

# Package ‘mixdist’

April 17, 2009

**Version** 0.5-2

**Date** 2008-05-02

**Title** Finite Mixture Distribution Models

**Author** Peter Macdonald <pdmmac@mcmaster.ca>, with contributions from Juan Du <duduyy@hotmail.com>

**Maintainer** Peter Macdonald <pdmmac@mcmaster.ca>

**Depends** R (>= 1.4.0)

**Description** This package contains functions for fitting finite mixture distribution models to grouped data and conditional data by the method of maximum likelihood using a combination of a Newton-type algorithm and the EM algorithm.

**License** GPL (>= 2)

**URL** <http://www.r-project.org>, <http://www.math.mcmaster.ca/peter/mix/mix.html>

**Repository** CRAN

**Date/Publication** 2008-05-03 06:53:36

## R topics documented:

anova.mix . . . . .	2
bindat . . . . .	3
binpar . . . . .	4
cassie . . . . .	5
coef.mix . . . . .	5
conditdat . . . . .	6
expdat . . . . .	7
fiftn80 . . . . .	8
fitted.mix . . . . .	9
groupstats . . . . .	10
mix . . . . .	11
mixconstr . . . . .	13

mixdata	14
mixgroup	15
mixparam	16
normals	17
pearson	18
pearsonpar	18
pikdat5	19
pike65	20
pike65sg	21
pikepar	22
pikeraw	23
plot.mix	24
plot.mixdata	25
poisdat	27
poispar	27
print.mix	28
summary.mix	29
testconstr	30
weibullpar	31
weibullparinv	32

## Index 33

---

anova.mix	<i>ANOVA Tables for Mixture Model Objects</i>
-----------	---

---

### Description

Compute analysis of variance tables for one or two mixture model objects.

### Usage

```
## S3 method for class 'mix':
anova(object, mixobj2, ...)
```

### Arguments

object	an object of class "mix", usually, a result of a call to the mixture model fitting function <code>mix</code> .
mixobj2	an object of the same type to be compared with <code>object</code> , which contains the results of fitting another model with more or fewer parameters fitted.
...	additional objects of the same type.

**Value**

An object of class "anova" inheriting from class "data.frame". When given a single argument this function produces a table which tests whether the model is significant. The table contains the residual degrees of freedom, Chi-square statistic and P value. If the class of the argument is not "mix", this function returns NULL. When given two objects, it tests the models against one another and lists them in the order of number of parameters fitted. For the model with fewer parameters fitted, the change in degrees of freedom is given. This only make statistical sense if the models are nested. If one of arguments does not belong to the class "mix", the function will give the anova table for the other argument; if both of them do not, it returns NULL.

**Warning**

The comparison between two models will only be valid if they are fitted to the same dataset. And the two models should be nested.

**See Also**

The model fitting function [mix](#), the generic function [anova](#).

**Examples**

```
data(pike65) # load the grouped data `pike65'
data(pikepar) # load the initial values of parameters for the data `pike65'
fitpikel1 <- mix(pike65, pikepar, "lnorm", mixconstr(conmu = "MFX",
          fixmu = c(FALSE, FALSE, FALSE, FALSE, TRUE), consigma = "CCV"), emstep = 3)
anova(fitpikel1)
fitpikel2 <- mix(pike65, pikepar, "lnorm", mixconstr(consigma = "CCV"), emsteps = 3)
anova(fitpikel2)
anova(fitpikel1, fitpikel2)
anova(fitpikel2, fitpikel1)
```

---

 bindat

*Grouped Binomial Data*


---

**Description**

We randomly generate four groups of binomial distribution data with means 4, 8, 12, 16, and corresponding variances 3.2, 4.8, 4.8 and 3.2. Then we mix the four data groups with 100 observations for each group, i.e., with equal proportions. After grouping the mixture data, we obtain the grouped data bindat.

The bindat data frame has 21 rows and 2 columns.

**Usage**

```
data(bindat)
```

**Format**

This data frame contains the following columns:

**x** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

**Examples**

```
data(bindat)
data(binpar)
plot.mixdata(bindat)
fit <- mix(bindat, binpar, "binom", mixconstr(conpi = "PFX",
      fixpi = c(TRUE, TRUE, TRUE, TRUE), consigma = "BINOM", size = c(20, 20, 20, 20)))
fit
plot(fit)
```

---

binpar

*Starting Values of Parameters for the Binomial Data Set*

---

**Description**

Starting values of parameters for fitting a mixture distribution to the data set `bindat`.

The `binpar` data frame has 4 rows and 3 columns.

**Usage**

```
data(binpar)
```

**Format**

This data frame contains the following columns:

**pi** the starting values for proportions.

**mu** the starting values for means.

**sigma** the starting values for standard deviations.

**Examples**

```
data(binpar)
```

---

`cassie`*Cassie's Length-Frequency Example*

---

**Description**

Data for Cassie's (1954) analysis of size frequency distributions.

The `cassie` data frame has 40 rows and 2 columns.

**Usage**

```
data(cassie)
```

**Format**

This data frame contains the following columns:

**length** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

**Source**

Cassie, R.M. (1954). *Some uses of probability paper in the analysis of size frequency distributions*. Aust. J. Mar. Freshwater Res. 5, 513-522.

The data, lengths (in) of 256 snapper (*Chrysophrys auratus* Forster) taken by a trawl with a mesh of about 1.5 in, are given in Table 5 of that paper. Cassie's results are given in his Table 1.

**References**

<http://www.math.mcmaster.ca/peter/mix/demex/excass.html>

**Examples**

```
data(cassie)
plot.mixdata(cassie)
```

---

`coef.mix`*Extract Mixture Model Coefficients*

---

**Description**

`coef.mix` is a function which extracts mixture model coefficients from objects returned by the model fitting function `mix`. It is called via the generic function `coef`.

