

# Package ‘gamlss.tr’

November 22, 2009

**Type** Package

**Title** Generating and fitting truncated (gamlss.family) distributions

**Version** 3.1-0

**Date** 2009-11-22

**Author** Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby  
<r.rigby@londonmet.ac.uk>

**Maintainer** Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

**LazyLoad** yes

**Depends** R (>= 2.2.1), gamlss

**Description** This is an add on package to GAMLSS. The purpose of this package is to allow users to defined truncated distributions in GAMLSS models. The main function `gen.trun()` generates truncated version of an existing GAMLSS family distribution.

**License** GPL-2 | GPL-3

**URL** <http://www.gamlss.com/>

**Repository** CRAN

**Date/Publication** 2009-11-22 16:17:28

## R topics documented:

gamlss.tr-package . . . . .	2
gen.trun . . . . .	3
trun . . . . .	4
trun.d . . . . .	6
trun.p . . . . .	8
trun.q . . . . .	10
trun.r . . . . .	12

<b>Index</b>	<b>15</b>
--------------	-----------

---

gamlss.tr-package *The GAMLSS Add On Package for Truncated Distributions*

---

## Description

The main purpose of this package is to allow the user of the GAMLSS models to fit truncated distributions.

## Details

Package: gamlss.tr  
Type: Package  
Version: 1.0  
Date: 2005-11-21  
License: GPL (version 2 or later)

~~ An overview of how to use the package, including the most important functions ~~

## Author(s)

Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>> and Bob Rigby <<r.rigby@londonmet.ac.uk>>

Maintainer: Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

## References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

## Examples

```
gen.trun(par=c(0,100),family="TF", name="0to100", type="both")
op <- par(mfrow = c(2, 2))
plot(function(x) dTF0to100(x, mu=80 ,sigma=20, nu=5), 0, 100)
plot(function(x) pTF0to100(x, mu=80 ,sigma=20, nu=5), 0, 100)
plot(function(x) qTF0to100(x, mu=80 ,sigma=20, nu=5), 0.01, .999)
hist(rTF0to100(1000, mu=80 ,sigma=20, nu=5))
op
```

---

`gen.trun`*Generates a Truncate Distribution from a gamlss.family*

---

**Description**

The `gen.trun()` function allows the user to generate the `d`, the `p`, the `q`, the `r` and the `gamlss` fitting functions of a truncated distribution given that the input `family` is a current GAMLSS family distribution.

**Usage**

```
gen.trun(par = c(0), family = "NO", name = "tr", type = c("left", "right", "both"),
```

**Arguments**

<code>par</code>	a vector with one (for left or right truncation) or two elements for both
<code>family</code>	a <code>gamlss.family</code> object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by <code>gamlss()</code> can be found in <code>gamlss.family</code> . Functions such as <code>BI()</code> (binomial) produce a family object.
<code>name</code>	the characters you want to add to the name of new truncated distribution, by default adds <code>tr</code>
<code>type</code>	whether left, right or in both sides truncation is required
<code>...</code>	for extra arguments

**Value**

Returns the `d`, the `p`, the `q` and the `r` functions of a truncated distribution

**Author(s)**

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

**References**

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

**See Also**

[trun.d](#), [trun.p](#), [trun.q](#), [trun.r](#)

## Examples

```
gen.trun(par=c(0,100),family="TF", name="0to100", type="both")
plot(function(x) dTF0to100(x, mu=80 ,sigma=20, nu=5), 0, 100)
plot(function(x) pTF0to100(x, mu=80 ,sigma=20, nu=5), 0, 100)
plot(function(x) qTF0to100(x, mu=80 ,sigma=20, nu=5), 0.01, .999)
hist(s1<-rTF0to100(1000, mu=80 ,sigma=20, nu=5))
m1<-gamlss(s1~1, family=TF0to100)
```

---

trun

*Fits a Truncate Distribution from a gamlss.family*


---

## Description

This function can be used to fit truncated distributions. It takes as an argument an existing GAMLSS family distribution and a parameter vector, of the type `c(left.value, right.value)`, and generates a `gamlss.family` object which then can be used to fit a truncated distribution.

