auditsampling

August 22, 2019

Title A auxiliary package to calculate some basic things in sampling. Most used para CGU auditors.

Version 0.0.0.9000

Description The package contains some functions the help auditors in sampling process.

Depends R (>= 3.4.3)

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1.9000

R topics documented:

`inference` *Outputs a table with SRS and Stratified based inferences*

Description

This function do Simple Random Sample and Stratified inferences from a sample data, given an alpha level. It can calculate estimates for means, totals, proportions and ratios too. It have option to print a Word *.docx table with the estimate parameters and two plotting graph types.

Usage

```
inference(sample.data, num.y = NULL, denom.x = NULL, total.x = NULL,
  cat.y = NULL, strat = NULL, post.strat = NULL, alpha, N = Inf,
  type = rep("mean", length(num.y) + length(cat.y)), fpc = NULL,
  labels = NULL, print.report = FALSE)
```

Arguments

`sample.data`	A data.frame representing a sample with the collected data.
`num.y`	The index number of the columns (vector) where the data is continuous.
`total.x`	If you are using ratio estimators and want to estimate totals, you gonna need a total of $\bar{X}$. Use vectors of the same length as your estimated variables for estimate totals. If you don't provide totals, they will be estimated from data.
`cat.y`	The index number of the columns where the data is categorical.
`strat`	The index number of the columns to be used as stratified design.
`post.strat`	The index number of the columns to used as a post stratified design.
`alpha`	1 - Confidence level.
`N`	Population size.
`type`	A vector of characters. Allowed means, totals or ratios. The function use `mean` as default.
`fpc`	The index number of the column to be used as the factor of population correction.
`labels`	An optional string vector of labels to be putted into printed table.
`print.report`	`FALSE` by default. If `TRUE`, prints a .docx table. Must have the same length of the sum of numerical columns and all levels of categorical columns.
`denom.y`	The index number of the columns (vector) that will be used as denominator for ratio estimators. Use NULL if you don't want to use ratio estimators or combine regular estimators with ratio ones.

Value

A data.frame containing the following:

parameter The label of the parameter.

n The effective sample size.

se Standard error.

point.estimate Point estimate.

interval Confidence interval.

neymanAllocation — *Returns the number of elements per strata based on Neyman optimum allocation*

Description

Based on data from sample and population, this function returns the optimum allocation of elements. Users can input strata vectors from sample and from population, but the last is not required. The output is a vector.

Usage

```
neymanAllocation(n, strat.vector = NULL, sample.vector, sd.vector)
```

Arguments

n	The size of sample.
strat.vector	The vector of strata from population data.
sample.vector	The vector of strata from sample data. Required.
sd.vector	The data sample vector from which the standards deviations will be calculated.

Value

Vector with the same length as the unique labels of strata containing the size of each strata for sampling.

nPilot — *Calculates the remaing sample size when using pilot sample*

Description

Based on a pilot sample, this function calculates the remain elements to the final sample given the variance considered.

Usage

```
nPilot(s2 = NULL, pq = NULL, n1, V)
```

Arguments

pq	The variance, as $P(1-P)$. Not necessary if s2 is not NULL.
n1	The size of the pilot sample.
V	The target variance, as $(d/t)^2$, where *d* is the margin of error, and *t* is the z-value.
s2	The variance, as σ^2. Not necessary if pq is not NULL.

Value

The final sample is

$$n = n_1 + n_2$$

Where n_2 is the value returned by this function.

References

COCHRAN, William Gemmell. Sampling techniques-3. 1977.

Examples

```
nPilot(s2=13.5,n1=30,V=(5/1.96)^2)
nPilot(pq=0.15,n1=30,V=(0.05/1.65)^2)
```

`nPlan` *Returns a data.frame with total resources to be spend in sampling process, based on the thresholds defined*

Description

This function returns a data.frame with the calculations of time to be spend on sampling process, listing respectives sampling parameters, like confidence levels and margin of errors. Need a sample size function to work.

Usage

```
nPlan(hh, max.hh = NULL, max.n = NULL, max.alpha = NULL,
  max.beta = NULL, fun = "nSRS", ...)
```

Arguments

`hh`	The cost, measured in time (HH), for collecting data for each sample unit.
`max.hh`	The cost threshold from which results and their parameters will be excluded.
`max.n`	The sample size threshold from which results and their parameters will be excluded.
`max.alpha`	The maximum z-value of α from which results and their parameters will be excluded.
`max.beta`	The maximum z-value of β from which results and their parameters will be excluded. Applicable only to `nPowerSample` function.
`fun`	Function used to calculate samplesize. Can be `nSRS`, `nStrata` or `nSampleSize`.
`...`	Others parameters passed to be used with the sample size functions. See examples.

