *deff*'s due to weighting (`deff.w`), clustering (`deff.c`), and stratification (`deff.s`) for each stratum. When `strvar` or `clvar` are `NULL` appropriate subsets of these are output.

`overall deff`
: Design effect for full sample accounting for weighting, clustering, and stratification. This is computed from Deff* in Chen-Rust (2017, top of p.117) and may differ from a directly computed deff based on the ratio of the complex sample variance to the simple random sample from, e.g., the `survey` package.

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Chen, S. and Rust, K. (2017). An Extension of Kish's Formula for Design Effects to Two- and Three-Stage Designs with Stratification. *Journal of Survey Statistics and Methodology*, 5(2), 111-130.

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 14). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`deff`, `deffH`, `deffK`, `deffS`

Examples

```
require(PracTools)
require(sampling)
require(reshape)
data(MDarea.popA)

Ni <- table(MDarea.popA$TRACT)
m <- 20
probi <- m*Ni / sum(Ni)
# select sample of clusters
set.seed(-780087528)
sam <- sampling::cluster(data=MDarea.popA, clustername="TRACT", size=m,
                         method="systematic",
                         pik=probi, description=TRUE)
# extract data for the sample clusters
samclus <- getdata(MDarea.popA, sam)
samclus <- rename(samclus, c("Prob" = "pi1"))
# treat sample clusters as strata and select srswor from each
nbar <- 8
s <- sampling::strata(data = as.data.frame(samclus), stratanames = "TRACT",
```

```
                      size = rep(nbar,m), method="srswor")
# extracts the observed data
samdat <- getdata(samclus,s)
samdat <- rename(samdat, c("Prob" = "pi2"))
  # add artificial stratum ID
H <- 4
nh <- m * nbar / H
strat1 <- NULL
for (h in 1:H){
  strat1 <- c(strat1, rep(h,nh))
}
wt <- 1/(samdat$pi1*samdat$pi2) * runif(m*nbar)
samdat <- cbind(subset(samdat, select = -c(Stratum)), strat1, wt)

  # 4 strata, SRS in each
deffCR(w = samdat$wt, strvar = samdat$strat1, clvar = NULL,
       Wh=NULL, y=samdat$y2, clnest = NULL, stages = c(1,1,1,1))

  # 4 strata + clusters; 2 stages in 4 strata
deffCR(w = samdat$wt, strvar = samdat$strat1, clvar = samdat$TRACT,
       Wh=NULL, y=samdat$y2, clnest = FALSE, stages = c(2,2,2,2))

  # 4 strata, SRS in strata 1 & 4, 2-stage in strata 2, 3
deffCR(w = samdat$wt, strvar = samdat$strat1,
       clvar = c(1:40, rep(1,20), rep(2,20), rep(1,20), rep(2,20), 1:40),
       Wh=NULL, y=samdat$y2, clnest = TRUE, stages = c(1,2,2,1))
```

---

deffH *Henry design effect for* pps *sampling and GREG estimation of totals*

Description

Compute the Henry design effect for single-stage samples when a general regression estimator is used for a total.

Usage

```
deffH(w, y, x)
```

Arguments

w	vector of inverses of selection probabilities for a sample
y	vector of the sample values of an analysis variable
x	matrix of covariates used to construct a GREG estimator of the total of y. This matrix does not include the intercept.

Details

The Henry design effect is the ratio of the variance of the general regression (GREG) estimator of a total of y to the variance of the estimated total in *srswr*. Calculations for the Henry *deff* are done as if the sample is selected in a single-stage and with replacement. Varying selection probabilities can be used. The model for the GREG is assumed to be $y = \alpha + \beta x + \epsilon$, i.e., the model has an intercept.

Value

numeric design effect

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Henry, K.A., and Valliant, R. (2015). A Design Effect Measure for Calibration Weighting in Single-stage Samples. *Survey Methodology*, 41, 315-331.

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 14). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

deff, deffCR, deffK, deffS

Examples

```
set.seed(-500398777)
    # generate population using HMT function
pop.dat <- as.data.frame(HMT())
mos <- pop.dat$x
pop.dat$prbs.1d <- mos / sum(mos)
    # select pps sample
require(sampling)
n <- 80
pk <- n * pop.dat$prbs.1d
sam <- UPrandomsystematic(pk)
sam <- sam==1
sam.dat <- pop.dat[sam, ]
dsgn.wts <- 1/pk[sam]
deffH(w=dsgn.wts, y=sam.dat$y, x=sam.dat$x)
```

deffK *Kish design effect*

Description

Compute the Kish design effect due to having unequal weights.

Usage

```
deffK(w)
```

Arguments

w	vector of inverses of selection probabilities for a sample

Details

The Kish design effect is the ratio of the variance of an estimated mean in stratified simple random sampling without replacement (*stsrswor*) to the variance of the estimated mean in *srswor*, assuming that all stratum unit variances are equal. In that case, proportional allocation with equal weighting is optimal. deffK equals $1+relvar(w)$ where $relvar$ is relvariance of the vector of survey weights. This measure is not appropriate in samples where unequal weighting is more efficient than equal weighting.

Value

numeric design effect

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Kish, L. (1965). *Survey Sampling*. New York: John Wiley & Sons.

Kish, L. (1992). Weighting for unequal Pi. *Journal of Official Statistics*, 8, 183-200.

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 14). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

deff, deffCR, deffH, deffS

Examples

```
data(nhis)
w <- nhis$svywt
deffK(w)
```

`deffS` *Spencer design effect for an estimated total from a* pps *sample*

Description

Compute the Spencer design effect for single-stage samples selected with probability proportional to a measure of size.

Usage

```
deffS(p, w, y)
```

Arguments

`p`	vector of 1-draw selection probabilities, i.e., the probability that each unit would be selected in a sample of size 1.
`w`	vector of inverses of selection probabilities for a sample
`y`	vector of the sample values of an analysis variable

Details

The Spencer design effect is the ratio of the variance of the *pwr*-estimator of the total of y, assuming that a single-stage sample is selected with replacement, to the variance of the total estimated in *srswr*. Varying selection probabilities can be used.

Value

numeric design effect

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Park, I., and Lee, H. (2004). Design Effects for the Weighted Mean and Total Estimators under Complex Survey Sampling. *Survey Methodology*, 30, 183-193.

Spencer, B. D. (2000). An Approximate Design Effect for Unequal Weighting When Measurements May Correlate With Selection Probabilities. *Survey Methodology*, 26, 137-138.

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 14). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`deff`, `deffCR`, `deffH`, `deffK`

Examples

```
set.seed(-500398777)
    # generate population using HMT function
pop.dat <- as.data.frame(HMT())
mos <- pop.dat$x
pop.dat$prbs.1d <- mos / sum(mos)
    # select pps sample
require(sampling)
n <- 80
pk <- pop.dat$prbs.1d
sam <- UPrandomsystematic(pk)
sam <- sam==1
sam.dat <- pop.dat[sam, ]
dsgn.wts <- 1/pk[sam]
deffS(p=sam.dat$prbs.1d, w=dsgn.wts, y=sam.dat$y)
```

Domainy1y2 *Domain data*

Description

Domainy1y2 is a small data file used for an exercise in sample size calculations.

Usage

```
data(Domainy1y2)
```

Format

A data frame with 30 observations on 2 variables.

y1 an artificial variable

y2 an artificial variable

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Examples

```
data(Domainy1y2)
str(Domainy1y2)
summary(Domainy1y2)
```

dub *Sample sizes for a double sampling design*

Description

Compute samples sizes at each phase of a two-phase design where strata are created using the first phase.

Usage

```
dub(c1, c2, Ctot, Nh, Sh, Yh.bar)
```

Arguments

c1	cost per unit in phase-1
c2	cost per unit in phase-2
Ctot	Total variable cost
Nh	Vector of stratum population counts or proportions
Sh	Vector of stratum population standard deviations
Yh.bar	Vector of stratum population means

Details

Compute the first and second phase sample sizes for a double sampling design. A first phase sample is selected by simple random sampling (*srs*). Strata are formed based on information collected in the first phase. The Neyman allocation to strata of the second phase sample is computed ignoring costs. Optimal total sample sizes are computed for the first and second phases, given per-unit costs for the first and second phases and a fixed total budget for both phases combined.

Value

A list object with elements:

V1	Variance component associated with phase-1
V2	Variance component associated with phase-2
n1	Phase-1 sample size
n2	Total phase-2 sample across all strata
"n2/n1"	Fraction that phase-2 is of phase-1
ney.alloc	Vector of stratum sample sizes for phase-2 sample
Vopt	Variance of mean with the calculated phase-1 and phase-2 sample sizes
nsrs	Size of an *srs* that has cost Ctot, assuming each unit costs c2
Vsrs	Variance of mean in an *srs* of cost Ctot, assuming each unit costs c2
Vratio	Ratio of Vopt to Vsrs
Ctot	Input value of total cost
cost.chk	Computed value of phase-1 plus phase-2 sample with optimal sample sizes; should agree with Ctot

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran W (1977, sect. 12.3) *Sampling Techniques*. New York: John Wiley & Sons, Inc.

Neyman J (1938) Contribution to the theory of sampling human populations. *Journal of the American Statistical Association*, 33(201), 101-116.

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 17.5.2). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Examples

```
Wh <- rep(0.25,4)
Ph <- c(0.02,0.12,0.37,0.54)
Sh <- sqrt(Ph*(1-Ph))
c1 <- 10
c2 <- 50
Ctot <- 20000
dub(c1, c2, Ctot, Nh=Wh, Sh, Yh.bar=Ph)
```

---

gamEst *Estimate variance model parameter γ*

Description

Regresses a y on a set of covariates X where $Var_M(y) = \sigma^2 x^\gamma$ and then regresses the squared residuals on $log(x)$ to estimate γ.

Usage

```
gamEst(X1, x1, y1, v1)
```

Arguments

X1	matrix of predictors in the linear model for y1
x1	vector of x's for individual units in the assumed specification of $Var_M(y)$
y1	vector of dependent variables for individual units
v1	vector proportional to $Var_M(y)$

Details

The function gamEst estimates the power γ in a model where the variance of the errors is proportional to x^γ for some covariate x. Values of γ are typically in [0,2]. The function is iteratively called by gammaFit, which is normally the function that an analyst should use.

Value

The estimate of γ.

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

gammaFit

Examples

```
data(hospital)
x <- hospital$x
y <- hospital$y

X <- cbind(sqrt(x), x)
gamEst(X1 = X, x1 = x, y1 = y, v1 = x)
```

---

gammaFit *Iteratively estimate variance model parameter γ*

Description

Iteratively computes estimate of γ in a model with $E_M(y) = x^T\beta$ and $Var_M(y) = \sigma^2 x^\gamma$.

Usage

```
gammaFit(X, x, y, maxiter = 100, show.iter = FALSE, tol = 0.001)
```

Arguments

X	matrix of predictors in the linear model for y
x	vector of x's for individual units in the assumed specification of $Var_M(y)$
y	vector of dependent variables for individual units
maxiter	maximum number of iterations allowed
show.iter	should values of γ be printed of each iteration? TRUE or FALSE
tol	size of relative difference in $\hat{\gamma}$'s between consecutive iterations used to determine convergence. Algorithm terminates when relative difference is less than tol.

Details

The function gammaFit estimates the power γ in a model where the variance of the errors is proportional to x^γ for some covariate x. Values of γ are typically in [0,2]. The function calls gamEst.

Value

A list with the components:

g.hat	estimate of γ when iterative procedure stopped
converged	TRUE or FALSE depending on whether convergence was obtained
steps	number of steps used by the algorithm

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

gamEst

Examples

```
data(hospital)
x <- hospital$x
y <- hospital$y

X <- cbind(sqrt(x), x)
gammaFit(X = X, x = x, y = y, maxiter=100, tol=0.002)
```

GeoDistMOS *Split geographic PSUs based on a measure of size threshold*

Description

Split geographic PSUs into new geographically contiguous PSUs based on a maximum measure of size for each PSU

Usage

```
GeoDistMOS(lat, long, psuID, n, MOS.var, MOS.takeall = 1, Input.ID = NULL)
```

Arguments

lat	latitude variable in an input file. Must be in decimal format.
long	longitude variable in an input file. Must be in decimal format.
psuID	PSU Cluster ID from an input file.
n	Sample size of PSUs; may be a preliminary value used in the computation to identify certainty PSUs
MOS.var	Variable used for probability proportional to size sampling
MOS.takeall	Threshold relative measure of size value for certainties; must satisfy 0 < MOS.takeall <= 1
Input.ID	ID variable from the input file

Details

GeoDistMOS splits geographic primary sampling units (PSUs) in the input object based on a variable which is used to create the measure of size for each PSU (MOS.var). The goal is to create PSUs of similarly sized MOS. The input file should have one row for each geographic unit, i.e. secondary sampling unit (SSU), with a PSU ID assigned. The latitude and longitude input vectors define the centroid of each input SSU. The complete linkage method for clustering is used. Accordingly, PSUs are split on a distance metric and not on the MOS threshold value. GeoDistMOS calls the function inclusionprobabilities from the sampling package to calculate the inclusion probability for each SSU within a PSU and distHaversine from the geosphere package to calculate the distances between centroids.

