Package ‘bfsl’

September 23, 2021

Title  Best-Fit Straight Line
Version  0.2.0
Date  2021-09-23
Description  How to fit a straight line through a set of points with errors in both coordinates? The 'bfsl' package implements the York regression (York, 2004 <doi:10.1119/1.1632486>). It provides unbiased estimates of the intercept, slope and standard errors for the best-fit straight line to independent points with (possibly correlated) normally distributed errors in both x and y. Other commonly used errors-in-variables methods, such as orthogonal distance regression, geometric mean regression or Deming regression are special cases of the 'bfsl' solution.

Depends  R (>= 3.5.0)
License  MIT + file LICENSE
URL  https://github.com/pasturm/bfsl
BugReports  https://github.com/pasturm/bfsl/issues
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.2
Suggests  testthat, tibble, dplyr
Imports  generics
NeedsCompilation  no
Author  Patrick Sturm [aut, cre]
Maintainer  Patrick Sturm <sturm@tofwerk.com>
Repository  CRAN
Date/Publication  2021-09-23 10:00:02 UTC
**R topics documented:**

- `augment.bfsl` .................................................. 2
- `bfsl` ............................................................... 3
- `bfsl_control` ................................................... 5
- `glance.bfsl` ....................................................... 6
- `pearson_york_data` ............................................. 7
- `plot.bfsl` ......................................................... 7
- `predict.bfsl` ...................................................... 8
- `print.bfsl` ......................................................... 9
- `print.summary.bfsl` ............................................ 9
- `summary.bfsl` .................................................... 10
- `tidy.bfsl` .......................................................... 10

**Index**

| augment.bfsl | Augment Data with Information from a bfsl Object |

**Description**

Broom tidier method to augment data with information from a bfsl object.

**Usage**

```r
## S3 method for class 'bfsl'
augment(x, data = x$data, newdata = NULL, ...)
```

**Arguments**

- `x` A `bfsl` object created by `bfsl::bfsl()`
- `data` A `base::data.frame()` or `tibble::tibble()` containing all the original predictors used to create `x`. Defaults to NULL, indicating that nothing has been passed to `newdata`. If `newdata` is specified, the data argument will be ignored.
- `newdata` A `base::data.frame()` or `tibble::tibble()` containing all the original predictors used to create `x`. Defaults to NULL, indicating that nothing has been passed to `newdata`. If `newdata` is specified, the data argument will be ignored.
- `...` Unused, included for generic consistency only.

**Value**

A `tibble::tibble()` with columns:

- `.fitted` Fitted or predicted value.
- `.se.fit` Standard errors of fitted values.
- `.resid` The residuals, that is `y` observations minus fitted values. (Only returned if `newdata = NULL`.)
bfsl

Examples

```r
fit = bfsl(pearson_york_data)

augment(fit)
```

---

bfsl  
*Calculates the Best-fit Straight Line*

---

**Description**

bfsl calculates the best-fit straight line to independent points with (possibly correlated) normally distributed errors in both coordinates.

**Usage**

```r
bfsl(...)  
## Default S3 method:
bfsl(x, y = NULL, sd_x = 0, sd_y = 1, r = 0, control = bfsl_control(), ...)

## S3 method for class 'formula'
bfsl(
  formula,
  data = parent.frame(),
  sd_x,
  sd_y,
  r = 0,
  control = bfsl_control(),
  ...
)
```

**Arguments**

- `...`  
  Further arguments passed to or from other methods.

- `x`  
  A vector of x observations or a data frame (or an object coercible by `as.data.frame` to a data frame) containing the named vectors `x`, `y`, and optionally `sd_x`, `sd_y` and `r`. If weights `w_x` and `w_y` are given, then `sd_x` and `sd_y` are calculated from `sd_x = 1/sqrt(w_x)` and `sd_y = 1/sqrt(w_y)`. Specifying `y`, `sd_x`, `sd_y` or `r` directly as function arguments overwrites these variables in the data structure.

- `y`  
  A vector of y observations.

- `sd_x`  
  A vector of x measurement error standard deviations. If it is of length one, all data points are assumed to have the same x standard deviation.

