

# Package ‘Rlab’

February 19, 2015

**Title** Functions and Datasets Required for ST370 class

**Version** 2.15.1

**Date** Aug. 18, 2012

**Author** Dennis D. Boos, Douglas Nychka

**Maintainer** Dennis Boos <boos@stat.ncsu.edu>

**Depends** R (>= 2.10), utils

**Description** Functions and Datasets Required for ST370 class

**License** GPL (>= 2)

**URL** <http://www.r-project.org>,  
<http://courses.ncsu.edu/ST370/distance/rlab>

**Repository** CRAN

**Date/Publication** 2012-08-22 05:44:15

**NeedsCompilation** no

## R topics documented:

actuator . . . . .	3
airplane . . . . .	4
alka . . . . .	4
Bernoulli . . . . .	5
bplot . . . . .	6
bplot.xy . . . . .	8
bread . . . . .	9
bread2 . . . . .	9
cancer . . . . .	10
capac . . . . .	10
cavendish . . . . .	11
climate . . . . .	11
college . . . . .	12
count . . . . .	13
draft . . . . .	13

drill	14
earthq	15
etruscan	15
Exponential	16
framerate	17
ftp	18
ftptime	18
Gamma	18
golf	20
hplot	21
insulate	22
jet	23
kurt	23
lplot	24
ls.rlab	25
magnet	26
marathon	26
means	27
metalcut	28
mfit	28
michelson	29
monarch	30
mplot	30
ncsu	31
oil	31
ozone	32
ping	32
popcorn	33
precip	34
quake	34
raleigh.snow	35
raleigh.temp	35
randomdata	35
Rlab	36
salt	36
skew	39
solar	40
SpGravity	40
stats	41
stats2	42
US	43
us.age	44
US.dat	44
us.pop	44
view	45
viscosity	45
vocab	46
webhost	46

<i>actuator</i>	3
Weibull . . . . .	47
wire . . . . .	48
world . . . . .	49
world.dat . . . . .	50
xline . . . . .	50
yarn . . . . .	51
yarnred . . . . .	51
yline . . . . .	52
<b>Index</b>	<b>53</b>

---

actuator	<i>Actuator force experiment</i>
----------	----------------------------------

---

### Description

Small propulsion units, called actuators, are used to maneuver space craft once they are in space. In order to control these motions accurately, the actuator needs to produce a precise amount of force. This data set represents an experiment to understand what factors effect the variability of the force produced by an actuator. The actuator is fired using compressed air, and the factors studied are the actuator used (*act*), the amount of pressure used (*press*), the length of the air supply line (*line*) and the nozzle type (*nozzle*).

### Format

A data frame with 16 observations on the following 6 variables.

- act** actuator used (A1 or A2)
- press** amount of pressure used (30psi or 100psi)
- line** length of the air supply line (20ft or 40ft)
- nozzle** nozzle type (rightang or straight)
- force** force produced
- order** experimental order

### Examples

```
bplot(actuator$force,by=actuator$act)
lplot(actuator$act,actuator$force,actuator$press)
anova( lm(force ~ (act+press+nozzle+line)^2, data=actuator) )
```

---

airplane	<i>Distances flown by paper airplanes in an experiment with four treatments.</i>
----------	--

---

### Description

The airplane data frame has 6 rows and 4 columns. Each data point is the distance flown by one of the of 24 airplanes randomly assigned to the four treatments described below.

### Format

A data frame with 24 observations on the following 2 variables.

**distance** : distance flown

**treatment** : one of four treatment values (treat1: no weighting of airplane nose, treat2: one paper clip on the nose, treat3: two paper clips on the nose or treat4: three paper clips on the nose)

### Source

Motivated by a class experiment (but artificial).

### Examples

```
# Make side by side boxplots of the four treatments:
bplot(airplane$distance,airplane$treatment)
```

---

alka	<i>Alka-Seltzer dissolving times</i>
------	--------------------------------------

---

### Description

This data set contains the times in seconds that it takes Alka-Seltzer tablets to dissolve in water and 7UP at two different temperatures.

### Format

A data frame with 8 observations on the following 4 variables.

**liquid** : liquid (7UP or water)

**temp** : temperature (cool or warm)

**time** : time to dissolve (in seconds)

**block** : blocking level for 2x2 factorial design

**Description**

Density, distribution function, quantile function and random generation for the Bernoulli distribution with parameter `prob`.

**Usage**

```
dbern(x, prob, log = FALSE)
pbern(q, prob, lower.tail = TRUE, log.p = FALSE)
qbern(p, prob, lower.tail = TRUE, log.p = FALSE)
rbern(n, prob)
```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>prob</code>	probability of success on each trial.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

The Bernoulli distribution with `prob = p` has density

$$p(x) = p^x(1 - p)^{1-x}$$

for  $x = 0$  or  $1$ .

If an element of `x` is not `0` or `1`, the result of `dbern` is zero, without a warning.  $p(x)$  is computed using Loader's algorithm, see the reference below.

The quantile is defined as the smallest value  $x$  such that  $F(x) \geq p$ , where  $F$  is the distribution function.

**Value**

`dbern` gives the density, `pbern` gives the distribution function, `qbern` gives the quantile function and `rbern` generates random deviates.

**References**

Catherine Loader (2000). *Fast and Accurate Computation of Binomial Probabilities*; manuscript available from <http://cm.bell-labs.com/cm/ms/departments/sia/catherine/dbinom>

**See Also**

[dbinom](#) for the binomial (Bernoulli is a special case of the binomial), and [dpois](#) for the Poisson distribution.

**Examples**

```
# Compute P(X=1) for X Bernoulli(0.7)
dbern(1, 0.7)
```

---

bplot

*boxplot*


---

**Description**

Plots boxplots of several groups of data and allows for placement at different horizontal or vertical positions. It is also flexible in the input object accepting either a list or matrix.

**Usage**

```
bplot(x, by, style = "tukey", outlier = TRUE, plot = TRUE, ...)
```

**Arguments**

<b>x</b>	Vector, matrix, list or data frame. A vector may be divided according to the by argument. Matrices and data frames are separated by columns and lists by components.
<b>by</b>	If x is a vector, an optional vector (either character or numerical) specifying the categories to divide x into separate data sets.
<b>style</b>	Type of boxplot default is "tukey". The other choice is "quantile" where the whiskers are drawn to the 5 and 95 percentiles instead being based on the inner fences.
<b>outlier</b>	If true outliers (points beyond outer fences) will be added to the plots.
<b>plot</b>	If false just returns a list with the statistics used for plotting the box plots.
<b>...</b>	Other arguments controlling the boxplots (passed to <code>bplot.obj</code> ) these are listed below. Other graphical plotting arguments not matched (e.g. <code>yaxt</code> ) are used in the call to plot to set up the initial plot if <code>add=TRUE</code> .

**pos** The boxplots will be plotted vertically and `pos` gives the x or y locations for their centers. If omitted the boxes are equally spaced at integer values.

**width** Width of boxplots (in user coordinates) if omitted then the width is a reasonable fraction of the distance between boxes and is set by the `space` argument.

**labels** Labels under each boxplot. If missing the columns names or components of x are used.

**srt** Rotate the labels (`srt=90` makes them vertical). Default is to put them horizontal. Sometimes long labels run into each if they are horizontal.

**add** If true, do not create a new plots just add the boxplots to a current plot. Note that the pos argument may be useful in this case and should be in the user coordinates of the parent plot.

**space** Space between boxplots.

**sort.names** If true plot the boxplot data set names are sorted in alphabetic order by their labels.

**xlab** Label for the x-axis

**ylab** Label for the y-axis

**label.cex** Boxplot label size where 1.0 is normal size characters. If zero labels will not be added.

**xaxt** Plotting parameter for x-axis generation. Default is not to produce an x-axis.

**horizontal** If true draw boxplots horizontally the default is false, produce vertical box plots.

## Details

This function was created as a complement to the usual S function for boxplots. The current function makes it possible to put the boxplots at unequal x or y positions. This is useful for visually grouping a large set of boxplots into several groups. Also placement of the boxplots with respect to the axis can add information to the plot. Another aspect is the emphasis on data structures for groups of data. One useful feature is the by option to break up the x vector into distinct groups. If 5 or less observations are in a group the points themselves are plotted instead of a box.

The function is broken into two steps: a call to stats.bplot to find the statistics and a call to bplot.obj to plot the resulting object. The user is referred to describe.bplot to modify the statistics used and to draw.bplot.obj to modify how the bplot is drawn.