**Usage**

```
## S3 method for class 'mix':
coef(object, natpar = FALSE, ...)
```

**Arguments**

`object` an object of class "mix", usually, the results returned by the model fitting function `mix`.

`natpar` a logical scalar specifying whether the natural parameters should be given.

`...` other arguments.

**Value**

A data frame containing three variables, which are, in order, the proportions, means, and standard deviations, respectively. If `natpar` is TRUE, then the natural parameters of component distributions are also displayed.

**See Also**

`mix` for model fitting.

**Examples**

```
data(pike65) # load the grouped data `pike65'
data(pikepar) # load the initial values of parameters for the data `pike65'
fit <- mix(pike65, pikepar, "lnorm", mixconstr(consigma = "CCV"), emsteps = 3)
coef(fit)
coef(fit, natpar = TRUE)
```

---

conditdat

---

*Add Conditional Data to Grouped Data*


---

**Description**

It combines automatically grouped data with conditional data when enter the conditional samples.

**Usage**

```
conditdat(mixdat, k, conditsamples)
```

**Arguments**

`mixdat` a data frame containing grouped data, whose first column should be the right boundaries of grouping intervals, and the second one should be the numbers of observations falling into each interval.

`k` the number of components.

conditsamples

a vector containing conditional data, which consists of the conditional samples, the first element of each sample is a number indicating which interval this sample comes from.

### Value

A data frame containing the grouped data with conditional data.

### See Also

[mixgroup](#) for constructing grouped and conditional data.

### Examples

```
data(pike65) # load the data set `pike65'
pike65 # display the data set `pike65'
conditdat(pike65, k = 5, conditsamples =
  c(c(4, 9, 2, 0, 0, 0), c(5, 8, 6, 0, 0, 0),
    c(12, 0, 2, 34, 0, 0), c(13, 0, 0, 21, 0, 0),
    c(15, 0, 0, 5, 5, 0), c(16, 0, 0, 6, 5, 1),
    c(17, 0, 0, 5, 7, 0), c(18, 0, 0, 4, 4, 3),
    c(19, 0, 0, 0, 8, 0), c(20, 0, 0, 0, 2, 1),
    c(21, 0, 0, 0, 1, 5), c(22, 0, 0, 0, 2, 4)))
# add conditional data to the grouped data `pike65'
```

---

expdat

*A Mixture Data of Three Exponential Distributions*

---

### Description

A total of 1000 observations was generated by computer to follow the mixture distribution  $1/3 E(1) + 1/3 E(4) + 1/3 E(16)$  where  $E(m)$  denotes an exponential distribution with mean  $m$ .

The expdat data frame has 25 rows and 2 columns.

### Usage

```
data(expdat)
```

### Format

This data frame contains the following columns:

**x** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

### Source

Macdonald, P.D.M. and Green, P.E.J. (1988) *User's Guide to Program MIX: An Interactive Program for Fitting Mixtures of Distributions*. ICHTHUS DATA SYSTEMS.

## References

Macdonald, P.D.M. and Green, P.E.J. (1988) *User's Guide to Program MIX: An Interactive Program for Fitting Mixtures of Distributions*. ICHTHUS DATA SYSTEMS.

<http://www.math.mcmaster.ca/peter/mix/demex/exexp.html>

## Examples

```
data(expdat)
plot.mixdata(expdat)
```

---

fiftn80

*A Mixed Data with Fifteen Normal Components*

---

## Description

Fifteen normal components grouped over eighty intervals.

The `fiftn80` data frame has 80 rows and 2 columns.

## Usage

```
data(fiftn80)
```

## Format

This data frame contains the following columns:

**x** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

## Details

A total of 820 observations were generated by computer to follow the distribution  $1/15 N(5, 1) + 1/15 N(10, 1) + \dots + 1/15 N(75, 1)$  where  $N(m, s)$  denotes a normal distribution with mean  $m$  and standard deviation  $s$ .

## Source

<http://www.math.mcmaster.ca/peter/mix/demex/ex1580.html>

## Examples

```
data(fiftn80)
plot.mixdata(fiftn80)
```

---

fitted.mix	<i>Compute Mixture Model Fitted Values</i>
------------	--

---

## Description

`fitted.mix` is a function which computes fitted values from objects returned by the modeling function `mix`. It is called via the generic function `fitted`.

## Usage

```
## S3 method for class 'mix':
fitted(object, digits = NULL, ...)
```

## Arguments

<code>object</code>	an object of class "mix", usually, the results returned by the model fitting function <code>mix</code> .
<code>digits</code>	a specified number of decimal places to be reserved.
<code>...</code>	other arguments.

## Value

List with the following components:

<code>mixed</code>	the estimated mixed data, that is, the fitted numbers of observations falling into each interval.
<code>joint</code>	the estimated joint data, that is, the fitted numbers of observations from each component falling into every interval.
<code>conditional</code>	the estimated conditional data to be returned if <code>usecondit</code> of <code>object</code> is <code>TRUE</code> , which are the fitted numbers of observations from given intervals belonging to each component.
<code>conditprob</code>	the estimated conditional probabilities of observations from given interval belonging to each component.

## See Also

`mix` for fitting mixture distributions.

## Examples

```
data(pike65)
data(pikepar)
fit1 <- mix(pike65, pikepar, "lnorm", mixconstr(consigma = "CCV"), emsteps = 3)
fitted(fit1)
data(pike65sg)
fit2 <- mix(pike65sg, pikepar, "gamma", mixconstr(consigma = "CCV"), usecondit = TRUE)
fitted(fit2, digits = 2)
```

---

`groupstats`*Estimate Parameters of One-Component Mixture Distribution*

---

### Description

`groupstats` is a function which estimates the proportion, mean and standard deviation for a mixture distribution with one component.

