Package: SimRDS (via r-universe)

November 9, 2024

Type Package

Title Simulation of Respondent Driven Samples

Version 2.0.0

Maintainer Ayesha Perera <gaayesha93@gmail.com>

Description Simulate populations with desired properties and extract respondent driven samples. To better understand the usage of the package and the algorithm used, please refer to Perera, A., and Ramanayake, A. (2019) <https://www.aimr.tirdiconference.com/assets/images/portfolio/Conference-Proceeding-AIMR-19.pdf>.

License GPL-2

Imports RDS, e1071, stats

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

Suggests knitr, rmarkdown

Depends R (>= 2.10)

NeedsCompilation no

Author Ayesha Perera [aut, cre]

Repository CRAN

Date/Publication 2024-11-08 15:10:09 UTC

Config/pak/sysreqs libglpk-dev make libicu-dev libxml2-dev

Contents

`population` *Generates a RDS Population*

Description

Generates RDS populations according to the user's preferred variability. Do note the computer capacities when using the function since it can affect the functionality of the methods used. A population size of 10,000 can be produced with a RAM of 4GB without an issue. But if higher population sizes are needed, then higher RAM is needed. The produced population consists of the degree size, a dichotomous independent variable, a dichotomous response variable and the IDs of the respondents in the network of each response.

Usage

```
population(
  N = 1000,
  p.ties = 0.33,
  minVal = 0,
  maxVal = 300,
  zeros = 2,
  dis_type = "rexp",
  skew = 0.05,
  pr = 0.33,
  pa = 0.33,
  atype_char = "NULL",
  atype_res = "NULL"
)
```

Arguments

`N`	Population size.
`p.ties`	Number of ties to be in the population.
`minVal`	The minimum degree size an individual in the population can have.
`maxVal`	The maximum degree size an individual in the population can have.
`zeros`	The maximum number of zeros the population can have
`dis_type`	Specify the base distribution of the degrees. ("rnorm, rexp or runif")
`skew`	is needed when the distribution is left- or right-skewed. When selecting the value for the skewness, it is advisable to first observe the range of values given from the rep function and to see whether the maximum value of the distribution is close to the defined maximal.
`pr`	Proportion of individuals that has '1' as the response in the response variable.
`pa`	Proportion of individuals that has '1' as their character in the independent variable.

atype_char — Defines how the independent variable is associated with other external factors. "NULL" when it needs to be randomly distributed. If it needs to be associated with the network size; use "net". It can be associated with network size only. You must note these abbreviations when you associate them with these variables: "network size = net."

atype_res — Defines how the response variable is associated with other external factors."NULL" is to be randomly distributed. If it needs to be associated with both independent v; use "char * net". Can associate with the network size and the independent variable abbreviations when needed to associate with these variables. "network size = net", "independent dichotomous variable = char."

Value

A simulated population with desired characteristics.

Examples

```
RDS.population <- population(N = 1000,p.ties = 0.6,minVal = 10,zeros = 0,
pr = .5,pa = .1, atype_char = "net",atype_res = "net*char")

# This function may take considerable time to produce output, depending on
# the computer's performance. For a quick reference to the expected result,
# a saved version of the output is available.
data(RDS.population)
```

RDS.population *The list that specifies the output of the* population *function.*

Description

This list contains the output of the function population so that it could be used in the sampling examples. Depending on the performance of the computer it could take a considerable amount of time to produce this output.

Usage

```
RDS.population
```

Format

An object of class list of length 4.

RDS.samplingEx *The dataframe that specifies the output of the* samplingEx *function.*

Description

The function samplingEx may take considerable time to produce output, depending on the computer's performance. For a quick reference to the expected result, a saved version of the output is available.

Usage

```
RDS.samplingEx
```

Format

An object of class rds.data.frame (inherits from data.frame) with 300 rows and 8 columns.

RDS.samplingP *The dataframe that specifies the output of the* samplingP *function.*

Description

The function samplingP may take considerable time to produce output, depending on the computer's performance. For a quick reference to the expected result, a saved version of the output is available.

Usage

```
RDS.samplingP
```

Format

An object of class rds.data.frame (inherits from data.frame) with 1899 rows and 7 columns.

samplingEx *Extracts samples from external populations*

Description

Extract samples from populations that the user inputs from an external source without using the inbuilt function 'population' to simulate a population. Data should be in .csv format. Inputs are recruited in the format of the output given in the inbuilt function 'population.'

Usage

```
samplingEx(
  population,
  response_column,
  degree_column,
  categorical_column,
  id_column,
  degree = NULL,
  net = NULL,
  seeds = 10,
  coupon = 2,
  waves = NULL,
  size = 300,
  prob = NULL,
  prop_sam = NULL,
  showids = TRUE,
  timeInterval = NULL
)
```

Arguments

population	.csv file containing the population details. The first column should contain the individual's ID.
response_column	Column including responses.
degree_column	Column including degrees.
categorical_column	Column including categories.
id_column	Column including IDs.
degree	.csv file containing the network sizes. If it's not provided, the degree size would be the number of individuals in the networks provided. If both the networks and the degrees are not provided, then it will randomly produce degrees for each respondent.
net	.csv file contains the individual's network. It should contain the same format as the output of the 'net size table' given from the output of the inbuilt function 'population'. If not provided, would randomly assign individuals considering the degree.

seeds	Number of seeds. It should be a positive integer.
coupon	Number of coupons. It should be a positive integer.
waves	Number of waves. It should be a positive integer.
size	Sample size. It should be a positive integer.
prob	Whether the seeds are selected probabilistically. If you want to be selected randomly then leave it as 'NULL' If you want to relate it to a variable, add the variable name, and it should be related. For example, if the seeds selected should depend on the network size, then the 'network' should be used. If it is inversely proportional, then it should enter as '1/network'.
prop_sam	If the individuals from the network of an individual should be chosen randomly use "NULL". If the selection should be done probabilistically, state the variable and how it should be related.
showids	If TRUE, it displays the IDs being recruited.
timeInterval	Days between recruitment waves. Zeroth wave is assumed to recruit at the present day.

Value

RDS sample

Examples

```
RDS.population <- population(N = 1000,p.ties = 0.6,minVal = 10,zeros = 0,
pr = .5,pa = .1, atype_char = "net",atype_res = "net*char")

# This should be replaced with a real world dataset.
RDS.samplingEx <- samplingEx(population=RDS.population$frame,
response_column="response", degree_column="network",
categorical_column="character", id_column="s", seeds = 40, coupon = 2,
waves =8)


# This function may take considerable time to produce output, depending on
# the computer's performance. For a quick reference to the expected result,
# a saved version of the output is available.
data(RDS.samplingEx)
```

---

samplingP *Extract samples from generated populations.*

Description

Extract samples from the generated populations using the 'SimRDS::population' function. Can directly use this output to the RDS estimates in the RDS package.

Usage

```
samplingP(
  population,
  seeds,
  coupon,
  waves = NULL,
  size = NULL,
  prob = NULL,
  character_sam = NULL,
  showids = TRUE,
  timeInterval = NULL
)
```

Arguments

population	Population generated using the 'SimRDS::population' function.
seeds	Number of seeds.
coupon	Number of coupons.
waves	Number of waves.
size	Sample size.
prob	Whether the seeds are selected probabilistically. If you want to be selected randomly, leave it as 'NULL' If you want to relate it to a variable, add the variable name. For example, if the seeds selected depend on the network size, then 'network' should be used. It should enter as '1/network' if it is inversely proportional.
character_sam	If the individuals from the network of an individual should be chosen randomly use "NULL". If the selection should be done probabilistically, state the variable and how it should be related. Possible inputs are NULL, netSize, character.
showids	If TRUE, it displays the IDs being recruited.
timeInterval	Days between recruitment waves. Zeroth wave is assumed to recruit at the present day.

Value

RDS sample

Examples

```
RDS.population <- population(N = 1000,p.ties = 0.6,minVal = 10,zeros = 0,
pr = .5,pa = .1, atype_char = "net",atype_res = "net*char")

RDS.samplingP <- samplingP(population=RDS.population, seeds = 40, coupon = 2,
waves = 8, prob = '1/network', character_sam = 'netSize')


# This function may take considerable time to produce output, depending on
# the computer's performance. For a quick reference to the expected result,
```

```
# a saved version of the output is available.
data(RDS.samplingP)
```

Index