Finally to bin data into groups based on a continuous variable and to make bplots of each group see bplot.xy.

## See Also

boxplot, bplot.xy, lplot, mplot, plot

## Examples

```
#
set.seed(123)
temp<- matrix( rnorm(12*8), ncol=12)
pos<- c(1:6,9:14)
bplot(temp)
#
bplot( temp, pos=pos, labels=paste( "D",1:12), horizontal=TRUE)
#
bplot( temp, pos=pos, label.cex=0, horizontal=TRUE)
# add an axis
axis( 2)
```

bplot.xy

*Boxplots for conditional distribution***Description**

Draws boxplots for y by binning on x. This gives a coarse, but quick, representation of the conditional distribution of [Y|X] in terms of boxplots.

**Usage**

```
bplot.xy(x, y, N = 10, breaks = pretty(x, N, eps.correct = 1),
         style = "tukey", outlier = TRUE, plot = TRUE, xaxt = "s", ...)
```

**Arguments**

x	Vector to use for bin membership
y	Vector to use for constructing boxplot statistics.
N	Number of bins on x. Default is 10.
breaks	Break points defining bin boundaries. These can be unequally spaced.
style	Type of boxplot default is "tukey". The other choice is "quantile" where the whiskers are drawn to the 5 and 95 percentiles instead being based on the inner fences.
xaxt	Plotting parameter for x-axis generation. Default is to produce an x-axis.
outlier	If true outliers (points beyond outer fences) will be added to the plots.
plot	If false just returns a list with the statistics used for plotting the box plots.
...	Any other optional arguments passed to the bplot.obj function see the help file for bplot for details.

**See Also**

bplot, boxplot

**Examples**

```
# bivariate normal corr= .6
set.seed( 123)
x<-rnorm( 1000)
y<- .6*x + sqrt( 1- .6**2)*rnorm( 1000)
#
#
bplot.xy( x,y, breaks=seq( -3, 3,,15) ,xlim =c(-4,4), ylim =c(-4,4))
points( x,y, pch=".", col=3)
```



---

bread	<i>Bread rising experiment</i>
-------	--------------------------------

---

**Description**

The data set bread contains height measurements of 48 cupcakes. A batch of Hodgson Mill Wholesome White Bread mix was divided into three parts and mixed with 0.75, 1.0, and 1.25 teaspoons of yeast, respectively. Each part was made into 8 different cupcakes and baked at 350 degrees. After baking, the height of each cupcake was measured. Then the experiment was repeated at 450 degrees.

**Format**

A data frame with 48 observations on the following 3 variables.

**yeast** : quantity of yeast (.75, 1 or 1.25 teaspoons)

**temp** : baking temperature (350 or 450 degrees)

**height** : cupcake height

---

bread2	<i>Bread rising experiment</i>
--------	--------------------------------

---

**Description**

The data set bread2 contains averaged measurements from the full data set, bread. The 8 cupcakes in each temp/yeast combination have been averaged.

**Format**

A data frame with 48 observations on the following 3 variables.

**yeast** : quantity of yeast (.75, 1 or 1.25 teaspoons)

**temp** : baking temperature (350 or 450 degrees)

**height** : cupcake height

---

cancer	<i>Cigarette smoking and mortality</i>
--------	--

---

**Description**

The data set cancer examines a relationship between lung cancer and cigarette smoking. The data consist of , a standardized measure of smoking amount (smoke) and the standardized mortality ratio (SMR) for males in England and Wales in 1970-72 who were working in 25 different broad groups of jobs such as textile workers, miners, etc.

**Format**

A data frame with 25 observations on the following 2 variables.

**smoke** : standardized measure of smoking amount

**SMR** : standardized mortality ratio

**Source**

A Handbook of Small Data Sets by Hand, et al. (1994, p.67).

---

capac	<i>Capacitance of different shaped capacitors.</i>
-------	--

---

**Description**

The capac data set measures the capacitance of a capacitor built with one of 5 shapes and 3 different sizes (area). Other covariates variables are perimeter length and number of discontinuities.

**Format**

A data frame with 15 observations on the following 5 variables.

**capac** : measured capacitance

**shape** : shape of the capacitor

**perim** : perimeter length of the capacitor

**area** : size of the capacitor

**discont** : number of discontinuities

**Examples**

```
# Make a means plot of capacitance by shape and area.  
mplot(capac$capac, capac$shape, capac$area, both=TRUE)
```

cavendish

*Cavendish's 1798 determinations of the density of the earth***Description**

Newton's law of gravitation states that the forces of attraction ( $f$ ) between two particles of matter is given by the formula  $f=mm'/(r^{**2})$ , where  $m$  and  $m'$  are their respective masses,  $r$  the distance between their centers of gravity, and  $G$  is the gravitational constant, independent of the kind of matter or intervening medium. From the late eighteenth through nineteenth centuries, a large number of experiments were performed in order to determine  $G$ . These experiments were usually designed to determine the earth's attraction of masses and described as experiments to determine the mean density of the earth: if the earth is supposed spherical with radius  $R$  and  $g$  is the acceleration toward the earth due to gravity, then Newton's law becomes  $dG=3g/(4(\pi)R)$ , where  $d$  is the mean density ( $g/ccm$ ) of the earth. Since  $g$  and  $R$  could be supposed known, determination of  $d$  could be viewed as equivalent to determination of  $G$ .

Of all these early experiments, that of Cavendish, performed in 1798 using a torsion balance devised by Michell, is generally considered the best. The completeness of his description of his experiments and the excellence of his methods are often described as an ideal example of scientific experimentation. Cavendish concluded his memoir by presenting 29 determinations of the mean density of the earth. After the 6th of these determinations, Cavendish changed his experimental apparatus by replacing a suspension wire by one that was stiffer. Another interesting feature of the data is that Cavendish calculated the sample mean incorrectly: somehow he used 5.88 instead of 4.88 for the 3rd value. This was first noticed by Baily in 1843 but overlooked by Laplace's analysis of the data in 1820. The "true value" of  $d$  is 5.517 (1977 Encyclopedia Britannica).

The data and above description were taken from Stigler (1977, *The Annals of Statistics*, p. 1055-1098) who obtained it from *The Laws of Gravitation* edited by A. S. Mackenzie.

**Format**

A numeric vector with 29 values.

**Examples**

```
plot(cavendish)
```

climate

*Climate and geographical data for 50 of the largest US cities.***Description**

This data frame contains information about 50 of the largest US cities, including location, rainfall, temperature and elevation.

**Format**

A data frame with 50 observations on the following 7 variables.

**lat** : latitude

**jan** : average minimum January temperature (degrees F)

**rain** : average rainfall in inches

**city** : city names

**jul** : average maximum July temperature

**elev** : elevation above sea level in KW

**lon** : longitude

**References**

The Universal Almanac (1992), ed. John W. Wright, Andrews and McNeel, Kansas City.

---

college

*Statistics on colleges in 15 states.*

---

**Description**

The data frame college contains statistics relating to colleges from 15 states. This is a sample of fifteen states and certain statistics taken from the Chronicle of Higher Education (most data is for 1992). All entries are in thousands so that Arkansas (first row) has a population of 2,399,000, a yearly per capita income of \$15,400, 85,700 undergraduates students, 7,000 graduate students, and average cost of tuition and fees at public universities of \$1,540, and is located in the south (s for south).

**Format**

A data frame with 15 observations on the following 7 variables (all data in thousands).

**school:** State in which school is located.

**pop:** State population.

**inc:** Yearly per capita income.

**undergrad:** Total number of undergraduate students.

**graduate:** Total number of graduate students.

**fees:** Average cost of tuition and fees.

**loc:** Area of the country (s for south, w for west, ne for northeast, mw for midwest).

---

count	<i>Counts elements which meet specified conditions</i>
-------	--

---

**Description**

Count the number of times the values in the vector meet the specified conditions.

**Usage**

```
count(x)
```

**Arguments**

x                      Vector and condition to count.

**See Also**

length, nchar

**Examples**

```
set.seed(1)
x <- rnorm(100)

# Count the number of times the values in x are greater than 0
count( x>0 )

# Count the number of times the values in x are within the 95% confidence interval
count( (x>-1.96) & (x<1.96) )
# Or could have used
count( abs(x)<1.96 )

# Count the number of times the values in x are the same as the first element
count( x==x[1] )
```

---

draft	<i>1970 Vietnam draft lottery summary</i>
-------	---

---

**Description**

The data set `draft` contains average lottery numbers by month for the 1970 Draft Lottery. In December of 1969 the U.S. randomly drew from the 366 possible birthdays without replacement. The draw order of each birthday determined the order by which men born between 1944-1950 (those eligible in the 1970 draft) were drafted. For example, a person with a birthday lottery number of 63 was drafted fairly early in 1970; a person with number 300 was not drafted at all. Sommers (2003, *Chance Magazine*) looked up deaths by age and birthday on the Vietnam Veterans Memorial. Thus, the data set has deaths by month as well.

