

# Package ‘sgr’

April 14, 2022

**Type** Package

**Title** Sample Generation by Replacement

**Version** 1.3.1

**Date** 2022-04-14

**Author** Massimiliano Pastore & Luigi Lombardi

**Depends** MASS

**Suggests** polycor

**Maintainer** Massimiliano Pastore <massimiliano.pastore@unipd.it>

**Description** Sample Generation by Replacement simulations (SGR; Lombardi & Pastore, 2014; Pastore & Lombardi, 2014). The package can be used to perform fake data analysis according to the sample generation by replacement approach. It includes functions for making simple inferences about discrete/ordinal fake data. The package allows to study the implications of fake data for empirical results.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-04-14 14:30:02 UTC

## R topics documented:

amse . . . . .	2
arb . . . . .	3
dgBeta . . . . .	4
dgBetaD . . . . .	5
model.fake.par . . . . .	7
partition.replacement . . . . .	8
pfake . . . . .	9
pfakebad . . . . .	12
pfakegood . . . . .	13
psydata . . . . .	14
rdatagen . . . . .	15
rdatarepl . . . . .	16

replacement.matrix . . . . .	18
smokers . . . . .	19

<b>Index</b>	<b>20</b>
--------------	-----------

amse *Average root mean square error*

## Description

Average root mean square error (AMSE).

## Usage

amse(Bpar, Bθ)

## Arguments

Bpar Matrix with dimension  $B$  (replicates)  $\times P$  (parameters).  
 Bθ Vector of true parameter values.

## Details

Let  $\hat{\theta}_{ij}$  be the estimated parameter value for the  $j$ th parameter in the  $i$ th sample (replicate),  $i = 1, 2, \dots, B$ ,  $j = 1, 2, \dots, P$ , and let  $\theta_j$  be the corresponding true parameter value, the Average root mean square error is defined as follows:

$$AMSE = \frac{1}{B} \sum_i \sqrt{\frac{1}{P} \sum_j \left[ \frac{\hat{\theta}_{ij} - \theta_j}{\theta_j} \right]^2}$$

## Value

Gives the AMSE value.

## Note

If  $\theta_j = 0$ , the ratio  $\left[ \frac{\hat{\theta}_{ij} - \theta_j}{\theta_j} \right]$  is modified as follows:  $\left[ \frac{\hat{\theta}_{ij} - 0}{1} \right]$

## Author(s)

Massimiliano Pastore & Luigi Lombardi

## References

Yang-Wallentin, F., Joreskog, K. G., Luo, H. (2010). Confirmatory Factor Analysis of Ordinal Variables With Misspecified Models, *Structural Equation Modeling: A Multidisciplinary Journal*, 17, 392-423.

**See Also**[arb](#)

arb

*Average relative bias***Description**

Average relative bias (ARB).

**Usage**

arb(Bpar, B0)

**Arguments**

Bpar                    Matrix with dimension  $B$  (replicates)  $\times P$  (parameters).  
 B0                        Vector of true parameter values.

**Details**

Let  $\hat{\theta}_{ij}$  be the estimated parameter value for the  $j$ th parameter in the  $i$ th sample (replicate),  $i = 1, 2, \dots, B$ ,  $j = 1, 2, \dots, P$ , and let  $\theta_j$  be the corresponding true parameter value, the Average relative bias is defined as follows:

$$ARB = \frac{100}{B} \sum_i \frac{1}{P} \sum_j \left( \frac{\hat{\theta}_{ij} - \theta_j}{\theta_j} \right)$$

**Value**

Gives the ARB value.

**Note**

If  $\theta_j = 0$ , the ratio  $\left( \frac{\hat{\theta}_{ij} - \theta_j}{\theta_j} \right)$  is modified as follows:  $\left( \frac{\hat{\theta}_{ij} - 0}{1} \right)$

**Author(s)**

Massimiliano Pastore &amp; Luigi Lombardi

**References**

Yang-Wallentin, F., Joreskog, K. G., Luo, H. (2010). Confirmatory Factor Analysis of Ordinal Variables With Misspecified Models, *Structural Equation Modeling: A Multidisciplinary Journal*, 17, 392-423.

