Package ‘sampler’

October 14, 2022

Type Package
Title Sample Design, Drawing & Data Analysis Using Data Frames
Version 0.2.4
Author Michael Baldassaro
Maintainer Michael Baldassaro <mbaldassaro@gmail.com>
Description Determine sample sizes, draw samples, and conduct data analysis using data frames. It specifically enables you to determine simple random sample sizes, stratified sample sizes, and complex stratified sample sizes using a secondary variable such as population; draw simple random samples and stratified random samples from sampling data frames; determine which observations are missing from a random sample, missing by strata, duplicated within a dataset; and perform data analysis, including proportions, margins of error and upper and lower bounds for simple, stratified and cluster sample designs.
License MIT + file LICENSE
URL https://github.com/mbaldassaro/sampler
BugReports https://github.com/mbaldassaro/sampler/issues
Encoding UTF-8
LazyData true
Imports dplyr, tidyr, reshape, purrr
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2019-09-15 15:40:02 UTC

R topics documented:

albania ................................................................. 2
cpro ................................................................. 4
dedupe .............................................................. 5
dupe ................................................................. 5
opening ............................................................. 6
Description

Data set containing 2017 Albania election results by polling station published by the Central Election Commission and opened by the Coalition of Domestic Observers & Democracy International.

Usage

albania

Format

A data frame with 5362 rows and 45 variables

qarku district, 12 in total
Q_ID geocode for district
bashkia municipality, 61 in total
BAS_ID geocode for municipality
zaz election area zone, 90 in total
njesiaAdministrative village, 373 in total
COM_ID geocode for village
qvKod polling station identifier
zgjedhes number of total registered voters
meshku number of male registered voters
femra number of female registered voters
totalSeats number of seats contested by district
vendndodhja name of polling center containing polling stations
ambienti type of polling center, 5 in total
totalVoters number of total registered voters that cast ballots
femVoters number of female registered voters that cast ballots
maleVoters number of male registered voters that cast ballots
unusedBallots number of ballots not used
damagedBallots number of ballots damaged
ballotsCast number of total ballots cast
invalidVotes number of ballots cast that were invalided
validVotes number of valid ballots cast
lsi number of ballots cast for LSI
ps number of ballots cast for PS
pkd number of ballots cast for PKD
sfida number of ballots cast for SFIDA
pr number of ballots cast for PR
pd number of ballots cast for PD
pbdksh number of ballots cast for PBDKSH
adk number of ballots cast for ADK
psd number of ballots cast for PSD
ad number of ballots cast for AD
frd number of ballots cast for FRD
pds number of ballots cast for PDS
pdiu number of ballots cast for PDIU
aak number of ballots cast for AAK
mega number of ballots cast for MEGA
pksh number of ballots cast for PKSH
apd number of ballots cast for APD
libra number of ballots cast for LIBRA
psSeats number of seats won by PS
pdSeats number of seats won by PD
lsiSeats number of seats won by LSI
pdiuSeats number of seats won by PDIU
psdSeats number of seats won by PSD

Source
https://albaniaelectiondata.herokuapp.com/
cpro  

**Calculate proportion and margin of error (unequal-sized cluster sample)**

---

**Description**

Calculate proportion and margin of error (unequal-sized cluster sample)

**Usage**

```r
cpro(df, numerator, denominator, ci = 95, na = "", N = 0)
```

**Arguments**

- `df`  
  object containing data frame on which to perform analysis
- `numerator`  
  variable in data frame for which you want to calculate proportion and margin of error
- `denominator`  
  variable in data frame containing population sizes of unequal clusters
- `ci`  
  (optional) confidence level for establishing a confidence interval using z-score (defaults to 95; restricted to 80, 85, 90, 95 or 99 as input)
- `na`  
  (optional) value that you want to filter and exclude (defaults to include everything)
- `N`  
  (optional) population universe (e.g. 10000, nrow(df)); if N value is passed as an argument, margin of error will be calculated using fpc

**Value**

Returns table of responses (n), proportions, margins of error, lower and upper bounds by factor for a given variable in a stratified sample

**References**


**Examples**

```r
alresults <- sssamp(albania, 890, qarku)
cpro(df=alresults, numerator=totalVoters, denominator=zgjedhes, ci=95)
cpro(df=alresults, numerator=pd, denominator=validVotes, ci=95, N=5361)
```
**dedupe**

*Removes duplicate observations within collected data*

**Description**

Removes duplicate observations within collected data.

**Usage**

```r
dedupe(df, col_name)
```

**Arguments**

- `df`: object containing data frame of collected data
- `col_name`: variable within data frame by which to filter for duplicate values

**Value**

Returns table of all data based on unique values within collected data.

**Examples**

```r
aldupe <- rsamp(df=albania, n=390, rep=TRUE)
dedupe(df=aldupe, col_name=qvKod)
```

---

**dupe**

*Identifies duplicate values within collected data*

**Description**

Identifies duplicate values within collected data.