### Usage

```
groupstats(mixdat)
```

### Arguments

`mixdat` A data frame containing grouped data, whose first column should be right boundaries of grouping intervals where the first and last intervals are open-ended; whose second column should consist of the frequencies indicating numbers of observations falling into each interval.

### Value

A list containing the following components:

`pi` the value is 1 because of only one component.  
`mu` the estimated mean of `mixdat`.  
`sigma` the estimated standard deviation of `mixdat`.

### See Also

[mixgroup](#) for grouping data, [mixparam](#) for constructing starting values of parameters.

### Examples

```
data(pike65)  
groupstats(pike65)
```

---

 mix

*Estimate Parameters of Mixture Distributions*


---

**Description**

Find a set of overlapping component distributions that gives the best fit to grouped data and conditional data, using a combination of a Newton-type method and EM algorithm.

**Usage**

```
mix(mixdat, mixpar, dist = "norm", constr = list(conpi = "NONE",
  conmu = "NONE", consigma = "NONE", fixpi = NULL, fixmu = NULL,
  fixsigma = NULL, cov = NULL, size = NULL), emsteps = 1,
  usecondit = FALSE, exptol = 5e-06, print.level = 0, ...)
```

**Arguments**

mixdat	A data frame containing grouped data, whose first column should be right boundaries of grouping intervals where the first and last intervals are open-ended; whose second column should consist of the frequencies indicating numbers of observations falling into each interval. If conditional data are available, this data frame should have $k + 2$ columns, where $k$ is the number of components, whose element in row $j$ and column $i + 2$ is the number of observations from the $j$ th interval belonging to the $i$ th component.
mixpar	A data frame containing starting values for parameters of component distributions, which are, in order, the proportions, means, and standard deviations.
dist	the distribution of components, it can be one of "norm", "lnorm", "gamma", "weibull", "binom", "nbinom" and "pois".
constr	a list of constraints on parameters of component distributions. See function <a href="#">mixconstr</a> .
emsteps	a non-negative integer specifying the number of EM steps to be performed.
usecondit	logical. If usecondit is TRUE and mixdat includes conditional data, then conditional data will be used with grouped data to estimate parameters of mixtures.
exptol	a positive scalar giving the tolerance at which the scaled fitted value is considered large enough to be a degree of freedom.
print.level	this argument determines the level of printing which is done during the optimization process. The default value of 0 means that no printing occurs, a value of 1 means that initial and final details are printed and a value of 2 means that full tracing information is printed.
...	additional arguments to the optimization function <a href="#">nlm</a>

**Value**

A list containing the following items:

parameters	A data frame containing estimated values for parameters of component distributions, which are, in order, the proportions, means, and standard deviations.
se	A data frame containing estimated values for standard errors of parameters of component distributions.
distribution	the distribution used to fit the data.
constraint	the constraints on parameters.
chisq	the goodness-of-fit chi-square statistic.
df	degrees of freedom of the fitted mixture model.
P	a significance level (P-value) for the goodness-of-fit test.
vmat	covariance matrix for the estimated parameters.
mixdata	the original data, i.e. the argument <code>mixdat</code> .
usecondit	the value of the argument <code>usecondit</code> .

**References**

Macdonald, P.D.M. and Green, P.E.J. (1988) *User's Guide to Program MIX: An Interactive Program for Fitting Mixtures of Distributions*. ICHTHUS DATA SYSTEMS.

**See Also**

[mixgroup](#) for grouping data, [mixparam](#) for organizing the parameter values, [mixconstr](#) for constructing constraints. [nlm](#) for additional arguments.

**Examples**

```
data(pike65)
data(pikepar)
fitpike1 <- mix(pike65, pikepar, "lnorm", constr = mixconstr(consigma = "CCV"), emsteps = 3)
fitpike1
plot(fitpike1)
data(pike65sg)
fitpike2 <- mix(pike65sg, pikepar, "lnorm", emsteps = 3, usecondit = TRUE)
fitpike2
plot(fitpike2)
data(bindat)
data(binpar)
fitbin1 <- mix(bindat, binpar, "binom", constr = mixconstr(consigma = "BINOM", size = c(20,
plot(fitbin1)
fitbin2 <- mix(bindat, binpar, "binom", constr = mixconstr(conpi = "PFX",
fixpi = c(TRUE, TRUE, TRUE, TRUE), consigma = "BINOM", size = c(20, 20, 20, 20),
plot(fitbin2)
```

---

 mixconstr

---

*Construct Constraints on Parameters*


---

**Description**

Construct constraints on parameters and check if the constraints are invalid. See the reference for details.

**Usage**

```
mixconstr(conpi = "NONE", conmu = "NONE", consigma = "NONE",
          fixpi = NULL, fixmu = NULL, fixsigma = NULL, cov = NULL,
          size = NULL)
```

**Arguments**

conpi	a constraint on proportions, it can be either "NONE" denoting no constraint on proportions, or "PFX" indicating some proportions being fixed.
conmu	a constraint on means, it can be "NONE", "MFX", "MEQ", "MES" and "MGC", which denote no constraint on means, specified means fixed, means equal, means with equal spaces and means lying along a growth curve, respectively.
consigma	a constraint on standard deviations, it can be "NONE", "SFX", "SEQ", "FCV", "CCV", "BINOM", "NBINOM" and "POIS", which denote no constraint on standard deviations, specified standard deviations fixed, standard deviations equal, fixed coefficient of variation, constant coefficient of variation, the means and standard deviations have the same relation as that of Binomial distribution, as that of Negative Binomial distribution and as that of Poisson distribution.
fixpi	NULL or a vector with TRUE and FALSE as its elements, indicating which proportions are fixed when conpi is "PFX". If an element is TRUE, the corresponding proportion is fixed at the starting value.
fixmu	similar to fixpi. NULL or a vector indicating which means are fixed when conmu is "MFX".
fixsigma	similar to fixpi. NULL or a vector indicating which standard deviations are fixed when consigma is "SFX".
cov	NULL or a scalar if consigma is "FCV", then the coefficients of variation are fixed at this scalar.
size	NULL or a vector of numbers of trials for each component when consigma is "BINOM" or "NBINOM".

**Value**

A list containing the following components, which are, in order, conpi, conmu, consigma, fixpi, fixmu, fixsigma, cov, size.

**References**

Macdonald, P.D.M. and Green, P.E.J. (1988) *User's Guide to Program MIX: An Interactive Program for Fitting Mixtures of Distributions*. ICHTHUS DATA SYSTEMS.

**See Also**

[mixgroup](#) for grouping data, [mixparam](#) for constructing starting values of parameters.

**Examples**

```

mixconstr()
mixconstr(conmu = "MEQ", consigma = "SFX", fixsigma = c(TRUE, FALSE, TRUE, TRUE, FALSE))
mixconstr(consigma = "BINOM", size = c(25, 25, 25))

```

---

 mixdata

*Mixed Data*


---

**Description**

`as.mixdata` checks if its argument is mixed data, if true, it returns the data with class "mixdata", if false, it returns NULL.

`is.mixdata` returns TRUE if its argument is of class "mixdata" and FALSE otherwise.

**Usage**

```

as.mixdata(x)
is.mixdata(x)

```

**Arguments**

`x` object to be tested.

**Details**

Mixed data consist of grouped data and conditional data (if available). Grouped data is either a data frame or a matrix, whose first column should be right boundaries of grouping intervals where the first and last intervals are open-ended; whose second column should consist of the frequencies indicating numbers of observations falling into each interval. If conditional data are available, mixed data should have  $k + 2$  columns, where  $k$  is the number of components, whose element in row  $j$  and column  $i + 2$  is the number of observations from the  $j$ th interval belonging to the  $i$ th component.

**See Also**

[mixgroup](#) to construct mixed data.

**Examples**

```

data(pike65) # load data set `pike65'
pike65 # display the mixed data `pike65'
data(pike65sg) # load data set `pike65sg'
pike65sg # display the mixed data `pike65sg'
data(pikepar)
as.mixdata(pikepar)
as.mixdata(pike65)
is.mixdata(pike65)
is.mixdata(as.mixdata(pike65))

```

---

mixgroup

---

*Construct Grouped Data from Raw Data*


---

**Description**

Group raw data in the form of numbers of observations over successive intervals.

**Usage**

```

mixgroup(x, breaks = NULL, xname = NULL, k = NULL, usecondit = FALSE)

```

**Arguments**

x	a data frame or matrix containing raw data, whose first column should be the measurements to be grouped, and second column, if available, includes the numbers indicating which component each individual belongs to.
breaks	one of: * a vector giving the boundaries of intervals which raw data are grouped into, * a single number giving the number of intervals, * a character string naming an algorithm to compute the number of intervals, * a function to compute the number of intervals. In the last three cases the number is a suggestion only.
xname	the name of measurement.
k	the number of components.
usecondit	if usecondit is TRUE and x has two columns, then conditional data will be displayed with grouped data.

**Value**

A data frame containing grouped data derived from raw data, whose first column includes the right boundaries of grouping intervals, where the first and last intervals are open-ended; whose second column consists of the frequencies which are the numbers of observations falling into each interval. If usecondit is TRUE and the numbers indicating which component the individual comes from are available, conditional data which can be regarded as a table, whose element in row j and column i is the number of observations from the jth interval belonging to the ith component, will be displayed with grouped data.

**See Also**

[hist](#) for more information about the argument `breaks`, [is.mixdata](#) for checking the class of data sets, [mixparam](#) for organizing the parameter values, [mixconstr](#) for constructing constraints.

**Examples**

```
data(pikeraw) # load raw data `pikeraw'
pikeraw # display the data set `pikeraw'
mixgroup(pikeraw) # group raw data
pikemd <- mixgroup(pikeraw, breaks = c(0, seq(19.75, 65.75, 2), 80))
plot(pikemd)
mixgroup(pikeraw, breaks = c(0, seq(19.75, 65.75, 2), 80), usecondit = TRUE, k = 5)
# construct grouped data associated with conditional data
mixgroup(pikeraw, usecondit = TRUE)
mixgroup(pikeraw, usecondit = TRUE, k = 3) # grouping data with a warning message
mixgroup(pikeraw, usecondit = TRUE, k = 8)
```

---

mixparam

*Construct Starting Values for Parameters*


---

**Description**

Construct starting values for parameters of a mixture model.

**Usage**

```
mixparam(mu, sigma, pi = NULL)
```

**Arguments**

<code>mu</code>	a vector of means of component distributions, which should be in ascending order.
<code>sigma</code>	a vector of standard deviations of component distributions, which are corresponding to the means. <code>sigmas</code> must be in ascending order when means are equal.
<code>pi</code>	the corresponding mixing proportions of components. If <code>NULL</code> , the proportions will be taken as $1/k$ , where <code>k</code> is the number of elements of <code>mu</code> .

**Value**

A data frame containing three variables, which are, in order, the proportions, means, and standard deviations.

**See Also**

[mixgroup](#) for grouping data, [mixconstr](#) for constructing constraints.

**Examples**

```
mixparam(mu = c(20, 30, 40), sigma = c(2, 3, 4))  
mixparam(c(20, 30, 40), c(3), c(0.15, 0.78, 0.07))
```

---

`normals`*Scale Mixture Data with Three Normal Components*

---

**Description**

Scale mixture of three normal distributions.

The `normals` data frame has 25 rows and 2 columns.

**Usage**

```
data(normals)
```

**Format**

This data frame contains the following columns:

**x** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

**Details**

A total of 249 observations were generated by computer to follow the mixture distribution  $1/3 N(12.5, 1) + 1/3 N(12.5, 3) + 1/3 N(12.5, 5)$  where  $N(m, s)$  denotes a normal distribution with mean  $m$  and standard deviation  $s$ .

**Source**

<http://www.math.mcmaster.ca/peter/mix/demex/exscle.html>

**Examples**

```
data(normals)  
plot.mixdata(normals)
```

pearson

*Karl Pearson's Crab Data*

---

**Description**

The data give the ratio of "forehead" breadth to body length for 1000 crabs sampled at Naples by Professor W.F.R. Weldon.

The `pearson` data frame has 29 rows and 2 columns.

**Usage**

```
data(pearson)
```

**Format**

This data frame contains the following columns:

**ratio** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

**Source**

Pearson, K. (1894). *Contributions to the mathematical theory of evolution*. Phil. Trans. Roy. Soc. London A 185, 71-110.

**References**

<http://www.math.mcmaster.ca/peter/mix/demex/excrabs.html>

**Examples**

```
data(pearson)
plot.mixdata(pearson)
```

---

pearsonpar

*Starting Values of Parameters for the Pearson's Data*

---

**Description**

Starting values of parameters for fitting a mixture distribution to the data set `pearson`.

The `pearsonpar` data frame has 2 rows and 3 columns.

**Usage**

```
data(pearsonpar)
```

**Format**

This data frame contains the following columns:

**pi** the starting values for proportions.

**mu** the starting values for means.

**sigma** the starting values for standard deviations.

**Source**

Pearson, K. (1894). *Contributions to the mathematical theory of evolution*. Phil. Trans. Roy. Soc. London A 185, 71-110.

**References**

<http://www.math.mcmaster.ca/peter/mix/demex/excrabs.html>

**Examples**

```
data(pearsonpar)
```

---

pikdat5

*Heming Lake Pike Data*

---

**Description**

The raw data `pikeraw` give the lengths of 523 pike (*Esox lucius*), and there are known to be five age-groups in the sample. We grouped the lengths over 25 intervals to obtain the grouped data given as separate samples for each age group determined by scale reading.

The `pikdat5` data frame has 25 rows and 6 columns.

**Usage**

```
data(pikdat5)
```

**Format**

This data frame contains the following columns:

**length** the boundaries of grouping intervals.

**age1** the numbers of observation from each interval belonging to the first age group.

**age2** the numbers of observation from each interval belonging to the second age group.

**age3** the numbers of observation from each interval belonging to the third age group.

**age4** the numbers of observation from each interval belonging to the fourth age group.

**age5** the numbers of observation from each interval belonging to the fifth age group.

**Source**

Macdonald, P.D.M. and T.J. Pitcher (1979). *Age-groups from size-frequency data: a versatile and efficient method of analysing distribution mixtures*. Journal of the Fisheries Research Board of Canada 36, 987-1001.

**References**

Macdonald, P.D.M. (1987). *Analysis of length-frequency distributions*. In R.C. Summerfelt and G.E. Hall [editors], *Age and Growth of Fish*, Iowa State University Press, Ames, Iowa. pp 371-384.

<http://www.math.mcmaster.ca/peter/mix/demex/expike.html>

**Examples**

```
data(pikdat5)
```

---

pike65

*Length-Frequency Data for Heming Lake Pike*

---

**Description**

The raw data `pikeraw` give the lengths of 523 pike (*Esox lucius*). We grouped the lengths over 25 intervals to obtain this length-frequency data.

The `pike65` data frame has 25 rows and 2 columns.

**Usage**

```
data(pike65)
```

**Format**

This data frame contains the following columns:

**length** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

**Source**

Macdonald, P.D.M. and T.J. Pitcher (1979). *Age-groups from size-frequency data: a versatile and efficient method of analysing distribution mixtures*. Journal of the Fisheries Research Board of Canada 36, 987-1001.

**References**

Macdonald, P.D.M. (1987). *Analysis of length-frequency distributions*. In R.C. Summerfelt and G.E. Hall [editors], *Age and Growth of Fish*, Iowa State University Press, Ames, Iowa. pp 371-384.

<http://www.math.mcmaster.ca/peter/mix/demex/expike.html>

## Examples

```
data(pike65)
data(pikepar)
plot.mixdata(pike65)
fit <- mix(pike65, pikepar, "lnorm", constr = mixconstr(consigma = "CCV"), emsteps = 3)
plot(fit)
```

---

pike65sg

*Length-Frequency Data with Subsamples for Heming Lake Pike*

---

## Description

The raw data `pikeraw` give the lengths of 523 pike (*Esox lucius*), and there are known to be five age-groups in the sample. After grouping the data, we take subsamples from some intervals to determine the age group, and then obtain this data set.

The `pike65sg` data frame has 25 rows and 7 columns.

## Usage

```
data(pike65sg)
```

## Format

This data frame contains the following columns:

**length** the boundaries of grouping intervals.

**freq** the frequencies of observation falling into each interval.

**age1** the numbers of observation in the subsamples belonging to the first age group.

**age2** the numbers of observation in the subsamples belonging to the second age group.

**age3** the numbers of observation in the subsamples belonging to the third age group.

**age4** the numbers of observation in the subsamples belonging to the fourth age group.

**age5** the numbers of observation in the subsamples belonging to the fifth age group.

## Source

Macdonald, P.D.M. and T.J. Pitcher (1979). *Age-groups from size-frequency data: a versatile and efficient method of analysing distribution mixtures*. Journal of the Fisheries Research Board of Canada 36, 987-1001.

## References

Macdonald, P.D.M. (1987). *Analysis of length-frequency distributions*. In R.C. Summerfelt and G.E. Hall [editors], *Age and Growth of Fish*, Iowa State University Press, Ames, Iowa. pp 371-384.

<http://www.math.mcmaster.ca/peter/mix/demex/expike.html>

### Examples

```
data(pike65sg)
data(pikepar)
fit1 <- mix(pike65sg, pikepar, "gamma", mixconstr(consigma = "CCV"), usecondit = TRUE)
plot(fit1)
fit2 <- mix(pike65sg, pikepar, "gamma", usecondit = TRUE)
plot(fit2)
```

---

pikepar

*Starting Values of Parameters for the Pike Data*

---

### Description

Starting values of parameters for fitting a mixture distribution to the data set `pike65`.

The `pikepar` data frame has 5 rows and 3 columns.

### Usage

```
data(pikepar)
```

### Format

This data frame contains the following columns:

**pi** the starting values for proportions.

**mu** the starting values for means.

**sigma** the starting values for standard deviations.

### Source

Macdonald, P.D.M. and T.J. Pitcher (1979). *Age-groups from size-frequency data: a versatile and efficient method of analysing distribution mixtures*. Journal of the Fisheries Research Board of Canada 36, 987-1001.

### References

Macdonald, P.D.M. (1987). *Analysis of length-frequency distributions*. In R.C. Summerfelt and G.E. Hall [editors], *Age and Growth of Fish*, Iowa State University Press, Ames, Iowa. pp 371-384.

<http://www.math.mcmaster.ca/peter/mix/demex/expike.html>

### Examples

```
data(pikepar)
```

---

pikeraw

*A Sample of Pike Lengths*

---

### Description

The data give the lengths of 523 pike (*Esox lucius*), sampled in 1965 from Heming Lake, Manitoba, Canada. There are known to be five age-groups in the sample. For each fish, the age group is determined by scale reading.

The `pikeraw` data frame has 523 rows and 2 columns.

### Usage

```
data(pikeraw)
```

### Format

This data frame contains the following columns:

**length** the lengths of 523 pike

**age** the age groups of 523 pike

### Source

Macdonald, P.D.M. and T.J. Pitcher (1979). *Age-groups from size-frequency data: a versatile and efficient method of analysing distribution mixtures*. Journal of the Fisheries Research Board of Canada 36, 987-1001.

### References

Macdonald, P.D.M. (1987). *Analysis of length-frequency distributions*. In R.C. Summerfelt and G.E. Hall [editors], *Age and Growth of Fish*, Iowa State University Press, Ames, Iowa. pp 371-384.

<http://www.math.mcmaster.ca/peter/mix/demex/expike.html>

### Examples

```
data(pikeraw)
```

plot.mix

*Mix Object Plotting***Description**

A function for plotting of Mix objects. It is called via the generic function `plot`.

**Usage**

```
## S3 method for class 'mix':
plot(x, mixpar = NULL, dist = "norm", root = FALSE, ytop = NULL,
      clwd = 1, main, sub, xlab, ylab, bty, BW = FALSE, ...)
```

**Arguments**

<code>x</code>	an object of class "mix", usually, the results returned by the model fitting function <code>mix</code> .
<code>mixpar</code>	NULL or a data frame containing the values for parameters of component distributions, which are, in order, the proportions, means, and standard deviations.
<code>dist</code>	the distribution of components, it can be "norm", "lnorm", "gamma", "weibull", "binom", "nbinom" and "pois".
<code>root</code>	if TRUE, a hanging rootogram will be displayed.
<code>ytop</code>	a scalar which determines the top of the y-axis.
<code>clwd</code>	a positive number denoting line width, defaulting to 1.
<code>main</code>	an overall title for the plot.
<code>sub</code>	a subtitle for the plot.
<code>xlab</code>	a title for the x-axis.
<code>ylab</code>	a title for the y-axis.
<code>bty</code>	A character string which determined the type of box which is drawn about plots. If <code>bty</code> is one of "o", "l", "7", "c", "u", or "]" the resulting box resembles the corresponding upper case letter. A value of "n" suppresses the box.
<code>BW</code>	logical; if TRUE the plot will be drawn in black and white.
<code>...</code>	additional arguments to the function <code>plot.default</code> .

**Details**

If the argument `x` gives an object of class "mix", the plot will be a histogram for the grouped data which come from the element `mixdata` of `x`. Although the leftmost (first) and rightmost (`mth`) intervals are always open-ended, on the histogram the first interval is shown as being twice the width of the second interval and the `mth` is shown as being twice the width of the `m - 1st` interval. When the fitted distribution is one of "lnorm", "gamma" and "weibull", the left boundary of the first interval will be taken zero since negative values and zeroes are not allowed for these distribution. For the distributions "binom", "nbinom" and "pois" negative data are

not permitted, so the left boundary of the first interval is taken -0.5. The component distributions weighted by their respect proportions and the mixture distribution are computed by the estimated parameter values from the element `parameters` of `x`, and superimposed on the histogram. The distribution of components will be taken the value of the element `distribution`. If `sub`, `xlab`, `ylab` and `bty` are not specified, the default values will be used. The positions of the means are indicated with triangles. When the argument `root` is `TRUE`, a hanging rootogram will be displayed, that is, if only grouped data are given, this option plots the histogram with the square root of relative frequency on the y-axis. If there is a model as well as data, not only is the y-axis the square root of relative frequency, also the bars of the histogram, instead of rising from 0, are shifted up or down so that the mid-point of the top of the bar is exactly on the curve indicating the mixture distribution and the bottom of the bar may therefore be above or below the x-axis. If the bar goes below the x-axis, the portion below is shown as a blue rectangle. If the bar does not reach the x-axis, the space between the bottom of the bar and the x-axis is shown as a blue rectangle. If the blue rectangles are almost above or below in an area of the x-axis, we may say that the mixture curve around that area is not fitting well.

### See Also

[mixparam](#) for organizing the parameter values, [mix](#) for fitting mixture model, [plot.mixdata](#) for plotting Mixdata objects, [plot.default](#) for additional arguments.

### Examples

```
data(pike65)
data(pikepar)
fit1 <- mix(pike65, pikepar, "lnorm", constr = mixconstr(consigma = "CCV"), emsteps = 3)
plot(fit1)
plot(fit1, root = TRUE)
data(bindat)
data(binpar)
fit2 <- mix(bindat, binpar, "binom", constr = mixconstr(consigma = "BINOM", size = c(20, 20),
plot(fit2)
plot(fit2, root = TRUE)
```

---

plot.mixdata

*Mixdata Object Plotting*

---

### Description

A function for plotting of Mixdata objects. It is called via the generic function `plot`.

### Usage

```
## S3 method for class 'mixdata':
plot(x, mixpar = NULL, dist = "norm", root = FALSE, ytop = NULL,
      clwd = 1, main, sub, xlab, ylab, bty, ...)
```

**Arguments**

x	an object of class "mixdata", usually, the results returned by the function <code>mixgroup</code> .
mixpar	NULL or a data frame containing the values for parameters of component distributions, which are, in order, the proportions, means, and standard deviations.
dist	the distribution of components, it can be "norm", "lnorm", "gamma", "weibull", "binom", "nbinom" and "pois".
root	if TRUE, a hanging rootogram will be displayed.
ytop	a scalar which determines the top of the y-axis.
clwd	a positive number denoting line width, defaulting to 1.
main	an overall title for the plot.
sub	a subtitle for the plot.
xlab	a title for the x-axis.
ylab	a title for the y-axis.
bty	A character string which determined the type of box which is drawn about plots. If <code>bty</code> is one of "o", "l", "7", "c", "u", or "]" the resulting box resembles the corresponding upper case letter. A value of "n" suppresses the box.
...	additional arguments to the function <code>plot.default</code> .

**Details**

If the argument `mixpar` is NULL, then only the histogram of the data will be displayed; if `mixpar` gives the values of parameters, the component distributions and the mixture distribution are computed from the parameter values and superimposed on the histogram.

**See Also**

`plot.mix` for plotting Mix objects, `plot.default` for additional arguments.

**Examples**

```
data(cassie)
as.mixdata(cassie) # if the result isn't `NULL`, then cassie is mixed data
plot.mixdata(cassie)
data(pikeraw)
data(pikepar)
pikemd <- mixgroup(pikeraw, breaks = c(0, seq(19.75, 65.75, 2), 80))
plot(pikemd)
plot(pikemd, pikepar, "lnorm")
fit <- mix(pikemd, pikepar, "lnorm", constr = mixconstr(consigma = "CCV"), emsteps = 3)
plot(fit)
plot(pikemd, pikepar, "lnorm", root = TRUE)
plot(fit, root = TRUE)
```

---

`poisdat`*Grouped Poisson Data*

---

**Description**

The `poisdat` data frame has 15 rows and 2 columns.

**Usage**

```
data(poisdat)
```

**Format**

This data frame contains the following columns:

**X** the boundaries of grouping intervals.

**samppois** the frequencies of observation falling into each interval.

**Examples**

```
data(poisdat)
plot.mixdata(poisdat)
```

---

`poispar`*Starting Values of Parameters for the Poisson Data Set*

---

**Description**

Starting values of parameters for fitting a mixture distribution to the data set `poisdat`.

The `poispar` data frame has 4 rows and 3 columns.

**Usage**

```
data(poispar)
```

**Format**

This data frame contains the following columns:

**pi** the starting values for proportions.

**mu** the starting values for means.

**sigma** the starting values for standard deviations.

**Examples**

```
data(poispar)
```

---

`print.mix`*Print Mix Object*

---

## Description

`print.mix` is a function which prints objects of class "mix" and returns it invisibly. It is called via the generic function `print`.

## Usage

```
## S3 method for class 'mix':  
print(x, digits = 4, ...)
```

## Arguments

<code>x</code>	an object of class "mix", usually, the results returned by the model fitting function <code>mix</code> .
<code>digits</code>	how many significant digits are to be used.
<code>...</code>	further arguments passed to or from other methods.

## Details

This function only prints information about the mixture model, which are the estimated parameters of the mixture, the distribution of components and the constraints on the parameters. Also, the values for the parameters are rounded to the specified number of decimal places (default 4). The whole object can be printed out using the function `print.default`.

## See Also

`mix` for model fitting. `print.default` for printing the whole object.

## Examples

```
data(pike65)  
data(pikepar)  
fit <- mix(pike65, pikepar, "gamma", mixconstr(consigma = "CCV"), emsteps = 3)  
fit  
print(fit)  
print.mix(fit)  
print.default(fit)
```

---

`summary.mix`*Summarizing Mixture Model Fits*

---

## Description

`summary` method for class "mix". It is called via the generic function `summary`.

## Usage

```
## S3 method for class 'mix':  
summary(object, digits = 4, ...)
```

## Arguments

<code>object</code>	an object of class "mix", usually, the results returned by the model fitting function <code>mix</code> .
<code>digits</code>	how many significant digits are to be used.
<code>...</code>	additional arguments affecting the summary produced.

## Value

A list containing the following items:

<code>parameters</code>	a data frame containing the values for parameters of component distributions, which are, in order, the proportions, means, and standard deviations.
<code>standard errors</code>	a data frame giving the standard errors of estimated parameters.
<code>anova table</code>	analysis of variance table for the <code>mixobj</code> , that is, the results from the function <code>anova.mix</code> .

## See Also

`mix` for model fitting, `summary` for summarizing other kinds of object. `anova.mix` for information about anova table.

## Examples

```
data(pike65)  
data(pikepar)  
fit <- mix(pike65, pikepar, "lnorm", mixconstr(consigma = "CCV"), emsteps = 3)  
fit  
summary(fit)
```

---

testconstr	<i>Check Constraints</i>
------------	--------------------------

---

## Description

Check if constraints on parameters are valid. See the reference for details.

## Usage

```
testconstr(mixdat, mixpar, dist, constr)
```

## Arguments

mixdat	a data frame containing grouped data, whose first column should be right boundaries of grouping intervals, whose second column should consist of the frequencies indicating numbers of observations falling into each interval. If conditional data are available, this data frame should have $k + 2$ columns, where $k$ is the number of components, whose element in row $j$ and column $i + 2$ is the number of observations from the $j$ th interval belonging to the $i$ th component.
mixpar	a data frame containing the values for parameters of component distributions, which are, in order, the proportions, means, and standard deviations.
dist	the distribution of components, it can be one of "norm", "lnorm", "gamma", "weibull", "binom", "nbinom" and "pois".
constr	a list of constraints on parameters of component distributions. See function <a href="#">mixconstr</a> .

## Value

If the constraints are valid, this function will give a logical value TRUE. If not, it will give an error message to illustrate the reason.

## References

Macdonald, P.D.M. and Green, P.E.J. (1988) *User's Guide to Program MIX: An Interactive Program for Fitting Mixtures of Distributions*. ICHTHUS DATA SYSTEMS.

## See Also

[mixgroup](#) for grouping data, [mixparam](#) for organizing the parameter values, [mixconstr](#) for constructing constraints.

## Examples

```
## Not run:
testconstr(pike65, pikepar, "lnorm", constr = mixconstr(consigma = "CCV"))
testconstr(bindat, binpar, "binom", constr = mixconstr())
testconstr(bindat, binpar, "binom", constr = mixconstr(consigma = "BINOM"))
testconstr(bindat, binpar, "pois", constr = mixconstr(conmu = "MEQ", consigma = "POIS"))
## End(Not run)
```

---

weibullpar

*Compute Shape and Scale Parameters for Weibull Distribution*

---

## Description

Compute the parameters shape and scale for Weibull distribution given the mean, standard deviation and location.

## Usage

```
weibullpar(mu, sigma, loc = 0)
```

## Arguments

mu	the mean of weibull distribution.
sigma	the standard deviation of weibull distribution.
loc	the location parameter of weibull distribution defaulting to 0.

## Value

A data frame containing three parameters, which are, in order, shape, scale, and location.

## See Also

[weibullparinv](#) for computing mean and standard deviation from the parameters shape, scale and location.

## Examples

```
weibullpar(2, 1.2)
weibullpar(2, 1.2, 1)
```

---

`weibullparinv`*Compute the Mean and Standard Deviation of Weibull Distribution*

---

**Description**

Compute mean and standard deviation of weibull distribution given the values of shape, scale and location.

**Usage**

```
weibullparinv(shape, scale, loc = 0)
```

**Arguments**

<code>shape</code>	the shape parameter of weibull distribution.
<code>scale</code>	the scale parameter of weibull distribution.
<code>loc</code>	the location parameter of weibull distribution defaulting to 0.

**Value**

A data frame containing three parameters, which are, in order, mean, standard deviation and location.

**See Also**

[weibullpar](#) for computing the parameters shape and scale from mean and standard deviation.

**Examples**

```
weibullparinv(weibullpar(2, 1.2)$shape, weibullpar(2, 1.2)$scale)
```

# Index

- \*Topic **attribute**
  - mixdata, 13
- \*Topic **classes**
  - mixdata, 13
- \*Topic **datasets**
  - bindat, 2
  - binpar, 3
  - cassie, 4
  - expdat, 6
  - fiftn80, 7
  - normals, 16
  - pearson, 17
  - pearsonpar, 17
  - pikdat5, 18
  - pike65, 19
  - pike65sg, 20
  - pikepar, 21
  - pikeraw, 22
  - poisdat, 26
  - poispar, 26
- \*Topic **dplot**
  - plot.mix, 23
  - plot.mixdata, 24
- \*Topic **misc**
  - conditdat, 5
  - groupstats, 9
  - mixconstr, 12
  - mixgroup, 14
  - mixparam, 15
  - testconstr, 29
  - weibullpar, 30
  - weibullparinv, 31
- \*Topic **models**
  - anova.mix, 1
  - coef.mix, 4
  - fitted.mix, 8
  - mix, 10
  - summary.mix, 28
- \*Topic **print**
  - print.mix, 27
- \*Topic **utilities**
  - conditdat, 5
  - groupstats, 9
  - mixconstr, 12
  - mixgroup, 14
  - mixparam, 15
  - testconstr, 29
  - weibullpar, 30
  - weibullparinv, 31
- anova, 2
- anova.mix, 1, 28
- as.mixdata (*mixdata*), 13
- bindat, 2
- binpar, 3
- cassie, 4
- coef, 4
- coef.mix, 4
- conditdat, 5
- expdat, 6
- fiftn80, 7
- fitted, 8
- fitted.mix, 8
- groupstats, 9
- hist, 15
- is.mixdata, 15
- is.mixdata (*mixdata*), 13
- mix, 2, 4, 5, 8, 10, 23, 24, 27, 28
- mixconstr, 10, 11, 12, 15, 29
- mixdata, 13
- mixgroup, 6, 9, 11, 13, 14, 15, 25, 29
- mixparam, 9, 11, 13, 15, 15, 24, 29

nlm, [10, 11](#)  
normals, [16](#)

pearson, [17](#)  
pearsonpar, [17](#)  
pikdat5, [18](#)  
pike65, [19](#)  
pike65sg, [20](#)  
pikepar, [21](#)  
pikeraw, [22](#)  
plot.default, [23–25](#)  
plot.mix, [23, 25](#)  
plot.mixdata, [24, 24](#)  
poisdat, [26](#)  
poispar, [26](#)  
print, [27](#)  
print.default, [27](#)  
print.mix, [27](#)

summary, [28](#)  
summary.mix, [28](#)

testconstr, [29](#)

weibullpar, [30, 31](#)  
weibullparinv, [30, 31](#)