**See Also**[amse](#)

---

 dgBeta

*Generalized Beta Distribution.*


---

### Description

The generalized beta distribution extends the classical beta distribution beyond the [0,1] range (Whitby, 1971).

### Usage

```
dgBeta(x, a = min(x), b = max(x), gam = 1, del = 1)
```

### Arguments

x	Vector of quantiles.
a	Minimum of range of r.v. $X$ .
b	Maximum of range of r.v. $X$ .
gam	Gamma parameter.
del	Delta parameter.

### Details

The Generalized Beta Distribution is defined as follows:

$$G(x; a, b, \gamma, \delta) = \frac{1}{B(\gamma, \delta)(b-a)^{\gamma+\delta-1}} (x-a)^{\gamma-1} (b-x)^{\delta-1}$$

where  $B(\gamma, \delta)$  is the beta function. The parameters  $a \in R$  and  $b \in R$  (with  $a < b$ ) are the left and right end points, respectively. The parameters  $\gamma > 0$  and  $\delta > 0$  are the governing shape parameters for  $a$  and  $b$  respectively. For all the values of the r.v.  $X$  that fall outside the interval  $[a, b]$ ,  $G$  simply takes value 0. The generalized beta distribution reduces to the beta distribution when  $a = 0$  and  $b = 1$ .

### Value

Gives the density.

### Author(s)

Massimiliano Pastore & Luigi Lombardi

### References

Whitby, O. (1971). *Estimation of parameters in the generalized beta distribution* (Technical Report NO. 29). Department of Statistics: Stanford University.

**See Also**[dgBetaD](#)**Examples**

```

curve(dgBeta(x))
curve(dgBeta(x,gam=3,del=3))
curve(dgBeta(x,gam=1.5,del=2.5))

x <- 1:7
GA <- c(1,3,1.5,8); DE <- c(1,3,4,2.5)
par(mfrow=c(2,2))
for (j in 1:4) {
  plot(x,dgBeta(x,gam=GA[j],del=DE[j]),type="h",
       panel.first=points(x,dgBeta(x,gam=GA[j],del=DE[j]),pch=19),
       main=paste("gamma=",GA[j]," delta=",DE[j],sep=""),ylim=c(0,.6),
       ylab="dgBeta(x)")
}

```

dgBetaD

*Generalized Beta distribution for discrete variables***Description**

Generalized Beta distribution for discrete variables.

**Usage**

```
dgBetaD(x, a = min(x), b = max(x), gam = 1, del = 1, ct = 1)
```

**Arguments**

x	Vector of quantiles.
a	Minimum of range of r.v. $X$ .
b	Maximum of range of r.v. $X$ .
gam	Gamma parameter.
del	Delta parameter.
ct	Correction term, default value: 1.

**Details**

Let  $X$  be a discrete r. v. with range

$$R_X = \{a, a + 1, a + 2, \dots, a + t - 1, a + t = b\}$$

and where  $a \in \mathbb{N} \cup \{0\}$  and  $t \in \mathbb{N}$ . The Generalized Discrete Beta Distribution for the r.v.  $X$  is defined as follows:

$$DG(x; a, b, \gamma, \delta) = \begin{cases} \frac{G^*(x; a, b, \gamma, \delta)}{\sum_{x' \in R_X} G^*(x'; a, b, \gamma, \delta)} & x \in R_X \\ 0 & x \notin R_X \end{cases}$$

where  $G^*$  is a modified version of the generalized beta distribution [dgBeta](#) defined as

$$G^*(x; a, b, \gamma, \delta) = \frac{1}{B(\gamma, \delta)(b - a + 2c)^{\gamma + \delta - 1}} (x - a + c)^{\gamma - 1} (b - x + c)^{\delta - 1}$$

### Value

Gives the density.