**Format**

A data frame with 12 observations on the following 4 variables.

**month** : Month of the birthday

**lottnum** : Average lottery number of all birthdays in the month

**deaths** : Total number of deaths by month

**order** : breaks months into 2 groups (first for Jan-June and second for July-Dec)

**References**

Death statistics available on-line at <http://thewall-usa.com/>.

---

drill

*Drill testing results*

---

**Description**

The data set `drill` contains the results of testing two types of drill bits in the manufacture of compressors. There were two brands considered (Besley and Cleveland), and the measurements are the number of holes drilled until the bit breaks. The tests were done under the same manufacturing conditions, and the influence on performance due to factors other than the brand was minimized.

**Format**

A data frame with 14 observations on the following 3 variables.

**brand** : drill manufacturer (Beasly or Cleveland)

**holes** : number of holes drilled before break

**price** : price of a bit

**Examples**

```
lplot(drill$brand,drill$price/drill$holes,  
main='Price per Hole for Drill Bits',ylab='Price per Hole')
```

---

earthq	<i>Earthquake data</i>
--------	------------------------

---

**Description**

The data set earthq includes the dominant frequency and magnitude of 148 earthquakes.

**Format**

A data frame with 148 observations on the following 5 variables.

**site** : location of the earthquake

**freq** : dominant frequency

**mag** : magnitude

**depth** : depth

**dist** : distance

**Source**

Earthquake Engineering and Structural Dynamics, Vol. 23, p. 583-597, 1994

---

etruscan	<i>Skull widths for ancient Etruscans and modern Italians</i>
----------	---

---

**Description**

The data set etruscan contains the maximum width for 84 skulls of Etruscan males and 70 modern Italian males. This data was gathered in attempt to determine if Etruscans were native Italians or immigrants from another land.

**Format**

A data frame with 154 observations on the following 2 variables.

**width** : skull width (in mm)

**group** : ancient or modern

**Source**

Medical Biology and Etruscan Origins, p. 136

**Description**

Density, distribution function, quantile function and random generation for the exponential distribution with mean beta or 1/rate).

This special Rlab implementation allows the parameter beta to be used, to match the function description often found in textbooks.

**Usage**

```
dexp(x, rate = 1, beta = 1/rate, log = FALSE)
pexp(q, rate = 1, beta = 1/rate, lower.tail = TRUE, log.p = FALSE)
qexp(p, rate = 1, beta = 1/rate, lower.tail = TRUE, log.p = FALSE)
rexp(n, rate = 1, beta = 1/rate)
```

**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.
beta	vector of means.
rate	vector of rates.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

If beta (or rate) is not specified, it assumes the default value of 1.

The exponential distribution with rate  $\lambda$  has density

$$f(x) = \lambda e^{-\lambda x}$$

for  $x \geq 0$ .

**Value**

dexp gives the density, pexp gives the distribution function, qexp gives the quantile function, and rexp generates random deviates.

**Note**

The cumulative hazard  $H(t) = -\log(1 - F(t))$  is `-pexp(t, r, lower = FALSE, log = TRUE)`.



## References

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

## See Also

[exp](#) for the exponential function, [dgamma](#) for the gamma distribution and [dweibull](#) for the Weibull distribution, both of which generalize the exponential.

## Examples

```
dexp(1) - exp(-1) #-> 0
```

---

framerate

*Video card frame rates*

---

## Description

The data set `framerate` contains processor speed, memory size, and screen resolution for Riva TNT video cards. The frame rates for these cards were measured using Quake II, a standard benchmarking program for 3D graphics.

## Format

A data frame with 36 observations on the following 6 variables.

**processor** : processor (Celeron 333 or Pentium II 450)

**memory** : memory size (64, 128 or 256 kB)

**resolution** : screen resolution (640x480, 800x600 or 1024x768)

**rate** : frames per second

**pixels** : total number of screen pixels

**pr01** : 0 for Celeron, 1 for Pentium

---

ftp	<i>File transfer times</i>
-----	----------------------------

---

**Description**

Several students studied the relationship between file size and transfer times using ftp (File Transfer Protocol) to retrieve files from two Internet locations. At each location three different files were transferred 5 times and averaged (to reduce variability).

**Format**

A data frame with 6 observations on the following 3 variables.

**size** : file size (in bytes)

**time** : transfer time (in seconds)

**loc** : internet location (0 or 1)

---

ftptime	<i>Forty FTP times</i>
---------	------------------------

---

**Description**

The data in ftptime are 40 ftp times for a file of 343285 bytes which was repeatedly obtained from a site in California.

**Format**

A numeric vector with 40 values.

---

Gamma	<i>The Gamma Distribution</i>
-------	-------------------------------

---

**Description**

Density, distribution function, quantile function and random generation for the Gamma distribution with parameters alpha (or shape) and beta (or scale or  $1/\text{rate}$ ).

This special Rlab implementation allows the parameters alpha and beta to be used, to match the function description often found in textbooks.

**Usage**

```

dgamma(x, shape, rate = 1, scale = 1/rate, alpha = shape,
       beta = scale, log = FALSE)
pgamma(q, shape, rate = 1, scale = 1/rate, alpha = shape,
       beta = scale, lower.tail = TRUE, log.p = FALSE)
qgamma(p, shape, rate = 1, scale = 1/rate, alpha = shape,
       beta = scale, lower.tail = TRUE, log.p = FALSE)
rgamma(n, shape, rate = 1, scale = 1/rate, alpha = shape,
       beta = scale)

```

**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.

`rate`                an alternative way to specify the scale.

`alpha`, `beta`        an alternative way to specify the shape and scale.

`shape`, `scale`      shape and scale parameters.

`log`, `log.p`        logical; if TRUE, probabilities `p` are given as `log(p)`.

`lower.tail`        logical; if TRUE (default), probabilities are  $P[X \leq x]$ , otherwise,  $P[X > x]$ .

**Details**

If `beta` (or `scale` or `rate`) is omitted, it assumes the default value of 1.

The Gamma distribution with parameters `alpha` (or `shape`) =  $\alpha$  and `beta` (or `scale`) =  $\sigma$  has density

$$f(x) = \frac{1}{\sigma^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\sigma}$$

for  $x > 0$ ,  $\alpha > 0$  and  $\sigma > 0$ . The mean and variance are  $E(X) = \alpha\sigma$  and  $Var(X) = \alpha\sigma^2$ .

`pgamma()` uses algorithm AS 239, see the references.

**Value**

`dgamma` gives the density, `pgamma` gives the distribution function `qgamma` gives the quantile function, and `rgamma` generates random deviates.

**Note**

The S parametrization is via `shape` and `rate`: S has no `scale` parameter.

The cumulative hazard  $H(t) = -\log(1-F(t))$  is `-pgamma(t, ..., lower = FALSE, log = TRUE)`.

`pgamma` is closely related to the incomplete gamma function. As defined by Abramowitz and Stegun 6.5.1

$$P(a, x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt$$

$P(a, x)$  is `pgamma(x, a)`. Other authors (for example Karl Pearson in his 1922 tables) omit the normalizing factor, defining the incomplete gamma function as `pgamma(x, a) * gamma(a)`.

## References

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

Shea, B. L. (1988) Algorithm AS 239, Chi-squared and Incomplete Gamma Integral, *Applied Statistics (JRSS C)* **37**, 466–473.

Abramowitz, M. and Stegun, I. A. (1972) *Handbook of Mathematical Functions*. New York: Dover. Chapter 6: Gamma and Related Functions.

## See Also

[gamma](#) for the Gamma function, [dbeta](#) for the Beta distribution and [dchisq](#) for the chi-squared distribution which is a special case of the Gamma distribution.

## Examples

```
-log(dgamma(1:4, alpha=1))
p <- (1:9)/10
pgamma(qgamma(p, alpha=2), alpha=2)
1 - 1/exp(qgamma(p, alpha=1))
```

---

golf

*Golf performance data*

---

## Description

The data set `golf` was taken from PGA Tour Records of 195 golf rounds by PGA players in an attempt to explain what golf attributes contribute the most to low scores.