Value

A list with two components:

PSU.ID.Max.MOS	A data frame containing the SSU ID value in character format (Input.ID), the original PSU ID (psuID.orig), and the new PSU ID after splitting for the maximum measure of size (psuID.new).
PSU.Max.MOS.Info	A data frame containing the new PSU ID (psuID.new) after splitting for the maximum Measure of Size, the inclusion probability of the PSU ID given the input sample size n (psuID.prob), the measure of size of the new PSU (MOS), the number of SSUs in the new PSU ID (Number.SSUs), and the means of the SSUs latitudes and longitudes that were combined to form the new PSU (PSU.Mean.Latitude and PSU.Mean.Longitude).

Author(s)

George Zipf, Richard Valliant

See Also

GeoDistPSU, GeoMinMOS

Examples

```
data(Test_Data_US)

   # Create PSU ID with GeoDistPSU
g <- GeoDistPSU(Test_Data_US$lat,
                Test_Data_US$long,
                "miles",
                100,
                Input.ID = Test_Data_US$ID)
   # Append PSU ID to input file
library(dplyr)
Test_Data_US <- dplyr::inner_join(Test_Data_US, g$PSU.ID, by=c("ID" = "Input.file.ID"))

   # Split PSUs with MOS above 0.80
m <- GeoDistMOS(lat         = Test_Data_US$lat,
                long        = Test_Data_US$long,
                psuID       = Test_Data_US$psuID,
                n           = 15,
                MOS.var     = Test_Data_US$Amount,
                MOS.takeall = 0.80,
                Input.ID    = Test_Data_US$ID)

   # Create histogram of Measure of Size Values
hist(m$PSU.Max.MOS.Info$psuID.prob,
     breaks = seq(0, 1, 0.1),
     main = "Histogram of PSU Inclusion Probabilities (Certainties = 1)",
     xlab = "Inclusion Probability",
     ylab = "Frequency")
```

`GeoDistPSU` *Form PSUs based on geographic distances*

Description

Combine geographic areas into primary sampling units to limit travel distances

Usage

```
GeoDistPSU(lat, long, dist.sw, max.dist, Input.ID = NULL)
```

Arguments

`lat`	latitude variable in an input file. Must be in decimal format.
`long`	longitude variable in an input file. Must be in decimal format.
`dist.sw`	units for distance; either `"miles"` or `"kms"` (for kilometers)
`max.dist`	maximum distance allowed within a PSU between centroids of geographic units
`Input.ID`	ID field in the input file if present

Details

GeoDistPSU combines geographic secondary sampling units (SSUs), like cities or census block groups, into primary sampling units (PSUs) given a maximum distance allowed between the centroids of the SSUs within each grouped PSU. The input file must have one row for each geographic unit. If the input file does not have an ID field, the function will create a sequential ID that is appended to the output. The latitude and longitude input vectors define the centroid of each input SSU. The complete linkage method for clustering is used. GeoDistPSU calls the functions distm and distHaversine from the geosphere package to calculate the distances between centroids.

Value

A list with two components:

PSU.ID	A data frame with the same number of rows as the input file. Column names are Input.file.ID and psuID. The psuID column contains the PSU number assigned to each geographic unit in the input file; multiple rows of the input file will typically be assigned to the same PSU.
PSU.Info	A data frame with the number of rows equal to the number of PSUs that are created. Column names are Num.SSUs, number of SSUs assigned to each PSU; PSU.Mean.Latitude, mean of the latitudes of the units assigned to a PSU; PSU.Mean.Longitude, mean of the longitudes of the units assigned to a PSU; PSU.Max.Dist, maximum distance among the SSUs in a PSU

.

Author(s)

George Zipf, Richard Valliant

See Also

GeoDistMOS, GeoMinMOS

Examples

```
data(Test_Data_US)
g <- GeoDistPSU(Test_Data_US$lat,
                Test_Data_US$long,
                "miles", 100,
                Input.ID = Test_Data_US$ID)
    # Plot GeoDistPSU output
plot(g$PSU.Info$PSU.Mean.Longitude,
     g$PSU.Info$PSU.Mean.Latitude,
     col  = 1:nrow(g$PSU.Info),
     pch  = 19,
     main = "Plot of PSU Centers",
     xlab = "Longitude",
     ylab = "Latitude")
grid(col = "grey40")

    # Plot GeoDistPSU output with map
```

```
## Not run:
  # install package sf to run usmap_transform
library(ggplot2)
library(sp)
library(usmap)
    # Transform PSUs into usmap projection
g.map  <- cbind(long = g$PSU.Info$PSU.Mean.Longitude,
                lat  = g$PSU.Info$PSU.Mean.Latitude)
g.map  <- as.data.frame(g.map)
g.proj <- usmap::usmap_transform(g.map,
                          input_names  = c("long", "lat"),
                          output_names = c("Long", "Lat"))
usmap::plot_usmap(color = "gray") +
  geom_point(data = g.proj,
             aes(x = Long,
                 y = Lat))
    # Create histogram of maximum distance
hist(g$PSU.Info$PSU.Max.Dist,
     main = "Histogram of Maximum Within-PSU Distance",
     xlab = "Distance",
     ylab = "Frequency")

## End(Not run)
```

`GeoMinMOS` *Check geographic PSUs to determine whether any are less than minimum measure of size threshold*

Description

Identify geographic PSUs whose measure of size is below a specified minimum and combine those PSUs with others

Usage

```
GeoMinMOS(lat, long, geo.var, MOS.var, MOS.min)
```

Arguments

`lat`	latitude variable in an input file. Must be in decimal format.
`long`	longitude variable in an input file. Must be in decimal format.
`geo.var`	Geographic variable ID for grouping
`MOS.var`	Measure of size (MOS) for each PSU
`MOS.min`	Minimum allowed MOS value

Details

GeoMinMOS is a utility function that should be run after using GeoDistMOS or GeoDistPSU. GeoMinMOS identifies each PSU whose measure of size, (MOS.var), is below the minimum specified by MOS.min. Distances to the latitude/longitude centroids of other PSUs are calculated. The undersized PSUs are then combined with the nearest PSUs in proximity order until the minimum MOS is met or exceeded. In some cases, *this can result in the same input PSU being assigned to more than one combined PSU*. Also, the distances between the centroids of the PSUs in a combination may be impractically large. Thus, the new combinations generated by GeoMinMOS should be treated as suggestions that should be manually reviewed and adjusted if desired.

Value

A list with four components:

`Parameter.Information`
: A data frame with three elements: Minimum.MOS = value of MOS.min; Geo.vars.start = total number of PSUs in the input data set; Geo.Vars.lt.min.MOS = number of PSUs whose MOS was less than the minimum.

`Input.Information`
: A data frame containing Geo.Var = geographic variable ID in the input data set used for grouping; Geo.MOS = MOS of each PSU in the input data set; Below.min.MOS = TRUE/FALSE indicator for whether a PSU's MOS was below the minimum in MOS.min.

`Geo.var.MOS.output`
: A data frame with PSUs that were formed by combining undersized PSUs with adequately-sized PSUs. Columns in the data frame are: Geo.Var = new geographic variable ID for a combined PSU. This is equal to geo.var for the undersized PSU used in a combination; New.Geo.MOS = input MOS for each PSU; Geo.Cum.MOS = cumulative MOS for a combined PSU. The last PSU in a combination will have the total size of the combined PSU; Geo.Var.ID = geographic variable ID for a PSU in the input data set; Geo.Var.Num = sequential number (1, 2, etc.) for the PSUs in a combination; Geo.Var.Kms = distance in kilometers of a PSU's centroid to the centroid of the undersized PSU in a combination. The undersized PSU will have a distance of 0; Geo.Var.Miles = distance in miles of a PSU's centroid to the centroid of the undersized PSU in a combination. The undersized PSU will have a distance of 0; Geo.Var.Lat = latitude of the PSU centroid; Geo.Var.Long = longitude of the PSU centroid.

`For.Review`
: A list of the geo.var's of PSUs that are used in more than one combination; these should be manually reviewed to determine which combination is preferred. The distances between PSU centroids in Geo.var.MOS.output can be helpful in the review.

Author(s)

George Zipf, Richard Valliant

See Also

GeoDistPSU, GeoDistMOS

Examples

```
library(PracTools)
library(dplyr)
g <- GeoDistPSU(Test_Data_US$lat,
                Test_Data_US$long,
                "miles",
                100,
                Input.ID = Test_Data_US$ID)
Test_Data_US <- inner_join(Test_Data_US, g$PSU.ID, by=c("ID" = "Input.file.ID"))
GeoMinMOS(lat      = Test_Data_US$lat,
          long     = Test_Data_US$long,
          geo.var  = Test_Data_US$psuID,
          MOS.var  = Test_Data_US$Amount,
          MOS.min  = 200000)
```

HMT *Generate an HMT population*

Description

Generate a population that follows the model in Hansen, Madow, and Tepping (1983)

Usage

```
HMT(N=5000, H=10)
```

Arguments

N	population size
H	number of strata

Details

HMT generates a population based on the model: $E(y) = \alpha + \beta x$, $Var(y) = \sigma^2 x^{3/2}$. Both x and y have gamma distributions. Strata are formed to have approximately the same total of x.

Value

N x 3 matrix with columns:

strat	stratum ID
x	auxiliary variable x
y	analysis variable y

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Hansen,M.H., Madow,W.G., and Tepping,B.J. (1983). An evaluation of model-dependent and probability sampling inferences in sample surveys. *Journal of the American Statistical Association*, 78, 776-793.

Examples

```
    # generate HMT population with 1000 units and 5 strata and plot results
pop <- HMT(N=1000, H=5)
plot(pop[,"x"],pop[,"y"])
```

---

hospital *Hospital Data*

Description

The `hospital` data file is a national sample of short-stay hospitals with fewer than 1000 beds.

Usage

```
data(hospital)
```

Format

A data frame with 393 observations on the following 2 variables.

`y` Number of patients discharged by the hospital in January 1968

`x` Number of inpatient beds in the hospital

Details

The `hospital` data are from the National Hospital Discharge Survey conducted by the U.S. National Center for Health Statistics. The survey collects characteristics of inpatients discharged from non-Federal short-stay hospitals in the United States. This population is from the January 1968 survey and contains observations on 393 hospitals.

Source

National Center for Health Statistics Hospital Discharge Survey of 1968.

References

Herson, J. (1976). An Investigation of Relative Efficiency of Least-Squares Prediction to Conventional Probability Sampling Plans. *Journal of the American Statistical Association*, 71, 700-703.

Examples

```
data(hospital)
str(hospital)
```

labor *Labor force population*

Description

A clustered population of persons extracted from the September 1976 Current Population Survey (CPS)

Usage

```
data(labor)
```

Format

A data frame with 478 persons on the following variables:

`h` stratum

`cluster` cluster (or segment) number

`person` person number

`age` age of person

`agecat` age category (1 = 19 years and under; 2 = 20-24; 3 = 25-34; 4 = 35-64; 5 = 65 years and over)

`race` race (1 = non-Black; 2 = Black)

`sex` Gender (1=Male; 2=Female)

`HoursPerWk` Usual number of hours worked per week

`WklyWage` Usual amount of weekly wages (in 1976 U.S. dollars)

`y` An artificial variable generated to follow a model with a common mean. Persons in the same cluster are correlated. Persons in different clusters are uncorrelated under the model.

Details

This population is a clustered population of 478 persons extracted from the September 1976 Current Population Survey (CPS) in the United States. The clusters are compact geographic areas used as one of the stages of sampling in the CPS and are typically composed of about 4 nearby households. The elements within clusters for this illustrative population are individual persons.

Source

Current Population Survey, `https://www.census.gov/programs-surveys/cps.html`

Examples

```
data(labor)
str(labor)
table(labor$h)
hist(labor$WklyWage)
```

MDarea.popA *Maryland area population*

Description

An artificial population of census tracts, block groups, and persons

Usage

```
data(MDarea.popA)
```

Format

A data frame with 343,398 persons on the following variables:

`PSU` A grouping of block groups (`BLKGROUP`) which has about 4290 persons

`SSU` A grouping of block groups which has about 1010 persons

`TRACT` A geographic area defined by the Census Bureau. Tracts generally have between 1,500 and 8,000 people but have a much wider range in Anne Arundel county.

`BLKGROUP` Block group. A geographic area defined by the Census Bureau. Block groups generally have between 600 and 3,000 people.

`Hispanic` Hispanic ethnicity (1=Hispanic; 2=Non-Hispanic)

`Gender` Gender (1=Male; 2=Female)

`Age` 23 level age category (1 = Under 5 years; 2 = 5 to 9 years; 3 = 10 to 14 years; 4 = 15 to 17 years; 5 = 18 and 19 years; 6 = 20 years; 7 = 21 years; 8 = 22 to 24 years; 9 = 25 to 29 years; 10 = 30 to 34 years; 11 = 35 to 39 years; 12 = 40 to 44 years; 13 = 45 to 49 years; 14 = 50 to 54 years; 15 = 55 to 59 years; 16 = 60 and 61 years; 17 = 62 to 64 years; 18 = 65 and 66 years; 19 = 67 to 69 years; 20 = 70 to 74 years; 21 = 75 to 79 years; 22 = 80 to 84 years; 23 = 85 years and over)

`person` Counter for person within tract/block group/Hispanic/Gender/Age combination

`y1` Artificial continuous variable

`y2` Artificial continuous variable

`y3` Artificial continuous variable

`ins.cov` Medical coverage (0 = person does not have medical insurance coverage; 1 = person has medical insurance coverage)

`hosp.stay` Overnight hospital stay (0 = person did not have an overnight hospital stay in last 12 months; 1 = person did have an overnight hospital stay in last 12 months)

Details

A dataset of 343,398 persons based on the 2000 decennial U.S. Census for Anne Arundel County in the US state of Maryland. Person records were generated based on counts from the 2000 census. Individual values for each person were generated using models. Groupings to form the variables PSU and SSU were done after sorting the census file by tract and block group within tract.