### Author(s)

Massimiliano Pastore & Luigi Lombardi

### References

Lombardi, L., Pastore, M. (2014). sgr: A Package for Simulating Conditional Fake Ordinal Data. *The R Journal*, 6, 164-177.

Pastore, M., Lombardi, L. (2014). The impact of faking on Cronbach's Alpha for dichotomous and ordered rating scores. *Quality & Quantity*, 48, 1191-1211.

### See Also

[dgBeta](#)

### Examples

```
x <- 1:7
GA <- c(1,3,1.5,8); DE <- c(1,3,4,2.5)
par(mfrow=c(2,2))
for (j in 1:4) {
  plot(x,dgBetaD(x,gam=GA[j],del=DE[j]),type="h",
       panel.first=points(x,dgBetaD(x,gam=GA[j],del=DE[j]),pch=19),
       main=paste("gamma=",GA[j]," delta=",DE[j],sep=""),ylim=c(0,.6),
       ylab="dgBetaD(x)")
}
```

---

model.fake.par	<i>Internal function.</i>
----------------	---------------------------

---

## Description

Set different instances of the conditional replacement distribution.

## Usage

```
model.fake.par(fake.model = c("uninformative", "average", "slight", "extreme"))
```

## Arguments

fake.model	A character string indicating the model for the conditional replacement distribution. The options are: uninformative (default option) [gam = c(1, 1) and del = c(1, 1)]; average [gam = c(3, 3) and del = c(3, 3)]; slight [gam = c(1.5, 4) and del = c(4, 1.5)]; extreme [gam = c(4, 1.5) and del = c(1.5, 4)].
------------	--

## Value

Gives a list with  $\gamma$  and  $\delta$  parameters.

## Author(s)

Massimiliano Pastore

## References

Lombardi, L., Pastore, M. (2014). sgr: A Package for Simulating Conditional Fake Ordinal Data. *The R Journal*, 6, 164-177.

Pastore, M., Lombardi, L. (2014). The impact of faking on Cronbach's Alpha for dichotomous and ordered rating scores. *Quality & Quantity*, 48, 1191-1211.

## Examples

```
model.fake.par() # default
model.fake.par("average")
```

---

partition.replacement *Internal function.*

---

### Description

This function allows to set different replacement distributions for different subsets of cells in the data matrix.

### Usage

```
partition.replacement(Dx, PM, Q = NULL, Pparam = NULL,
  fake.model = NULL, p = NULL)
```

### Arguments

Dx	Data frame or matrix to be replaced.
PM	Partition matrix with size $\dim(Dx)$ . See details.
Q	Max value in the discrete r.v. range: $1, \dots, Q$ .
Pparam	List of replacement parameters for each class in the replacement partition. See details.
fake.model	A character string indicating the model for the conditional replacement distribution, see <a href="#">model.fake.par</a> .
p	Overall probability of replacement. Must be a matrix with $P$ rows and two columns. See details.

### Details

PM has size  $\dim(Dx)$  and contains a numeric code for each distinct class in the partition. If a cell of the partition matrix PM contains 0, then the corresponding Dx cell value is not modified (no replacements condition class).

Pparam is a list containing three elements. Each element is a  $P \times 2$  matrix where  $P$  is the total number of classes in the partition (see examples for further details).

p: Overall probability of replacement:  $p[, 1]$  indicates the faking good probability,  $p[, 2]$  indicates the faking bad probability.

gam: Gamma parameter:  $gam[, 1]$  and  $gam[, 2]$  indicate the faking good and the faking bad parameters for the lower bound a.

del: Delta parameter:  $del[, 1]$  and  $del[, 2]$  indicate the faking good and the faking bad parameters for the upper bound b.

Note that it is possible to define a faking model using the fake.model assignment. In such cases the user must specify also the overall probability of replacement using parameter p.

### Value

Returns the fake data matrix.