## Format

A data frame with 195 observations on the following 7 variables.

**score** : score on the 18 holes (par 72)

**distance** : average distance of drive on two holes in opposite direction (to balance out wind)

**accur** : percentage of times that dirve was in the fairway for par 4 and par 5 holes

**putts** : average number of putts in the round for holes where the green was hit in regulation

- sand** : an estimate of sand trap play accuracy  
based on the residuals from regressing percentage  
of successful pars from traps on putts
- chip** : based on the residuals from regressing the score  
on par three holes on putts, sand, and chip
- irons** : an estimate of chipping accuracy  
based on the residuals from regressing percentage  
of successful pars on holes not hit in regulation  
on putts

### Source

"Drive for Show and Putt for Dough" by Scott M. Berry,  
Chance, Vol. 12, No. 4, p. 50-55, 1999

---

hplot

*Histogram allowing forced number of bins*

---

### Description

Plots a histogram in the same manner as `hist`, but with the following changes: `freq = FALSE` by default, to print the density instead of the frequency and `nclass` specifies the exact number of bins to use (calculated by equally separating the distance between the min and max value to be graphed)

### Usage

```
hplot(x, breaks = "Sturges", freq = FALSE, nclass = NULL, col = 8, ...)
```

### Arguments

- |                     |   |
|---------------------|---|
| <code>x</code>      | a vector of values for which the histogram is desired.  |
| <code>breaks</code> | see <code>hist</code> for the use of this option. If both <code>breaks</code> and <code>nclass</code> are specified, then <code>breaks</code> is ignored.   |
| <code>freq</code>   | logical; if 'FALSE' (default), relative frequencies ("probabilities"), component 'density', are plotted; if 'TRUE', the histogram graphic is a representation of frequencies, the 'counts' component of the result. |
| <code>nclass</code> | numeric (integer); the number of bins for the histogram. If both <code>breaks</code> and <code>nclass</code> are specified, then <code>breaks</code> is ignored.  |
| <code>col</code>    | color of the histogram bars (8, the default, is grey).  |
| <code>...</code>    | Other arguments controlling the plot. Many graphical plotting arguments may be used. See help on <code>hist</code> or <code>plot</code> or <code>par</code> for more information.                                   |

### See Also

`hist`, `plot`

### Examples

```
# Create and graph some Normal data
set.seed(100)
set.panel(3,1)
z<- rnorm(100)
hplot(z, nclass=5, main="Standard Normal", xlim=c(-10,10), ylim=c(0,.4))
z<- rnorm(100, sd=2)
hplot(z, nclass=10, main="Std Dev of 2", xlim=c(-10,10), ylim=c(0,.4))
z<- rnorm(100, sd=3)
hplot(z, nclass=15, main="Std Dev of 3", xlim=c(-10,10), ylim=c(0,.4))
```

---

insulate

*Effects of house insulation*

---

### Description

The data set `insulate` is one person's record of weekly gas consumption (gas) and outside temperature (temp), before (insulation=0) and after (insulation=1) insulating a house. The house thermostat was set at 20 degrees Celsius during the 26 weeks before and 30 weeks after insulating.

### Format

A data frame with 56 observations on the following 3 variables.

**insulation** : before insulation (0) or after (1)

**temp** : outside temperature (in degrees Celsius)

**gas** : gas consumption (in 1000 cubic feet)

### Source

A Handbook of Small Data Sets

---

jet	<i>Jet (actuator) force experiment</i>
-----	--

---

**Description**

This data set is a subset of the actuator data set without the line or nozzle factors.

**Format**

A data frame with 16 observations on the following 4 variables.

**act** actuator used (A1 or A2)

**press** amount of pressure used (30psi or 100psi)

**force** force produced

**order** experimental order

**Examples**

```
bplot(jet$force,by=jet$act)
mplot(jet$force,jet$act,jet$press,both=TRUE)
anova( lm(force ~ act+press+act:press, data=jet) )
```

---

kurt	<i>4th moment kurtosis ratio</i>
------	----------------------------------

---

**Description**

4th moment kurtosis ratio

**Usage**

```
kurt(x)
```

**Arguments**

x                    vector

**See Also**

skew

**Examples**

```
set.seed(1)
x <- rexp(100)

# Get kurtosis coefficient estimate for exponential distribution
kurt(x)
```

lplot

*Label plot***Description**

Plots x versus y with optional labels. The x or y variable may be a character vector, but not both.

**Usage**

```
lplot(x, y, labels = "*", srt = 0, tcex = 0.7, ...)
```

**Arguments**

x	Vector to be graphed on x-axis. May be a character vector, if y is numeric.
y	Vector to be graphed on y-axis. May be a character vector, if x is numeric.
labels	Character vector containing the labels for individual points.
srt	A numerical value specifying (in degrees) how strings should be rotated. It is unwise to expect values other than multiples of 90 to work. See help on par for more information.
tcex	A numerical value giving the amount by which the labels text or symbols should be scaled relative to the default.
...	Other arguments controlling the plot. Many graphical plotting arguments may be used. See help on plot or par for more information.

**See Also**

plot, bplot, boxplot, mplot

**Examples**

```
# Create some Normal data
set.seed(123)
temp<- data.frame(matrix(rnorm(12*8), ncol=12))
pos<- c(1:6,9:14)
lplot(temp)

# Now see some labels
lplot(temp, labels=paste("Y",1:12), tcex=.5)

# Create a data set with two factors (age and gender)
race<-data.frame(c(1.02,.99,1.11,1.30,1.09,1.26,1.21,1.19,1.30,1.45,1.34,1.49),
                 c('M','M','M','M','M','M','F','F','F','F','F','F'),
                 c('under 50','under 50','under 50','over 50','over 50','over 50',
                   'under 50','under 50','under 50','over 50','over 50','over 50'))
names(race)<-c("time","gender","age")
```



```
# Plot the data to see the factors
lplot(race$gender, race$time, race$age)
```

---

ls.rlab *List of objects in Rlab*

---

### Description

List the objects in Rlab. By default the Rlab datasets are listed, however "functions" or "all" can be specified to list only the Rlab functions or everything in Rlab.

### Usage

```
ls.rlab(what="data")
ls.summary.rlab(what="data")
```

### Arguments

what	character string specifying which Rlab object to list, which may be one of "data" or "d" (default) : lists datasets "functions" or "f" : lists functions "all" or "a" : lists everything "ex" or "e" : lists the files which can be viewed with the ex method
------	---

### Details

The `ls.summary.rlab` function will list various object attributes, such as class and size.

### See Also

ls, search

### Examples

```
# list all Rlab datasets and their sizes
ls.summary.rlab()

# list all Rlab functions
ls.rlab("functions")
```

---

magnet	<i>Magnetic force of an electromagnet as a function of voltage and number of wire turns.</i>
--------	--

---

### Description

The magnet dataset is from an experiment concerning the magnetic force of an electromagnet as a function of voltage and number of wire turns. The device was a wire wrapped around a core and measured at a variety of voltages. The statistical design here is actually a randomized complete block design where the three electromagnets are the blocks, and the three voltages are levels of the factor voltage.

### Format

**volt:** Voltage applied (1.5 or 3.0 volts).

**turns:** The number of wire turns (100, 200, or 300, as factors).

**force:** The magnetic force.

---

marathon	<i>Winning times from New York City marathon</i>
----------	--

---

### Description

The data consists of the winning times (in minutes) for men and women at the New York Marathon, 1978-1998, along with the temperature in Fahrenheit.

### Format

A data frame with 21 observations on the following 4 variables.

**year** : 4-digit year

**temp** : temperature (in Fahrenheit)

**mtime** : men's winning time (in seconds)

**wtime** : women's winning time (in seconds)

### Source

"The Effects of Temperature on Marathon Runners' Performance" by David E. Martin and John F. Buoncristiani, *Chance*, Vol. 12, No. 4, 1999

---

means	<i>Computes one-way and two-way means tables</i>
-------	--

---

### Description

Calculates means for individual factors and two-way factor combinations. Any number of factors may be input and the individual factor means as well as all possible two-way means will be shown for each factor. Three-way, four-way, etc. means are not shown, even when more than 2 factors are given.

### Usage

```
means(y, ..., dec = 3)
```

### Arguments

y	Vector of responses whose means are shown.
...	Vectors of independent variables on which the responses' means are broken down.
dec	Number of decimal places to print.