Note that MDarea.popA is different from the dataset, MDarea.pop, that is used in the book by Valliant, Dever, and Kreuter (2018). MDarea.pop is larger with 403,997 persons. MDarea.popA was created by taking an equal probability, systematic subsample from MDarea.pop. MDarea.popA does have the same numbers of TRACTs, PSUs, and SSUs as MDarea.pop. The smaller data set was created to meet the CRAN size limit on installed packages. The full population, MDarea.pop, can be downloaded from https://umd.app.box.com/v/PracTools2ndEdition.

Source

2000 U.S. decennial census, http://www.census.gov/main/www/cen2000.html

References

Valliant, R., Dever, J., Kreuter, F. (2018). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Examples

```
data(MDarea.popA)
str(MDarea.popA)
table(MDarea.popA$TRACT)
table(MDarea.popA$TRACT, MDarea.popA$Hispanic)
```

---

mibrfss *Michigan Behavioral Risk Factor Surveillance Survey*

Description

Demographic and health related variables from a U.S. household survey in the state of Michigan

Usage

```
data(mibrfss)
```

Format

A data frame with 2485 observations on persons for the following 21 variables.

SMOKE100 Smoked 100 or more cigarettes in lifetime (1 = Yes; 2 = No)

BMICAT3 Body mass index category (1 = Neither overweight nor obese (BMI < 25); 2 = Overweight (25 <= BMI <= 30); 3 = Obese (BMI > 30))

AGECAT Age group (1 = 18-24 years; 2 = 25-34 years; 3 = 35-44 years; 4 = 45-54 years; 5 = 55-64 years; 6 = 65+)

GENHLTH General health (self-reported) (1 = Excellent; 2 = Very good; 3 = Good; 4 = Fair; 5 = Poor)

PHYSACT Physical activity: In last month participated in activities such as running, calisthenics, golf, gardening, or walking for exercise (1 = Yes; 2 = No)

HIGHBP High blood pressure: Have you ever been told by a doctor, nurse, or other health professional that you have high blood pressure? (1 = Yes; 2 = No)

ASTHMA Asthma: Have you ever been told by a doctor, nurse, or other health professional that you have asthma? (1 = Yes; 2 = No)

HISPANIC Hispanic ethnicity (1 = Yes; 2 = No; 7 = Missing)

WEIGHT Body weight in pounds

GENDER Gender (1 = Male; 2 = Female)

CELLPHON Has a wireless phone (1 = Yes; 2 = No)

INETHOME Has access to the Internet at home (1 = Yes; 2 = No)

WEBUSE How often do you use the Internet at home? Would you say, at least once a day, five to six times a week, two to four times a week, about once a week, less than once a week, or have you not used the Internet in the last month? (1 = At least once a day; 2 = 5-6 times a week; 3 = 2-4 times a week; 4 = About once a week; 5 = Less than once a week; 6 = Not in the last month)

RACECAT Race (1 = White; 2 = African American; 3 = Other)

EDCAT Education level (1 = Did not graduate high school; 2 = Graduated high school; 3 = Attended college or technical school; 4 = Graduated from college or technical school)

INCOMC3 Income category (1 = Less than $15000; 2 = $15000 to less than $25000; 3 = $25000 to less than $35000; 4 = $35000 to less than $50000; 5 = $50000 or more)

DIABETE2 Diabetes: Have you ever been told by a doctor, nurse, or other health professional that you have diabetes? (1 = Yes; 2 = No)

CHOLCHK Cholesterol check: Blood cholesterol is a fatty substance found in the blood. Have you ever had your blood cholesterol checked? (1 = Yes; 2 = No)

BMI Body mass index (continuous)

BINGE2 Binge drinking: At risk for binge drinking based on alcohol consumption responses (1 = Yes; 2 = No)

ARTHRIT Arthritis: Have you ever been told by a doctor, nurse, or other health professional that you have some form of arthritis, rheumatoid arthritis, gout, lupus, or fibromyalgia, or have joint symptoms of arthritis? (1 = Yes; 2 = No; 3 = Don't know, not sure, or refused)

Details

The Michigan Behavioral Risk Factor Surveillance Survey (MIBRFSS) is part of a national state-by-state system of surveys used to monitor health conditions in the U.S. Data are collected through telephone household interviews. Demographic variables and a few health related variables are included in this subset. The `mibrfss` data set contains observations on 2845 persons and is extracted from the 2003 U.S. survey. The file contains only persons 18 years and older.

Source

Michigan Behavioral Risk Factor Surveillance Survey of 2003 sponsored by the U.S. Center for Disease Control. https://www.cdc.gov/brfss/

See Also

nhis, nhis.large

Examples

```
data(mibrfss)
str(mibrfss)
table(mibrfss$SMOKE100, useNA = "always")
table(mibrfss$BMICAT3, useNA="always")
```

---

nAuditAttr *Sample sizes for an attribute sample in an audit*

Description

Compute a sample size for an audit where the goal is to control the probability of observing only a small number of errors given an underlying error rate in the population. Auditors refer to this as an attribute sample.

Usage

```
nAuditAttr(TolRate=0.05, AccDev, CL, N=5000)
```

Arguments

TolRate	Proportion of units in the population with an attribute, e.g., errors in an audit. Auditors term this the 'tolerable rate of deviation' in the population to be tested.
AccDev	Acceptable deviation, which is the number of units with the attribute (i.e., the number of errors) that would be acceptable in the sample. The largest proportion of errors that would be deemed to be acceptable in an audit would be AccDev/N.
CL	Probability that the sample will contain an acceptable number of errors. Auditors refer to this as 'confidence level'. The probability that the sample will contain AccDev errors or fewer is 1-CL.
N	Size of the population of records to be audited.

Details

nAuditAttr computes the minimum sample size required so that the probability, 1-CL of detecting less than or equal to a specified number of errors in the sample, is controlled. The sample is assumed to be selected with equal probabilities. AccDev is the largest number of errors in the sample that will be considered as meeting the audit standards. TolRate is the underlying population error rate, which is typically set to be larger than AccDev/N. The sample size is computed in two ways: (1) using the hypergeometric distribution, which accounts for the size of the population and (2) with the binomial distribution, which will be appropriate if the population is very large. When N is large and the sampling fraction is small, both sample sizes will be approximately the same.

Value

List object with values:

Pop.Size — population size

Tol.Dev.Rate — proportion of records with errors in population

Acceptable.Errors — largest number of errors, found in the sample, that will meet audit standards

Sample.Size.Hypergeometric — minimum sample size needed to detect AccDev errors in the sample computed via the hypergeometric distribution

Sample.Size.Binomial — minimum sample size needed to detect AccDev errors in the sample computed via the binomial distribution

Author(s)

George Zipf, Richard Valliant

References

GAO (2020). Financial Audit Manual, Volume 1, section 450.08. Washington DC; https://www.gao.gov/assets/gao-18-601g.pdf

Stewart, Trevor R. (2012). *Technical Notes on the AICPA Audit Guide: Audit Sampling*. American Institute of Certified Public Accountants, Inc. New York, NY 10036-8775; https://www.aicpa-cima.com/home

Examples

```
# Examples from the US GAO Financial Audit Manual (2020), Figure 450.1, Table 1
nAuditAttr(AccDev = 0, CL = .90)
nAuditAttr(AccDev = 1, CL = .90)
nAuditAttr(AccDev = 2, CL = .90)
nAuditAttr(AccDev = 3, CL = .90)
nAuditAttr(AccDev = 4, CL = .90)
```

nAuditMUS *Sample sizes for a Monetary Unit Sample in an audit*

Description

Compute a sample size for an audit where the goal is to control the probability of observing only a small number of errors given an underlying error rate in the population. The sample will be selected with probabilities proportional to a measure of size (MOS). When the MOS of each record is a monetary unit, auditors refer to this as an monetary unit sampling or dollar unit sampling.

Usage

```
nAuditMUS(MUSVar, Value.sw, CL = 0.90, Error.sw, Tol.Error, Exp.Error = 0)
```

Arguments

MUSVar	The measure of size for monetary unit sampling (MUS)
Value.sw	Determines whether the monetary unit sample is based on positive values, negative values, or absolute values. If Value.sw = "Positive" or "Pos", only positive values of MUSVar are used; if "Negative" or "Neg" only negative values are used; if "Absolute" or "Abs", all values of MUSVar are used after taking their absolute values.
CL	Probability that the sample will contain an acceptable number of errors. Auditors refer to this as 'confidence level'. The probability that the sample will contain the tolerable number of errors or fewer is 1-CL. The range of CL is 0 to 1.
Error.sw	Determines whether errors are based on monetary amounts or percentages, i.e., whether Tol.error is interpreted as a dollar amount (Error.sw = "Absolute" or "Amt") or as a percent (Error.sw = "Percent" or "Pct").
Tol.Error	The amount of error expressed as a value or a percentage that the auditor considers tolerable. If Error.sw is "Percent" or "Pct", Tol.Error is a percent between 0 and 100. If Error.sw = "Absolute" or "Amt", Tol.Error is interpreted as a dollar amount.
Exp.Error	The amount of error, expressed as a value or a percentage, that the auditor expects in the population. If Error.sw is "Percent" or "Pct", Exp.Error is a percent between 0 and 100. If Error.sw = "Absolute" or "Amt", Exp.Error is interpreted as a dollar amount.

Details

nAuditMUS computes the minimum sample size required for a given population, tolerable error rate or misstatement, and desired confidence level. If the expected error or misstatement is 0, (i.e. Exp.Error = 0), then the sample size is computed using the hypergeometric distribution where the acceptable number of deviations is 0. If the expected error is greater than 0, then sample size is computed by first calculating the maximum sample size where the number of deviations divided by the sample size is less than the expected error, then calculating the minimum sample size where

the number of deviations divided by the sample size is greater than the expected error, and finally performing a straight line interpolation between these two values where the interpolated value is the specified expected error. The returned sample size calculation is the ceiling of that interpolated sample size.

Value

List object with values:

`Value.Range`	Whether the MUS variable is for positive, negative, or absolute values as defined by `Value.sw`
`Error.Type`	Amount or Percent as defined by `Error.sw`
`Tol.Error.Rate`	The tolerable error expressed as a percentage of items if `Error.sw` = "Percent" or "Pct" or as a percentage of total monetary value otherwise
`Exp.Error.Rate`	The expected error expressed as a percentage of items if `Error.sw` = "Percent" or "Pct" or as a percentage of total monetary value otherwise
`Number.Records`	The population count of records in the value range based on selecting ones with positive, negative or absolute value of `MUSVar`
`Sample.Size`	Minimum sample size needed to meet tolerable and expected error rate requirements
`Number.HighVal`	Number of records that are high value (exceed the interval used for systematic sampling) and will be certainties in the sample
`Positive.Pop.Dollars`	The absolute value of the total dollar (or other monetary unit) amount in the population in the value range
`Conf.level`	Probability that the sample will meet MUS requirements
`Sampling.Interval`	Spacing or skip interval that would be used in a systematic probability proportional to monetary unit sampling

Author(s)

George Zipf, Richard Valliant

References

GAO (2020). Financial Audit Manual, Volume 1, section 480.21-480.26. Washington DC; https://www.gao.gov/assets/gao-18-601g.pdf

See Also

`nAuditAttr`

Examples

```
    # generate an artificial population with some negative monetary amounts
EX <- 1000
relvar <- 2
alpha <- 1/relvar
sigma <- EX * relvar
lowval <- 100                   # minimum positive X's allowed
prop.neg <- 0.05                # proportion of pop with negative values
N.neg <- floor(1000 * prop.neg)    # number of negative X's allowed
X <- rgamma(n=1000, shape=alpha, scale=sigma)
Xlow <- sort(X)[1:N.neg]
xneg <- -Xlow - lowval
xpos <- X[N.neg:length(X)]
X <- c(xneg, xpos)

nAuditMUS(X, Value.sw = "Pos", Error.sw = "Amount", Tol.Error = 180000, Exp.Error = 10000)
nAuditMUS(X, Value.sw = "Pos", Error.sw = "Pct", Tol.Error = 18, Exp.Error = 3)
nAuditMUS(X, Value.sw = "Abs", Error.sw = "Amount", Tol.Error = 180000, Exp.Error = 10000)
```

---

nCont *Compute a simple random sample size for an estimated mean*

Description

Compute a simple random sample size using either a target coefficient of variation, CV_0, or target variance, V_0, for an estimated mean.

Usage

```
nCont(CV0=NULL, V0=NULL, S2=NULL, ybarU=NULL, N=Inf, CVpop=NULL)
```

Arguments

CV0	target value of coefficient of variation of $\bar{y}_s$
V0	target value of variance of $\bar{y}_s$
S2	unit (population) variance
ybarU	population mean of target variable
N	number of units in finite population
CVpop	unit (population) coefficient of variation

Details

If CV_0 is the desired target, then the unit CV, CVpop, or the population mean and variance, ybarU and S2, must also be provided. If V_0 is the constrained value, then S2 must be also be included in the function call.