**Usage**

```r
dupe(df, col_name)
```

**Arguments**

- `df`: object containing data frame of collected data
- `col_name`: variable within data frame by which to filter for duplicate values

**Value**

Returns table of duplicate values within collected data.
Examples

```r
aldupe <- rsamp(df=albania, n=390, rep=TRUE)
dupe(df=aldupe, col_name=qvKod)
```

opening Albania 2017 CDO Election Observation Data Findings on Opening Process

Description

Data set containing 2017 Albania election observation findings on polling station opening process by the Coalition of Domestic Observers (CDO) CDO conducted a statistically-based observation (SBO) exercise, deploying observers to a random sample of polling stations for the 25 June 2017 Albanian elections. This is a subset of observation data collected by CDO observers that includes data that was used to perform statistical analysis.

Usage

opening

Format

A data frame with 524 rows and 19 variables

- qarku: district, 12 in total
- psID: polling station identifier
- votersList: number of registered voters at the polling station
- ballotPapers: number of ballot papers at the polling station
- pubPriv: type of polling station, public or private
- openTime: time when polling station opening, in 30 minute ranges
- numKommish: number of commissioners present at polling station
- secrecyOpen: yes-no if polling station enabled voters to cast ballots in secrecy, po or jo
- movementOpen: yes-no if polling station provided sufficient space to vote, po or jo
- removeMatInside: yes-no if campaign materials were removed from inside polling station, po or jo
- removeMatOutside: yes-no if campaign materials were removed from outside polling station, po or jo
- pvComplete: yes-no if commissioners completed the opening record checklist sheet, po or jo
- boxChecked: yes-no if commissioners checked to ensure the ballot box was empty before opening, po or jo
- boxSealed: yes-no if commissioners sealed the ballot box to prevent ballot tampering, po or jo
- recordBox: yes-no if commissioners recorded the seal number on the ballot box, po or jo
- centerMat: yes-no if there were all election materials were available at the polling station, po or jo
blindTools  yes-no if the polling station was equipped for blind voters, po or jo

disabledTools  yes-no-partially if the polling station was equipped for disabled voters, po or jo or

overallOpen  very good-good-problematic-very problematic an overall assessment of the opening

process, shummir,mir,meprob,shumprob

Source

https://ona.io/cdo/35080/216662

---

psampcalc  

Determines sample size by strata using sub-units

Description

Determines sample size by strata using sub-units

Usage

psampcalc(df, n, strata, unit, over = 0)

Arguments

df  object containing full sampling data frame (e.g. data)
n  sample size (integer) or object containing sample size
strata  variable in sampling data frame by which to stratify (e.g. region)
unit  variable in sampling data frame containing sub-units (e.g. population)
over  (optional) desired oversampling proportion (defaults to 0; takes value between
        0 and 1 as input)

Value

Returns sample size per strata based on sub-units (rounded up to nearest integer)

References

[1] Sampling Design & Analysis, S. Lohr, 1999, 4.4
**rmissing**

*Identifies missing points between sample and collected data*

**Description**

Identifies missing points between sample and collected data

**Usage**

```r
rmissing(sampdf, colldf, col_name)
```

**Arguments**

- `sampdf`: object containing data frame of sample points
- `colldf`: object containing data frame of collected data
- `col_name`: common variable (i.e. key) in data frames by which to check for missing points

**Value**

Returns table of sample points missing from collected data

**References**

Simplified wrapper around dplyr::anti_join()

**Examples**

```r
alsample <- rsamp(df=albania, 544)
alreceived <- rsamp(df=alsample, 390)
rmissing(sampdf=alsample, colldf=alreceived, col_name=qvKod)
```

---

**rpro**

*Calculate proportion and margin of error (simple random sample)*

**Description**

Calculate proportion and margin of error (simple random sample)

**Usage**

```r
rpro(df, col_name, ci = 95, na = "", N = 0)
```
**Arguments**

- **df**: object containing data frame on which to perform analysis (e.g. data)
- **col_name**: variable in data frame for which you want to calculate proportion and margin of error
- **ci** *(optional)*: confidence level for establishing a confidence interval using z-score (defaults to 95; restricted to 80, 85, 90, 95 or 99 as input)
- **na** *(optional)*: value that you want to filter and exclude (defaults to include everything)
- **N** *(optional)*: population universe (e.g. 10000, nrow(df)); if N value is passed as an argument, margin of error will be calculated using fpc

**Value**

Returns table of responses (n), proportions, margins of error, lower and upper bounds by factor for a given variable

**References**

[1] Sampling Design & Analysis, S. Lohr, 1999, Equation 2.15

**Examples**

rpro(df=opening, col_name=openTime, ci=95, na="n/a", N=5361)

---

**rsamp**

*Draws simple random sample without replacement*

**Description**

Draws simple random sample without replacement

**Usage**

rsamp(df, n, over = 0, rep = FALSE)