### See Also

mplot, mean

### Examples

```
# Create a data set with two factors (age and gender)
race<-data.frame(c(1.02,.99,1.11,1.30,1.09,1.26,1.21,1.19,1.30,1.45,1.34,1.49),
                 c('M','M','M','M','M','M','F','F','F','F','F','F'),
                 c('under 50','under 50','under 50','over 50','over 50','over 50',
                   'under 50','under 50','under 50','over 50','over 50','over 50'))
names(race)<-c("time","gender","age")

# Show mean times broken by age, gender and age & gender
means(race$time, race$age, race$gender)
```

---

 metalcut

*Metal cutting performance*


---

### Description

The data set metalcut is an attempt to determine which cutting method (vertical or horizontal) yields the quickest and smoothest cuts for three types of metal stock (angle, flat, round). Students from the Biological Engineering Department measured cutting times and quality for the six combinations of method and stock.

### Format

A data frame with 18 observations on the following 4 variables.

**cut** : cutting method (hcut for horizontal or vcut for vertical)

**stock** : type of metal stock (angle, flat or round)

**time** : cutting time (in seconds)

**quality** : smooth (0) or rough (1)

---

 mfit

*Computes main and interaction fitted effects*


---

### Description

Calculates main fitted effects for individual factors and two-way interaction fitted effects for all pairs of factors. Any number of factors may be input. Three-way, four-way, etc. fitted effects are not shown, even when more than two factors are given.

### Usage

```
mfit(y, ..., dec = 3)
```

### Arguments

y	Vector of responses whose fitted effects are shown.
...	Vectors of different factors.
dec	Number of decimal places to print.

### See Also

lm, fitted

**Examples**

```
# Create a data set with two factors (age and gender)
race<-data.frame(c(1.02,.99,1.11,1.30,1.09,1.26,1.21,1.19,1.30,1.45,1.34,1.49),
                 c('M','M','M','M','M','M','F','F','F','F','F','F'),
                 c('under 50','under 50','under 50','over 50','over 50','over 50',
                   'under 50','under 50','under 50','over 50','over 50','over 50'))
names(race)<-c("time","gender","age")

# Show fitted effects for age, gender and age & gender
means(race$time, race$age, race$gender)
```

---

michelson

*Results from Michelson's determination of the velocity of light in air.*


---

**Description**

These data are actually measurements obtained by Michelson between June 5, 1879, and July 2, 1879. The data are in km/sec if 299000 is added to each value. Working backwards from the current 'true value' of the velocity of light in vacuum (299,792.5 km/sec) and using Michelson's own adjustment for the effect of air, the comparable 'true value' for these data is 734.5 (considerably smaller than the actual measurements). Michelson used a modification of Foucault's 1850 experimental method which consisted of passing light from a source off a rapidly rotating mirror to a distant fixed mirror, and back to the rotating mirror. Presumably the five sets of 20 measurements are in time sequence. From Stigler (1977 *Annals of Statistics*, p.1073-1076, Table 6).

**Format**

A data frame with 100 observations on the following 2 variables.

**velocity** : measured velocity of light as described above

**set** : the set in which the measurement was taken

**Examples**

```
lplot(michelson$velocity,michelson$set)
bplot(michelson$velocity,michelson$set)
```

---

monarch	<i>Post coronation lifespan</i>
---------	---------------------------------

---

**Description**

The data set monarch contains the years lived after inauguration, election, or coronation of popes, U.S. presidents, and British monarchs from 1690 to 1970.

**Format**

A data frame with 72 observations on the following 3 variables.

**group** : group (K&Qs, popes or pres)

**years** : year lived after coronation, inauguration or election

**name** : name of the monarch, pope or president (no spaces)

**Source**

*Computer-Active Data Analysis* by Lunn and McNeil (1991)

---

mplot	<i>Plots factor means</i>
-------	---------------------------

---

**Description**

Graphs means for two-way factor combinations (interaction plots). Any number of factors may be included and all possible two factor combinations will be plotted.

**Usage**

```
mplot(y, ..., both = FALSE)
```

**Arguments**

y	Vector of responses whose means are graphed.
...	Vectors of independent variables on which the responses' means are broken down.
both	If TRUE, creates additional plots with the opposite factor on the x-axis.

**See Also**

interaction.plot, means, mean

## Examples

```
# Create a data set with three factors (age, gender and number of water breaks)
race<-data.frame(c(1.02,.99,1.11,1.30,1.09,1.26,1.21,1.19,1.30,1.45,1.34,1.49),
  c('M','M','M','M','M','M','F','F','F','F','F','F'),
  c('under 50','under 50','under 50','over 50','over 50','over 50',
    'under 50','under 50','under 50','over 50','over 50','over 50'),
  c(1,0,2,2,0,1,2,1,0,2,1,0))
names(race)<-c("time","gender","age","water")

# Show mean times broken by age, gender and age & gender
mplot(race$time, race$age, race$gender)

# Show 2 plots, with age and then gender along the x-axis
mplot(race$time, race$age, race$gender, both=TRUE)

# Now also consider water breaks
mplot(race$time, race$age, race$gender, race$water, both=TRUE)

# Print the means for the above plots
means(race$time, race$age, race$gender, race$water)
```

---

 ncsu

*Number of degrees granted at ncsu by year.*


---

## Description

The ncsu data frame has 92 rows and 2 columns. It gives the number of degrees granted at North Carolina State University (NCSU) from 1894 to 1983.

## Format

A data frame with 92 observations on the following 2 variables.

**year** : 4 digit year

**degree** : number of degrees granted

---

 oil

*Synthetic versus Conventional Oil*


---

## Description

Synthetic versus Conventional Oil

**Format**

A data frame with 8 observations on the following 3 variables.

**type** : type of oil (conv or syn)

**vis** : viscosity

**time** : time

---

ozone	<i>Monthly mean concentrations of ozone.</i>
-------	--

---

**Description**

The ozone data frame has 552 rows and 3 columns. The first column is the mean monthly ozone concentration in Dobson units of the ozone layer at Arosa, Switzerland, from 1926 to 1971.

**Format**

A data frame with 518 observations on the following 3 variables.

**ozone** : mean monthly ozone in Dobson units

**year** : year in which measurements were taken (4 digits)

**month** : month in which measurements were taken (3 letter abbreviation)

---

ping	<i>Internet ping times</i>
------	----------------------------

---

**Description**

In Spring 2000 a team measured ping times of Internet servers at various distances from Raleigh using a software program called NeoTrace. They actually measured ping times at 4 different times of the day, but since there was very little difference over time, we have averaged over the times of day.

**Format**

A data frame with 12 observations on the following 2 variables.

**dist** : distance from Raleigh

**time** : ping response time



---

popcorn

*A 2x4 experiment to evaluate popcorn preparation.*

---

### Description

The popcorn data frame has 16 rows and 6 columns. Four factors are varied to see what produces the largest volume of popcorn. The order that the observations were made is in the column order.

### Format

A data frame with 16 observations on the following 5 variables.

**brand** : brand of popcorn used (Orville Reddenbach or Jolly Time)

**temp** : temperature of oil (hot or cold)

**quantity** : quantity of oil used (3 Tsp or 1 Tsp)

**shake** : whether the pan was shaken (yes or no)

**volume** : volume of popcorn produced

### Source

This experiment was designed and carried out by Stan Spencer for the NCSU statistics class ST516. Here is an excerpt of his report:

#### INTRODUCTION

Popcorn has always been a crucial element of sustenance in my life and I've always wondered what effects certain factors have in the making of a good batch. Now, having acquired some of the basic tools of statistical experimentation I have been able to optimize my frequent ritual of popping popcorn as well as understand exactly how much of an effect these factors have on the desired outcome. The purpose of the experiment was to optimize the factors involved for the maximum volume of popcorn. I focused on stovetop popping and didn't look at microwave or air-pop methods.

#### DESIGN AND TEST CONDITIONS

I chose the two major popcorn brands with the motive of trying to prove if Orville's claims are true. The second factor refers to the temperature of the oil at the time the popcorn was put in the pan. For the experiments requiring cold oil I added the popcorn to the oil before putting the pan on the stove. For the hot oil treatments I let it heat for 20 seconds before adding the popcorn. The quantity of oil factor required either one or three tablespoons. The last factor I thought was important was to either shake or not shake the pan during cooking. The conditions of the test that were kept constant for each treatment are: 1/2 cup of popcorn was used, the pan was cooled and washed between treatments, the gas flame was set to a constant, and the same pan and oil type were used for each treatment. The volume was measured with a measuring cup with units in mL.