Value

numeric sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nLogOdds, nProp, nPropMoe, nWilson

Examples

```
nCont(CV0=0.05, CVpop=2)
nCont(CV0=0.05, CVpop=2, N=500)
nCont(CV0=0.10/1.645, CVpop=1)

    # Compute sample size for a ratio estimator in smho98 population
    # Identify large units to select with certainty first
data(smho98)
cert <- smho98[,"BEDS"] > 2000
tmp <- smho98[!cert, ]
tmp <- tmp[tmp[, "BEDS"] > 0, ]

x <- tmp[,"BEDS"]
y <- tmp[, "EXPTOTAL"]
m <- lm(y ~ 0 + x, weights = 1/x)
ybarU <- mean(y)
S2R <- sum(m$residuals^2/(length(x)-1))
nCont(CV0=0.15, S2=S2R, ybarU=ybarU)
```

---

nContMoe *Compute a simple random sample size for an estimated mean of a continuous variable based on margin of error*

Description

Compute a simple random sample size using a margin of error specified as the half-width of a normal approximation confidence interval or the half-width relative to the population mean.

Usage

```
nContMoe(moe.sw, e, alpha=0.05, CVpop=NULL, S2=NULL, ybarU=NULL, N=Inf)
```

Arguments

moe.sw	switch for setting desired margin of error (1 = CI half-width on the mean; 2 = CI half-width on the mean divided by $\bar{y}_U$)
e	desired margin of error; either $e = z_{1-\alpha/2}\sqrt{V(\bar{y}_s)}$ or $e = z_{1-\alpha/2}CV(\bar{y}_s)$
alpha	1 - (confidence level)
CVpop	unit (population) coefficient of variation
S2	population variance of the target variable
ybarU	population mean of target variable
N	number of units in finite population

Details

If moe.sw=1, then S2 must be provided. If moe.sw=2, then either (i) CVpop or (ii) S2 and ybarU must be provided.

Value

numeric sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nCont, nLogOdds, nProp, nPropMoe, nWilson

Examples

```
nContMoe(moe.sw=1, e=0.05, alpha=0.05, S2=2)
nContMoe(moe.sw=1, e=0.05, alpha=0.05, S2=2, N=200)
nContMoe(moe.sw=2, e=0.05, alpha=0.05, CVpop=2)
nContMoe(moe.sw=2, e=0.05, alpha=0.05, CVpop=2, N=200)
nContMoe(moe.sw=2, e=0.05, alpha=0.05, S2=4, ybarU=2)
```

nContOpt *Compute the sample size required to estimate the mean of a continuous variable by optimizing the numbers of take-alls and non-take-all units selected by probability sampling*

Description

Compute a sample size required to achieve a precision target for an estimated mean, $\hat{\bar{y}}_s$, based on splitting the sample between take-alls and non-take-alls. The sample design for non-take-alls can be either simple random sampling or probability proportional to size sampling.

Usage

```
nContOpt(X, Y = NULL, CV0 = NULL, V0 = NULL, design = NULL)
```

Arguments

X	population variable used for determining take-all cutoff and for selecting a probability proportional to size sample if design = "PPS". X is a vector that contains a value for every unit in the population.
Y	variable used for computing a population variance; required if design = "PPS". Y is ignored if design = "SRS". X is a vector that contains a value for every unit in the population and is the same length as Y.
CV0	target value of coefficient of variation of $\hat{\bar{y}}_s$
V0	target value of variance of $\hat{\bar{y}}_s$
design	Sample design to be used for non-take-alls; must be either "SRS" or "PPS".

Details

Compute a sample size based on splitting the sample between take-alls and non-take-alls in a way that achieves either a target coefficient of variation or a target variance for an estimated mean. The function sorts the file in descending order by X and then systematically designates units as take-alls (certainty selections) starting from largest to smallest, and computes the sample size of non-take-alls needed to achieve the precision target. Initially, no unit in the ordered list is a certainty, and if design = "SRS", the first value in nContOpt.curve is the same as nCont produces under identical inputs. In each pass, the algorithm increases the number of certainties. In the second pass, the first value is taken as a certainty and the non-take-all sample size is based on units 2:N, where N is the population size. On the third pass, the first two values are taken as certainties and the non-take-all sample size is based on units 3:N. The function cycles through units 1:(N-1) with take-alls increasing by 1 each cycle, and determines the minimum total sample size needed to achieve the specified precision target. The optimum sample size nContOpt.n combines certainties and non-certainties for its value.

The sample design can be either simple random sampling or probability proportional to size sampling. When design = "SRS", calculations are based only on X. The SRS variance formula is for without replacement sampling so that a finite population correction factor (*fpc*) is included. When design = "PPS", X is used for the measure of size and Y is the variable for computing the variance

used to determine the sample size. The PPS variance is computed for a with-replacement design, but an ad hoc *fpc* is included. Either `CV0` or `V0` must be provided but not both.

Value

A list with five components:

`nContOpt.Curve`	The sample size for the given inputs based on the number of take-alls incrementing from 1 to N-1
`Take.alls`	A TRUE/FALSE vector for whether the element in the `X` vector is a take-all
`nContOpt.n`	The minimum sample size (take-alls + non-take-alls) required for the given inputs, rounded to 4 decimal places
`Min.Takeall.Val`	The minimum value of `X` for the take-alls
`n.Take.all`	The number of take-all units in the optimal sample

Author(s)

George Zipf, Richard Valliant

See Also

`nCont`, `nContMoe`

Examples

```
nContOpt(X = TPV$Total.Pot.Value, CV0 = 0.05, design = "SRS")
nContOpt(X = TPV$Total.Pot.Value, V0  = 5e+14, design = "SRS")
g <- nContOpt(X = TPV$Total.Pot.Value, CV0 = 0.05, design = "SRS")
plot(g$nContOpt.Curve,
     type = "o",
     main = "Sample Size Curve",
     xlab = "Take-all / Sample Split Starting Value",
     ylab = "Total sample size (take-alls + non-tale-alls)" )
nContOpt(X = TPV$Total.Pot.Value, Y = TPV$Y, CV0 = 0.05, design = "PPS")
```

---

nDep2sam *Simple random sample size for difference in means*

Description

Compute a simple random sample size for estimating the difference in means when samples overlap

Usage

```
nDep2sam(S2x, S2y, g, r, rho, alt, del, sig.level=0.05, pow=0.80)
```

Arguments

S2x	unit variance of analysis variable *x* in sample 1
S2y	unit variance of analysis variable *y* in sample 2
g	proportion of sample 1 that is in the overlap with sample 2
r	ratio of the size of sample 1 to that of sample 2
rho	unit-level correlation between *x* and *y*
alt	should the test be 1-sided or 2-sided; allowable values are alt="one.sided" or alt="two.sided".
del	size of the difference between the means to be detected
sig.level	significance level of the hypothesis test
pow	desired power of the test

Details

nDep2sam computes sample sizes in two groups that are required for testing whether the difference in group means is significant. The power of the test is one of the input parameters. The samples have a specified proportion of units in common. Both samples are assumed to be selected via simple random sampling.

Value

List with values:

n1	sample size in group 1
n2	sample size in group 2
S2x.S2y	unit variances in groups 1 and 2
delta	difference in group means to be detected
gamma	proportion of sample 1 that is in the overlap with sample 2
r	ratio of the size of sample 1 to that of sample 2
rho	unit-level correlation between analysis variables in groups 1 and 2
alt	type of test: one-sided or two-sided
sig.level	significance level of test
power	power of the test

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 4). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Woodward, M. (1992). Formulas for Sample Size, Power, and Minimum Detectable Relative Risk in Medical Studies. *The Statistician*, 41, 185-196.

See Also

nProp2sam

Examples

```
nDep2sam(S2x=200, S2y=200,
            g=0.75, r=1, rho=0.9,
            alt="one.sided", del=5,
            sig.level=0.05, pow=0.80)
```

---

nDomain *Compute a simple random sample size for an estimated mean or total for a domain*

Description

Compute a simple random sample size using either a target coefficient of variation, $CV_0(d)$, or target variance, $V_0(d)$, for an estimated mean or total for a domain.

Usage

```
nDomain(CV0d=NULL, V0d=NULL, S2d=NULL, ybarUd=NULL, N=Inf, CVpopd=NULL, Pd, est.type)
```

Arguments

CV0d	target value of coefficient of variation of estimated domain mean or total
V0d	target value of variance of estimated domain mean or total
S2d	unit (population) variance for domain units
ybarUd	population mean of target variable for domain units
N	number of units in full finite population (not just the domain population)
CVpopd	unit (population) coefficient of variation for domain units
Pd	proportion of units in the population that are in the domain
est.type	type of estimate; allowable values are "mean" or "total"

Details

If CV0d is the desired target, then the unit CV, CVpopd, or the domain population mean and variance, ybarUd and S2d, must also be provided. If V0d is the constrained value, then ybarUd must be also be included in the function call. CV0d will then be computed as sqrt(V0d)/ybarUd.

Value

numeric sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, sec. 3.5.2). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nCont, nLogOdds, nProp, nPropMoe, nWilson

Examples

```
nDomain(CV0d=0.05, N=Inf, CVpopd=1, Pd=0.5, est.type="total")
nDomain(CV0d=0.05, N=Inf, CVpopd=1, Pd=0.5, est.type="mean")
nDomain(V0d=50, ybarUd=50, S2d=100, N=Inf, Pd=0.5, est.type="total")
nDomain(CV0d=0.05, ybarUd=50, S2d=100, N=Inf, Pd=0.5, est.type="total")
nDomain(CV0d=0.05, ybarUd=50, S2d=100, N=Inf, Pd=0.5, est.type="mean")
```

---

nEdge — *Compute the total sample size for a stratified, simple random sample based on an Edgeworth approximation*

Description

Compute the total stratified, simple random sample size for various allocations that is large enough to insure adequate coverage of a normal approximation confidence interval (CI) for a population mean.

Usage

```
nEdge(ci.lev, side, epsilon = 0.005, dat, pop.sw = TRUE, wts = NULL, hcol=NULL, ycol,
      alloc = NULL, Costh = NULL, nh.lowerbd = NULL
)
```

Arguments

ci.lev	desired confidence level for a 1- or 2-sided normal approximation confidence interval based on an estimated mean; must be in the interval (0,1)
side	either "two.sided", "one.sided.upper" or "one.sided.lower" for type of confidence interval
epsilon	tolerance on coverage probability; the sample should be large enough that CI coverage is within $\pm$ epsilon of ci.lev; must be in the interval (0,1)
dat	either a population or sample data frame
pop.sw	TRUE if dat is for a full population; FALSE if dat is for a sample

wts	vector of weights if dat is a sample; if dat is for a population, wts = NULL
hcol	column of dat that contains the stratum ID; strata can be character or numeric
ycol	column of dat that contains the analysis variable; must be numeric
alloc	allocation to the strata; must be one of prop, equal, neyman, totcost, totvar, or NULL
Costh	vector of costs per unit in each stratum; these exclude fixed costs that do not vary with the sample size
nh.lowerbd	Scalar value of the minimum value of the sample size allowed for any stratum

Details

nEdge computes the total sample size needed in either a stratified or unstratified simple random sample so that the coverage probability of a confidence interval is within a specified tolerance (epsilon) of a nominal confidence level (ci.lev). The calculation assumes that there is a single estimated mean of the variable ycol that is of key importance in a sample. Confidence intervals for the finite population mean are usually computed using the normal approximation whose accuracy depends on the underlying structure of the analytic variable and the total sample size.

In some applications, assuring that CIs have near nominal coverage is critical. For example, for some items on business tax returns the US Internal Revenue Service allows sample estimates to be used but sets precision standards based on the lower (or upper) limit of a 1-sided CI. In some cases the computed sample size from nEdge may be too small to meet other goals of a sample (such as a precision requirement) and may need to be adjusted.

Using a three-term Edgeworth approximation to the distribution of the estimated overall mean in Qing & Valliant (2026), which extends results in Qing & Valliant (2025), nEdge computes the total sample size needed so that a CI will have coverage equal to the nominal value in ci.lev plus or minus the tolerance epsilon. The remainder term in the approximation used in nEdge is $O(n^{-3/2})$. In previous versions of PracTools, the remainder term in the approximation used in nEdge was $O(n^{-1})$, i.e. this version of nEdge provides a more accurate sample size approximation. See Qing & Valliant (2026) for comparisons between the two levels of approximation.

This was also the level of approximation used in the function nEdgeSRS in previous versions of Practools. nEdgeSRS has been deleted from the package because nEdge also can be used for simple random sampling. The calculation in nEdge assumes that the sampling fraction in each stratum is negligible. The total sample size returned by nEdge is based on the overall Edgeworth criterion; the resulting stratum sample sizes may not be large enough so that the normal approximation is adequate for each stratum estimator.

When dat is a sample, the weights (wts) used in the estimator of the mean are assumed to be scaled for estimating population totals. They can be inverse selection probabilities, i.e. ones used in the π-estimator, or weights that have been adjusted to account for nonresponse or coverage errors.

The value of nh.lowerbd is usually set to be 2 or more to permit variances to be estimated.

Value

If there are no strata, the value is a list with components:

CI type	one-sided lower, one-sided upper, or two-sided
epsilon	tolerance on CI coverage

`Total sample size`
: numeric sample size

`allocation`
: type of allocation to strata or NULL if no strata are used

`g1`
: skewness of y

`g2`
: excess kurtosis of y, i.e., 4th central moment divided by the square of the variance minus 3

`q1`
: 1st polynomial in Edgeworth expansion of the t-statistic

`q2`
: 2nd polynomial in Edgeworth expansion of the t-statistic

If the sample design has strata, the value is a list with the first four items above followed by

`Stratum values`
: Data frame with columns for stratum, number of population units in each stratum, number of sample units allocated to each stratum (`nh`), value of `ch` which depends on the value of `alloc` (see Qing & Valliant 2026), stratum skewness `g1h`, stratum excess kurtosis `g2h`

If `alloc` = "totcost" or "totvar", the total cost of the allocation is returned and `Stratum values` contains a column for `Costh`.