Value

A data.frame with

alpha The α.

moe The margin of error, when using `nSRS` or `nStrata`.

beta The α, when using `nSampleSize`.

n Respective sample size.

hh Cost, in time, as the argument passed to the function.

hh_total Total cost, in time.

Examples

```
################### nSRS
nPlan(hh = (10/60), fun = "nSRS", max.alpha = 0.1, max.n = 176, max.hh = 60,
      moe = seq(0.01,0.1,by=0.01), pq = 0.25, N = 16548)

################### nPowerSample
nPlan(hh = (5/60), fun = "nPowerSample", max.hh = 40, max.alpha = 0.1, max.beta = 0.2,
      p=0.02, p0=0.05, alternative = 1)
```

`nPowerSample` *Calculates the sample size for hypotesis tests*

Description

This function calculates de sample size for hypotesys tests considering the following hypotesis:

$$H_0 : \mu \neq \mu_0$$

$$H_0 : \bar{p} \neq p_0$$

$$H_0 : \mu \leq or \geq \mu_0$$

$$H_0 : \bar{p} \leq or \geq p_0$$

Usage

```
nPowerSample(p = NULL, p0 = NULL, mu = NULL, mu0 = NULL,
  sd = NULL, alpha = 0.05, beta = 0.2, alternative = 1)
```

Arguments

`p`	Population proportion.
`p0`	Hypothesized proportion.
`mu`	Population mean.
`mu0`	Hypothesized mean.
`alpha`	Level of significance.
`alternative`	1 if is a greater/lesser test and 2 for equal/different one.

Value

The sample size.

Examples

```
nPowerSample(p=0.03, p0=0.05)
nPowerSample(p=0.02, p0=0.05)
nPowerSample(mu=100, mu0=120,sd=24)
nPowerSample(mu=100, mu0=110,sd=24,beta=0.05,alternative = 2)
```

---

nSRS *Calculates the sample size for SRS*

Description

This function calculates de sample size in a SRS design based on one of three possible arguments: σ^2, pq or CV. Provide just one of then.

Usage

```
nSRS(moe, alpha, CV = NULL, S2 = NULL, pq = NULL, N = Inf)
```

Arguments

moe	Margin of error.
alpha	1 - (confidence level).
CV	Variaton coefficient.
S2	σ^2, population variance.
pq	$P(1-P)$, population variance for proportions.
N	Population size.

Value

The value of n.

Examples

```
nSRS(moe=0.05,alpha=0.05,S2=6,N=1500)
nSRS(moe=0.05,alpha=0.05,CV=0.2,N=1500)
nSRS(moe=0.05,alpha=0.05,pq=0.25,N=1500)
```

nStrata *Calculates the sample size in a stratified desgin*

Description

This function returns the number of elements to be sampled in a stratified based survey. If the method choosen is 'prop' or 'optimum', the function return the allocation too.

Usage

```
nStrata(data, stratanames, alpha, moe, S2 = NULL, pq = NULL,
  V = NULL, N, method = c("prop", "optimum"))
```

Arguments

data	The data.frame containing the variable denoted as the cluster.
stratanames	The variable name of the considered strata. Can be a vector of variable names.
alpha	(1 - confidence level).
moe	Margin of error.
S2	The variance, as σ^2. Not necessary if pq or V is not NULL.
pq	The variance, as $P(1-P)$. Not necessary if S2 or V is not NULL.
V	The variance, as $(d/t)^2$. Not necessary if S2 or pq is not NULL.
N	Population size.
method	A string with prop or optimum. With optimum, it uses Neyman allocation.

Value

The value (integer) of the number of elements to sample in stratified sampling. If prop or optimum where used, the function returns a list with the elements per strata.

n Total number of elements in the sample.

nh The allocation in each strata.

Examples

```
library(survey)
data("api")
nStrata(apipop,stratanames = "stype", alpha = 0.05, moe = 0.05,
        pq = 0.25, N = nrow(apipop), method = "prop")
nStrata(apipop,stratanames = "stype", alpha = 0.05, moe = 0.05,
        pq = 0.25, N = nrow(apipop), method = "optimum")
### Cochran's example (Cochran, W. G. (2007). Sampling techniques. John Wiley & Sons. p. 106-107)
data_cochran=data.frame(stratum=rep(c(1:6),c(13,18,26,42,73,24)))
nStrata(data_cochran,stratanames = "stratum", N=196, method = "optimum",
        S2=c(325^2,190^2,189^2,82^2,86^2,190^2),V=7974976/196^2)
```

NSubpop — *Calculates the new subpopulation due elements that not belong to the populational universe*

Description

This function returns the number of the new subpopulation.