---

precip	<i>Average monthly rainfall</i>
--------	---------------------------------

---

**Description**

Average monthly rainfall in Raleigh, NC for 1948-1992. Values are recorded only for February, March, May, June, and August.

**Format**

A data frame with 45 observations on the following 6 variables.

**year** : 2 digit year

**feb** : February rainfall (in inches)

**mar** : March rainfall (in inches)

**jun** : June rainfall (in inches)

**aug** : August rainfall (in inches)

**may** : May rainfall (in inches)

---

quake	<i>Earthquake strengths and locations</i>
-------	---

---

**Description**

The data set quake contains the strengths of earthquakes measured at the earth's continental plates. Much of the earth's seismic activity is due to motion of the large plates that make up the crust of the earth. Earthquakes occur when a buildup in tension between two layers of rock is suddenly released. For this reason many earthquakes occur at plate boundaries.

**Format**

A data frame with 496 observations on the following 5 variables.

**lat** : latitude of the event

**lon** : longitude of the event

**direction** : direction of earthquake

**strength** : strength of the earthquake (Richter scale)

**plateid** : numerical code for plate boundary

---

raleigh.snow	<i>Annual snowfall in Raleigh, NC</i>
--------------	---------------------------------------

---

**Description**

The data set raleigh.snow contains the annual snowfall totals for Raleigh, NC from the 1962-63 season through the 1992-92 season.

**Format**

A data frame with 30 observations on the following 2 variables.

**year** : 2 digit year

**snow** : annual snowfall (in inches)

---

raleigh.temp	<i>Average monthly temperatures, 1949-1988.</i>
--------------	---

---

**Description**

The raleigh.temp data frame has 480 rows and 3 columns. Each year has 12 rows of data, one for each month. The measurement is likely to be the average of the average daily temperature, where the average daily temperature = (daily high+daily low)/2.

**Format**

A data frame with 480 observations on the following 3 variables.

**temp** : temperature = (daily high+daily low)/2

**month** : the month the measurements were taken during

**year** : the year the measurements were taken during

---

randomdata	<i>Random data used in Lab 6</i>
------------	----------------------------------

---

**Description**

Random data used in Lab 6

---

Rlab	<i>Rlab - functions and datasets for ST370 at North Carolina State University</i>
------	---

---

**Description**

Rlab is a collection of functions and datasets to be used in the class ST370-Probability and Statistics for Engineers at North Carolina State University. For more information see the class labs at: <http://www.courses.ncsu.edu/ST370/distance/rlab/rlab.html>

Some major methods include:

- `bplot` - customized boxplot
- `hplot` - customized boxplot
- `lplot` - label plot (allows character z or y)
- `means` - 2-way means
- `mfit` - 2-way interaction fit
- `mplot` - means plot
- `stats` - variety of statistics
- `US` - plot of the United States
- `world` - plot of the world

These labs are based on Slab and Mlab by Doug Nychka and Dennis Boos.

**DISCLAIMER:**

This is software for statistical research and not for commercial uses. The authors do not guarantee the correctness of any function or program in this package. Any changes to the software should not be made without the authors permission.

---

salt	<i>The effect of salt on ice melting</i>
------	--

---

**Description**

When salt comes into contact with ice, it tends to break apart into individual ions which then interact with the frozen water and disrupt hydrogen bonds that have formed between ice molecules. This lowers the melting temperature of ice, and it was hypothesized that the melting process would be hastened. The data set `salt` was collected during an experiment to determine whether varying the type and amount of salt applied to a specific amount of ice has an effect on the interval required to melt that ice.

**Usage**

```
data(salt)
```

**Format**

A data frame with 24 observations on the following 3 variables.

**type** : type of salt (rock salt or table salt)

**amount** : amount of salt used (in teaspoons)

**time** : time for ice to melt (in seconds)

**Details**

Background: The Effect of Salt on the Rate at Which Ice Melts

In those sections of the country that experience winter as a time of snow and ice, salt is often spread on roadways in an attempt to counter the hazardous consequences of accumulated ice. Ice is formed when the relatively disordered molecules in liquid water reach a temperature of 32 degrees F (0 degrees C) and begin to "nucleate" or form solid ice crystals consisting of ordered water molecules. Salt, when in contact with ice, tends to break apart into individual ions (i.e. sodium and chloride) which then interact with the water and disrupt the hydrogen bonds that have formed between water molecules. Since no covalent bonds are broken or formed, the resulting chemical "solvation" is not considered to be a chemical reaction. However, the end result from the introduction of salt is that the ice crystals are disrupted and liquid water is achieved.

The purpose of the current experiment is to study the effect of salt on the rate at which ice melts. More specifically, the experiment is being conducted to answer the following questions:

1. Does varying the amount of salt applied to a constant quantity of ice result in a change in the rate of melting?
2. Does the type of salt used have an effect on the melting rate?

The first question is of interest as it relates to issues such as the cost of salt and the potential harmful effects of its use on pavement. If increasing the amount of salt applied to a given quantity of ice is not accompanied by an increase in melting rate, any application of salt beyond minimal amounts would constitute a waste of public money and possibly cause unnecessary damage to public roadways. It is hypothesized that the relationship between amount of salt used and the

time required to completely melt a given quantity of ice is negative and significant.

Likewise, the second question seeks to address the possibility that dissimilar forms of salt may produce different rates of melting. To answer this question, table salt and rock salt were included in the experimental design. Although both are chemically similar, rock salt consists of larger crystals than does the typical table salt bought in local supermarkets. Given the greater density and more efficient packing of NaCl molecules within the larger rock salt crystals, a specified volume of rock salt will likely contain a greater number of salt molecules than a similar volume of the less tightly packed table salt crystals. Therefore, it is hypothesized that rock salt will result in faster melting times than table salt.

#### Materials

Tap water

42 - 6 ounce plastic cups (paper cups tend to break at the seam as the contents freeze)

Morton brand table salt

Morton brand rock salt

1/2 cup measure

Stop Watch

#### Procedure

To answer the questions posed above, a balanced 2 x 4 factorial design was employed with amount of salt identified as a factor consisting of four levels (i.e. no salt, 1/2 tsp, 1 tsp, 1 tbsp), and the other factor being type of salt with two levels (i.e. table salt, rock salt). Three replications were conducted within each cell for a total of 24 runs. A p-level of .05 was identified for statistical significance prior to the data collection phase of the project.

Twenty-four small plastic cups were each labeled with a number designating type of salt, and a letter A-D indicating amount of salt. Each plastic cup was then filled with 4 ounces of tap water and placed in the freezer overnight (approximately 16 hours).

Since salt could not be emptied into all of the ice cups simultaneously, the remaining 18 plastic cups were each labeled and then used to hold an amount and type of salt corresponding to one of the experimental conditions. After the ice cups had been removed from the

freezer, each salt cup was quickly emptied into a corresponding ice cup with matching identification so as to minimize the time interval between the application of salt to the first and last cups.

After the last cup of salt had been emptied into the appropriate ice cup, the stopwatch was started. Room temperature during the data collection phase was approximately 72 degrees Fahrenheit. The time was recorded for each cup when ice was no longer visible in that cup.

### Source

Taken from a 1999 project by Wayde D. Johnson

---

skew	<i>3rd moment skewness ratio</i>
------	----------------------------------

---

### Description

3rd moment skewness ratio

### Usage

```
skew(x)
```

### Arguments

x                      vector

### See Also

kurt

### Examples

```
set.seed(1)
x <- rexp(100)

# Get skewness coefficient estimate for exponential distribution
skew(x)
```

---

solar	<i>Solar cell's output voltage</i>
-------	------------------------------------

---

**Description**

The data set `solar` is from an experiment investigating the effect of surface area (cell), distance, and light intensity on the output voltage of photovoltaic cells (solar cells).

**Format**

A data frame with 18 observations on the following 4 variables.

**cell** : surface area (0 for 8 sq. cm and 1 for 3 sq. cm)

**light** : light intensity

**distance** : distance

**volt** : solar cell's output voltage

---

SpGravity	<i>Specific gravity measurements</i>
-----------	--------------------------------------

---

**Description**

Data collected by a team of civil engineering students in an attempt to determine if the suggested 24 hour soak time for measuring the specific gravity and absorption of course aggregate (e.g., granite or limestone) was really necessary. They obtained samples of course aggregate from eight quarries and measured three types of specific gravity (BDSG, SSDSG and ASG), and absorption (abs) at five soak times (6, 18, 24, 48, and 72 hours).

**Format**

A data frame with 40 observations on the following 7 variables.

