In [1]:
```r
print("hello world")
```

[1] "hello world"

Basic data structure is a 'vector'. There are a few ways to create a vectors

In [2]:
```r
x = c(1,3,6)
```

In [3]:
```r
x
```

Out[3]:

1 · 3 · 6

In [4]:
```r
x[2]
```

Out[4]: 3

In [5]:
```r
1:5
```

Out[5]:

1 · 2 · 3 · 4 · 5

In [6]:
```r
5:1
```

Out[6]:

5 · 4 · 3 · 2 · 1

In [8]:
```r
seq(1,10, by=2)
```

Out[8]:

1 · 3 · 5 · 7 · 9

sampling random variables. ex: normal distribution.

In [9]:
```r
?rnorm
```

Out[9]:

Normal {stats}	R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```r
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are *P[X ≤ x]* otherwise, *P[X > x]*.

Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has density

f(x) = 1/(√(2 π) σ) e^-((x - μ)^2/(2 σ^2))

where *μ* is the mean of the distribution and *σ* the standard deviation.

Value

`dnorm` gives the density, `pnorm` gives the distribution function, `qnorm` gives the quantile function, and `rnorm` generates random deviates.

The length of the result is determined by `n` for `rnorm`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

For `sd = 0` this gives the limit as `sd` decreases to 0, a point mass at `mu`. `sd < 0` is an error and returns `NaN`.

Source

For `pnorm`, based on

Cody, W. D. (1993) Algorithm 715: SPECFUN – A portable FORTRAN package of special function routines and test drivers. *ACM Transactions on Mathematical Software* **19**, 22–32.

For `qnorm`, the code is a C translation of

Wichura, M. J. (1988) Algorithm AS 241: The percentage points of the normal distribution. *Applied Statistics*, **37**, 477–484.

which provides precise results up to about 16 digits.

For `rnorm`, see RNG for how to select the algorithm and for references to the supplied methods.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) *Continuous Univariate Distributions*, volume 1, chapter 13. Wiley, New York.

See Also

Distributions for other standard distributions, including `dlnorm` for the *Log*normal distribution.

Examples

```r
require(graphics)

dnorm(0) == 1/sqrt(2*pi)
dnorm(1) == exp(-1/2)/sqrt(2*pi)
dnorm(1) == 1/sqrt(2*pi*exp(1))

## Using "log = TRUE" for an extended range :
par(mfrow = c(2,1))
plot(function(x) dnorm(x, log = TRUE), -60, 50,
     main = "log { Normal density }")
curve(log(dnorm(x)), add = TRUE, col = "red", lwd = 2)
mtext("dnorm(x, log=TRUE)", adj = 0)
mtext("log(dnorm(x))", col = "red", adj = 1)

plot(function(x) pnorm(x, log.p = TRUE), -50, 10,
     main = "log { Normal Cumulative }")
curve(log(pnorm(x)), add = TRUE, col = "red", lwd = 2)
mtext("pnorm(x, log=TRUE)", adj = 0)
mtext("log(pnorm(x))", col = "red", adj = 1)

## if you want the so-called 'error function'
erf <- function(x) 2 * pnorm(x * sqrt(2)) - 1
## (see Abramowitz and Stegun 29.2.29)
## and the so-called 'complementary error function'
erfc <- function(x) 2 * pnorm(x * sqrt(2), lower = FALSE)
## and the inverses
erfinv <- function (x) qnorm((1 + x)/2)/sqrt(2)
erfcinv <- function (x) qnorm(x/2, lower = FALSE)/sqrt(2)
```

[Package *stats* version 3.6.1]

In [10]:
```r
y = rnorm(1000)
```

In [11]:
```r
y
```

Out[11]:

1.75708023217534 · -0.472922251553035 · 1.36473324838116 · 0.301701034419927 · -1.62291237219315 · -0.276848490054476 ·
-0.424877043324269 · 0.115881146398079 · -0.555036069974077 · -1.39246130996118 · -0.29474980224971 · -0.367819686567105 ·
0.939038810122164 · -0.543875285077413 · 1.37848726316564 · -1.38457532711118 · -0.941606014055414 · 0.823130933616846 ·
-0.446535109665506 · -0.567020835625362 · -0.517352665110286 · 0.845059790841002 · 0.341381197017128 · 0.126920289370804 ·
0.118032123982894 · 0.520664233349521 · -1.01823651975713 · 2.61376669161072 · -1.49190245055323 · 0.207806868744911 ·
-0.220733556521028 · 0.174511038963446 · 0.453418190857039 · 0.157866119655613 · 0.270857278374199 · -0.166934952291068 ·
-0.127311547845744 · -2.59614279016931 · 1.49538281616418 · 1.43589533128516 · -0.0333338134697863 · 1.11627611340537 ·
0.598378469510751 · -0.0483217886186 · -0.373591450547762 · 1.39692342021931 · -0.973795983820675 · -1.33387289571712 ·
0.505562685170549 · -1.31774948464202 · -0.162137988325389 · -1.96014311325505 · -1.29831581253101 · 0.706249663639491 ·
-0.0940376975722098 · 1.0815531769211 · -0.398585077444649 · 1.03434712644556 · 0.608369896408536 · -0.672325729731182 ·
0.601028435118091 · -0.77904883015311 · -0.8380114737635 · 0.784179230566884 · -0.954952445874917 · 1.58488812176166 ·
0.0233168738154237 · -0.960671661337233 · 0.736036616899853 · 0.633979657135348 · 0.0806791050309606 · -0.937384775775112 ·
-0.405256435468931 · -1.06240159288451 · -1.07730520024809 · -2.61639783875292 · -0.662741082218592 · 1.6867409258749 ·
1.0813410100035 · -1.38993084389949 · -1.27613049105705 · 0.650583721977513 · 1.41741719370548 · -0.529630260519075 ·
0.00235105490421119 · 0.121500003520571 · 1.35352611565842 · 1.78265811892405 · 0.487134841935969 · -1.17580841392797 ·
-0.450140269249115 · 0.290414612695073 · 0.299154912895126 · -1.70167173107841 · 0.227298995904431 · 0.307188231979846 ·
-0.247353294846483 · -0.851356938244688 · -0.632986204048138 · 1.76931318163378 · 1.31786054985188 · -0.845902824780116 ·
-0.0901455883761352 · 0.0193194387007662 · 0.892383446453958 · 1.75298223998419 · 0.2777025530256 · 0.609052827435193 ·
-0.876538940919775 · -1.77802050333164 · 0.381528396091163 · -0.746262038003389 · -0.659343887591061 · 0.306727342855517 ·
0.679888054168078 · -1.10173676399174 · 0.595669815822562 · -0.341059314775449 · 0.224240951126324 · 0.841779511822225 ·
0.699212713590001 · -0.335640180828712 · 0.76473224206224 · -0.00482744842836181 · 0.64000195961567 · 1.98004533005066 ·
2.29251424642833 · -1.59213510556282 · 1.27349649265339 · 0.066067650011007 · 0.078715992987876 · -0.601704252905694 ·
0.575342019016853 · -0.518259629755606 · 2.19560012936515 · 0.587050550972328 · 0.0568367156597451 · 1.15530698354065 ·
-0.812022037544742 · -0.150788654685758 · -0.826725021917667 · 0.930744004890931 · 1.19515939415592 · 0.150688738460062 ·
1.26726299266499 · 0.0885504248001475 · 0.580083980446058 · -0.211922283455206 · 2.73093761779056 · -0.588361258713122 ·
0.573951820864251 · -0.366781742519017 · 0.850106675693276 · 0.339866739453973 · -1.48807300067469 · 1.33613878801183 ·
0.0919782608166941 · -0.694082527507077 · 1.30515031595776 · -0.26969594046875 · -0.250467318454389 · 2.28760082971581 ·
-1.45567113489381 · -2.30643656089853 · -0.269337936965089 · -0.596653227852353 · 0.837965064012512 · 0.0814063866231444 ·
0.367547411663011 · 0.521250018512507 · -1.9205556187829 · 0.905635459298853 · 0.590008757075195 · -0.644143338694076 ·
0.493822233501936 · 0.0413236761106942 · 1.46799238741839 · -1.4024633572806 · 2.158593821686 · 0.214426265325158 ·
1.61457996919859 · -1.50894207611319 · -0.0741751599483956 · -1.41718640673985 · 2.208652331742 · 0.202068736552205 ·
0.985982178393549 · 0.751579601082118 · 0.41547730189019 · -0.069541728401202 · 0.10162707458115 · 0.965826184306508 ·
-0.850478102461908 · 0.13508061151728 · 0.23719850747539 · 0.0800509646275888 · -1.73436341392006 · 0.18135430938562 ·