Author(s)

Richard Valliant, Siyu Qing

References

Qing, S. and Valliant, R. (2025). Extending Cochran's Sample Size Rule to Stratified Simple Random Sampling with Applications to Audit Sampling. *Journal of Official Statistics*, 41(1), 309-328.

Qing, S. and Valliant, R. (2026). Minimum Sample Size Rule for STSRS: A Three-Term Edgeworth Expansion for the Studentized Sample Mean, accepted by *Journal of Survey Statistics and Methodology*.

U.S. Internal Revenue Service (2011). 26 CFR 601.105: Examination of returns and claims for refund, credit or abatement: determination of correct tax liability. Washington DC. https://www.irs.gov/pub/irs-drop/rp-11-42.pdf

See Also

nCont, nLogOdds, nProp, nPropMoe, nWilson

Examples

```
require(PracTools)
set.seed(1289129963)
pop <- HMT(N=10000, H=5)
    # run for full population, 2-sided CI, Neyman allocation
nEdge(ci.lev = 0.95, side = "two.sided", epsilon = 0.005, dat = pop,
      pop.sw = TRUE, wts = NULL, hcol = "strat", ycol = "y", alloc = "neyman",
      Costh = NULL)
nEdge(ci.lev = 0.95, side = "two.sided", epsilon = 0.005,dat = pop,
      pop.sw = TRUE, wts = NULL, hcol = "strat", ycol = "y", alloc = "totcost",
```

```
      Costh = c(5, 10, 15, 20, 25)
)

    # run for a stratified sample, 1-sided lower CI, equal allocation
require(sampling)
sam <- strata(data=pop, stratanames="strat", size=c(30, 40, 50, 60, 70), method=c("srswor"),
              description=TRUE)
samdat <- pop[sam$ID_unit,]
w = 1/sam$Prob
nEdge(ci.lev = 0.99, side = "one.sided.lower", epsilon = 0.01, dat = samdat,
      pop.sw = FALSE, wts = w, hcol = "strat", ycol = "y", alloc = "equal",
      Costh = NULL, nh.lowerbd = 2)

    # simple radom sample size
nEdge(ci.lev = 0.95, side = "two.sided", epsilon = 0.005, dat = pop,
      pop.sw = TRUE, wts = NULL, hcol = NULL, ycol = "y")
```

`nhis` *National Health Interview Survey: Demographic variables*

Description

Demographic variables from a U.S. national household survey

Usage

```
data(nhis)
```

Format

A data frame with 3,911 observations on the following 16 variables.

`ID` Identification variable

`stratum` Sample design stratum

`psu` Primary sampling unit, numbered within each stratum (1,2)

`svywt` survey weight

`sex` Gender (1 = male; 2 = female)

`age` Age, continuous

`age_r` Recoded age (3 = 18-24 years; 4 = 25-44 years; 5 = 45-64 years; 6 = 65-69 years; 7 = 70-74 years; 8 = 75 years and older)

`hisp` Hispanic ethnicity (1 = Hispanic; 2 = Non-Hispanic)

`marital` Marital status (1 = Separated; 2 = Divorced; 3 = Married; 4 = Single/never married; 5 = Widowed; 9 = Unknown marital status)

`parents` Parent(s) of sample person present in the family (1 = Mother, no father; 2 = Father, no mother; 3 = Mother and father; 4 = Neither mother nor father)

`parents_r` Parent(s) of sample person present in the family recode (1 = Yes; 2 = No)

`educ` Education (1 = 8th grade or less; 2 = 9-12th grade, no high school diploma; 3 = High school graduate; 4 = General education development (GED) degree recipient; 5 = Some college, no degree; 6 = Associate's degree, technical or vocational; 7 = Associate's degree, academic program; 8 = Bachelor's degree (BA, BS, AB, BBA); 9 = Master's, professional, or doctoral degree)

`educ_r` Education recode (1 = High school, general education development degree (GED), or less; 2 = Some college; 3 = Bachelor's or associate's degree; 4 = Master's degree & higher)

`race` Race (1 = White; 2 = Black; 3 = Other)

`resp` Respondent (0 = nonrespondent; 1 = respondent)

Details

The National Health Interview Survey (NHIS) is used to monitor health conditions in the U.S. Data are collected through personal household interviews. Only demographic variables are included in this subset which was collected in 2003. The `nhis` data set contains observations for 3,911 persons. The file contains only persons 18 years and older.

Source

National Health Interview Survey of 2003 conducted by the U.S. National Center for Health Statistics. https://www.cdc.gov/nchs/nhis.htm

See Also

`nhis.large`

Examples

```
data(nhis)
str(nhis)
table(nhis$sex,nhis$age_r)
```

---

nhis.large *National Health Interview Survey: Demographic and health variables*

Description

Demographic and health related variables from a U.S. national household survey

Usage

```
data(nhis.large)
```

Format

A data frame with 21,588 observations on the following 18 variables.

`ID` Identification variable

`stratum` Sample design stratum

`psu` Primary sampling unit, numbered within each stratum (1,2)

`svywt` survey weight

`sex` Gender (1 = male; 2 = female)

`age.grp` Age group (1 = < 18 years; 2 = 18-24 years; 3 = 25-44 years; 4 = 45-64 years; 5 = 65+)

`hisp` Hispanic ethnicity (1 = Hispanic; 2 = Non-Hispanic White; 3 = Non-Hispanic Black; 4 = Non-Hispanic All other race groups)

`parents` Parents present in the household (1 = mother, father, or both present; 2 = neither present)

`educ` Highest level of education attained (1 = High school graduate, graduate equivalence degree, or less; 2 = Some college; 3 = Bachelor's or associate's degree; 4 = Master's degree or higher; NA = missing)

`race` Race (1 = White; 2 = Black; 3 = All other race groups)

`inc.grp` Family income group (1 = < $20K; 2 = $20000-$24999; 3 = $25000-$34999; 4 = $35000-$44999; 5 = $45000-$54999; 6 = $55000-$64999; 7 = $65000-$74999; 8 = $75K+; NA = missing)

`delay.med` Delayed medical care in last 12 months because of cost (1 = Yes; 2 = No; NA = missing)

`hosp.stay` Had an overnight hospital stay in last 12 months (1 = Yes; 2 = No; NA = missing)

`doc.visit` During 2 WEEKS before interview, did (person) see a doctor or other health care professional at a doctor's office, a clinic, an emergency room, or some other place? (excluding overnight hospital stay)? (1 = Yes; 2 = No)

`medicaid` Covered by medicaid, a governmental subsidy program for the poor (1 = Yes; 2 = No; NA = missing)

`notcov` Not covered by any type of health insurance (1 = Yes; 2 = No; NA = missing)

`doing.lw` What was person doing last week? (1 = Working for pay at a job or business; 2 = With a job or business but not at work; 3 = Looking for work; 4 = Working, but not for pay, at a job or business; 5 = Not working and not looking for work; NA = missing)

`limited` Is the person limited in any way in any activities because of physical, mental or emotional problems? (1 = Limited in some way; 2 = Not limited in any way; NA = missing)

Details

The National Health Interview Survey (NHIS) is used to monitor health conditions in the U.S. Data are collected through personal household interviews. Demographic variables and a few health related variables are included in this subset. The `nhis.large` data set contains observations on 21,588 persons extracted from the 2003 U.S. NHIS survey. The file contains only persons 18 years and older.

Source

National Health Interview Survey of 2003 conducted by the U.S. National Center for Health Statistics. https://www.cdc.gov/nchs/nhis.htm

See Also

nhis

Examples

```
data(nhis.large)
str(nhis.large)
table(nhis.large$stratum, nhis.large$psu)
table(nhis.large$delay.med, useNA="always")
```

---

nhispart *National Health Interview Survey data from 2003: socioeconomic variables*

Description

Socioeconomic variables from a U.S. national household survey

Usage

```
data(nhispart)
```

Format

A data frame with 3,924 observations on the following variables.

`HHX` Household identification variable

`PX` Person identifier within household

`STRATUM` Sample design stratum

`PSU` Primary sampling unit, numbered within each stratum (1,2)

`WTFA` survey weight

`SEX` Gender (1 = male; 2 = female)

`AGE_P` Age of persons; values are 18-85 (85 includes age 85 and older)

`R_AGE1` Age group (3 = 18-24 years; 4 = 25-44 years; 5 = 45-64 years; 6 = 65-69 years; 7 = 70-74 years; 8 = 75 years and over)

`ORIGIN_I` Hispanic ethnicity (1 = Hispanic; 2 = Non-Hispanic)

`RACERPI2` Race grouped (1 = White only; 2 = Black/African American only; 3 = American Indian or Alaska native (AIAN) only; 4 = Asian only; 5 = Race group not releasable; 6 = Multiple race)

`MRACRPI2` Race detailed (1 = White; 2 = Black/African American; 3 = Indian (American), Alaska Native; 9 = Asian Indian; 10 = Chinese; 11 = Filipino; 15 = Other Asian; 16 = Primary race not releasable; 17 = Multiple race, no primary race selected)

`RACRECI2` White/Black (1 = White; 2 Black; 3 All other race groups)

R_MARITL Marital status (1 = Married - spouse in household; 2 = Married - spouse not in household; 3 = Married - unknown whether spouse in household; 4 = Widowed; 5 = Divorced; 6 = Separated; 7 = Never married; 8 = Living with partner; 9 = Unknown marital status)

CDCMSTAT CDC marital status (1 = Mother, no father; 2 = Father, no mother; 3 = Mother and father; 4 = Neither mother nor father)

INCGRP Total combined family income group (1 = 0-$4999; 2 = $5000-$9999; 3 = $10000-$14999; 4 = $15000-$19999; 5 = $20000-$24999; 6 = $25000-$34999; 7 = $35000-$44999; 8 = $45000-$54999; 9 = $55000-$64999; 10 = $65000-$74999; 11 = $75000 and over; 12 = $20000 or more (no detail); 13 = Less than $20000 (no detail); 97 = Refused; 98 = Not ascertained; 99 = Don't know)

PARENTS Parent(s) present in the family (1 = Mother, no father; 2 = Father, no mother; 3 = Mother and father; 4 = Neither mother nor father)

EDUC_R1 Highest level of education attained (1 = Less than high school graduate; 3 = High school graduate or general education development degree (GED); 5 = Some college, no degree; 6 = AA degree, technical or vocational or AA degree, academic program or Bachelor's degree (BA, BS, AB, BBA); 9 = Master's, professional, or doctoral degree)

RAT_CAT Ratio of family income to poverty level (1 = Under 0.50; 2 = 0.50 to 0.74; 3 = 0.75 to 0.99; 4 = 1.00 to 1.24; 5 = 1.25 to 1.49; 6 = 1.50 to 1.74; 7 = 1.75 to 1.99; 8 = 2.00 to 2.49; 9 = 2.50 to 2.99; 10 = 3.00 to 3.49; 11 = 3.50 to 3.99; 12 = 4.00 to 4.49; 13 = 4.50 to 4.99; 14 = 5.00 and over; 99 = Unknown)

Details

The National Health Interview Survey (NHIS) is used to monitor health conditions in the U.S. Data are collected through personal household interviews. Socioeconomic variables are included in this subset along with household and person codes. The `nhispart` data set contains observations on 3,924 persons extracted from the 2003 U.S. survey. The file contains only persons 18 years and older.

Source

National Health Interview Survey of 2003 conducted by the U.S. National Center for Health Statistics. https://www.cdc.gov/nchs/nhis.htm

Examples

```
data(nhispart)
str(nhispart)
table(nhispart$STRATUM, nhispart$PSU)
table(nhispart$RACERPI2, nhispart$RACRECI2, useNA="always")
```

nLogOdds *Calculate simple random sample size for estimating a proportion*

Description

Calculate the simple random sample size for estimating a proportion using the log-odds transformation.

Usage

```
nLogOdds(moe.sw, e, alpha=0.05, pU, N=Inf)
```

Arguments

moe.sw	switch for setting desired margin of error (1 = CI half-width on the proportion; 2 = CI half-width on a proportion divided by pU)
e	desired margin of error
alpha	1 - (confidence level)
pU	population proportion
N	number of units in finite population

Details

The function accepts five parameters, which are the same ones as accepted by nPropMoe. The desired margin of error can be specified as the CI half-width on the proportion (moe.sw=1) or as the CI half-width as a proportion of the population value pU (moe.sw=2).

Value

numeric sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nProp, nPropMoe, nWilson, nCont

Examples

```
nLogOdds(moe.sw=1, e=0.05, alpha=0.05, pU=0.2, N=Inf)
nLogOdds(moe.sw=2, e=0.05, alpha=0.05, pU=0.2, N=Inf)
```

nPPS *Calculate the sample size for a probability proportional to size (PPS) sample*

Description

Calculate the sample size for a probability proportional to size (PPS) sample, assuming the sample is selected with replacement.

Usage

```
nPPS(X = NULL, Y = NULL, CV0 = NULL, V0 = NULL, N = NULL, V1 = NULL, ybarU = NULL)
```

Arguments

X	variable used for computing 1-draw probabilities; length is N, the population size; must be numeric
Y	variable used for variance calculation; length is N, the population size; must be numeric
CV0	target value of the coefficient of variation of the estimated total of Y
V0	target value of the variance of the estimated total of Y; only one of CV0 and V0 can be specified
N	population size; required if X or Y is NULL
V1	unit variance for PPS calculation
ybarU	population mean of Y (or an estimate of it)

Details

nPPS computes the sample size needed for a probability proportional to size sample or, more generally, a sample selected with varying probabilities, assuming the sample is selected with replacement (WR). Although these samples are rarely selected WR, the variance formula for WR samples is simple and convenient for sample size calculations. Population vectors can be input of X, a measure of size for selecting the sample, and Y, an analysis variable. Alternatively, the population size, N, the unit variance, V1, and the population mean of Y, ybarU can be inputs.

Value

A list with four components:

N	Size of the population
V1	Population variance of Y appropriate for a sample selected with varying probabilities and with replacement; see Valliant, Dever, and Kreuter (2018, sec. 3.4).
ybarU	Population mean of Y
n	Calculated sample size

Author(s)

George Zipf, Richard Valliant

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nCont, nContMoe, nContOpt, unitVar

Examples

```
library(PracTools)
data("smho.N874")
y <- smho.N874[,"EXPTOTAL"]
x <- smho.N874[, "BEDS"]
y <- y[x>0]
x <- x[x>0]
nPPS(X = x, Y = y, CV0 = 0.15)
nPPS(X = x, Y = y,  V0 = 2000000^2)
nPPS(CV0 = 0.15, N = length(y), V1 = (10^21), ybarU = mean(y))
```

---

nProp *Compute simple random sample size for estimating a proportion*

Description

Compute the simple random sample size for estimating a proportion based on different precision requirements.

Usage

```
nProp(CV0 = NULL, V0 = NULL, pU = NULL, N = Inf)
```

Arguments

CV0	target value of coefficient of variation of the estimated proportion
V0	target value of variance of the estimated proportion
pU	population proportion
N	number of units in finite population

Details

The precision requirement of p_s can be set based on either a target coefficient of variation, CV_0, or a target variance, V_0. In either case, a value of p_U must be supplied.

Value

numeric sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nCont, nLogOdds, nPropMoe, nWilson

Examples

```
# srs sample size so that CV of estimated proportion is 0.05
# assuming the population is large and pU=0.01
# Both examples below are equivalent
nProp(V0=0.0005^2, N=Inf, pU=0.01) #or
nProp(CV0=0.05, N=Inf, pU=0.01)

# srswor sample size so that half-width of 2-sided 95% CI is 0.005
nProp(V0=(0.005/1.96)^2, N=Inf, pU=0.01)
```

---

nProp2sam *Simple random sample size for difference in proportions*

Description

Compute a simple random sample size for estimating the difference in proportions when samples overlap

Usage

```
nProp2sam(px, py, pxy, g, r, alt, sig.level=0.05, pow=0.80)
```

Arguments

px	proportion in group 1
py	proportion in group 2
pxy	proportion in the overlap has the characteristic in both samples
g	proportion of sample 1 that is in the overlap with sample 2
r	ratio of the size of sample 1 to that of sample 2

alt	should the test be 1-sided or 2-sided; allowable values are alt="one.sided" or alt="two.sided".
sig.level	significance level of the hypothesis test
pow	desired power of the test

Details

nProp2sam computes sample sizes in two groups that are required for testing whether the difference in group proportions is significant. The power of the test is one of the input parameters. The samples have a specified proportion of units in common.

Value

List with values:

n1	sample size in group 1
n2	sample size in group 2
px.py.pxy	input values of the px, py, pxy parameters
gamma	proportion of sample 1 that is in the overlap with sample 2
r	ratio of the size of sample 1 to that of sample 2
alt	type of test: one-sided or two-sided
sig.level	significance level of test
power	power of the test

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 4). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Woodward, M. (1992). Formulas for Sample Size, Power, and Minimum Detectable Relative Risk in Medical Studies. *The Statistician*, 41, 185-196.

See Also

nDep2sam

Examples

```
nProp2sam(px=0.5, py=0.55, pxy=0.45, g=0.5, r=1, alt="two.sided")
```

nPropMoe *Simple random sample size for a proportion based on margin of error*

Description

Calculates a simple random sample size based on a specified margin of error.

Usage

```
nPropMoe(moe.sw, e, alpha = 0.05, pU, N = Inf)
```

Arguments

moe.sw	switch for setting desired margin of error (1 = CI half-width on the proportion; 2 = CI half-width on a proportion divided by p_U)
e	desired margin of error; either $e = z_{1-\alpha/2}\sqrt{V(p_s)}$ or $e = z_{1-\alpha/2}CV(p_s)$
alpha	1 - (confidence level)
pU	population proportion
N	number of units in finite population

Details

The margin of error can be set as the half-width of a normal approximation confidence interval, $e = z_{1-\alpha/2}\sqrt{V(p_s)}$, or as the half-width of a normal approximation confidence interval divided by the population proportion, $e = z_{1-\alpha/2}CV(p_s)$. The type of margin of error is selected by the parameter moe.sw where moe.sw=1 sets $e = z_{1-\alpha/2}\sqrt{V(p_s)}$ and moe.sw=2 sets i.e., $e = \frac{z_{1-\alpha/2}\sqrt{V(p_s)}}{p_U}$.

Value

numeric sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nCont, nLogOdds, nProp, nWilson

Examples

```
# srs sample size so that half-width of a 95% CI is 0.01
# population is large and population proportion is 0.04
nPropMoe(moe.sw=1, e=0.01, alpha=0.05, pU=0.04, N=Inf)

# srswor sample size for a range of margins of error defined as
# half-width of a 95% CI
nPropMoe(moe.sw=1, e=seq(0.01,0.08,0.01), alpha=0.05, pU=0.5)

# srswor sample size for a range of margins of error defined as
# the proportion that the half-width of a 95% CI is of pU
nPropMoe(moe.sw=2, e=seq(0.05,0.1,0.2), alpha=0.05, pU=0.5)
```

NRadjClass *Class-based nonresponse adjustments*

Description

Compute separate nonresponse adjustments in a set of classes.

Usage

```
NRadjClass(ID, NRclass, resp, preds=NULL, wts=NULL, type)
```

Arguments

ID	identification value for a unit
NRclass	vector of classes to use for nonresponse adjustment. Length is number of respondents plus nonrespondents
resp	indicator for whether unit is a nonrespondent (must be coded 0) or respondent (must be coded 1)
preds	response probabilities, typically estimated from a binary regression model as in pclass
wts	vector of survey weights, typically base weights or base weights adjusted for unknown eligibility
type	type of adjustment computed within each value of NRclass. Allowable codes are 1, 2, 3, 4, or 5. (1 = unweighted average of response propensities, i.e., preds; 2 = weighted average response propensity; 3 = unweighted response rate; 4 = weighted response rate; 5 = median response propensity)

Details

The input vectors should include both respondents and nonrespondents in a sample. A single value between 0 and 1 is computed in each nonresponse adjustment class to be used as a nonresponse adjustment. Five alternatives are available for computing the adjustment based on the value of type. The value of the adjustment is merged with individual unit data and stored in the RR field of the output data frame.

Value

A data frame of respondents only with four columns:

NRcl.no	number of the nonresponse adjustment class for each unit
ID	identification value for a unit
resp	value of the resp variable (always 1)
RR	nonresponse adjustment for each unit

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 13). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

pclass

Examples

```
require(PracTools)
data(nhis)
out <- pclass(formula = resp ~ age + as.factor(sex) + as.factor(hisp) + as.factor(race),
         data = nhis, type = "unwtd", link="logit", numcl=5)
    # unweighted average of response propensities within each class
zz <- NRadjClass(ID=nhis[,"ID"], NRclass = as.numeric(out$p.class), resp=nhis[,"resp"],
           preds=out$propensities, wts=NULL, type=1)
```

NRFUopt *Sample sizes for a nonresponse follow-up study*

Description

Compute optimal values of the first-phase sample size and the second-phase sampling fraction in a two-phase sample.

Usage

```
NRFUopt(Ctot=NULL, c1, c2, theta, CV0=NULL, CVpop=NULL, N=Inf, type.sw)
```

Arguments

Ctot	total variable cost
c1	cost per unit in phase-1
c2	cost per unit in phase-2
theta	probability of response for each unit
CV0	target coefficient of variation for the estimated total or mean
CVpop	Unit coefficient of variation
N	Population size; default is Inf
type.sw	type of allocation; "cost" = target total variable cost, "cv" = target coefficient of variation

Details

NRFUopt computes the optimal values of the first-phase sample size and the second-phase sampling fraction in a two-phase sample. Both stages are assumed to be selected using simple random sampling without replacement. If type.sw="cost", the optima are computed for a target total, expected cost across both phases. If type.sw="cv", the optima are computed for a target coefficient of variation for an estimated mean.

Value

List object with values:

allocation	type of allocation: either "fixed cost" or "fixed CV"
"Total variable cost"	expected total cost: fixed cost if type.sw="cost" or computed cost if type.sw="cv"; unrounded sample sizes are used in calculation
"Response rate"	first-phase response rate
CV	anticipated coefficient of variation (CV) if type.sw="cost" or target CV if type.sw="cv"
v.opt	optimal fraction of first-phase nonrespondents to select for second-phase follow-up
n1.opt	optimal number of units to sample at first-phase
"Expected n2"	expected number of respondents obtained at second-phase
"srs sample for same cv"	size of single-phase simple random sample (*srs*) needed to obtain same CV as the two-phase sample
"Cost Ratio: Two phase to srs"	ratio of expected cost for two-phase sample to cost of single-phase *srs*

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Saerndal, C.E., Swensson, B., and Wretman, J. (1992, examples 15.4.4 and 15.4.5). *Model Assisted Survey Sampling*. New York: Springer.

Valliant, R., Dever, J., Kreuter, F. (2018, chap.17). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Examples

```
    # optima for fixed target CV
NRFUopt(Ctot=NULL, c1=50, c2=200, theta=0.5, CV0=0.05, CVpop=1, type.sw = "cv")
    # optima for fixed total cost
NRFUopt(Ctot=100000, c1=50, c2=200, theta=0.5, CV0=NULL, CVpop=1, type.sw = "cost")
```

---

nWilson *Calculate a simple random sample size for estimating a proportion*

Description

Calculate a simple random sample size for estimating a proportion using the Wilson method.

Usage

```
nWilson(moe.sw, alpha = 0.05, pU, e)
```

Arguments

moe.sw	switch for setting desired margin of error (1 = CI half-width on the proportion; 2 = CI half-width on a proportion divided by pU)
alpha	1 - (confidence level)
pU	population proportion
e	desired margin of error; either the value of CI half-width or the value of the half-width divided by pU

Details

Calculate a simple random sample size using the Wilson (1927) method. A margin of error can be set as the CI half-width on the proportion (moe.sw=1) or as the CI half-width as a proportion of the population value p_U (moe.sw=2).

Value

n.sam	numeric sample size
"CI lower limit"	lower limit of Wilson confidence interval with computed sample size
"CI upper limit"	upper limit of Wilson confidence interval with computed sample size
"length of CI"	length of Wilson confidence interval with computed sample size

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter,F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

Wilson, E.B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22, 209-212.

See Also

nCont, nLogOdds, nProp, nPropMoe

Examples

```
# srs sample size using Wilson method so that half-width of a 95% CI
# is 0.01. Population proportion is 0.04
nWilson(moe.sw = 1, pU = 0.04, e = 0.01)
```

---

pclass *Form nonresponse adjustment classes based on propensity scores*

Description

Fit a binary regression model for response probabilities and divide units into a specified number of classes.

Usage

```
pclass(formula, data, link="logit", numcl=5, type, design=NULL, seed=NULL, rng.preds=0.05)
```

Arguments

formula	symbolic description of the binary regression model to be fitted as used in glm
data	an optional data frame containing columns for a response variable and covariates used in the model to predict response propensities; must be specified if type="unwtd"
link	a specification for the model link function; allowable values are "logit", "probit", or "cloglog"
numcl	number of classes into which units are split based on estimated propensities
type	whether an unweighted or weighted binary regression should be fit; allowable values are "unwtd" or "wtd"
design	sample design object; required if type="wtd"
seed	random number seed
rng.preds	cutoff value for the range of the predicted response probabilities

Details

A typical formula has the form response ~ terms where response is a two-level variable coded as 0 or 1, or is a factor where the first level denotes nonresponse and the second level is response. If type="unwtd", glm is used to fit an unweighted regression. If type="wtd", svyglm in the survey package is used to fit a survey-weighted regression. The jitter function is used to perturb regression predictions slightly. To reproduce the same set of weights in different runs, set seed to some non-null integer. Recommended seeds can be found in positions 3:626 of the built-in global vector .Random.seed.

If the range (defined as max - min) of the predicted response probabilities is less than rng.preds, the predictions are considered to have no meaningful variation and a single response adjustment class is returned. rng.preds is adjustable to accomodate different user preferences.