**quarry** : quarry sample was taken from

**rock** : type of rock (Granite or Limestone)

**time** : soak time (6, 18, 24, 48, or 72 hours)

**BDSG** : dry specific gravity

**SSDSG** : SSD specific gravity

**ASG** : apparent specific gravity

**abs** : absorption



---

stats *Calculate summary statistics*

---

## Description

Various summary statistics are calculated for different types of data.

## Usage

```
stats(x, by)
```

## Arguments

x	The data structure to compute the statistics. This can either be a vector, matrix (data sets are the columns), or a list (data sets are the components).
by	If x is a vector, an optional vector (either character or numerical) specifying the categories to divide x into separate data sets.

## Details

Stats breaks x up into separate data sets and then calls describe to calculate the statistics. Statistics are found by columns for matrices, by components for a list and by the relevant groups when a numeric vector and a by vector are given. The default set of statistics are the number of (non-missing) observations, mean, standard deviation, minimum, lower quartile, median, upper quartile, maximum, and number of missing observations. If any data set is nonnumeric, missing values are returned for the statistics. The by argument is a useful way to calculate statistics on parts of a data set according to different cases.

## Value

A matrix where rows index the summary statistics and the columns index the separate data sets.

## See Also

stats.bplot, mean, sd

## Examples

```
#Statistics for 8 normal random samples:
zork<- matrix( rnorm(200), ncol=8)
stats(zork)

zork<- rnorm( 200)
id<- sample( 1:8, 200, replace=TRUE)
stats( zork, by=id)
```

---

`stats2`*Calculate summary statistics*

---

**Description**

Various summary statistics are calculated for different types of data. Same as stats with addition of skewness and kurtosis.

**Usage**

```
stats2(x, by, digits=8)
```

**Arguments**

<code>x</code>	The data structure to compute the statistics. This can either be a vector, matrix (data sets are the columns), or a list (data sets are the components).
<code>by</code>	If <code>x</code> is a vector, an optional vector (either character or numerical) specifying the categories to divide <code>x</code> into separate data sets.
<code>digits</code>	Default number of digits is 8. This allows it to be set smaller.

**Details**

Stats breaks `x` up into separate data sets and then calls `describe` to calculate the statistics. Statistics are found by columns for matrices, by components for a list and by the relevant groups when a numeric vector and a `by` vector are given. The default set of statistics are the number of (nonmissing) observations, mean, standard deviation, skewness, kurtosis, minimum, lower quartile, median, upper quartile, maximum, and number of missing observations. If any data set is nonnumeric, missing values are returned for the statistics. The `by` argument is a useful way to calculate statistics on parts of a data set according to different cases.

**Value**

A matrix where rows index the summary statistics and the columns index the separate data sets.

**See Also**

`stats`, `stats.bplot`, `mean`, `sd`

**Examples**

```
#Statistics for 8 normal random samples:
zork<- matrix( rnorm(200), ncol=8)
stats2(zork)

zork<- rnorm( 200)
id<- sample( 1:8, 200, replace=TRUE)
stats2( zork, by=id)
```

---

US *Plot of the US with state boundaries*

---

### Description

Plots quickly, medium resolution outlines of the US with the states and bodies of water.

### Usage

```
US( xlim=c(-124.7, -67.1), ylim = c(25.2, 49.4),  
    add=FALSE, shift=FALSE, asp=1, ...)
```

### Arguments

<code>ylim</code>	range of latitudes
<code>xlim</code>	range of longitudes
<code>add</code>	If true will add the world map to current plot
<code>shift</code>	If TRUE shifts to be centered on the Dateline and longitude runs from 0 to 360. If FALSE centers on Prime Meridian and longitude runs from -180 to 180.
<code>asp</code>	aspect ratio used if add is false, see <code>plot.default</code> .
<code>...</code>	These are graphical arguments that are passed to the <code>lines</code> function that draws outline.

### Details

This function uses the Rlab dataset `US.dat` for the coordinates.

### See Also

`world`

### Examples

```
# Draw map in device color # 3  
US( col=3)
```

---

us.age	<i>American age data</i>
--------	--------------------------

---

**Description**

The data set `us.age` contains the average age for all Americans (`all`), females (`f`), and males (`m`) for the years 1990-1999. Actually, the data from the US Census Bureau are based on the 1990 census and then updated yearly.

**Format**

A data frame with 10 observations on the following 4 variables.

**all** : average age of all Americans  
**f** : average age of female Americans  
**m** : average age of male Americans  
**year** : 4 digit year

**Source**

US Census Bureau

---

US.dat	<i>Outline of coterminous US and states.</i>
--------	--

---

**Description**

This data set is used by the `rlab` function `US` to draw a map. It is the medium resolution outline that is produced by drawing the US from the `maps` package.

---

us.pop	<i>American population data</i>
--------	---------------------------------

---

**Description**

The data set `us.pop` contains the United States population for the years 1900-1999.

**Format**

A data frame with 100 observations on the following 2 variables.

**year** : 2 digit year  
**pop** : population (in millions)

---

view	<i>View first rows of a data set</i>
------	--------------------------------------

---

**Description**

View the first X rows (10, by default) rows of a data set. Columns names are displayed if appropriate.

**Usage**

```
view(x, maxlines = 10)
```

**Arguments**

x	data set to be viewed; can be data.frame, matrix, list or vector.
maxlines	maximum number of rows to be displayed.

**Details**

If the data set contains more rows than maxlines, then a message indicating the number unviewed rows id displayed. If the data set contains fewer rows the maxlines, only those rows are displayed.

**See Also**

ls, objects

---

viscosity	<i>Results from students experiments using water, oil and shampoo.</i>
-----------	--

---

**Description**

The viscosity data frame was the result of the following ST370 experiment performed in the fall of 1996. The students' description is as follows.

For this experiment we used three different liquids: water, cooking oil and shampoo. First we placed a cup of shampoo in a microwave oven, and heated it for 50 seconds. Immediately after that we transfered the liquid to a dishwasher container. We turned this container upside down with the spout closed and poked a hole on the bottom part of it. Then we placed a half cup measuring container beneath the dishwasher container. Then we opened the spout of the dishwashing container, and measured the time it took for liquid to come out and fill the half cup container. We repeated the same procedure with each liquid three times. Then we placed the liquids at room temperature in the container and repeated the above pcedure three times as well. Then we placed each liquid in the freezer 10 minutes at a time and repeated the prior procedure three times.

**Format**

A data frame with 26 observations on the following 3 variables.

**liquid** : liquid used (shampoo, oil or water)

**temp** : temperature (hot or cold)

**time** : time (in seconds)

---

vocab

*Children's vocabulary by age*

---

**Description**

The data set vocab contains the average oral vocabulary size (words) for children at different ages (age).

**Format**

A data frame with 10 observations on the following 2 variables.

**age** : age (in years)

**words** : vocabulary (in number of words)

**Source**

Discovering Psychology by Weiner, 1977

---

webhost

*Comparison of four different web hosts*

---

**Description**

The data set webhost contains the results of a comparison of different webhosts. A student team decided to compare four free hosting services in the spring of 2000: go.com, angelfire.com, geocities.com, and xoom.com. They uploaded four pages:

\* one with text only (100k),

\* one with text only (100k) and one 20k jpeg image

\* one with text only (100k) and two 20k jpeg images

\* one with text only (100k) and three 20k jpeg images

The last page wouldn't load for xoom; so they had 15 data points with load times for the response variable and the number of graphic images for a quantitative independent variable.

**Format**

A data frame with 15 observations on the following 3 variables.

**graphics** : number of 29k jpeg images on uploaded page

**time** : time to up load page (in seconds)

**host** : web host (angelfire, geocities, go or xoom)

Weibull

*The Weibull Distribution***Description**

Density, distribution function, quantile function and random generation for the Weibull distribution with parameters alpha (or shape) and beta (or scale).

This special Rlab implementation allows the parameters alpha and beta to be used, to match the function description often found in textbooks.