Usage

```
NSubpop(N, n, p, z = 3)
```

Arguments

N	Population size.
n	Sample size.
p	Proportion or the number of elements that belong to the universe intended.
z	The z-value of α.

Value

The value (integer) of the new population (subpopulation) number.

Examples

```
NSubpop(45000,100,59)
NSubpop(45000,100,5)
```

selectSample — *Select sample and returns the* data.frame

Description

Sample and outputs a data.frame with desired technique and size.

Usage

```
selectSample(data, stratum = NULL, method, size, seed)
```

Arguments

data	The data.frame containing the population data.
method	Select between srswr, srswor or systematic.
size	The size of sample. Must provide a vector if the sample is stratified.
seed	The seed for the random generator.
stratanames	A vector with strata variable names.

Value

A data.frame additionally with Prob and fpc columns. If the systematic method of selection has been chosen, returns the k and cumulated k also.

siasgdw *Data from brazillian government bid processes*

Description

Data from Sistema Integrado de Administracao de Servicos Gerais (SIASG) from goods and services purchases by the Brazilian government

A dataset containing all bids made in 2018 It has some simulated data, to be used for learning purposes.

Usage

```
data(siasgdw)
```

Format

A dataset of bids made by Brazillian government offices, totalizing 164372 observations, on the following variables

id_compra ID of the bidding process

no_processo Number of the processo protocol

dt_compra Reference date of the purchase

id_unidde ID of the department of a public agency responsible for the purchase

no_unidade Name of the department a public agency responsible for the purchase

orgao Name of the public agency to which the one responsible for the purchase is subordinate

orgao_sup Name of the supervisor public agency of the respective responsible for the purchase

modalidae The category of juridic rules and procedures involved in the bidding process

valor_compra The purchase amount involved in the bidding process

qtd_forn The number of bidders involved in the bidding process

qtd_itens The number of different goods and/or services demanded in the bidding process

controle_01 / 12 Internal controls failure occurrences (binary)

risco_01 / 05 Risk occurrences (binary)

risco_06 / 08 Risk occurrences (financial losses)

tempo_total Total time spent, in days, for the conclusion of the bidding process

tempo_fase_interna Total time spent, in days, for the planning of the bidding process

Source

Sistema Integrado de Administração de Serviços Gerais(SIASG)

Examples

```
library(auditsampling)
data('siasgdw')
############# Subscripting
uasg_cgu <- siasgdw[siasgdw$id_unidade == 370003,]
summary(uasg_cgu)
```

`test` *Calculates the statistics for hypotesis tests*

Description

This function calculates de statistics for hypotesys tests considering the following hypotesis:

$$H_0 : \mu \neq \mu_0$$

$$H_0 : \bar{p} \neq p_0$$

$$H_0 : \mu \leq or \geq \mu_0$$

$$H_0 : \bar{p} \leq or \geq p_0$$

Usage

```
test(sample.data, num.col = NULL, prop.col = NULL, mu0 = NULL,
  p0 = NULL, alpha, alternative = NULL, labels = NULL,
  plot.graph = FALSE)
```

Arguments

`sample.data`	The source `data.frame` from which the funtion will search the columns.
`num.col`	Vector of the numeric column numbers.
`prop.col`	Vector of the binary column numbers.
`mu0`	Hypothesized mean.
`p0`	Hypothesized proportion.
`alpha`	Level of significance.
`alternative`	`1` if is a greater/lesser test and `2` for equal/different one.
`labels`	Label vector for the outputs.
`plot.graph`	Plot a confidence intervals for proportions only. Not implemented yet.

Value

`data.frame` containing the following:

var The label of the variable/parameter.

point Point estimate.

tolerance The hypothesized mean/proportion.

sdd_err Standard error.

z_value Statistics z.

pval P-value.

power Test power, defined as $(1 - \beta)$.

Examples

```
n=168
set.seed(101)
dados <- data.frame(CONTROLE1=rbinom(n,1,prob = 0.4),
                    CONTROLE2=rbinom(n,1,prob = 0.02),
                    CONTROLE3=rbinom(n,1,prob = 0.1),
                    CONTROLE4=rbinom(n,1,prob = 0.01),
                    CONTROLE5=rbinom(n,1,prob = 0.2),
                    CONTROLE6=rnorm(n,10,3),
                    CONTROLE7=rnorm(n,250,20))
test(sample.data = dados, num.col = 6:7, prop.col = 1:5, alpha = 0.05,
     mu0 = c(10,100),p0=c(0.05,0.05,0.10,0.10,0.1),
     alternative = c(rep(1,5),rep(2,2)))
```

Index