Value

A list with components:

p.class	propensity class for each unit
propensities	estimated response probability for each unit

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 13). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

NRadjClass

Examples

```
    # classes based on unweighted logistic regression
require(PracTools)
data(nhis)
out <- pclass(formula = resp ~ age + as.factor(sex) + as.factor(hisp) + as.factor(race),
           data = nhis, type = "unwtd", link = "logit", numcl = 5, seed = 1387014860,
           rng.preds = 0.05)
table(out$p.class, useNA="always")
summary(out$propensities)
    # classes based on survey-weighted logistic regression
require(survey)
nhis.dsgn <- svydesign(ids = ~psu, strata = ~stratum, data = nhis, nest = TRUE, weights = ~svywt)
out <- pclass(formula = resp ~ age + as.factor(sex) + as.factor(hisp) + as.factor(race),
           type = "wtd", design = nhis.dsgn, link="logit", numcl=5)
table(out$p.class, useNA="always")
summary(out$propensities)
    # degenerate case where all response propensities are the same
```

```
df <- data.frame(resp = rbinom(500, 1, 0.75), x = rnorm(500))
out1 <- pclass(resp ~ 1, data = df, type = "unwtd", numcl = 5, rng.preds = 0.05)
```

---

quad_roots *Compute the roots of a quadratic equation*

Description

Compute the roots of a quadratic equation

Usage

```
quad_roots(a, b, c)
```

Arguments

a	coefficient of the quadratic term
b	coefficient of the linear term
c	coefficient of the constant term

Details

quad_roots computes the roots of a quadratic equation of the form $ax^2 + bx + c = 0$.

Value

vector with the two roots

Examples

```
quad_roots(1, -8, 12)
```

SampStop *Stopping rule for surveys*

Description

Compute the probability that continuing data collection will lead to a change in the value of an estimated mean.

Usage

```
SampStop(lm.obj, formula, n1.data, yvar, n2.data, p = NULL, delta = NULL, seed = NULL)
```

Arguments

lm.obj	object of class lm from a regression predicting y based on n1.data
formula	righthand side of the formula in lm.obj; it excludes the dependent variable y; no quotes are used.
n1.data	data frame containing units in the part of the sample that has been completed; includes y and the covariates in formula.
yvar	name or number of column in n1.data containing y.
n2.data	data frame containing units in the part of the sample that is yet to be completed; includes only covariates in formula.
p	Vector of anticipated response probabilities for the n2 sample; 0 < p < 1.
delta	vector of potential differences in the estimated means for the n1 and n2 samples.
seed	random number seed for selecting sample from incomplete cases.

Details

SampStop allows an evaluation to be made of whether data collection can be stopped, without substantially affecting the value of an estimated mean, prior to completing collection for all units. Suppose that a sample of size n is divided between the n_1 units whose collection has been completed and the remaining $n_2 = n - n_1$ units that are yet to be completed. The function computes $Pr(|e_1 - e_2| < \delta)$ where $e_1 - e_2$ is the potential difference (delta) between the estimated mean based on the completed sample and the estimated mean for the full sample if all units were to be completed. For e_1 the mean is estimated after imputing the y's for the n_2 incomplete units. The estimated mean e_2 is computed assuming that an additional $n_2 * p$ units are completed, and the y's for the remaining $n_2 - n_2 * p$ incomplete units are imputed. Estimating the variance of $e_1 - e_2$ involves selecting a sample from n2.data using the random number seed in seed.

The parameter p is the response rate that is anticipated for the n_2 uncompleted units. The usual situation is that there is some uncertainty about p which can be accounted for by inputting a vector of p's. δ is a difference in estimates that, if not exceeded, would lead to stopping data collection. For an acceptably small value of delta, if $Pr(|e_1 - e_2| < \delta)$ is large enough, the decision can be made to stop data collection. The variable y in yvar is assumed to follow the linear model in lm.obj. A model with independent errors (or a simple random sample) is assumed for calculations.

Value

A list with two components:

Input	Data frame with columns: No. of current respondents, number of rows in n1.data; No. of current nonrespondents, number of rows in n2.data; Formula, regression model from lm.obj
Output	Data frame with length(p)*length{delta} rows and columns: Pr(response), Probability of response by each of the remaining n_2 cases; Exp no. resps, Expected number of respondents among the remaining n_2 cases, i.e. $n_2 * p$; y1 mean, Mean of the n_1 respondents; diff in means, Value of the input parameter delta; se of diff, Standard error of the difference delta; z-score, Z-score for computing $Pr(\|e_1 - e_2\| < \delta)$; Pr(smaller diff), $Pr(\|e_1 - e_2\| < \delta)$ for the inputs of p and delta

Author(s)

George Zipf, Richard Valliant

References

Wagner, J. and Raghunathan, T. (2010). A new stopping rule for surveys. *Statistics in Medicine*, 29(9), 1014-1024.

Examples

```
library(PracTools)
library(kableExtra)
    # Model with quantitative covariates
HOSP <- PracTools::hospital
HOSP$sqrt.x <- sqrt(HOSP$x)
sam <- sample(nrow(HOSP), 50)
N1.resp <- HOSP[sam, ]
N2.nonresp  <- HOSP[-sam, ]

## Create lm object using "known" data; no intercept model
lm.obj  <- lm(y ~ 0 + sqrt.x + x, data = N1.resp)
del <- mean(HOSP$y) - mean(HOSP$y) * seq(.6, 1, by=0.05)
SampStop(lm.obj  = lm.obj,
                    formula = ~ 0 + sqrt.x + x,
                    n1.data = N1.resp,
                    yvar    = "y",
                    n2.data = N2.nonresp,
                    p       = seq(0.2, 0.6, by=0.05),
                    delta   = del,
                    seed = .Random.seed[413])
    # Model with factors
data(PracTools::labor)
Labor <- PracTools::labor
sam   <- sample(nrow(Labor), 50)
n1.vars <- c("WklyWage", "HoursPerWk", "agecat", "sex")
n2.vars <- c("HoursPerWk", "agecat", "sex")
N1.resp <- Labor[sam, n1.vars]
N2.nonresp <- Labor[-sam, n2.vars]
lm.obj  <- lm(WklyWage ~ HoursPerWk + as.factor(agecat) + as.factor(sex), data = Labor)
del <- mean(N1.resp$WklyWage) - mean(N1.resp$WklyWage) * seq(.75, .95, by=0.05)
S <- SampStop(lm.obj  = lm.obj,
                    formula = ~ HoursPerWk + as.factor(agecat) + as.factor(sex),
                    n1.data = N1.resp,
                    yvar    = "WklyWage",
                    n2.data = N2.nonresp,
                    p       = seq(0.2, 0.4, by=0.05),
                    delta   = del,
                    seed = .Random.seed[78])

kableExtra::kable(S$Input,  caption = "SampStop Input")
kableExtra::kable(head(S$Output, n=15),
      caption = "SampStop Output: First 15 Observations")
```

```
S1 <- as.data.frame(S$Output)
p.nresp <- paste(S1$`Pr(response)`, S1$`Exp no. resps`, sep=", ")
library(ggplot2)
ggplot(S1, aes(x = `diff in means`,
               y = `Pr(smaller diff)`,
               colour = factor(p.nresp))) +
  geom_point() +
  geom_line(linewidth=1.1) +
  labs(x = "delta", y = "Pr(|e1-e2|<= delta)", colour = "Pr(resp), n.resp")
```

---

smho.N874 *Survey of Mental Health Organizations Data*

Description

Data from the 1998 Survey of Mental Health Organizations (SMHO)

Usage

```
data(smho.N874)
```

Format

A data frame with 874 observations on the following 6 variables.

`EXPTOTAL` Total expenditures in 1998

`BEDS` Total inpatient beds

`SEENCNT` Unduplicated client/patient count seen during year

`EOYCNT` End of year count of patients on an institution's roll

`FINDIRCT` Hospital receives money from the state mental health agency (1=Yes; 2=No)

`hosp.type` Hospital type (1 = Psychiatric; 2 = Residential or veterans; 3 = General; 4 = Outpatient, partial care; 5 = Multi-service, substance abuse)

Details

The 1998 SMHO was conducted by the U.S. Substance Abuse and Mental Health Services Administration. It collected data on mental health care organizations and general hospitals that provide mental health care services, with an objective to develop national and state level estimates for total expenditure, full time equivalent staff, bed count, and total caseload by type of organization. The population omits one extreme observation in the `smho98` population and has fewer variables than `smho98`. `smho.N874` contains observations on 874 facilities.

Source

Substance Abuse and Mental Health Services Administration

References

Manderscheid, R.W. and Henderson, M.J. (2002). Mental Health, United States, 2002. DHHS Publication No. SMA04-3938. Rockville MD USA: Substance Abuse and Mental Health Services Administration.

See Also

smho98

Examples

```
data(smho.N874)
str(smho.N874)
```

---

smho98 *Survey of Mental Health Organizations Data*

Description

Data from the 1998 Survey of Mental Health Organizations (SMHO)

Usage

```
data(smho98)
```

Format

A data frame with 875 observations on the following variables.

`STRATUM` Sample design stratum (1 = Psychiatric Hospital, private; 2 = Psychiatric Hospital, public; 3 = Residential, children; 4 = Residential, adults; 5 = General Hospital, public, inpatient or residential care; 6 = General Hospital, public, outpatient care only; 7 = General Hospital, private, inpatient or residential care; 8 = General Hospital, private, outpatient care only; 9 = Military Veterans, inpatient or residential care; 10 = Military Veterans, outpatient care only; 11 = Partial Care 12 = Outpatient care, private; 13 = Outpatient care, public; 14 = Multi-service, private; 15 = Multi-service, public; 16 = Substance Abuse)

`BEDS` Total inpatient beds

`EXPTOTAL` Total expenditures in 1998

`SEENCNT` Unduplicated client/patient count seen during year

`EOYCNT` End of year count of patients on the role

`Y_IP` Number of inpatient visits during year

`OPCSFRST` Number of outpatients on the rolls on the first day of the reporting year

`OPCSADDS` Number of outpatients admitted, readmitted, or transferred to the organization during the reporting year for less than a 24 hour period and not overnight

`OPCSVIST` Number of outpatient visits during the reporting year for less than a 24 hour period and not overnight

`EMGWALK` Number of emergency walk-ins during the reporting year

`PSYREHAB` Number of visits for psychiatric rehabilitation services

`IPCSADDS` Number of residential patients added during the reporting year or patients admitted for more than a 24 hour period

Details

The 1998 SMHO was conducted by the U.S. Substance Abuse and Mental Health Services Administration. It collected data on mental health care organizations and general hospitals that provide mental health care services, with an objective to develop national and state level estimates for total expenditure, full time equivalent staff, bed count, and total caseload by type of organization.

Source

Substance Abuse and Mental Health Services Administration

References

Manderscheid, R.W. and Henderson, M.J. (2002). Mental Health, United States, 2002. DHHS Publication No. SMA04-3938. Rockville MD USA: Substance Abuse and Mental Health Services Administration.

See Also

`smho.N874`

Examples

```
data(smho98)
str(smho98)
summary(smho98)
```

---

strAlloc *Allocate a sample to strata*

Description

Compute the proportional, Neyman, cost-constrained, and variance-constrained allocations in a stratified simple random sample.

Usage

```
strAlloc(n.tot = NULL, Nh = NULL, Sh = NULL, cost = NULL, ch = NULL,
     V0 = NULL, CV0 = NULL, ybarU = NULL, alloc)
```

Arguments

n.tot	fixed total sample size
Nh	vector of population stratum sizes (N_h) or pop stratum proportions (W_h)
Sh	stratum unit standard deviations (S_h), required unless alloc = "prop"
cost	total variable cost
ch	vector of costs per unit in stratum h (c_h)
V0	fixed variance target for estimated mean
CV0	fixed CV target for estimated mean
ybarU	population mean of y ($\bar{y}_U$)
alloc	type of allocation; must be one of "prop", "neyman", "totcost", "totvar"

Details

alloc="prop" computes the proportional allocation of a fixed total sample size, n.tot, to the strata. alloc="neyman" computes the allocation of a fixed total sample size, n.tot, to the strata that minimizes the variance of an estimated mean. alloc="totcost" computes the allocation of a fixed total sample size, n.tot, to the strata that minimizes the variance of an estimated mean subject to the fixed total cost. alloc="totvar" computes the allocation that minimizes total cost subject to the target coefficient of variation, CV0, or the target variance, V0, of the estimated mean.

Value

For proportional allocation, a list with values:

alloc	type of allocation: "prop", "neyman", "totcost", "totvar"
Nh	vector of population sizes (N_h) or pop stratum proportions (W_h)
nh	vector of stratum sample sizes
"nh/n"	proportion of sample allocated to each stratum

For other allocations, the three components above plus:

Sh	stratum unit standard deviations (S_h)
"anticipated SE of estimated mean"	Anticipated SE of the estimated mean for the computed allocation

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Cochran, W.G. (1977). *Sampling Techniques*. John Wiley & Sons.

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. Springer.

See Also

nCont, nLogOdds, nProp, nPropMoe, nWilson

Examples

```
# Neyman allocation
Nh <- c(215, 65, 252, 50, 149, 144)
Sh <- c(26787207, 10645109, 6909676, 11085034, 9817762, 44553355)
strAlloc(n.tot = 100, Nh = Nh, Sh = Sh, alloc = "neyman")

# cost constrained allocation
ch <- c(1400, 200, 300, 600, 450, 1000)
strAlloc(Nh = Nh, Sh = Sh, cost = 100000, ch = ch, alloc = "totcost")

# allocation with CV target of 0.05
strAlloc(Nh = Nh, Sh = Sh, CV0 = 0.05, ch = ch, ybarU = 11664181, alloc = "totvar")
```

Test_Data_US *Accounting data for some US cities with latitude and longitude of the city centroids*

Description

A list of US cities with their latitude and longitude centroids and other data

Usage

```
data(Test_Data_US)
```

Format

A data frame with 381 cities with the following variables:

`ID` Sequential ID field

`State` State name

`City` City name

`Count` Number of records in city

`Amount` Total dollar amount of records

`lat` latitude of the city center

`long` longitude of the city center

`Y` Artificial analysis variable

Details

This population has 381 US cities with the latitude and longitude of the city center. It is used to illustrate the use of the `GeoDistPSU` and `GeoDistMOS` functions.