**Arguments**

- **df**: object containing full sampling data frame (e.g. data)
- **n**: sample size (integer) or object containing sample size
- **over** *(optional)*: desired oversampling proportion (defaults to 0; takes value between 0 and 1 as input)
- **rep** *(optional)*

**Value**

Returns simple random sample without replacement
RSAMP\_CALC

References
Simplified wrapper around dplyr::sample_n()

Examples
rsamp(albania, n=360, over=0.1, rep=FALSE)

size <- rsampcalc(nrow(albania), 3, 95, 0.5)
randomsample <- rsamp(albania, size)

rsampcalc

Determines random sample size

Description
Determines random sample size

Usage
rsampcalc(N, e, ci = 95, p = 0.5, over = 0)

Arguments
N population universe (e.g. 10000, nrow(df))
e tolerable margin of error (integer or float, e.g. 5, 2.5)
ci (optional) confidence level for establishing a confidence interval using z-score (defaults to 95; restricted to 80, 85, 90, 95 or 99 as input)
p (optional) anticipated response distribution (defaults to 0.5; takes value between 0 and 1 as input)
over (optional) desired oversampling proportion (defaults to 0; takes value between 0 and 1 as input)

Value
Returns appropriate sample size (rounded up to nearest integer)

References

Examples
rsampcalc(N=5361, e=3, ci=95, p=0.5, over=0.1)
rsampcalc(nrow(data), 3)
smissing

Identifies number of missing points by strata between sample and collected data

**Description**

Identifies number of missing points by strata between sample and collected data

**Usage**

`smissing(sampdf, colldf, strata, col_name)`

**Arguments**

- `sampdf`: object containing data frame of sample points
- `colldf`: object containing data frame of collected data
- `strata`: variable in both data frames by which to stratify
- `col_name`: common variable (i.e. key) in data frames by which to check for missing points

**Value**

Returns table of number of sample points by strata missing from collected data

**References**

Simplified wrapper around `dplyr::anti_join()`

**Examples**

```r
alsample <- rsamp(df=albania, 544)
alreceived <- rsamp(df=alsample, 390)
smissing(sampdf=alsample, colldf=alreceived, strata=qarku, col_name=qvKod)
```

spro

Calculate proportion and margin of error (stratified sample)

**Description**

Calculate proportion and margin of error (stratified sample)

**Usage**

`spro(fulldf, sampdf, strata, col_name, ci = 95, na = "")`
Arguments

- `fulldf`: object containing original data frame used to draw sample
- `sampdf`: object containing data frame on which to perform analysis
- `strata`: variable in both data frames by which to stratify
- `col_name`: variable in data frame for which you want to calculate proportion and margin of error
- `ci`: (optional) confidence level for establishing a confidence interval using z-score (defaults to 95; restricted to 80, 85, 90, 95 or 99 as input)
- `na`: (optional) value that you want to filter and exclude (defaults to include everything)

Value

Returns table of responses (n), proportions, margins of error, lower and upper bounds by factor for a given variable in a stratified sample

References


Examples

```r
spro(fulldf=albania, sampdf=opening, strata=qarku, col_name=openTime, ci=95, na="n/a")
```

---

**ssamp**

*Draws stratified sample without replacement using proportional allocation*

**Description**

Draws stratified sample without replacement using proportional allocation

**Usage**

```r
ssamp(df, n, strata, over = 1)
```

**Arguments**

- `df`: object containing full sampling data frame (e.g. data)
- `n`: sample size (integer) or object containing sample size
- `strata`: variable in sampling data frame by which to stratify (e.g. region)
- `over`: (optional) desired oversampling proportion (defaults to 0; takes value between 0 and 1 as input)
ssampcalc

Value

Returns stratified sample without replacement

Examples

ssamp(df=albania, n=360, strata=qarku, over=0.1)

size <- rsampcalc(nrow(albania), 3, 95, 0.5)
stratifiedsample <- ssamp(albania, size, qarku)

ssampcalc

Determines sample size by strata using proportional allocation

Description

Determines sample size by strata using proportional allocation

Usage

ssampcalc(df, n, strata, over = 0)

Arguments

df object containing sampling data frame (e.g. data)
n sample size (integer) or object containing sample size
strata variable in sampling data frame by which to stratify (e.g. region)
over (optional) desired oversampling proportion (defaults to 0; takes value between 0 and 1 as input)

Value

Returns proportional sample size per strata (rounded up to nearest integer)

References

[1] Sampling Design & Analysis, S. Lohr, 1999, 4.4

Examples

ssampcalc(df=albania, n=544, strata=qarku, over=0.05)

size <- rsampcalc(nrow(albania), 3, 95, 0.5)
ssampcalc(albania, size, qarku)
Index

* datasets
  albania, 2
  opening, 6

albania, 2

cpro, 4

dedupe, 5
dupe, 5

opening, 6

psampcalc, 7

rmissing, 8
rpro, 8
rsamp, 9
rsampcalc, 10

smissing, 11
spro, 11
ssamp, 12
ssampcalc, 13