## Usage

```
trun(par = c(0), family = "NO", type = c("left", "right", "both"), name = "tr",
     local = TRUE, delta=NULL, ...)
```

## Arguments

<code>par</code>	a scalar for left and right truncation or a vector of the type <code>c(left.value, right.value)</code> for interval truncation
<code>family</code>	an existing <code>gamlss.family</code> distribution
<code>type</code>	what type of truncation is required, left, right or both. If both the <code>par</code> should be a vector of length two. (the default is left truncation)
<code>name</code>	a character string to be added to name of the created object i.e. with <code>family=TF</code> and <code>name=trZero</code> the <code>gamlss.family</code> object will be called <code>TFtrZero</code>
<code>local</code>	if <code>TRUE</code> the function will try to find the environment of <code>gamlss</code> to generate the <code>d</code> and <code>p</code> functions required for the fitting, if <code>FALSE</code> the functions will be generated in the global environment
<code>delta</code>	the delta increment used in the numerical derivatives
<code>...</code>	for extra arguments

## Details

This function is created to help the user to fit a truncated form of existing `gamlss` distribution. It does this by taking an existing `gamlss.family` and changing some of the components of the distribution to help the fitting process. It particular it i) creates a pdf (`d`) and a cdf (`p`) function within `gamlss`, ii) changes the global deviance function `G.dev.incr`, the first derivative functions (see note below) and the quantile residual function.

**Value**

It returns a `gamlss.family` object which has all the components needed for fitting a distribution in `gamlss`.

**Note**

This function is experimental and could be changed. The function `trun` changes the first derivatives of the original `gamlss` family `d` function to numerical derivatives for the new truncated `d` function. The default increment `delta`, for this numerical derivatives function, is `eps * pmax(abs(x), 1)` where `eps<-sqrt(.Machine$double.eps)`. The default `delta` could be inappropriate for specific applications and can be overwritten by using the argument `delta`.

**Author(s)**

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

**References**

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

**See Also**

[trun.d](#), [trun.p](#), [trun.q](#), [trun.r](#), [gen.trun](#)

**Examples**

```
# generate a left truncated zero t family
gen.trun(0,family="TF")
# take a random sample of 1000 observations
sam<-rTFtr(1000,mu=10,sigma=5, nu=5 )
hist(sam)
# fit the distribution to the data
mod1<-gamlss(sam~1, family=trun(0,TF))
mod1
# now create a gamlss.family object before the fitting
Ttruc.Zero<- trun(par=0,family=TF, local=FALSE)
mod2<-gamlss(sam~1, family=Ttruc.Zero)
# now check the sensitivity of delta
Ttruc.Zero<- trun(par=0,family=TF, local=FALSE, delta=c(0.01,0.01, 0.01))
mod3<-gamlss(sam~1, family=Ttruc.Zero)
```

---

trun.d	<i>Truncated Probability Density Function of a gamlss.family Distribution</i>
--------	---

---

## Description

Creates a truncated probability density function version from a current GAMLSS family distribution

## Usage

```
trun.d(par, family = "NO", type = c("left", "right", "both"), ...)
```

## Arguments

par	a vector with one (for left or right truncation) or two elements for both
family	a <code>gamlss.family</code> object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by <code>gamlss()</code> can be found in <code>gamlss.family</code> . Functions such as <code>BI()</code> (binomial) produce a family object.
type	whether left, right or in both sides truncation is required, (left is the default).
...	for extra arguments

## Value

Returns a d family function

## Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

## References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

## See Also

[trun.p](#), [trun.q](#), [trun.r](#), [gen.trun](#)