**Author(s)**

Massimiliano Pastore

**See Also**[rdatarepl](#), [replacement.matrix](#)**Examples**

```

require(MASS)
set.seed(20130207)
R <- matrix(c(1, .3, .3, 1), 2, 2)
Dx <- rdatagen(n=20, R=R, Q=5)$data

## partition matrix
PM <- matrix(0, nrow(Dx), ncol(Dx))
PM[3:10, 2] <- 1
PM[3:10, 1] <- 1
partition.replacement(Dx, PM) # warning! zero replacements

## using fake.model
partition.replacement(Dx, PM, fake.model="uninformative", p=matrix(c(.3, .2), ncol=2))

###
p <- c(.5, 0)
gam <- c(1, 1)
del <- c(1, 1)
Pparam <- list(p=p, gam=gam, del=del)
partition.replacement(Dx, PM, Pparam=Pparam)

### another partition
PM[11:15, 2] <- 2
(Pparam <- list(p=matrix(c(0, .5, .5, 0), 2, 2),
  gam=matrix(c(1, 4, 1, 4), 2, 2), del=matrix(c(1, 2, 1, 2), 2, 2)))
partition.replacement(Dx, PM, Pparam=Pparam)

```

pfake

*Probability of faking.***Description**

The function gives the conditional probability of replacement  $p(f = k | d = h, \theta_F)$  for discrete values in the range  $1, \dots, Q$ .

**Usage**

```

pfake(k, h = k, p = c(0, 0), Q = 5, gam = c(1, 1), del = c(1, 1),
  fake.model = c("uninformative", "average", "slight", "extreme"))

```

**Arguments**

k	A fake value.
h	An observed original value.
p	Overall probability of replacement: p[1] indicates the faking good probability, p[2] indicates the faking bad probability.
Q	Max value in the discrete r.v. range: 1, ..., Q.
gam	Gamma parameter: gam[, 1] indicates the faking good parameter $\gamma_+$ , gam[, 2] indicates the faking bad parameter $\gamma_-$ .
del	Delta parameter: del[, 1] indicates the faking good parameter $\delta_+$ , del[, 2] indicates the faking bad parameter $\delta_-$ .
fake.model	A character string indicating the model for the conditional replacement distribution. The options are: uninformative (default option) [gam = c(1, 1) and del = c(1, 1)]; average [gam = c(3, 3) and del = c(3, 3)]; slight [gam = c(1.5, 4) and del = c(4, 1.5)]; extreme [gam = c(4, 1.5) and del = c(1.5, 4)].

**Value**

Gives the conditional probability distribution based on the following equation

$$p(f = k | d = h, \theta_F) = \begin{cases} DG(k; h + 1, Q, \gamma_+, \delta_+) \pi_+ & 1 \leq h < k \leq Q \\ DG(k; q, h - 1, \gamma_-, \delta_-) \pi_- & 1 \leq k < h \leq Q \\ 1 - (\pi_+ + \pi_-) & 1 < h = k < Q \\ 1 - \pi_+ & k = h = 1 \\ 1 - \pi_- & k = h = Q \end{cases}$$

with  $\theta_F$  and  $DG$  being the parameter vector  $(\gamma_+, \gamma_-, \delta_+, \delta_-, \pi_+, \pi_-)$  and the generalized Beta distribution for discrete variables (`dgBetaD`) with bounds  $a = h + 1$  (resp.  $a = 1$ ) and  $b = Q$  (resp.  $b = h - 1$ ). The parameter  $\pi_+$  denotes the probability of faking good,  $\pi_-$  indicates the probability of faking bad. Note that the faking probabilities must satisfy the following condition:  $\pi_+ + \pi_- \leq 1$ .

**Author(s)**

Massimiliano Pastore & Luigi Lombardi

**References**

- Lombardi, L., Pastore, M. (2014). sgr: A Package for Simulating Conditional Fake Ordinal Data. *The R Journal*, 6, 164-177.
- Pastore, M., Lombardi, L. (2014). The impact of faking on Cronbach's Alpha for dichotomous and ordered rating scores. *Quality & Quantity*, 48, 1191-1211.

**Examples**

```
x <- 1:7
GA <- c(1,3,1.5,8); DE <- c(1,3,4,2.5)

### fake good
```

```

par(mfrow=c(2,2))
for (j in 1:4) {
  y <- NULL
  for (i in x) y <- c(y,pfake(x[i],h=4,Q=7,
    gam=c(GA[j],GA[j]),del=c(DE[j],DE[j]),p=c(.4,0)))
  plot(x,y,type="h",panel.first=points(x,y,pch=19),
    main=paste("gamma=",GA[j]," delta=",DE[j],sep=""),ylim=c(0,.7),
    ylab="Replacement probability")
}

### fake bad
for (j in 1:4) {
  y <- NULL
  for (i in x) y <- c(y,pfake(x[i],h=4,Q=7,
    gam=c(GA[j],GA[j]),del=c(DE[j],DE[j]),p=c(0,.4)))
  plot(x,y,type="h",panel.first=points(x,y,pch=19),
    main=paste("gamma=",GA[j]," delta=",DE[j],sep=""),ylim=c(0,.7),
    ylab="Replacement probability")
}

### fake good and fake bad
P = c(.4,.4)
par(mfrow=c(2,4))
for (j in x) {
  y <- NULL
  for (i in x) {
    y <- c(y,pfake(x[i],h=x[j],Q=max(x),gam=c(GA[1],GA[1]),del=c(DE[1],DE[1]),p=P))
  }
  plot(x,y,type="h",panel.first=points(x,y,pch=19),
    main=paste("h=",x[j],sep=""),ylim=c(0,1),
    ylab="Replacement probability")
  print(sum(y,na.rm=TRUE))
}

### using the fake.model argument
x <- 1:5
models <- c("uninformative","average","slight","extreme")
par(mfrow=c(2,2))
for (j in 1:4) {
  y <- NULL
  for (i in x) y <- c(y,pfake(x[i],h=2,Q=max(x),
    fake.model=models[j],p=c(.45,0))) # fake good
  plot(x,y,type="h",panel.first=points(x,y,pch=19),
    main=paste(models[j],"model"),ylim=c(0,1),
    ylab="Replacement probability")
}

par(mfrow=c(2,2))
for (j in 1:4) {
  y <- NULL
  for (i in x) y <- c(y,pfake(x[i],h=4,Q=max(x),
    fake.model=models[j],p=c(0,.45))) # fake bad
  plot(x,y,type="h",panel.first=points(x,y,pch=19),

```

```

    main=paste(models[j], "model"), ylim=c(0,1),
    ylab="Replacement probability")
}

```

---

pfakebad

*Probability of faking bad.*

---

### Description

The function gives the conditional probability of replacement  $p(f = k|d = h, \theta_F)$  for discrete values in the range  $1, \dots, Q$ .

### Usage

```
pfakebad(k, h = k, p = 0, Q = 5, gam = 1, del = 1)
```

### Arguments

k	A fake value.
h	An observed original value.
p	Overall probability of replacement.
Q	Max value in the discrete r.v. range: $1, \dots, Q$ .
gam	Gamma parameter.
del	Delta parameter.

### Value

Gives the conditional probability based on the following equation

$$p(f = k|d = h, \theta_F) = \begin{cases} 1 & h = k = 1 \\ GD(k; 1, h - 1, \gamma, \delta)\pi & 1 \leq k < h \leq Q \\ 1 - \pi & 1 < h = k \leq Q \\ 0 & 1 \leq h < k \leq Q \end{cases}$$

with  $\theta_F$  and  $GD$  being the parameter vector  $(\gamma, \delta, \pi)$  and the generalized Beta distribution for discrete variables ([dgBetaD](#)) with bounds  $a = h + 1$  and  $b = Q$ . The parameter  $\pi$  denotes the probability of faking bad.

### Author(s)

Massimiliano Pastore & Luigi Lombardi

### References

Pastore, M., Lombardi, L. (2014). The impact of faking on Cronbach's Alpha for dichotomous and ordered rating scores. *Quality & Quantity*, 48, 1191-1211.

**Examples**

```

x <- 1:7
GA <- c(1,3,1.5,8); DE <- c(1,3,4,2.5)
par(mfrow=c(2,2))
for (j in 1:4) {
  y <- NULL
  for (i in x) y <- c(y,pfakebad(x[i],h=5,Q=7,gam=GA[j],del=DE[j],p=.4))
  plot(x,y,type="h",panel.first=points(x,y,pch=19),
       main=paste("gamma=",GA[j]," delta=",DE[j],sep=""),ylim=c(0,.7),
       ylab="Replacement probability")
}

```

pfakegood

*Probability of faking good.***Description**

The function gives the conditional probability of replacement  $p(f = k|d = h, \theta_F)$  for discrete values in the range  $1, \dots, Q$ .

**Usage**

```
pfakegood(k, h = k, p = 0, Q = 5, gam = 1, del = 1)
```

**Arguments**

k	A fake value.
h	An observed original value.
p	Overall probability of replacement.
Q	Max value in the discrete r.v. range: $1, \dots, Q$ .
gam	Gamma parameter.
del	Delta parameter.

**Value**

Gives the conditional probability based on the following equation

$$p(f = k|d = h, \theta_F) = \begin{cases} 1 & h = k = Q \\ GD(k; h + 1, Q, \gamma, \delta)\pi & 1 \leq h < k \leq Q \\ 1 - \pi & 1 \leq k = h < Q \\ 0 & 1 \leq k < h \leq Q \end{cases}$$

with  $\theta_F$  and  $GD$  being the parameter vector  $(\gamma, \delta, \pi)$  and the generalized Beta distribution for discrete variables ([dgBetaD](#)) with bounds  $a = h + 1$  and  $b = Q$ . The parameter  $\pi$  denotes the probability of faking good.

**Author(s)**

Massimiliano Pastore & Luigi Lombardi

**References**

Pastore, M., Lombardi, L. (2014). The impact of faking on Cronbach's Alpha for dichotomous and ordered rating scores. *Quality & Quantity*, 48, 1191-1211.

**Examples**

```
x <- 1:7
GA <- c(1,3,1.5,8); DE <- c(1,3,4,2.5)
par(mfrow=c(2,2))
for (j in 1:4) {
  y <- NULL
  for (i in x) y <- c(y,pfakegood(x[i],h=3,Q=7,gam=GA[j],del=DE[j],p=.4))
  plot(x,y,type="h",panel.first=points(x,y,pch=19),
       main=paste("gamma=",GA[j]," delta=",DE[j],sep=""),ylim=c(0,.7),
       ylab="Replacement probability")
}
```

---

psydata

*Data set*

---

**Description**

The psydata data frame has 744 rows (observations) and 22 columns (variables).

**Usage**

```
data(psydata)
```

**Format**

This data frame contains the following variables:

- nsogg: int, subject number.
- vers: Factor, questionnaire version: V1 fake-motivating version, V3 honest-motivating version e V4 neutral version.
- sex: Factor, gender.
- eta: int, age.
- resid: Factor, residence.
- dipl: Factor, education.
- voto: int, high school's final score.
- votomax: int, maximum value for voto.
- cd1: Factor, a character string indicating the type of undergraduate program.
- aep. .: int, 12 items of the AEP/A scale.
- tot: int, total score.

**Author(s)**

Andrea Bobbio, Massimo Nucci, Massimiliano Pastore

---

rdatagen	<i>Simulate discrete data.</i>
----------	--------------------------------

---

**Description**

Simulate discrete data from either a correlation matrix or thresholds or probabilities.

**Usage**

```
rdatagen(n = 100, R = diag(1,2), Q = NULL, th = NULL, probs = NULL)
```

**Arguments**

n	Number of observations.
R	Correlation matrix.
Q	Number of discrete values in the random variables. It is a single value or a vector. If Q is set to 1 (default), the function returns continuous data distributed according to the normal standardized distribution.
th	List of thresholds; each element contains Q+1 values.
probs	List of probabilities; each elements contains Q values.

**Value**

Returns a list with four elements:

data	The simulated data matrix.
R	Correlation matrix.
thresholds	The list of thresholds.
probs	The list of probabilities.

**Note**

Defaults work like in the `mvrnorm` function of the MASS package.

**Author(s)**

Massimiliano Pastore, Luigi Lombardi & Marco Bressan

**References**

Lombardi, L., Pastore, M. (2014). `sgr`: A Package for Simulating Conditional Fake Ordinal Data. *The R Journal*, 6, 164-177.

Pastore, M., Lombardi, L. (2014). The impact of faking on Cronbach's Alpha for dichotomous and ordered rating scores. *Quality & Quantity*, 48, 1191-1211.

**Examples**

```

require(MASS)
## only default
rdatagen()

## set correlations only
R <- matrix(c(1, .4, .4, 1), 2, 2)
Dx <- rdatagen(n=10000, R=R)$data

par(mfrow=c(1,2))
for (j in 1:ncol(Dx)) hist(Dx[,j])

## set correlations and Q
Dx <- rdatagen(n=10000, R=R, Q=2)$data

par(mfrow=c(1,2))
for (j in 1:ncol(Dx)) barplot(table(Dx[,j])/nrow(Dx))

## set correlations and thresholds
th <- list(c(-Inf, qchisq(pbinom(0:3, 4, .5), 1), Inf),
           c(-Inf, qnorm(pbinom(0:2, 3, .5)), Inf))
Dx <- rdatagen(n=10000, R=R, th=th)$data

par(mfrow=c(1,2))
for (j in 1:ncol(Dx)) barplot(table(Dx[,j])/nrow(Dx))

## set correlations and probabilities [1]
probs <- list(c(.125, .375, .375, .125), c(.125, .375, .375, .125))
Dx <- rdatagen(n=10000, R=R, probs=probs)$data

par(mfrow=c(1,2))
for (j in 1:ncol(Dx)) barplot(table(Dx[,j])/nrow(Dx))

## set correlations and probabilities [2]
probs <- c(.125, .375, .375, .125)
Dx <- rdatagen(n=10000, R=R, probs=probs)$data

par(mfrow=c(1,2))
for (j in 1:ncol(Dx)) barplot(table(Dx[,j])/nrow(Dx))

## set different values for Q
Dx <- rdatagen(n=1000, Q=c(2, 4), R=R)$data

par(mfrow=c(1,2))
for (j in 1:ncol(Dx)) barplot(table(Dx[,j])/nrow(Dx))

```

**Description**

Replaces data in the original data matrix using a specified replacement matrix.

**Usage**

```
rdatarepl(Dx, RM, printfp = TRUE)
```

**Arguments**

Dx	Data frame or matrix to be replaced.
RM	Replacement matrix.
printfp	Logical, if TRUE (the default), it prints the percentage of data replaced.

**Details**

Replacement matrices can be obtained from the [replacement.matrix](#) function.

**Value**

Returns a list with two elements:

Fx	The replaced (fake) data matrix.
Fperc	Percentage of replaced data.

**Author(s)**

Massimiliano Pastore

**See Also**

[replacement.matrix](#)

**Examples**

```
require(MASS)
set.seed(20130207)
Dx <- rdatagen(R=matrix(c(1, .3, .3, 1), 2, 2), Q=5)$data
RM <- replacement.matrix(p=c(.6, 0))
Fx <- rdatarepl(Dx, RM)$Fx

par(mfrow=c(2, 2))
for (j in 1:ncol(Dx)) barplot(table(Dx[, j]), main="original data")
for (j in 1:ncol(Fx)) barplot(table(Fx[, j]), main="replaced data")
```

---

replacement.matrix      *Replacement matrix.*

---

### Description

Builds the replacement matrix.