-1.27780167321918 · -0.966235681914948 · ⋯ · -0.87668605560055 · -1.38759272542824 · 1.46374785789079 · 1.24679142973345 ·
1.66569888311579 · -0.234693471929191 · 0.336952576617175 · -0.000559327959247255 · -1.20862998526529 · 0.000107770611350725 ·
0.463458899036942 · -0.167032420816823 · 0.988444615167564 · -0.108617131508176 · 0.440970544588608 · -0.344276989980575 ·
0.47561696702679 · 0.49499912440834 · -0.78107266499217 · 0.448016045589501 · 0.341428716785019 · 0.966154247000007 ·
0.14820046592999 · 0.590236588878661 · -0.631017459365196 · -0.887240647388256 · -1.57452979483853 · 0.740204494588522 ·
-1.31948610753852 · -0.625436498025391 · 0.981656730006338 · 0.61998352675648 · 0.0605403529502834 · -0.560788127547777 ·
1.09205225353207 · -0.138097014675494 · -0.818127192065799 · -2.15154159176638 · 1.39773387226736 · -2.23809487322732 ·
-0.202982708700433 · 0.650845677499466 · -1.33700311798119 · -0.0907395009205478 · 1.74186561878432 · -2.49141620269406 ·
3.55284598518224 · -1.59578253956644 · 0.0254525077535446 · 3.43229517969561 · -0.0820674433727965 · -0.99706886896898 ·
0.872913255576474 · -1.7035410936329 · 0.592154689712943 · 0.602063723510227 · 2.91632735655045 · -0.796663260959157 ·
0.270005282256476 · 0.0675426352240251 · 0.988530668133634 · -0.722203195179389 · 0.980146856689706 · 1.10347260813444 ·
-2.39689570937364 · 0.185778204083632 · -0.0391857986283266 · -0.0218829513968317 · -0.595283528230629 · 1.84832165724594 ·
0.222938118547176 · -0.783209239358566 · 1.92402829336316 · -0.858083065223 · -2.71900684341815 · 0.522663375643773 ·
0.0322312154621763 · 0.745477365480472 · -1.37612044852036 · 0.646600203977752 · -0.24943082637038 · 0.788783379219893 ·
0.489197635714229 · 0.807073723039565 · -0.951552542086375 · 0.83984655189687 · 0.0545715786812735 · 1.73675997565552 ·
0.706502143630047 · -0.00468792331555382 · 0.367542154110679 · 0.41222711736266 · -0.487484683697723 · -2.60809399874147 ·
-1.08851400947072 · -1.63686014216183 · -0.759792593311371 · 1.63478691909461 · -0.577523937432809 · 0.331672347353518 ·
1.59924240679198 · 2.03524354109431 · 0.0167747259414836 · 0.309125132040201 · 0.796705594758197 · 1.33919895378198 ·
0.242004043175645 · 0.258843689768752 · -0.784750823156262 · 0.449680995790252 · -0.266649499854558 · -0.905580252704892 ·
0.0957179090693952 · 0.498948652224197 · 1.94509820118269 · -1.43166194347052 · -0.896932842547887 · -1.42574252722375 ·
1.9450910095675 · -0.383971081142665 · 0.188531742189538 · 0.511799487443055 · -0.648142175296639 · -0.52430815590467 ·
-1.07795773911588 · -1.63363743945198 · -0.431386660432526 · -0.704851709080822 · -0.507722939290992 · -2.21746200675973 ·
-1.62803551080839 · -0.471822680511738 · 0.563644741315249 · 0.833160017841024 · 1.98229179575803 · -0.343012986824976 ·
0.115179297033761 · 0.388909165755052 · 0.395614452169547 · -0.338405007222038 · -1.98900158465774 · 1.1175788015803 ·
-2.19838173357806 · 0.701457070138702 · -0.0169529378508091 · -0.102201934909467 · 1.00006680027753 · 0.566161113631215 ·
-1.3929807514483 · -0.291322053063508 · -0.353266455290697 · -0.176465185074209 · 1.43168638940845 · 1.06946399922188 ·
0.593427938705786 · 0.954931472991037 · 0.67538021295531 · -0.957207797733529 · 2.31492322053277 · 0.358209720530272 ·
0.699150534656223 · 0.206307053033449 · 0.75676594152242 · -1.30840949225014 · 1.75738977692296 · -0.515894163558753 ·
-0.0921122426018146 · 1.32406221911224 · 0.0918008983012535 · -2.19798268780607 · 0.677669552030046 · 0.723167436772127 ·
0.202235512057779 · -0.301523790682919 · -0.55592270329706 · 1.43787609460749 · -1.22915202437029 · -0.0659398440487359 ·
-1.2903965975185 · -1.81586341220854 · -0.136464512405556 · -0.297846800362635 · -0.340241014252728 · -0.228707844980931 ·
-1.25045717604226 · 0.86273789031242 · -0.490900570580532 · -1.05761352209152 · 1.49691787040858 · 0.346407919629826 ·
1.95774684049667 · -0.923041319588123 · 1.65635986199948 · 2.24614822062599 · 0.941403892155112 · 0.685875172351946 ·
-0.828998169547084 · -0.278294982126978 · 1.84667053811176 · 0.876372167319415