## Examples

```

# continuous distribution
# left truncation
test1<-trun.d(par=c(0), family="TF", type="left")
test1(1)
dTF(1)/(1-pTF(0))
if(abs(test1(1)-(dTF(1)/pTF(0)))>0.00001) stop("error in left truncation")
test1(1, log=TRUE)
log(dTF(1)/(1-pTF(0)))
if(abs(test1(1, log=TRUE)-log(dTF(1)/pTF(0)))>0.00001) stop("error in left truncation")
#if (!identical(test1(1, log=TRUE), log(dTF(1)/(1-pTF(0)))) stop("error in left truncation")
  integrate(function(x) test1(x, mu=-2, sigma=1, nu=1),0,Inf) # the pdf is defined even with
  integrate(function(x) test1(x, mu=0, sigma=10, nu=1),0,Inf)
  integrate(function(x) test1(x, mu=5, sigma=5, nu=10),0,Inf)
plot(function(x) test1(x, mu=-3, sigma=1, nu=1),0,10)
plot(function(x) test1(x, mu=3, sigma=5, nu=10),0,10)
#-----
# right truncation
test2<-trun.d(par=c(10), family="BCT", type="right")
test2(1)
dBCT(1)/(pBCT(10))
#if(abs(test2(1)-(dBCT(1)/pBCT(10)))>0.00001) stop("error in right truncation")
test2(1, log=TRUE)
log(dBCT(1)/(pBCT(10)))
if(abs(test2(1, log=TRUE)-log(dBCT(1)/(pBCT(10))))>0.00001) stop("error in right truncation")
integrate(function(x) test2(x, mu=2, sigma=1, nu=1),0,10)
integrate(function(x) test2(x, mu=2, sigma=.1, nu=1),0,10)
integrate(function(x) test2(x, mu=2, sigma=.1, nu=10),0,10)
plot(function(x) test2(x, mu=2, sigma=.1, nu=1),0,10)
plot(function(x) test2(x, mu=2, sigma=1, nu=1),0,10)
#-----
# both left and right truncation
test3<-trun.d(par=c(-3,3), family="TF", type="both")
test3(0)
dTF(0)/(pTF(3)-pTF(-3))
if(abs(test3(0)-dTF(0)/(pTF(3)-pTF(-3)))>0.00001) stop("error in right truncation")
test3(0, log=TRUE)
log(dTF(0)/(pTF(3)-pTF(-3)))
if(abs(test3(0, log=TRUE)-log(dTF(0)/(pTF(3)-pTF(-3))))>0.00001) stop("error in both truncati")
plot(function(x) test3(x, mu=0, sigma=1, nu=1),-3,3)
integrate(function(x) test3(x, mu=2, sigma=1, nu=1),-3,3)
#-----
# discrete distribution
# left
# Poisson truncated at zero means zero is excluded
test4<-trun.d(par=c(0), family="PO", type="left")
test4(1)
dPO(1)/(1-pPO(0))
if(abs(test4(1)-dPO(1)/(1-pPO(0)))>0.00001) stop("error in left truncation")
test4(1, log=TRUE)
log(dPO(1)/(1-pPO(0)))
if(abs(test4(1, log=TRUE)-log(dPO(1)/(1-pPO(0))))>0.00001) stop("error in left truncation")

```

```

sum(test4(x=1:20, mu=2)) #
sum(test4(x=1:200, mu=80)) #
plot(function(x) test4(x, mu=20), from=1, to=51, n=50+1, type="h") # pdf
# right truncated at 10 means 10 is included
test5<-trun.d(par=c(10), family="NBI", type="right")
test5(2)
dNBI(2)/(pNBI(10))
if(abs(test5(1)-dNBI(1)/(pNBI(10)))>0.00001) stop("error in right truncation")
test5(1, log=TRUE)
log(dNBI(1)/(pNBI(10)))
if(abs(test5(1, log=TRUE)-log(dNBI(1)/(pNBI(10))))>0.00001) stop("error in right truncation")
sum(test5(x=0:10, mu=2, sigma=2)) #
sum(test5(x=0:10, mu=300, sigma=5)) # can have mu > parameter
plot(function(x) test5(x, mu=20, sigma=3), from=0, to=10, n=10+1, type="h") # pdf
plot(function(x) test5(x, mu=300, sigma=5), from=0, to=10, n=10+1, type="h") # pdf
#-----
# both
test6<-trun.d(par=c(0,10), family="NBI", type="both")
test6(2)
dNBI(2)/(pNBI(10)-pNBI(0))
if(abs(test6(2)-dNBI(2)/(pNBI(10)-pNBI(0)))>0.00001) stop("error in right truncation")
test6(1, log=TRUE)
log(dNBI(1)/(pNBI(10)-pNBI(0)))
if(abs(test6(1, log=TRUE)-log(dNBI(1)/(pNBI(10)-pNBI(0))))>0.00001)
  stop("error in right truncation")
sum(test6(x=1:10, mu=2, sigma=2)) #
sum(test6(x=1:10, mu=100, sigma=5)) # can have mu > parameter
plot(function(x) test6(x, mu=20, sigma=3), from=1, to=10, n=10, type="h") # pdf
plot(function(x) test6(x, mu=300, sigma=.4), from=1, to=10, n=10, type="h") # pdf
#-----

```

trun.p

*Truncated Cumulative Density Function of a gamlss.family Distribution*

## Description

Creates a truncated cumulative density function version from a current GAMLSS family distribution

## Usage

```
trun.p(par, family = "NO", type = c("left", "right", "both"), ...)
```

## Arguments

**par** a vector with one (for left or right truncation) or two elements for both

**family** a `gamlss.family` object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by `gamlss()` can be found in `gamlss.family`. Functions such as `BI()` (binomial) produce a family object.

type            whether left, right or in both sides truncation is required, (left is the default)  
 ...            for extra arguments

### Value

Return a p family function

### Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

### References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

### See Also

[trun.d](#), [trun.q](#), [trun.r](#), [gen.trun](#)