### Usage

```
replacement.matrix(Q = 5, p = c(0,0), gam = c(1,1), del = c(1,1),
  fake.model = c("uninformative", "average", "slight", "extreme"))
```

### Arguments

Q	Max value in the discrete r.v. range: $1, \dots, Q$ .
p	Overall probability of replacement: p[1] indicates the faking good probability, p[2] indicates the faking bad probability.
gam	Gamma parameter: gam[, 1] indicates the faking good parameter $\gamma_+$ , gam[, 2] indicates the faking bad parameter $\gamma_-$ .
del	Delta parameter: del[, 1] indicates the faking good parameter $\delta_+$ , del[, 2] indicates the faking bad parameter $\delta_-$ .
fake.model	A character string indicating the model for the conditional replacement distribution. The options are: <i>uninformative</i> (default option) [gam = c(1, 1) and del = c(1, 1)]; <i>average</i> [gam = c(3, 3) and del = c(3, 3)]; <i>slight</i> [gam = c(1.5, 4) and del = c(4, 1.5)]; <i>extreme</i> [gam = c(4, 1.5) and del = c(1.5, 4)].

### Value

Gives a  $Q \times Q$  matrix with replacement probabilities. Each row  $r$  ( $1 \leq r \leq Q$ ) in the matrix indicates the conditional probability distribution

$$p(k = r | h = c, \pi), \quad h = 1, \dots, Q$$

$\pi$  (p) denotes the overall replacement probability.

### Author(s)

Massimiliano Pastore

### See Also

[dgBetaD](#), [pfake](#), [pfakegood](#), [pfakebad](#)

**Examples**

```
## no replacements
replacement.matrix(Q=7)

## faking good
replacement.matrix(Q=7,p=c(.5,0))
replacement.matrix(Q=7,p=c(.5,0),gam=8,del=2.5)

## faking bad
replacement.matrix(Q=7,p=c(0,.5))
replacement.matrix(Q=7,p=c(0,.5),gam=8,del=2.5)

## faking good and faking bad
replacement.matrix(Q=7,p=c(.3,.5),gam=c(8,8),del=c(2.5,2.5))

## using the fake.model argument
replacement.matrix(Q=7,p=c(0,.4),fake.model="extreme")
replacement.matrix(Q=7,p=c(.4,0),fake.model="extreme")
replacement.matrix(Q=7,p=c(.4,.4),fake.model="slight")
```

---

smokers

*Data set*

---

**Description**

Data about smoking and drug consumption among young people.

**Usage**

```
data(smokers)
```

**Format**

This data frame contains the following columns:

- age: int, 1 = adults, 2 = minors.
- smoking: int, 1 = yes, 2 = no.
- drug: int, drug addiction, 1 = yes, 2 = no.
- druguse: int, drug consumption, 1 = never, 2 = once, 3 = some times, 4 = often.

**Source**

Pastore, M., Lombardi, L., Mereu, F. (2007). Effects of malingering in self-report measures: A scenario analysis approach; in C. H. Skiadas (Ed.), *Recent Advances in Stochastic Modeling and Data Analysis*, pp. 483-491, World Scientific Publishing.

# Index

- \* **datagen**
  - rdatagen, 15
- \* **data**
  - psydata, 14
  - smokers, 19
- \* **distribution**
  - dgBeta, 4
  - dgBetaD, 5
  - pfake, 9
  - pfakebad, 12
  - pfakegood, 13
- \* **utility**
  - amse, 2
  - arb, 3
  - model.fake.par, 7
  - partition.replacement, 8
  - rdatarepl, 16
  - replacement.matrix, 18

amse, 2, 3

arb, 3, 3

dgBeta, 4, 6

dgBetaD, 5, 5, 10, 12, 13, 18

model.fake.par, 7, 8

partition.replacement, 8

pfake, 9, 18

pfakebad, 12, 18

pfakegood, 13, 18

psydata, 14

rdatagen, 15

rdatarepl, 9, 16

replacement.matrix, 9, 17, 18

smokers, 19