**Usage**

```
dweibull(x, shape, scale = 1, alpha = shape, beta = scale, log = FALSE)
pweibull(q, shape, scale = 1, alpha = shape, beta = scale,
         lower.tail = TRUE, log.p = FALSE)
qweibull(p, shape, scale = 1, alpha = shape, beta = scale,
         lower.tail = TRUE, log.p = FALSE)
rweibull(n, shape, scale = 1, alpha = shape, beta = scale)
```

**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.
shape, scale	shape and scale parameters, the latter defaulting to 1.
alpha, beta	alpha and beta parameters, the latter defaulting to 1.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

The Weibull distribution with alpha (or shape) parameter  $a$  and beta (or scale) parameter  $\sigma$  has density given by

$$f(x) = (a/\sigma)(x/\sigma)^{a-1} \exp(-(x/\sigma)^a)$$

for  $x > 0$ . The cumulative is  $F(x) = 1 - \exp(-(x/\sigma)^a)$ , the mean is  $E(X) = \sigma\Gamma(1 + 1/a)$ , and the  $Var(X) = \sigma^2(\Gamma(1 + 2/a) - (\Gamma(1 + 1/a))^2)$ .

**Value**

dweibull gives the density, pweibull gives the distribution function, qweibull gives the quantile function, and rweibull generates random deviates.

**Note**

The cumulative hazard  $H(t) = -\log(1-F(t))$  is `-pweibull(t, a, b, lower = FALSE, log = TRUE)` which is just  $H(t) = (t/b)^a$ .

**See Also**

[dexp](#) for the Exponential which is a special case of a Weibull distribution.

**Examples**

```
x <- c(0,rlnorm(50))
all.equal(dweibull(x, alpha = 1), dexp(x))
all.equal(pweibull(x, alpha = 1, beta = pi), pexp(x, rate = 1/pi))
## Cumulative hazard H():
all.equal(pweibull(x, 2.5, pi, lower=FALSE, log=TRUE), -(x/pi)^2.5, tol=1e-15)
all.equal(qweibull(x/11, alpha = 1, beta = pi), qexp(x/11, rate = 1/pi))
```

---

 wire

*Wire resistance experiment*


---

**Description**

The data set `wire` contains the results of an experiment to study the relationship between the resistance of a wire and its gauge and its length.

**Format**

A data frame with 27 observations on the following 3 variables.

**gauge** : wire gauge (in AWG units)

**length** : wire length (in feet)

**resistance** : resistance (in ohms)



---

world	<i>Plot of the world</i>
-------	--------------------------

---

**Description**

Plots quickly, medium resolution outlines of large land masses and bodies of water.

**Usage**

```
world(ylim=c(-90, 90), xlim=c(-180, 180), add=FALSE, asp = 1,
      xlab = "", ylab = "", xaxt = "n", yaxt = "n", eps=.1,
      shift=FALSE,...)
```

**Arguments**

ylim	range of latitudes
xlim	range of longitudes
add	logical; if true will add the world map to current plot.
asp	aspect ratio used if add is false, see plot.default.
xlab,ylab	labels for x- and y-axis; empty by default.
xaxt,yaxt	axis type for x- and y-axis; empty by default.
eps	Tolerance to decide when to insert line break about 0 if map is to be shifted. (leave this at .1)
shift	If TRUE shifts to be centered on the Dateline and longitude runs from 0 to 360. If FALSE centers on Prime Meridian and longitude runs from -180 to 180.
...	These are graphical arguments that are passed to the lines (and plot if add is false) function that draws the outline.

**Details**

This function uses the Rlab dataset world.dat for the coordinates. The main advantage of this function is that it is fast and easy to modify. The shift option to center over the dateline is useful because often plots of ocean and atmosphere information center the map this way.

**See Also**

US

**Examples**

```
# Draw map in device color number 2
world( col=2 )
# add the US
US( add=TRUE )

## Western Europe (*which* big island is missing ?)
```

```
## with a (light color!) coordinate grid:
world(xlim=c(-10,18),ylim=c(36,60), xaxt = "s", yaxt = "s")
grid()
```

---

world.dat	<i>Outline of world map.</i>
-----------	------------------------------

---

### Description

This data set is used by the rlab function world to draw a map.

---

xline	<i>Draw a vertical line</i>
-------	-----------------------------

---

### Description

Adds vertical lines in the plot region.

### Usage

```
xline(x, ...)
```

### Arguments

x	Values on x axis specifying location of vertical lines.
...	Any plotting options for abline.

### See Also

yline, abline

### Examples

```
plot( 1:10)
xline( 6.5, col=2)

world( col=3)
yline( seq( -80,80,10),col=4, lty=2)
xline( seq( -180,180,10),col=4,lty=2)
yline( 0, lwd=2, col=4)
```



---

yline *Draw horizontal lines*

---

**Description**

Adds horizontal lines in the plot region.

**Usage**

```
yline(y, ...)
```

**Arguments**

y	Values on y axis specifying location of vertical lines.
...	Any plotting options for abline.

**See Also**

xline, abline

**Examples**

```
world( col=3)
yline( seq( -80,80,10),col=4, lty=2)
xline( seq( -180,180,10),col=4,lty=2)
yline( 0, lwd=2, col=4)
```

# Index

## \*Topic **aplot**

xline, 50  
yline, 52

## \*Topic **datasets**

actuator, 3  
airplane, 4  
alka, 4  
bread, 9  
bread2, 9  
cancer, 10  
capac, 10  
cavendish, 11  
climate, 11  
college, 12  
draft, 13  
drill, 14  
earthq, 15  
etruscan, 15  
framerate, 17  
ftp, 18  
ftptime, 18  
golf, 20  
insulate, 22  
jet, 23  
magnet, 26  
marathon, 26  
metalcut, 28  
michelson, 29  
monarch, 30  
ncsu, 31  
oil, 31  
ozone, 32  
ping, 32  
popcorn, 33  
precip, 34  
quake, 34  
raleigh.snow, 35  
raleigh.temp, 35  
randomdata, 35

Rlab, 36  
salt, 36  
solar, 40  
SpGravity, 40  
us.age, 44  
US.dat, 44  
us.pop, 44  
viscosity, 45  
vocab, 46  
webhost, 46  
wire, 48  
world.dat, 50  
yarn, 51  
yarnred, 51

## \*Topic **distribution**

Bernoulli, 5  
Exponential, 16  
Gamma, 18  
Weibull, 47

## \*Topic **hplot**

bplot, 6  
bplot.xy, 8  
hplot, 21  
lplot, 24  
mplot, 30  
US, 43  
world, 49

## \*Topic **list**

ls.rlab, 25  
view, 45

## \*Topic **univar**

count, 13  
kurt, 23  
means, 27  
mfit, 28  
skew, 39  
stats, 41  
stats2, 42

actuator, 3

- airplane, 4
- alka, 4
- Bernoulli, 5
- bplot, 6
- bplot.xy, 8
- bread, 9
- bread2, 9
- cancer, 10
- capac, 10
- cavendish, 11
- climate, 11
- college, 12
- count, 13
- dbern (Bernoulli), 5
- dbeta, 20
- dbinom, 6
- dchisq, 20
- dexp, 48
- dexp (Exponential), 16
- dgamma, 17
- dgamma (Gamma), 18
- dpois, 6
- draft, 13
- drill, 14
- dweibull, 17
- dweibull (Weibull), 47
- earthq, 15
- etruscan, 15
- exp, 17
- Exponential, 16
- framerate, 17
- ftp, 18
- ftpstime, 18
- Gamma, 18
- gamma, 20
- golf, 20
- hplot, 21
- insulate, 22
- jet, 23
- kurt, 23
- lplot, 24
- ls.rlab, 25
- ls.summary.rlab (ls.rlab), 25
- magnet, 26
- marathon, 26
- means, 27
- metalcut, 28
- mfit, 28
- micelson, 29
- monarch, 30
- mplot, 30
- ncsu, 31
- oil, 31
- ozone, 32
- pbern (Bernoulli), 5
- pexp (Exponential), 16
- pgamma (Gamma), 18
- ping, 32
- popcorn, 33
- precip, 34
- pweibull (Weibull), 47
- qbern (Bernoulli), 5
- qexp (Exponential), 16
- qgamma (Gamma), 18
- quake, 34
- qweibull (Weibull), 47
- raleigh.snow, 35
- raleigh.temp, 35
- randomdata, 35
- rbern (Bernoulli), 5
- rexp (Exponential), 16
- rgamma (Gamma), 18
- Rlab, 36
- rweibull (Weibull), 47
- salt, 36
- skew, 39
- solar, 40
- SpGravity, 40
- stats, 41
- stats2, 42
- US, 43
- us.age, 44

US.dat, [44](#)  
us.pop, [44](#)

view, [45](#)  
viscosity, [45](#)  
vocab, [46](#)

webhost, [46](#)  
Weibull, [47](#)  
wire, [48](#)  
world, [49](#)  
world.dat, [50](#)

xline, [50](#)

yarn, [51](#)  
yarnred, [51](#)  
yline, [52](#)