### Examples

```
# truncated p continuous function
# continuous
#-----
# left
test1<-trun.p(par=c(0), family="TF", type="left")
test1(1)
(pTF(1)-pTF(0))/(1-pTF(0))
if(abs(test1(1)-(pTF(1)-pTF(0))/(1-pTF(0)))>0.00001) stop("error in left truncation of p")
plot(function(x) test1(x, mu=2, sigma=1, nu=2),0,10)
#-----
# right
test5 <- trun.p(par=c(10), family="BCT", type="right")
test5(1)
pBCT(1)/pBCT(10)
if(abs(test5(1)-pBCT(1)/pBCT(10))>0.00001) stop("error in right truncation")
test5(1, lower.tail=FALSE)
1-pBCT(1)/pBCT(10)
if(abs(test5(1, lower.tail=FALSE)-(1-pBCT(1)/pBCT(10)))>0.00001) stop("error in right truncation")
test5(1, log.p=TRUE)
log(pBCT(1)/pBCT(10))
if(abs(test5(1, log.p=TRUE)-log(pBCT(1)/pBCT(10)))>0.00001) stop("error in right truncation")
plot(function(x) test5(x, mu=2, sigma=1, nu=2, tau=2),0,10)
plot(function(x) test5(x, mu=2, sigma=1, nu=2, tau=2, lower.tail=FALSE),0,10)
#-----
# both
```

```

test3<-trun.p(par=c(-3,3), family="TF", type="both")
test3(1)
(pTF(1)-pTF(-3))/(pTF(3)-pTF(-3))
if(abs(test3(1)-(pTF(1)-pTF(-3))/(pTF(3)-pTF(-3)))>0.00001) stop("error in right truncation")
test3(1, lower.tail=FALSE)
1-(pTF(1)-pTF(-3))/(pTF(3)-pTF(-3))
if(abs(test3(0, lower.tail=FALSE)-(1-(pTF(0)-pTF(-3))/(pTF(3)-pTF(-3))))>0.00001)
  stop("error in right truncation")
plot(function(x) test3(x, mu=2, sigma=1, nu=2, ),-3,3)
plot(function(x) test3(x, mu=2, sigma=1, nu=2, lower.tail=FALSE),-3,3)
#-----
# Discrete
#-----
# truncated p function
# left
test4<-trun.p(par=c(0), family="PO", type="left")
test4(1)
(pPO(1)-pPO(0))/(1-pPO(0))
if(abs(test4(1)-(pPO(1)-pPO(0))/(1-pPO(0)))>0.00001) stop("error in left truncation of p")
plot(function(x) test4(x, mu=2), from=1, to=10, n=10, type="h")
cdf <- stepfun(1:40, test4(1:41, mu=5), f = 0)
plot(cdf, main="cdf", ylab="cdf(x)", do.points=FALSE )
#-----
# right
test2<-trun.p(par=c(10), family="NBI", type="right")
test2(2)
pNBI(2)/(pNBI(10))
if(abs(test2(2)-(pNBI(2)/(pNBI(10))))>0.00001) stop("error in right truncation of p")
plot(function(x) test2(x, mu=2), from=0, to=10, n=10, type="h")
cdf <- stepfun(0:9, test2(0:10, mu=5), f = 0)
plot(cdf, main="cdf", ylab="cdf(x)", do.points=FALSE )
#-----
# both
test6<-trun.p(par=c(0,10), family="NBI", type="both")
test6(2)
(pNBI(2)-pNBI(0))/(pNBI(10)-pNBI(0))
if(abs(test6(2)-(pNBI(2)-pNBI(0))/(pNBI(10)-pNBI(0)))>0.00001) stop("error in the both trunca
test6(1, log=TRUE)
log((pNBI(1)-pNBI(0))/(pNBI(10)-pNBI(0)))
if(abs(test6(1, log=TRUE)-log((pNBI(1)-pNBI(0))/(pNBI(10)-pNBI(0))))>0.00001) stop("error in
plot(function(y) test6(y, mu=20, sigma=3), from=1, to=10, n=10, type="h") # cdf
plot(function(y) test6(y, mu=300, sigma=.4), from=1, to=10, n=10, type="h") # cdf
cdf <- stepfun(1:9, test6(1:10, mu=5), f = 0)
plot(cdf, main="cdf", ylab="cdf(x)", do.points=FALSE )
#-----

```

**Description**

Creates a function to produce the inverse of a truncated cumulative density function generated from a current GAMLSS family distribution