See Also

GeoDistPSU,GeoDistMOS

Examples

```
data(Test_Data_US)
str(Test_Data_US)
```

ThirdGrade *Third grade population*

Description

The ThirdGrade data file is a population of students who participated in the Third International Mathematics and Science Study (TIMSS).

Usage

```
data(ThirdGrade)
```

Format

A data frame with 2,427 students on the following variables:

`region` Geographic region of the U.S. (1 = Northeast; 2 = South; 3 = Central; 4 = West)

`school.id` School identifier (1 - 135)

`student.id` Student identifier (1 - 2427)

`sex` Sex of student (1 = female; 2 = male)

`language` Is language of test spoken at home? (1 = always; 2 = sometimes; 3 = never)

`math` Mathematics test score

`ethnicity` Ethnicity of student (1 = White, non-Hispanic; 2 = Black; 3 = Hispanic; 4 = Asian; 5 = Native American; 6 = Other)

`science` Science test score

`community` Type of location of school (2 = village or rural area; 3 = outskirts of a town or city; 4 = close to center of a town or city)

`enrollment` Number of students in entire school

Details

The Third Grade population consists of 2,427 students in the U.S. who participated in the Third International Mathematics and Science Study (Caslyn, Gonzales, Frase 1999). The methods used in conducting the original study are given in TIMSS International Study Center (1996). Clusters are schools while units within clusters are the students.

Source

TIMSS International Study Center 1996.

References

Caslyn, C., Gonzales, P., Frase, M. (1999). *Highlights from TIMSS*. National Center for Education Statistics, Washington DC.

TIMSS International Study Center (1996). *Third International Mathematics and Science Study: Technical Report, Volume 1 Design and Development*. Boston College: Chestnut Hill MA.

Examples

```
data(ThirdGrade)
str(ThirdGrade)
```

TPV *TPV Data*

Description

TPV is an example data file for illustrating the use of certainty (take-all) units in sampling

Usage

```
data(TPV)
```

Format

A data frame with 67 observations on the following 2 variables:

`Total.Pot.Value` a measure of size for each unit; for example, maximum potential amount spent on a contract, i.e. base price plus all options.

`Y` an analytic variable for each unit

Details

The TPV data are used as an example for `nContOpt` which determines the optimal split of a sample between take-all and non-take-all units.

See Also

nContOpt

Examples

```
data(TPV)
str(TPV)
```

unitVar *Compute the unit (population) variance for a variable*

Description

Compute the unit (population) variance for a variable based on either a full population file or a sample from a finite population.

Usage

```
unitVar(pop.sw = NULL, w = NULL, p = NULL, y = NULL)
```

Arguments

pop.sw	TRUE if the full population is input; FALSE if a sample is input
w	vector of sample weights if y is a sample; used only if pop.sw = FALSE
p	vector of 1-draw selection probabilities; optionally provided if pop.sw = TRUE
y	vector of values of an analysis variable; must be numeric

Details

unitVar computes unit (population) variances of an analysis variable y from either a population or a sample. S2 is the unweighted population variance, $S^2 = \sum_{i \in U}(y_i - \bar{y}_U)^2/(N-1)$ where U is the universe of elements, N is the population size, and $\bar{y}_U$ is the population mean. If the input is a sample, S2 is estimated as $\hat{S}^2 = (n/(n-1))\sum_{i \in s} w_i(y_i - \bar{y}_w)^2/(\sum_{i \in s} w_i)$ where s is the set of sample elements, n is the sample size, and $\bar{y}_w$ is the weighted sample mean.

V1 is a weighted population variance used in calculations for samples where elements are selected with varying probabilities. If the y is a population vector, $V_1 = \sum_U p_i(y_i/p_i - t_U)^2$ where p_i is the 1-draw probability for element i and t_U is the population total of y. If y is for a sample, $\hat{V}_1 = \sum_s (y_i/p_i - n^{-1}\sum_k y_k/p_k)^2/(n-1)$ with p_i computed as $1/(nw_i)$.

Value

A list with three or four components:

Note	Describes whether output was computed from a full population or estimated from a sample.
Pop size N	Size of the population; included if y is for the full population.
S2	Unit variance of y; if pop.sw = TRUE, S2 is computed from the full population; if pop.sw = FALSE, S2 is estimated from the sample using the w weights.
V1	Population variance of y appropriate for a sample selected with varying probabilities; see Valliant, Dever, and Kreuter (VDK; 2018, sec. 3.4). If pop.sw = TRUE and p is provided, V1 is computed with equation (3.32) in VDK. If pop.sw = FALSE, V1 is estimated with equation (3.41) in VDK.

Author(s)

Richard Valliant

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 3). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

nCont, nContMoe, nContOpt, nPPS

Examples

```
library(PracTools)
data("smho.N874")
y <- smho.N874[,"EXPTOTAL"]
x <- smho.N874[, "BEDS"]
y <- y[x>0]
x <- x[x>0]
pik <- x/sum(x)
require(sampling)
n <- 50
sam <- UPrandomsystematic(n * pik)
wts <- 1/(n*pik[sam==1])
unitVar(pop.sw = TRUE, w = NULL, p = pik, y=y)
unitVar(pop.sw = FALSE, w = wts, p = NULL, y=y[sam==1])
```

---

wtd.moments *Compute moments of a variable from either a population or sample*

Description

Compute the 2nd, 3rd, 4th moments, skewness, and kurtosis of a variable from either population or sample input

Usage

```
wtd.moments(y, w=NULL, pop.sw=TRUE)
```

Arguments

y	variable to be analyzed
w	vector of weights if the input is a sample
pop.sw	is the input for a population (pop.sw=TRUE) or for a sample (pop.sw=FALSE)?

Details

The r^{th} population moment is defined as $m_r = (1/N)\sum_{k \in U}(y_k - \bar{y}_U)^r$ where U is the set of population units, N is the population size, and $\bar{y}_U$ is the population mean. When the input is for the whole population, `wtd.moments` evaluates this directly for $r = 2, 3, 4$. When the input is for a sample, the r^{th} moment is estimated as $\hat{m}_r = (K/\hat{N})\sum_{k \in s}(w_k(y_k - \hat{\bar{y}}_U)^r), r = 2, 3, 4$ where s is the set of sample units, w_k is the weight for sample unit k, $\hat{N} = \sum_s w_k$, and $\hat{\bar{y}}_U = \sum_{k \in s} w_k y_k / \hat{N}$. When $r = 2$, $K = n/(n-1)$ so that the estimator equals the unbiased variance estimator if the sample is a simple random sample; if $r = 3, 4$, then $K = 1$. The function also computes or estimates the population skewness, defined as $m_3/m_2^{3/2}$ and the population kurtosis, m_4/m_2^2.

The weights should be scaled for estimating population totals. The sample can be obtained from any complex design. The estimators produced by `wtd.moments` are consistent for general sample designs but not unbiased (except for $r = 2$ and SRS).

Value

Vector with values:

`m2`	2nd moment
`m3`	3rd moment
`m4`	4th moment
`skewness`	skewness
`kurtosis`	kurtosis

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, sect. 3.4). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. New York: Springer.

See Also

`wtdvar`

Examples

```
require(PracTools)
wtd.moments(y = hospital$y, w = NULL)
require(sampling)
sam <- strata(data = labor, stratanames = "h", size = c(30, 20, 10), method = c("srswor"),
              description=TRUE)
samdat <- labor[sam$ID_unit,]
wtd.moments(y = samdat$WklyWage, w = 1/sam$Prob, pop.sw=FALSE)
```

wtdvar *Compute weighted variance*

Description

Compute an estimate of a population unit variance from a complex sample with survey weights.

Usage

```
wtdvar(x,w,na.rm=TRUE)
```

Arguments

x	data vector
w	vector of survey weights; must be same length as x
na.rm	remove missing values (TRUE or FALSE)

Details

wtdvar is also used by BW3stagePPSe in estimating relvariance components.

Value

numeric estimate of population unit variance

Author(s)

Richard Valliant, Jill A. Dever, Frauke Kreuter

References

Valliant, R., Dever, J., Kreuter, F. (2018, chap. 9). *Practical Tools for Designing and Weighting Survey Samples, 2nd edition*. Springer.

Examples

```
x <- c(1:3)
wts <- c(4, 6, 8)
wtdvar(x=x, w=wts)
```

Index

∗ **datasets**
Domainy1y2, 35
hospital, 46
labor, 47
MDarea.popA, 48
mibrfss, 49
nhis, 65
nhis.large, 66
nhispart, 68
smho.N874, 85
smho98, 86
Test_Data_US, 89
ThirdGrade, 90
TPV, 91
∗ **methods**
BW2stagePPS, 3
BW2stagePPSe, 5
BW2stageSRS, 7
BW3stagePPS, 9
BW3stagePPSe, 11
clusOpt2, 14
clusOpt2fixedPSU, 16
clusOpt3, 17
clusOpt3fixedPSU, 19
CompMOS, 21
CVcalc2, 23
CVcalc3, 24
deff, 26
deffCR, 29
deffH, 31
deffK, 33
deffS, 34
dub, 36
GeoDistMOS, 39
GeoDistPSU, 41
GeoMinMOS, 43
HMT, 45
nAuditAttr, 51
nAuditMUS, 53
nCont, 55
nContMoe, 56
nContOpt, 58
nDep2sam, 59
nDomain, 61
nEdge, 62
nLogOdds, 70
nPPS, 71
nProp, 72
nProp2sam, 73
nPropMoe, 75
NRadjClass, 76
NRFUopt, 77
nWilson, 79
pclass, 80
quad_roots, 82
SampStop, 82
strAlloc, 87
unitVar, 92
wtd.moments, 93
wtdvar, 95
∗ **models**
gamEst, 37
gammaFit, 38
∗ **survey**
BW2stagePPS, 3
BW2stagePPSe, 5
BW2stageSRS, 7
BW3stagePPS, 9
BW3stagePPSe, 11
clusOpt2, 14
clusOpt2fixedPSU, 16
clusOpt3, 17
clusOpt3fixedPSU, 19
CompMOS, 21
CVcalc2, 23
CVcalc3, 24
deff, 26
deffCR, 29

deffH, 31
deffK, 33
deffS, 34
dub, 36
GeoDistMOS, 39
GeoDistPSU, 41
GeoMinMOS, 43
HMT, 45
nAuditAttr, 51
nAuditMUS, 53
nCont, 55
nContMoe, 56
nContOpt, 58
nDep2sam, 59
nDomain, 61
nEdge, 62
nLogOdds, 70
nPPS, 71
nProp, 72
nProp2sam, 73
nPropMoe, 75
NRadjClass, 76
NRFUopt, 77
nWilson, 79
pclass, 80
quad_roots, 82
SampStop, 82
strAlloc, 87
unitVar, 92
wtd.moments, 93
wtdvar, 95

BW2stagePPS, 3, *6*, *8*, *10*, *12*
BW2stagePPSe, *4*, 5, *8*, *10*, *12*
BW2stageSRS, *4*, *6*, 7, *10*, *12*
BW3stagePPS, *4*, *6*, *8*, 9, *12*
BW3stagePPSe, *4*, *6*, *8*, *10*, 11, *95*

clusOpt2, 14, *17, 18*, *20*
clusOpt2fixedPSU, *15*, 16, *18*, *20*
clusOpt3, *15*, *17*, 17, *20*
clusOpt3fixedPSU, *15*, *17, 18*, 19
CompMOS, 21
CVcalc2, 23
CVcalc3, *24*, 24, *25*

deff, 26, *30*, *32–34*
deffCR, *27*, 29, *32–34*
deffH, *27*, *30*, 31, *33, 34*
deffK, *27*, *30*, *32*, 33, *34*
deffS, *27*, *30*, *32, 33*, 34
Domainy1y2, 35
dub, 36

gamEst, 37, *39*
gammaFit, *37, 38*, 38
GeoDistMOS, 39, *42*, *44*, *90*
GeoDistPSU, *40*, 41, *44*, *90*
GeoMinMOS, *40*, *42*, 43

HMT, 45
hospital, 46

labor, 47

MDarea.popA, 48
mibrfss, 49

nAuditAttr, 51, *54*
nAuditMUS, 53
nCont, 55, *57*, *59*, *62*, *64*, *70*, *72, 73*, *75*, *80*, *89*, *93*
nContMoe, 56, *59*, *72*, *93*
nContOpt, 58, *72*, *91*, *93*
nDep2sam, 59, *74*
nDomain, 61
nEdge, 62
nhis, *51*, 65, *68*
nhis.large, *51*, *66*, 66
nhispart, 68
nLogOdds, *56, 57*, *62*, *64*, 70, *73*, *75*, *80*, *89*
nPPS, 71, *93*
nProp, *56, 57*, *62*, *64*, *70*, 72, *75*, *80*, *89*
nProp2sam, *61*, 73
nPropMoe, *56, 57*, *62*, *64*, *70*, *73*, 75, *80*, *89*
NRadjClass, 76, *81*
NRFUopt, 77
nWilson, *56, 57*, *62*, *64*, *70*, *73*, *75*, 79, *89*

pclass, *77*, 80

quad_roots, 82

SampStop, 82
smho.N874, 85, *87*
smho98, *86*, 86
strAlloc, 87

Test_Data_US, 89

ThirdGrade, 90
TPV, 91

unitVar, *72*, 92

wtd.moments, 93
wtdvar, *94*, 95