

# Package: LifeHist (via r-universe)

August 28, 2024

**Type** Package

**Title** Life History Models of Individuals

**LazyLoad** yes

**LazyData** yes

**Version** 1.0-1

**Date** 2015-08-31

**Author** Ruben H. Roa-Ureta

**Maintainer** Ruben H. Roa-Ureta <ruben.roa.ureta@mail.com>

**Depends** R (>= 3.0.0), Hmisc

**Imports** optimx (>= 2013.8.6), BB

**Description** Likelihood-based estimation of individual growth and sexual maturity models for organisms, usually fish and invertebrates. It includes methods for data organization, plotting standard exploratory and analytical plots, predictions.

**License** GPL (>= 2)

**NeedsCompilation** no

**Date/Publication** 2015-09-03 12:20:03

**Repository** <https://ruben-hernan.r-universe.dev>

**RemoteUrl** <https://github.com/cran/LifeHist>

**RemoteRef** HEAD

**RemoteSha** 01c5f5f5114d6a168371f26555e61ae943cbd9

## Contents

LifeHist-package . . . . .	2
as.GroAgeData . . . . .	4
BlackAngler.agematdat . . . . .	6
BlackAngler.lenmatdat . . . . .	7
BlackAngler.wgtmatdat . . . . .	7

deltamethod . . . . .	8
GroAgeData . . . . .	9
GroAgeExp . . . . .	10
groageexp . . . . .	10
GroAgeFit . . . . .	12
GroAgeMod . . . . .	15
groagemod . . . . .	16
ksbream . . . . .	17
MatCount . . . . .	18
MatData . . . . .	19
MatFit . . . . .	20
plot.GroAgeData . . . . .	22
plot.GroAgeExp . . . . .	22
plot.GroAgeMod . . . . .	23
plot.MatData . . . . .	24
<b>Index</b>	<b>25</b>

---

LifeHist-package	<i>Life History Models of Individuals</i>
------------------	---

---

## Description

Likelihood-based estimation of individual growth and sexual maturity models for organisms, usually fish and invertebrates. It includes methods for data organization, plotting standard exploratory and analytical plots, predictions.

## Details

The DESCRIPTION file:

```

Package:      LifeHist
Type:         Package
Title:        Life History Models of Individuals
LazyLoad:    yes
LazyData:    yes
Version:      1.0-1
Date:         2015-08-31
Author:       Ruben H. Roa-Ureta
Maintainer:   Ruben H. Roa-Ureta <ruben.roa.ureta@mail.com>
Depends:     R (>= 3.0.0), Hmisc
Imports:      optimx (>= 2013.8.6), BB
Description:  Likelihood-based estimation of individual growth and sexual maturity models for organisms, usually fish and in
License:      GPL (>= 2)

```

Index of help topics:

BlackAngler.agematdat	Black Anglerfish Age, Sex, Month, Maturity Stage Data
BlackAngler.lenmatdat	Black Anglerfish Length, Sex, Month, Maturity Stage Data
BlackAngler.wgtmatdat	Black Anglerfish Body Weight, Sex, Month, Maturity Stage Data
GroAgeData	Class Attribute of LifeHist Data Object
GroAgeExp	Class Attribute of LifeHist Exploratory Model Object
GroAgeFit	Fit Growth Model with Age and Length Data by Maximum Likelihood
GroAgeMod	Class Attribute of LifeHist Fitted Model Object
LifeHist-package	Life History Models of Individuals
MatCount	Build Databases for Sexual Maturity Modeling by Logistic Regression
MatData	Class Attribute of LifeHist Data Object
MatFit	Wrapper Function to Estimate a 2- or 3-Parameters Logistic Regression of Sexual Maturity
as.GroAgeData	Create a Data Object to Fit Individual Growth Models with Age-Length Observations
deltamethod	First-order Taylor Series Expansion of Functions of Random Variables
groageexp	Exploratory Growth Modeling with Age and Length Data
groagemod	Predictions from an Object of Class GroAgeMod
ksbream	King Soldier Bream Individual Biological Data
plot.GroAgeData	Exploratory Analysis of Ager and Length Individual Data
plot.GroAgeExp	Exploratory Modeling of Individual Growth with Age and Length Data
plot.GroAgeMod	Examination of Results from Fitting Growth Models
plot.MatData	Exploratory Analysis of a Logistic Model on a Graphical Display

Create data objects for maturity and growth modeling with `MatCount()` and `as.GroAgeData()`, respectively. Fit sexual maturity models with `MatFit()` for two types of models and any numerical method accepted by `optimx()` from package `optimx`. Explore raw growth data with generic function `plot()` applied on an object of class `'GroAgeData'` created by `as.GroAgeData()`. Create predictions with initial parameter values for growth models using `groageexp()` and examine how good these initial values are with generic function `plot()` applied on an object of class `'GroAgeExp'` created by `groageexp()`. Fit any of four type of models with any of five likelihood models and any numerical method accepted by `optimx()` from package `optimx`, using `GroAgeFit()`. Graphically examines results with generic function `plot()` applied on an object of class `'GroAgeMod'` created by `GroAgeFit()`.

**Author(s)**

Ruben H. Roa-Ureta

Maintainer: Ruben H. Roa-Ureta <ruben.roa.ureta@mail.com>

**References**

R. Roa, B. Ernst, F. Tapia. 1999. Estimation of size at sexual maturity: evaluation of analytical and resampling procedures. *Fishery Bulletin (US)* 97:570-580.

Arkhipkin, A., Roa-Ureta, R. 2005. Identification of growth models for squids. *Marine and Freshwater Research* 56:371-386.

Roa-Ureta, R. 2010. A Likelihood-based Model of Fish Growth with Multiple Length Frequency Data. *Journal of Agricultural, Biological and Environmental Statistics* 15:416-429.

**Examples**

```
#See examples for functions MatFit() and GroAgeFit()
```

---

as.GroAgeData	<i>Create a Data Object to Fit Individual Growth Models with Age-Length Observations</i>
---------------	--

---

**Description**

From a data.frame identify columns with data on length and age from a single sex category, two sexes, or two sexes plus unsexed individuals, and optionally include date, body weight, liver weight, and gonad weight, to create a data object with a class attribute.

**Usage**

```
as.GroAgeData(x, sex, maleskey = NULL, femaleskey = NULL, colsex = NULL,
              colage, collen, colbw = NULL, colliver = NULL, colgonad = NULL,
              coldate = NULL, lentype, unitsage, unitslen, unitsbw = NULL,
              unitsliver = NULL, unitsgonad = NULL, spec)
```

**Arguments**

x	Data.frame containing the data, where each line is an individual.
sex	Character, either "Females", "Males", "Pooled", "Both", or "Total".
maleskey	Usually a numeric key indicating whether the individual is male.
femaleskey	Usually a numeric key indicating whether the individual is female.
colsex	Integer. The column in 'x' with the sex key.
colage	Integer. The column in 'x' with the age data.
collen	Integer. The column in 'x' with the length data.
colbw	Integer. Optionally, the column in 'x' with the body weight data.

colliver	Integer. Optionally, the column in 'x' with the liver weight data.
colgonad	Integer. Optionally, the column in 'x' with the gonad weight data.
coldate	Integer. Optionally, the column in 'x' with the collection date data.
lentye	Character. In the case of fish, this indicates whether the length is total, fork, or standard.
unitsage	Character. The units of age.
unitslen	Character. The units of length.
unitsbw	Character. Optionally, the units of body weight.
unitsliver	Character. Optionally, the units of liver weight.
unitsgonad	Character. Optionally, the units of gonad weight.
spec	Character. A species identification key.

### Details

When 'sex' is "Females" or "Males", the data.frame 'x' contains data for a single sex. When 'sex' is "Pooled" the sex has not been determined and both sexes might be mixed. When 'sex' is "Both" all individuals have been sexed. Finally, when 'sex' is 'Total' some individuals have been sexed but some have been not so there is a third sex category which implicitly corresponds to unsexed individuals. For instance this could happen with very young individuals for which their sex could not be determined.

### Value

A list with class attribute 'GroAgeData' with the following components:

Properties	A data.frame with properties of the data, such as the species key, the type of sex, units of age, units of length, and the other optional properties
Sex 1	A data.frame with an individual key, the age, the length, and optionally other individual data
Sex 2	Optionally, the same as the previous component but for the other sex
Sex 3	Optionally, the same as the previous components but for the unsexed individuals

### Note

NAs are allowed in all data column and later ignored in the modeling functions.

### Author(s)

Ruben H. Roa-Ureta

### Examples

```
data(ksbream)
KSBream.AgeLen <- as.GroAgeData(x=ksbream,
                                sex="Total",
                                maleskey=1,
                                femaleskey=2,
```

```

                                coldate=1,
                                colsex=9,
                                colage=11,
                                collen=2,
                                colbw=5,
                                colliver=7,
                                colgonad=8,
                                lentype="Total",
                                unitsage="Years",
                                unitslen="mm",
                                unitsbw="g",
                                unitsliver="g",
                                unitsgonad="g",
                                spec="KSBream")
plot(KSBream.AgeLen)

```

---

BlackAngler.agematdat *Black Anglerfish Age, Sex, Month, Maturity Stage Data*

---

### Description

Black anglerfish individual-based data from the Bay of Bizcay sampled from the catch of an industrial fishing fleet.

### Usage

```
data("BlackAngler.agematdat")
```

### Format

A data frame with 711 observations on the following 4 variables.

pred Integer, age in years

sex Integer, 1 for males and 2 for females

month Integer, the month of capture

stages Integer, stages of maturity encompassing a gradual advance from juvenile to sexually mature individual

### Details

Age has been fixed by biologists examining rings in otoliths. Maturity stages have been fixed by biologists using macroscopic aspect of the fish gonads.

### Source

<http://www.azti.es>

### Examples

```
data(BlackAngler.agematdat)
```

---

BlackAngler.lenmatdat *Black Anglerfish Length, Sex, Month, Maturity Stage Data*

---

**Description**

Black anglerfish individual-based data from the Bay of Bizcay sampled from the catch of an industrial fishing fleet.

**Usage**

```
data("BlackAngler.lenmatdat")
```

**Format**

A data frame with 1367 observations on the following 4 variables.

pred Numeric, total fish length in cm

sex Integer, 1 for males and 2 for females

month Integer, the month of capture

stages Integer, stages of maturity encompassing a gradual advance from juvenile to sexually mature individual

**Details**

Maturity stages have been fixed by biologists using macroscopic aspect of the fish gonads.

**Source**

<http://www.azti.es>

**Examples**

```
data(BlackAngler.lenmatdat)
```

---

BlackAngler.wgtmatdat *Black Anglerfish Body Weight, Sex, Month, Maturity Stage Data*

---

**Description**

Black anglerfish individual-based data from the Bay of Bizcay sampled from the catch of an industrial fishing fleet.

**Usage**

```
data("BlackAngler.wgtmatdat")
```

**Format**

A data frame with 479 observations on the following 4 variables.

pred Numeric, fish body weight length in g

sex Integer, 1 for males and 2 for females

month Integer, the month of capture

stages Integer, stages of maturity encompassing a gradual advance from juvenile to sexually matur individual

**Details**

Maturity stages have been fixed by biologists using macroscopic aspect of the fish gonads.

**Source**

<http://www.azti.es>

**Examples**

```
data(BlackAngler.wgtmatdat)
```

---

deltamethod

*First-order Taylor Series Expansion of Functions of Random Variables*

---

**Description**

The delta method for approximating the standard error of a transformation  $g(X)$  of a random variable  $X = (x_1, x_2, \dots)$ , given estimates of the mean and covariance matrix of  $X$ .

**Usage**

```
deltamethod(g, mean, cov, ses = TRUE)
```

**Arguments**

g	A formula representing the transformation. The variables must be labelled $x_1, x_2, \dots$ . For example, $\sim 1 / (x_1 + x_2)$ . If the transformation returns a vector, then a list of formulae representing $(g_1, g_2, \dots)$ can be provided, for example <code>list( <math>\sim x_1 + x_2, \sim x_1 / (x_1 + x_2)</math> )</code> .
mean	The estimated mean of $X$ .
cov	The estimated covariance matrix of $X$ .
ses	If TRUE, then the standard errors of $g_1(X), g_2(X), \dots$ are returned. Otherwise the covariance matrix of $g(X)$ is returned.



**Details**

This function was copied from package `msm`. It is used in `CatDyn` to backtransform from the logarithm because `CatDyn` parameters are all estimated in the log scale to improve numerical performance. For more details see the help pages for function `deltamethod` of package `msm`.

**Value**

A vector containing the standard errors of  $g_1(X)$ ,  $g_2(X)$ , ... or a matrix containing the covariance of  $g(X)$ .

**Author(s)**

C. H. Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

**Examples**

```
#See the examples in package msm.
```

---

GroAgeData

*Class Attribute of LifeHist Data Object*

---

**Description**

To be used by `plot.GroAgeData()` to examine raw length, age data before modeling.

**Usage**

```
GroAgeData(x, ...)
```

**Arguments**

<code>x</code>	A list object coming from <code>as.GroAgeData()</code> .
<code>...</code>	Not used.

**Value**

A class attribute

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
#See example for as.GroAgeData().
```

---

GroAgeExp

*Class Attribute of LifeHist Exploratory Model Object*

---

### Description

To be used by plot.GroAgeExp() to examine the quality of initial parameter values before fitting a growth model.

### Usage

```
GroAgeExp(x, ...)
```

### Arguments

x	A list object coming from groageexp().
...	Not used.

### Value

A class attribute

### Author(s)

Ruben H. Roa-Ureta

### Examples

```
#See examples for plot.AgeGroExp()
```

---

groageexp

*Exploratory Growth Modeling with Age and Length Data*

---

### Description

For a data object of class 'GroAgeData' examine the goodness of initial guessed values of model parameters by making predictions and calculating residuals.

### Usage

```
groageexp(x, m, unsex.action = NULL, par, distr)
```

**Arguments**

<code>x</code>	An object of class 'GroAgeData'.
<code>m</code>	Character, the specific growth model to be examined, either 'vonBer' (von Bertalanffy), 'schnute1', 'schnute2', or 'schnute3'
<code>unsex.action</code>	Character, optionally, when there are unsexed individual, what to do with these, either 'split' (random assignment of half to males and half to females), 'males' (complete assignment to males), 'females' (complete assignment to females), or 'ignore'.
<code>par</code>	Numeric, the logarithm of the vector of initial parameter values.
<code>distr</code>	Character, the distribution of the data, either 'apnormal' (adjusted profile approximation to the normal likelihood), 'aplnormal' adjusted profile approximation to the lognormal likelihood), 'normal', 'lognormal', or 'gamma'.

**Details**

When the distribution of the data is 'normal', 'lognormal', or 'gamma', an additional dispersion parameter has to be provided along with the growth model parameters, and placed at the end of the parameter vector.

**Value**

A list with class attribute 'GroAgeExp' with the following components:

Properties	Properties of the data, passed from 'x'
Data	A data.frame with age, observed length, predicted length, and 'deviance' residuals
Data	Optionally, the same as previous component but for the other sex

**Note**

Model types 'schnute1', 'schnute2', 'schnute3' have 5, 3, and 2 parameters respectively. 'schnute1' is described in Arkhipkin and Roa-Ureta (2005) whereas 'schnute2' and 'schnute3' are described in Roa-Ureta (2010).

**Author(s)**

Ruben H. Roa-Ureta

**References**

- Arkhipkin, A., Roa-Ureta, R. 2005. Identification of growth models for squids. *Marine and Freshwater Research* 56:371-386.
- Roa-Ureta, R. 2010. A likelihood-based model of fish growth with multiple length frequency data. *Journal of Agricultural, Biological and Environmental Statistics* 15:416-429

**Examples**

```

#Schnute - Normal
data(ksbream)
KSBream.AgeLen <- as.GroAgeData(x=ksbream,
                                sex="Total",
                                maleskey=1,
                                femaleskey=2,
                                coldate=1,
                                colsex=9,
                                colage=11,
                                collen=2,
                                colbw=5,
                                colliver=7,
                                colgonad=8,
                                lentype="Total",
                                unitsage="Years",
                                unitslen="mm",
                                unitsbw="g",
                                unitsliver="g",
                                unitsgonad="g",
                                spec="KSBream")

#par ini
l1.f <- 175
a1.f <- 1
mu.f <- 250
g1.f <- 0.15
g2.f <- 0.1
l1.m <- 175
a1.m <- 1
mu.m <- 210
g1.m <- 0.25
g2.m <- 0.1
psi.f <- 0.5
psi.m <- 0.45
par.ini <- log(c(l1.f,a1.f,mu.f,g1.f,g2.f,l1.m,a1.m,mu.m,g1.m,g2.m,psi.f,psi.m))
KSBream.AgeLen.exp <- groageexp(x=KSBream.AgeLen,
                                unsex.action="split",
                                m="schnute1",
                                par=par.ini,
                                distr="normal")

plot(KSBream.AgeLen.exp)

```

**Description**

A wrapper and post-processing tool that calls `optimx()` (from package `optimx`) with any of four possible versions of the growth models and any of 5 possible versions of likelihood models (as internal functions), in addition to possible multiple numerical methods for optimization and then it post-processes `optimx()` results and join all results in a list of lists.

**Usage**

```
GroAgeFit(x, m, unsex.action = NULL, par, distr, method, control = list(),
          hessian = TRUE, itnmax)
```

**Arguments**

<code>x</code>	An object of class <code>GroAgeData</code> . See <code>as.GroAgeData</code> .
<code>m</code>	Character, the specific growth model to be examined, either <code>'vonBer'</code> (von Bertalanffy), <code>'schnute1'</code> , <code>'schnute2'</code> , or <code>'schnute3'</code>
<code>unsex.action</code>	Character, optionally, when there are unsexed individual, what to do with these, either <code>'split'</code> (random assignment of half to males and half to females), <code>'males'</code> (complete assignment to males), <code>'females'</code> (complete assignment to females), or <code>'ignore'</code> .
<code>par</code>	Numeric, the logarithm of the vector of initial parameter values.
<code>distr</code>	Character, the distribution of the data, either <code>'apnormal'</code> (adjusted profile approximation to the normal likelihood), <code>'aplnormal'</code> adjusted profile approximation to the lognormal likelihood), <code>'normal'</code> , <code>'lognormal'</code> , or <code>'gamma'</code> .
<code>method</code>	Character. Any method accepted by <code>optimx()</code> can be used, but some may return warnings or errors.
<code>control</code>	A list of control arguments to be passed to <code>optimx()</code> .
<code>hessian</code>	Logical. Defaults to <code>TRUE</code> . If set to <code>FALSE</code> all numerical methods tried will fail.
<code>itnmax</code>	Numeric. Maximum number of iterations, to pass to <code>optimx()</code> .

**Details**

When `'x'` has been set with `'sex'` equal to `"Both"` or `"Total"` then `GroAgeFit` will automatically fit separate models for females, males, and for both sexes pooled.

Care should be taken in selecting good initial values to pass in the `par` argument. To accomplish this `LifeHist` includes the `GroAgeExp` class, and the `groageexp()` and the `plot.GroAgeExp()` functions to graphically fine tune the initial values for model parameters.

Initial parameter values must be passed log-transformed by the user. `GroAgeFit()` will backtransform the maximum likelihood estimates and its numerical Hessian matrix without user intervention using the delta method.

The difference between `"normal"` and `"apnormal"`, `"lognormal"` and `"aplnormal"` is that in the former the dispersion parameters is included in the likelihood function and it is a free parameter to be estimated along with the parameters of the generalized depletion model (and therefore an initial value for the dispersion has to be provided) whereas in the latter the dispersion is eliminated by using the adjusted profile likelihood approximation.

**Value**

A list of length four with a class attribute `'groage'`.

<code>Data</code>	The original data and its properties
-------------------	--------------------------------------

Initial	Initial parameter values in their original scale
Methods	The numerical methods passed to optimx()
Model	A list of length equal to one when 'sex' is 'Females', 'Males', or 'Pooled' or of length equal to three when 'sex' is 'Both' or 'Total'. Each component is a list of length equal to the number of numerical methods. The list for each numerical method is in turn a list with the type of growth model, the chosen distribution for the data, the integer code describing the success or not of convergence returned by the method, the Karush Kuhn Tucker conditions, hopefully TRUE and TRUE, the value of the Akaike Information Criterion, not comparable between different distributions, the back-transformed (from log) maximum likelihood estimates, the numerical gradients at each maximum likelihood estimate, the standard errors of backtransformed (from log) maximum likelihood estimates, and the correlation matrix of the back-transformed (from log) maximum likelihood estimates.

**Note**

Some effort has been made to avoid being kicked out of numerical optimization by just one numerical method that fails, so that optimization continues with other methods, but there may remain some cases when the whole optimization process is aborted by failure in just one method. Try taking out some suspicious methods and optimize again.

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
data(ksbream)
KSBream.AgeLen <- as.GroAgeData(x=ksbream,
                                sex="Total",
                                maleskey=1,
                                femaleskey=2,
                                coldate=1,
                                colsex=9,
                                colage=11,
                                collen=2,
                                colbw=5,
                                colliver=7,
                                colgonad=8,
                                lentype="Total",
                                unitsage="Years",
                                unitslen="mm",
                                unitsbw="g",
                                unitsliver="g",
                                unitsgonad="g",
                                spec="KSBream")

l1.f <- 175
a1.f <- 1
mu.f <- 250
```

```

g1.f <- 0.15
g2.f <- 0.1
l1.m <- 175
a1.m <- 1
mu.m <- 210
g1.m <- 0.25
g2.m <- 0.1
psi.f <- 0.5
psi.m <- 0.45
par.ini <- log(c(l1.f,a1.f,mu.f,g1.f,g2.f,l1.m,a1.m,mu.m,g1.m,g2.m,psi.f,psi.m))
KSBream.AgeLen.fit.n <- GroAgeFit(x=KSBream.AgeLen,
                                m="schnute1",
                                unsex.action="split",
                                par=par.ini,
                                distr="normal",
                                method=c("CG", "spg", "Nelder-Mead"),
                                itnmax=100)
KSBream.AgeLen.pred.n.spg <- groagemod(x=KSBream.AgeLen.fit.n,method="spg")
plot(KSBream.AgeLen.pred.n.spg)

```

---

GroAgeMod

*Class Attribute of LifeHist Fitted Model Object*


---

### Description

To be used by plot.GroAgeMod() to examine results of fitting a growth model.

### Usage

```
GroAgeMod(x, ...)
```

### Arguments

x	A list object coming from GroAgeFit().
...	Not used.

### Value

A class attribute

### Author(s)

Ruben H. Roa-Ureta

### Examples

```
#See examples for plot.AgeGroMod()
```

---

`groagemod`*Predictions from an Object of Class GroAgeMod*

---

**Description**

For a data object of class 'GroAgeMod' examine the results of model fit by making predictions and calculating residuals.

**Usage**

```
groagemod(x, method)
```

**Arguments**

<code>x</code>	An object of class 'GroAgeMod'.
<code>method</code>	Character, one of the numerical methods used to fit the model with GroAgeFit.

**Details**

If the sex is 'Both' or 'Total' predictions will be calculated for females, males and both sexes pooled, and if the numerical has failed to fit any of these three arrangements of the data then predictions will be calculated only for the successful fits.

**Value**

A list of length 3.

Properties	The properties of the data, information carried over from the creation of the data object with <code>as.GroAgeData</code> .
Model	The type of growth model, the distribution for the data, and the numerical method.
Predictions	A list of length one, two or three, with each component a data.frame with the age and length data, predicted length by the model, and the 'deviance' residuals.

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
#See examples in help pages of function GroAgeFit()
```



---

ksbream	<i>King Soldier Bream Individual Biological Data</i>
---------	--

---

**Description**

King soldier bream individual-based data from the Arabian Gulf sampled from the catch of two artisanal fishing fleets.

**Usage**

```
data("ksbream")
```

**Format**

A data frame with 797 observations on the following 14 variables.

CollectionDate Date of capture

TL.mm Total length in mm

FL.mm Fork length in mm

SL.mm Standard length in mm

TW.g Body weight in g

Ewt.g Eviscerated body weight in g

Liver.Wt.g Liver weight in g

GW.g Gonad weight in g

Sex 1 for males, 2 for females

Mature 1 for mature, 0 for juvenile

Age.yr Age in years

MaturityStage Macroscopic maturity stages

**Examples**

```
data(ksbream)
```

---

 MatCount

*Build Databases for Sexual Maturity Modeling by Logistic Regression*


---

### Description

From a data.frame of one continuous predictor (length, age, etc), and keys for ordinal predictor sex, month, and maturity stage, this function will build a list of variables ready to be used in a 2- or 3-parameter logistic model

### Usage

```
MatCount(matdat, fem.key, mal.key, stage.key, season.key)
```

### Arguments

matdat	A data.frame where each row is an individual fish, and the columns are a continuous predictor, and ordinal keys for sex, month, and maturity stage
fem.key	Integer, the ordinal key identifying females
mal.key	Integer, the ordinal key identifying males
stage.key	Integer, the ordinal key identifying the last maturity stage of juveniles. This key will split the individual into immature (all having stage $\leq$ stage.key) and mature.
season.key	Integer vector with two components, the first and last month of the reproductive season

### Details

The year is divided into two seasons, the reproductive season and the non-reproductive season

### Value

mal.nrep	A data.frame with the continuous predictor duplicated and sorted, a binary for mature individuals, a binary for juveniles, and the count of males in the non-reproductive season
mal.rep	A data.frame with the continuous predictor duplicated and sorted, a binary for mature individuals, a binary for juveniles, and the count of males in the reproductive season
fem.nrep	A data.frame with the continuous predictor duplicated and sorted, a binary for mature individuals, a binary for juveniles, and the count of females in the non-reproductive season
fem.rep	A data.frame with the continuous predictor duplicated and sorted, a binary for mature individuals, a binary for juveniles, and the count of females in the reproductive season

**Author(s)**

Ruben H. Roa-Ureta

**References**

R. Roa, B. Ernst, F. Tapia. 1999. Estimation of size at sexual maturity: evaluation of analytical and resampling procedures. *Fishery Bulletin (US)* 97:570-580.

**Examples**

```
data(BlackAngler.lenmatdat)
MatCount(matdat=BlackAngler.lenmatdat, fem.key=2, mal.key=1, stage.key=2,
         season.key=c(5,7))
```

---

MatData

*Class Attribute of LifeHist Data Object*

---

**Description**

To be used by `plot.MatData()` to examine data versus model predictions.

**Usage**

```
MatData(x)
```

**Arguments**

x                    An object of class `MatData`, generated by function `MatCount()`.

**Details**

This generic function will be used to give a class to the four components of the list output by function `MatCount()`.

**Value**

A class attribute.

**Note**

Objects of class `MatData` are plotted for exploration of parameter values (initial or maximum likelihood) in 2- or 3-parameter logistic regression

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
#See example in man pages for plot.MatData().
```

---

MatFit	<i>Wrapper Function to Estimate a 2- or 3-Parameters Logistic Regression of Sexual Maturity</i>
--------	---

---

### Description

Define the logistic model, pass the initial parameter values, the data, and the numerical optimization method(s) to estimate the model and to organize results in a list.

### Usage

```
MatFit(p, par, matdat, method, control = ls(), itnmax)
```

### Arguments

p	Integer, either 2 or 3, determining if the asymptotic proportion is assumed known at 1 or estimated.
par	Numeric vector of 2 or 3 components with initial parameter values.
matdat	A data.frame where each row is an individual fish, and the columns are a continuous predictor, and ordinal keys for sex, month, and maturity stage.
method	Character or character vector, defining the list of numerical methods to use. See help pages for function <code>optimx()</code> .
control	List, optimization control parameters to be passed to <code>optimx()</code> . See help for function <code>optimx()</code> .
itnmax	Integer, maximum number of iterations, to be passed to <code>optimx()</code> .

### Details

The function controls the estimation process. From a valid value of `p`, it will define the proper process model, either 2-parameters or 3-parameters logistic regression. The 3-parameter model might be useful outside the reproductive season when not all adult individuals are ready to reproduce, whereas the 2-parameter model assumes that above certain value of the continuous predictor all individuals are reproducing.

The function will re-organize `optimx()`'s output by adding some items (AIC, standard errors, correlation matrix) and ignoring other items. The output is a list of results in a list of optimization methods.

### Value

model	Type of model, <code>matlik.2p</code> or <code>matlik.3p</code>
method	Name of numerical optimization method
converg	Convergence message
kkt	The Karush-Kuhn-Tucker optimality conditions
AIC	The Akaike Information Criterion

pars	Maximum likelihood estimates of model parameters
num.grads	Numerical gradients at the maximum likelihood estimates
stdev	Estimated standard deviations of maximum likelihood estimates of model parameters
Cor	The estimated correlation matrix of maximum likelihood estimates of model parameters

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```

data(BlackAngler.lenmatdat)
BA.matlen <- MatCount(matdat=BlackAngler.lenmatdat,
                      fem.key=2,
                      mal.key=1,
                      stage.key=2,
                      season.key=c(5,7))

plot(BA.matlen[[2]],
     pred.50=35,
     pred.95=55,
     pred.unit="Length (cm)",
     top.text="Males Rep Season",
     lwd=2, xlab="", ylab="")
par(mfrow=c(2,2), mar=c(4,4,1,1), oma=c(4,4,1,1))
plot(BA.matlen[[1]],pred.50=35,pred.95=55,pred.unit="Length (cm)",
     top.text="Males NoRep Season", lwd=2, xlab="", ylab="")
plot(BA.matlen[[2]],pred.50=35,pred.95=55,pred.unit="Length (cm)",
     top.text="Males Rep Season", lwd=2, xlab="", ylab="")
plot(BA.matlen[[3]],pred.50=85,pred.95=110,pred.unit="Length (cm)",
     top.text="Females NoRep Season", lwd=2, xlab="", ylab="")
plot(BA.matlen[[4]],pred.50=55,pred.95=75,pred.unit="Length (cm)",
     top.text="Females Rep Season", lwd=2, xlab="", ylab="")
require(optimx)
BA.matlen.mal <- MatFit(p=2,
                       par=c(35,55),
                       matdat=BA.matlen[[2]],
                       method=c("spg", "CG", "Nelder-Mead"),
                       itnmax=100)

#
plot(BA.matlen[[2]],
     pred.50=BA.matlen.mal[[1]]$par.mle[1],
     pred.95=BA.matlen.mal[[1]]$par.mle[2],
     pred.unit="Length (cm)",
     top.text="Males Reproductive Season",
     lwd=2,
     xlab="Length (cm)",
     ylab="Proportion Mature")

```

---

plot.GroAgeData      *Exploratory Analysis of Ager and Length Individual Data*

---

### Description

Allows examining the relation between age and length in order to estimate individual growth models, and optionally other individual processes such as the relation between length and body weight and the timing of organ growth along the annual season.

### Usage

```
## S3 method for class 'GroAgeData'
plot(x, ...)
```

### Arguments

x                    An object of class 'GroAgeData' produced by function as.GroAgeData().  
 ...                  Not used.

### Value

A multipanel plot.

### Author(s)

Ruben H. Roa-Ureta

### Examples

```
#See examples in help pages for function as.GroAgeData()
```

---

plot.GroAgeExp      *Exploratory Modeling of Individual Growth with Age and Length Data*

---

### Description

Evaluate and refine the goodness of initial parameter values before fitting growth models to data.

### Usage

```
## S3 method for class 'GroAgeExp'
plot(x, ...)
```

### Arguments

x                    An object of class 'GroAgeExp' created by function groageexp().  
 ...                  Not used.

**Value**

A multipanel plot.

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
#See examples for function groageexp()
```

---

plot.GroAgeMod

*Examination of Results from Fitting Growth Models*

---

**Description**

After model fit and prediction, examine model results on a graphical display.

**Usage**

```
## S3 method for class 'GroAgeMod'  
plot(x, ...)
```

**Arguments**

x                    An object of class 'GroAgeMod' created by function GroAgeFit().  
...                   Not used.

**Value**

A multipanel plot.

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
#See examples in help pages of function AgeGroFit()
```

---

plot.MatData

*Exploratory Analysis of a Logistic Model on a Graphical Display*


---

**Description**

A one panel plot with data on proportion of sexual maturity and a line built with initial parameter values or with maximum likelihood estimates

**Usage**

```
## S3 method for class 'MatData'
plot(x, pred.50, pred.95, top.text, ...)
```

**Arguments**

x	An object of class MatData.
pred.50	Numeric, the predictor value at 0.5 proportion mature.
pred.95	Numeric, the predictor value at 0.95 proportion mature.
top.text	Character, to be used as main for plot.
...	Pass additional parameters to plot().

**Value**

A plot of data and model

**Author(s)**

Ruben H. Roa-Ureta

**Examples**

```
data(BlackAngler.lenmatdat)
BA.matlen <- MatCount(matdat=BlackAngler.lenmatdat,
                      fem.key=2,
                      mal.key=1,
                      stage.key=2,
                      season.key=c(5,7))
plot(BA.matlen[[2]], pred.50=35, pred.95=55, top.text="Males Rep Season")
```



# Index

- \* **& &nonlinear**
  - plot.MatData, 24
- \* **~classes**
  - GroAgeData, 9
  - GroAgeExp, 10
  - GroAgeMod, 15
- \* **~datasets**
  - GroAgeFit, 12
- \* **~distribution**
  - GroAgeFit, 12
- \* **~iplot**
  - GroAgeFit, 12
  - plot.GroAgeData, 22
  - plot.GroAgeExp, 22
  - plot.GroAgeMod, 23
- \* **~manip**
  - as.GroAgeData, 4
  - GroAgeFit, 12
- \* **~models**
  - groageexp, 10
  - GroAgeFit, 12
  - groagemod, 16
- \* **~optimize**
  - GroAgeFit, 12
- \* **datasets**
  - BlackAngler.agematdat, 6
  - BlackAngler.lenmatdat, 7
  - BlackAngler.wgtmatdat, 7
  - ksbream, 17
- \* **manip**
  - MatCount, 18
- \* **models**
  - MatFit, 20
  - plot.MatData, 24
- \* **optimize**
  - MatFit, 20
  - plot.MatData, 24
- \* **package**
  - LifeHist-package, 2
- as.GroAgeData, 4
- BlackAngler.agematdat, 6
- BlackAngler.lenmatdat, 7
- BlackAngler.wgtmatdat, 7
- deltamethod, 8
- GroAgeData, 9
- GroAgeExp, 10
- groageexp, 10
- GroAgeFit, 12
- GroAgeMod, 15
- groagemod, 16
- ksbream, 17
- LifeHist (LifeHist-package), 2
- LifeHist-package, 2
- MatCount, 18
- MatData, 19
- MatFit, 20
- plot.GroAgeData, 22
- plot.GroAgeExp, 22
- plot.GroAgeMod, 23
- plot.MatData, 24