**Usage**

```
trun.q(par, family = "NO", type = c("left", "right", "both"), ...)
```

**Arguments**

par	a vector with one (for left or right truncation) or two elements for both
family	a <code>gamlss.family</code> object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by <code>gamlss()</code> can be found in <code>gamlss.family</code> . Functions such as <code>BI()</code> (binomial) produce a family object.
type	whether left, right or in both sides truncation is required, (left is the default)
...	for extra arguments

**Value**

Returns a q family function

**Author(s)**

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

**References**

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

**See Also**

[trun.d](#), [trun.q](#), [trun.r](#), [gen.trun](#)

**Examples**

```
# continuous
#-----
# left
test1<-trun.q(par=c(0), family="TF", type="left")
test1(.6)
qTF(pTF(0)+0.6*(1-pTF(0)))
#-----
# right
```

```

test2 <- trun.q(par=c(10), family="BCT", type="right")
test2(.6)
qBCT(0.6*pBCT(10))
#-----
# both
test3<-trun.q(par=c(-3,3), family="TF", type="both")
test3(.6)
qTF(0.6*(pTF(3)-pTF(-3))+pTF(-3))
#-----
# FOR DISCRETE DISTRIBUTIONS
# truncated q function
# left
test4<-trun.q(par=c(0), family="PO", type="left")
test4(.6)
qPO(pPO(0)+0.6*(1-pPO(0)))
#-----
# right
test5 <- trun.q(par=c(10), family="NBI", type="right")
test5(.6)
qNBI(0.6*pNBI(10))
test5(.6, mu=10, sigma=2)
qNBI(0.6*pNBI(10, mu=10, sigma=2), mu=10, sigma=2)
#-----
# both
test6<-trun.q(par=c(0,10), family="NBI", type="both")
test6(.6)
qNBI(0.6*(pNBI(10)-pNBI(0))+pNBI(0))
#-----

```

---

trun.r

*Generates Random Values from a Truncated Density Function of a  
gamlss.family Distribution*

---

### Description

Creates a function to generate random values from a truncated probability density function created from a current GAMLSS family distribution

### Usage

```
trun.r(par, family = "NO", type = c("left", "right", "both"), ...)
```

### Arguments

**par** a vector with one (for left or right truncation) or two elements for both

**family** a `gamlss.family` object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by `gamlss()` can be found in `gamlss.family`. Functions such as `BI()` (binomial) produce a family object.

type	whether left, right or in both sides truncation is required, (left is the default)
...	for extra arguments

**Value**

Returns a r family function

**Author(s)**

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

**References**

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also <http://www.gamlss.com/>).

**See Also**

[trun.p](#), [trun.q](#), [trun.d](#), [gen.trun](#)

**Examples**

```
# truncated r function
# continuous
#-----
# left
test1<-trun.r(par=c(0), family="TF", type="left")
rr<-test1(1000)
hist(rr)
#-----
# right
test2 <- trun.r(par=c(10), family="BCT", type="right")
rr<-test2(1000)
hist(rr)
#-----
# both
test3<-trun.r(par=c(-3,3), family="TF", type="both")
rr<-test3(1000)
hist(rr)
#-----
# discrete
# truncated r function
# left
test1<-trun.r(par=c(0), family="PO", type="left")
tN <- table(Ni <- test1(1000))
r <- barplot(tN, col='lightblue')
#-----
```

```
# right
test2 <- trun.r(par=c(10), family="NBI", type="right")
tN <- table(Ni <- test2(1000))
  r <- barplot(tN, col='lightblue')
tN <- table(Ni <- test2(1000,mu=5))
  r <- barplot(tN, col='lightblue')
tN <- table(Ni <- test2(1000,mu=10, sigma=.1))
  r <- barplot(tN, col='lightblue')
#-----
# both
test3<-trun.r(par=c(0,10), family="NBI", type="both")
tN <- table(Ni <- test3(1000,mu=5))
  r <- barplot(tN, col='lightblue')
```

# Index

## \*Topic **distribution**

- gamlss.tr-package, 1
- gen.trun, 2
- trun, 4
- trun.d, 5
- trun.p, 8
- trun.q, 10
- trun.r, 12

## \*Topic **package**

- gamlss.tr-package, 1

## \*Topic **regression**

- gamlss.tr-package, 1
- gen.trun, 2
- trun, 4
- trun.d, 5
- trun.p, 8
- trun.q, 10
- trun.r, 12

gamlss.tr (*gamlss.tr-package*), 1

gamlss.tr-package, 1

gen.trun, 2, 5, 6, 9, 11, 13

trun, 4

trun.d, 3, 5, 5, 9, 11, 13

trun.p, 3, 5, 6, 8, 13

trun.q, 3, 5, 6, 9, 10, 11, 13

trun.r, 3, 5, 6, 9, 11, 12