- `sd_y`  
  A vector of y measurement error standard deviations. If it is of length one, all data points are assumed to have the same y standard deviation.
A vector of correlation coefficients between errors in x and y. If it is of length one, all data points are assumed to have the same correlation coefficient.

control
A list of control settings. See `bfsl.control` for the names of the settable control values and their effect.

formula
A formula specifying the bivariate model (as in `lm`, but here only `y ~ x` makes sense).

data
A data.frame containing the variables of the model.

Details

`bfsl` provides the general least-squares estimation solution to the problem of fitting a straight line to independent data with (possibly correlated) normally distributed errors in both x and y.

With `sd_x = 0` the (weighted) ordinary least squares solution is obtained. The calculated standard errors of the slope and intercept multiplied with `sqrt(chisq)` correspond to the ordinary least squares standard errors.

With `sd_x = c`, `sd_y = d`, where c and d are positive numbers, and `r = 0` the Deming regression solution is obtained. If additionally `c = d`, the orthogonal distance regression solution, also known as major axis regression, is obtained.

Setting `sd_x = sd(x)`, `sd_y = sd(y)` and `r = 0` leads to the geometric mean regression solution, also known as reduced major axis regression or standardised major axis regression.

The goodness of fit metric `chisq` is a weighted reduced chi-squared statistic. It compares the deviations of the points from the fit line to the assigned measurement error standard deviations. If x and y are indeed related by a straight line, and if the assigned measurement errors are correct (and normally distributed), then `chisq` will equal 1. A `chisq > 1` indicates underfitting: the fit does not fully capture the data or the measurement errors have been underestimated. A `chisq < 1` indicates overfitting: either the model is improperly fitting noise, or the measurement errors have been overestimated.

Value

An object of class "bfsl", which is a list containing the following components:

- `coefficients` A 2x2 matrix with columns of the fitted coefficients (intercept and slope) and their standard errors.
- `chisq` The goodness of fit (see Details).
- `fitted.values` The fitted mean values.
- `residuals` The residuals, that is y observations minus fitted values.
- `df.residual` The residual degrees of freedom.
- `cov.ab` The covariance of the slope and intercept.
- `control` The control list used, see the control argument.
- `convInfo` A list with convergence information.
- `call` The matched call.
- `data` A list containing x, y, sd_x, sd_y and r.
bfsl_control

References

Examples

```r
x = pearson_york_data$x
y = pearson_york_data$y
sd_x = 1/sqrt(pearson_york_data$w_x)
sd_y = 1/sqrt(pearson_york_data$w_y)
bfsl(x, y, sd_x, sd_y)
bfsl(y~x, pearson_york_data, sd_x, sd_y)

fit = bfsl(pearson_york_data)
plot(fit)
```

### Description

*bfsl_control* allows the user to set some characteristics of the *bfsl* best-fit straight line algorithm.

### Usage

```r
bfsl_control(tol = 1e-10, maxit = 100)
```

### Arguments

- **tol**
  A positive numeric value specifying the tolerance level for the convergence criterion

- **maxit**
  A positive integer specifying the maximum number of iterations allowed.

### Value

A list with two components named as the arguments.

### See Also

*bfsl*

### Examples

```r
bfsl_control(tol = 1e-8, maxit = 1000)
```
**glance.bfsl**  
*Glance at a bfsl Object*

**Description**

Broom tidier method to glance at a bfsl object.

**Usage**

```r
## S3 method for class 'bfsl'
glance(x, ...)
```

**Arguments**

- `x` A `bfsl` object.
- `...` Unused, included for generic consistency only.

**Value**

A `[tibble::tibble()]` with one row and columns:

- `chisq` The goodness of fit.
- `p.value` P-value.
- `df.residual` Residual degrees of freedom.
- `nobs` Number of observations.
- `isConv` Did the fit converge?
- `iter` Number of iterations.
- `finTol` Final tolerance.

**Examples**

```r
fit = bfsl(pearson_york_data)
glance(fit)
```
**Description**


**Usage**

```r
pearson_york_data
```

**Format**

A data frame with 10 rows and 4 variables:

- `x` x observations
- `w_x` weights of x
- `y` y observations
- `w_y` weights of y

**References**


**Examples**

```r
bfsl(pearson_york_data)
```

---

**plot.bfsl**

*Plot Method for bfsl Results*

**Description**

`plot.bfsl` plots the data points with error bars and the calculated best-fit straight line.

**Usage**

```r
## S3 method for class 'bfsl'
plot(x, grid = TRUE, ...)
```
Arguments

x An object of class "bfsl".
grid If TRUE (default) grid lines are plotted.
... Further parameters to be passed to the plotting routines.

predict.bfsl Predict Method for bfsl Model Fits

Description

predict.bfsl predicts future values based on the bfsl fit.

Usage

## S3 method for class 'bfsl'
predict(
  object, 
  newdata, 
  interval = c("none", "confidence"), 
  level = 0.95, 
  se.fit = FALSE, 
  ... 
)

Arguments

object Object of class "bfsl".
newdata A data frame with variable x to predict. If omitted, the fitted values are used.
interval Type of interval calculation. "none" or "confidence".
level Confidence level.
se.fit A switch indicating if standard errors are returned.
... Further arguments passed to or from other methods.

Value

predict.bfsl produces a vector of predictions or a matrix of predictions and bounds with column
names fit, lwr, and upr if interval is set to "confidence".

If se.fit is TRUE, a list with the following components is returned:

fit Vector or matrix as above
se.fit Standard error of predicted means
Examples

    fit = bfsl(pearson_york_data)
    predict(fit, interval = "confidence")
    new = data.frame(x = seq(0, 8, 0.5))
    predict(fit, new, se.fit = TRUE)

    pred.clim = predict(fit, new, interval = "confidence")
    matplot(new$x, pred.clim, lty = c(1, 2, 2), type = "l", xlab = "x", ylab = "y")
    df = fit$data
    points(df$x, df$y)
    arrows(df$x, df$y-df$sd_y, df$x, df$y+df$sd_y,
           length = 0.05, angle = 90, code = 3)
    arrows(df$x-df$sd_x, df$y, df$x+df$sd_x, df$y,
           length = 0.05, angle = 90, code = 3)

print.bfsl

Print Method for bfsl Results

Description

print method for class "bfsl".

Usage

    ## S3 method for class 'bfsl'
    print(x, digits = max(3L,getOption("digits") - 3L), ...)

Arguments

x         An object of class "bfsl".
digits    The number of significant digits to use when printing.
...        Further arguments passed to print.default.

print.summary.bfsl

Print Method for summary.bfsl Objects

Description

print method for class "summary.bfsl".

Usage

    ## S3 method for class 'summary.bfsl'
    print(x, digits = max(3L,getOption("digits") - 3L), ...)

Arguments

x         An object of class "summary.bfsl".
digits    The number of significant digits to use when printing.
...        Further arguments passed to print.default.
Arguments

x  An object of class "summary.bfsl".
digits  The number of significant digits to use when printing.
...
Further arguments passed to print.default.

summary.bfsl  

Summary Method for bfsl Results

Description

summary method for class "bfsl".

Usage

```r
## S3 method for class 'bfsl'
summary(object, ...)
```

Arguments

object  An object of class "bfsl".
...
Further arguments passed to summary.default.

tidy.bfsl  

Tidy a bfsl Object

Description

Broom tidier method to tidy a bfsl object.

Usage

```r
## S3 method for class 'bfsl'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x  A 'bfsl' object.
conf.int  Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level  The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
...
Unused, included for generic consistency only.
Value

A tidy [tibble::tibble()] summarizing component-level information about the model

Examples

```r
fit = bfsl(pearson_york_data)

tidy(fit)
```
Index

* datasets
  pearson_york_data, 7

as.data.frame, 3
augment.bfs1, 2
bfs1, 3, 5
bfs1_control, 4, 5
glance.bfs1, 6
lm, 4
pearson_york_data, 7
plot.bfs1, 7
predict.bfs1, 8
print.bfs1, 9
print.summary.bfs1, 9
summary.bfs1, 10
tidy.bfs1, 10