In [12]:
```r
hist(y)
```

Out[12]:

In [13]:
```r
mean(y)
```

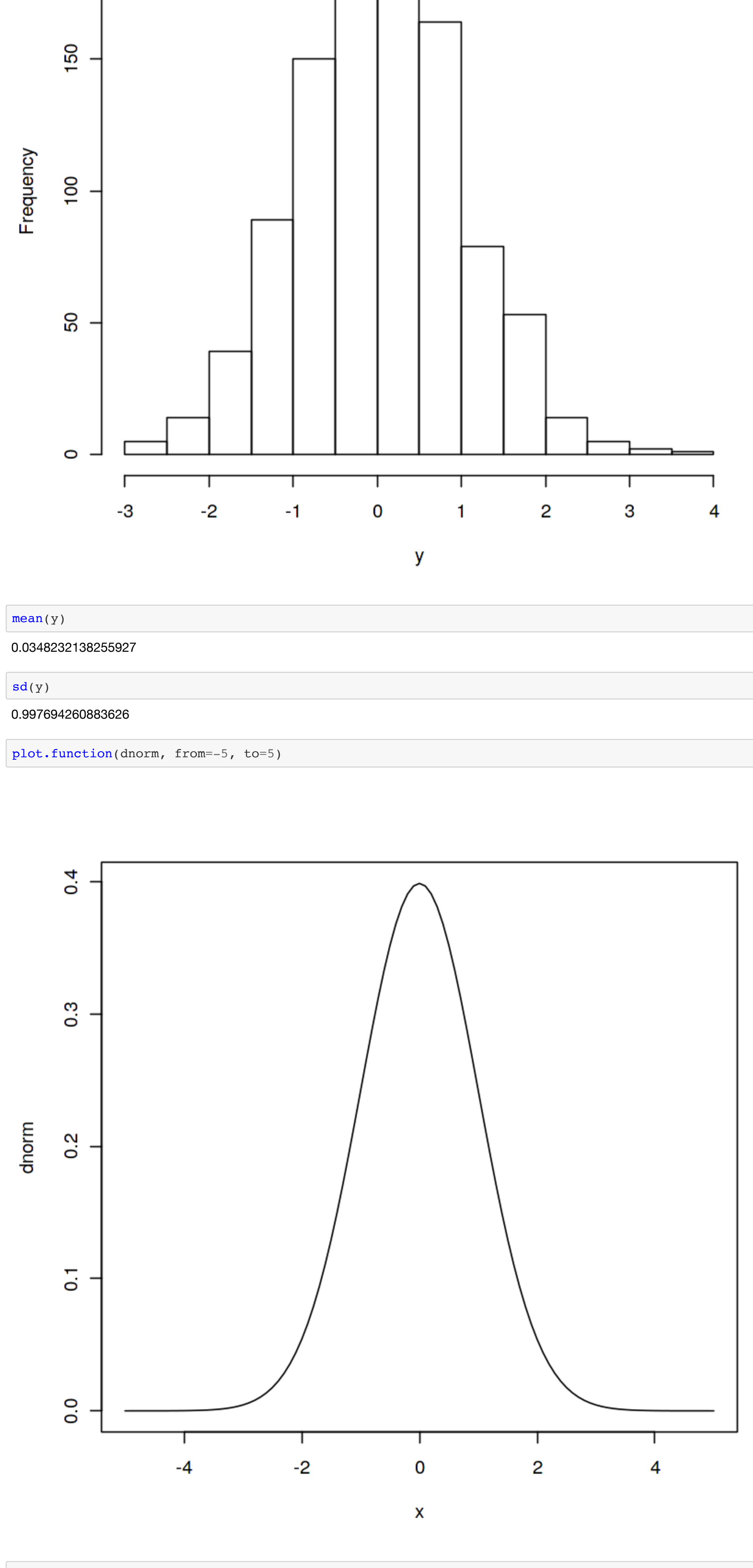
Out[13]: 0.0348232138255927

In [14]:
```r
sd(y)
```

Out[14]: 0.997694260883626

In [15]:
```r
plot.function(dnorm, from=-5, to=5)
```

Out[15]:



In [0]: