Package ‘bife’

October 30, 2020

Type Package
Title Binary Choice Models with Fixed Effects
Version 0.7.1
Description Estimates fixed effects binary choice models (logit and probit) with potentially many individual fixed effects and computes average partial effects. Incidental parameter bias can be reduced with an asymptotic bias correction proposed by Fernandez-Val (2009) <doi:10.1016/j.jeconom.2009.02.007>.
License GPL (>= 2)
Depends R (>= 3.1.0)
Imports data.table, Formula, Rcpp, stats
LinkingTo Rcpp, RcppArmadillo
URL https://github.com/amrei-stammann/bife
BugReports https://github.com/amrei-stammann/bife/issues
RoxygenNote 7.1.1
LazyData true
Suggests alpaca, knitr, rmarkdown
VignetteBuilder knitr
Encoding UTF-8
NeedsCompilation yes
Author Amrei Stammann [aut, cre],
Daniel Czarnowske [aut] (<https://orcid.org/0000-0002-0030-929X>),
Florian Heiss [aut],
Daniel McFadden [ctb]
Maintainer Amrei Stammann <amrei.stammann@rub.de>
Repository CRAN
Date/Publication 2020-10-30 09:40:02 UTC
bias_corr

Asymptotic bias correction for binary choice Models with fixed effects

Description

bias_corr is a post-estimation routine that can be used to substantially reduce the incidental parameter bias problem (Neyman and Scott (1948)) present in non-linear fixed effects models (see Fernández-Val and Weidner (2018) for an overview). The command applies the analytical bias correction derived by Fernández-Val (2009) to obtain bias-corrected estimates of the structural parameters.

Remark: Fernández-Val (2009) further refined the bias correction of Hahn and Newey (2004). The correction is now also applicable to dynamic models.

Usage

bias_corr(object, L = 0L)

Arguments

object an object of class "bife".
L unsigned integer indicating a bandwidth for the estimation of spectral densities proposed by Hahn and Kuersteiner (2011). Default is zero, which should be used if all regressors are assumed to be strictly exogenous. In the presence of weakly exogenous or predetermined regressors, Fernández-Val and Weidner (2018) suggest to choose a bandwidth not higher than four.
Value

The function `bias_corr` returns a named list of class "bife".

References


See Also

`bife`

Examples

```r
# Load 'psid' dataset
library(bife)
dataset <- psid

# Fit a static logit model
mod <- bife(LFP ~ I(AGE^2) + log(INCH) + KID1 + KID2 + KID3 + factor(TIME) | ID, dataset)
summary(mod)

# Apply analytical bias correction
mod_bc <- bias_corr(mod)
summary(mod_bc)
```

bife

Efficiently fit binary choice models with fixed effects
Description

*bife* can be used to fit fixed effects binary choice models (logit and probit) based on an unconditional maximum likelihood approach. It is tailored for the fast estimation of binary choice models with potentially many individual fixed effects. The routine is based on a special pseudo demeaning algorithm derived by Stammann, Heiss, and McFadden (2016). The estimates obtained are identical to the ones of *glm*, but the computation time of *bife* is much lower.

**Remark:** The term fixed effect is used in econometrician’s sense of having a full set of individual specific intercepts. All other parameters in the model are referred to as structural parameters.

Usage

```r
bife(
  formula,
  data = list(),
  model = c("logit", "probit"),
  beta_start = NULL,
  control = list(),
  bias_corr = NULL,
  tol_demeaning = NULL,
  iter_demeaning = NULL,
  tol_offset = NULL,
  iter_offset = NULL
)
```

Arguments

- **formula**: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. formula must be of type \( y \times id \) where the id refers to an individual identifier (fixed effect category).
- **data**: an object of class "data.frame" containing the variables in the model.
- **model**: the description of the error distribution and link function to be used in the model. For *bife* this has to be a character string naming the model function. Default is "logit".
- **beta_start**: an optional vector of starting values used for the structural parameters in the optimization algorithm. Default is zero for all structural parameters.
- **control**: a named list of parameters for controlling the fitting process. See *bife_control* for details.
- **bias_corr**: deprecated; see *bias_corr*.
- **tol_demeaning, iter_demeaning, tol_offset, iter_offset**: deprecated; see *bife_control*.

Details

*bife* drops all observations of cross-sectional units (individuals) with non-varying response. This can be done because these observations do not contribute to the identification of the structural parameters (perfect classification).
If `bife` does not converge this is usually a sign of linear dependence between one or more regressors and the fixed effects. In this case, you should carefully inspect your model specification.

**Value**

The function `bife` returns a named list of class "bife".

**References**


**Examples**

```r
# Load 'psid' dataset
library(bife)
dataset <- psid

# Fit a static logit model
mod <- bife(LFP ~ I(AGE^2) + log(INCH) + KID1 + KID2 + KID3 + factor(TIME) | ID, dataset)
summary(mod)
```

---

**bife_control**  
*Set bife Control Parameters*

**Description**

Set and change parameters used for fitting `bife`.

**Usage**

```r
bife_control(
  dev_tol = 1e-08,
  iter_max = 25L,
  trace = FALSE,
  rho_tol = NULL,
  conv_tol = NULL
)
```

**Arguments**

- `dev_tol` tolerance level for the first stopping condition of the maximization routine. The stopping condition is based on the relative change of the deviance in iteration $r$ and can be expressed as follows: $|dev_r - dev_{r-1}|/(0.1 + |dev_r|) < tol$. Default is $1.0e-08$.

- `iter_max` unsigned integer indicating the maximum number of iterations in the maximization routine. Default is $25L$. 

trace logical indicating if output should be produced in each iteration. Default is FALSE.
conv_tol, rho_tol
deprecated; step-halving is now similar to glm.fit2.

Value

The function bife_control returns a named list of control parameters.

See Also

bife

---

**coef.bife**

*Extract estimates of structural parameters or fixed effects*

**Description**

*coef.bife* is a generic function which extracts estimates of the structural parameters or fixed effects from objects returned by *bife*.

**Usage**

```r
## S3 method for class 'bife'
coef(object, type = c("sp", "fe"), corrected = NULL, fixed = NULL, ...)
```

**Arguments**

- **object**: an object of class "bife".
- **type**: the type of parameter estimates that should be returned; structural parameters or fixed effects. Default is "sp" referring to the structural parameters.
- **corrected, fixed**: deprecated.
- **...**: other arguments.

**Value**

The function *coef.bife* returns a named vector of estimates of the requested parameters.

**See Also**

*bife*
**Description**

*coef.bifeAPEs* is a generic function which extracts estimates of the average partial effects from objects returned by *get_APEs*.

**Usage**

```r
## S3 method for class 'bifeAPEs'
coef(object, ...)
```

**Arguments**

- `object` an object of class "APEs".
- `...` other arguments.

**Value**

The function *coef.bifeAPEs* returns a named vector of estimates of the average partial effects.

**See Also**

*get_APEs*

---

**Description**

*fitted.bife* is a generic function which extracts fitted values from an object returned by *bife*.

**Usage**

```r
## S3 method for class 'bife'
fitted(object, ...)
```

**Arguments**

- `object` an object of class "bife".
- `...` other arguments.

**Value**

The function *fitted.bife* returns a vector of fitted values.
See Also

bife

get_APEs
Compute average partial effects for binary choice models with fixed effects

Description

get_APEs is a post-estimation routine that can be used to estimate average partial effects with respect to all covariates in the model and the corresponding covariance matrix. The estimation of the covariance is based on a linear approximation (delta method). Note that the command automatically determines which of the regressors are continuous or binary.

Remark: The routine currently does not allow to compute average partial effects based on functional forms like interactions and polynomials.

Note: apeff_bife is deprecated and will be removed soon.

Usage

get_APEs(
  object,
  n_pop = NULL,
  sampling_fe = c("independence", "unrestricted"),
  weak_exo = FALSE
)
apeff_bife(...)

Arguments

object an object of class "bife".
n_pop unsigned integer indicating a finite population correction for the estimation of the covariance matrix of the average partial effects proposed by Cruz-Gonzalez, Fernández-Val, and Weidner (2017). The correction factor is computed as follows: \((n^* - n)/(n^* - 1)\), where \(n^*\) and \(n\) are the size of the entire population and the full sample size. Default is NULL, which refers to a factor of one and is equal to an infinitely large population.
sampling_fe a string equal to "independence" or "unrestricted" which imposes sampling assumptions about the unobserved effects. "independence" imposes that all unobserved effects are mutually independent sequences. "unrestricted" does not impose any sampling assumptions. Note that this option only affects the estimation of the covariance. Default is "independence".
weak_exo logical indicating if some of the regressors are assumed to be weakly exogenous (e.g. predetermined). If object is returned by bias_corr, the option will be automatically set to TRUE if the chosen bandwidth parameter is larger than zero. Note that this option only affects the estimation of the covariance matrix. Default is FALSE, which assumes that all regressors are strictly exogenous.

arguments passed to the deprecated function aeff_bife.

Value
The function get_APEs returns a named list of class "bifeAPEs".

References

See Also
bias_corr, bife

Examples

```r
# Load 'psid' dataset
library(bife)
dataset <- psid

# Fit a static logit model
mod <- bife(LFP ~ I(AGE^2) + log(INCH) + KID1 + KID2 + KID3 + factor(TIME) | ID, dataset)
summary(mod)

# Compute average partial effects
mod_ape <- get_APEs(mod)
summary(mod_ape)

# Apply analytical bias correction
mod_bc <- bias_corr(mod)
summary(mod_bc)

# Compute bias-corrected average partial effects
mod_ape_bc <- get_APEs(mod_bc)
```
logLik.bife  

Extract log-likelihood

Description

logLik.bife extracts the sum of the log-likelihood from an object returned by bife.

Usage

## S3 method for class 'bife'
logLik(object, ...)

Arguments

object  
an object of class "bife".

...  
other arguments.

Value

The function logLik.bife returns the sum of the log-likelihood.

See Also

bife

predict.bife  

Predict method for bife fits

Description

predict.bife is a generic function which obtains predictions from an object returned by bife.

Usage

## S3 method for class 'bife'
predict(
  object,
  type = c("link", "response"),
  X_new = NULL,
  alpha_new = NULL,
  corrected = NULL,
  ...
)


print.bife

Arguments

object  
an object of class "bife".

type  
the type of prediction required. "link" is on the scale of the linear predictor whereas "response" is on the scale of the response variable. Default is "link".

X_new  
a data.frame or a regressor matrix for predictions. If not supplied predictions are based on the regressor matrix returned by the object bife. See Details.

alpha_new  
a scalar or vector of fixed effects. If not supplied predictions are based on the vector of fixed effects returned by bife or the average. See Details.

corrected  
deprecated.

...  
other arguments

Details

The model frame returned by the object bife only includes individuals that were not dropped before the fitting process (due to perfect classification). The linear predictors of perfectly classified observations are equal to -Inf or Inf whereas the predicted probabilities are equal to their response. In-sample predictions are only based on non-perfectly classified observations.

If alpha_new is supplied as a scalar the linear predictor is computed using the same value of the fixed effect for each observation. If alpha_new is supplied as a vector it has to be of same length as the rows of the corresponding regressor matrix. If only X_new is provided but not alpha_new, we use the average.

Value

The function predict.bife returns a vector of predictions.

See Also

bife

Description

print.bife is a generic function which displays some minimal information from objects returned by bife.

Usage

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

x  an object of class "bife".
digits  unsigned integer indicating the number of decimal places. Default is max(3L,getOption("digits") - 3L).
...
  other arguments.

See Also

bife

Description

print.bifeAPEs is a generic function which displays some minimal information from objects returned by get_APEs.

Usage

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

Arguments

x  an object of class "bifeAPEs".
digits  unsigned integer indicating the number of decimal places. Default is max(3L,getOption("digits") - 3L).
...
  other arguments.

See Also

get_APEs
**Description**

`print.summary.bife` is a generic function which displays summary statistics from objects returned by `summary.bife`.

**Usage**

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

**Arguments**

- `x`: an object of class "summary.bife".
- `digits`: unsigned integer indicating the number of decimal places. Default is `max(3L,getOption("digits")-3L)`.
- `...`: other arguments.

**See Also**

`bife`
See Also

get_APEs

<table>
<thead>
<tr>
<th>psid</th>
<th>Female labor force participation</th>
</tr>
</thead>
</table>

Description

The sample was obtained from the "Panel Study of Income Dynamics" and contains information about \( N = 1461 \) women that were observed over \( T = 9 \) years.

Usage

psid

Format

A data frame with 13,149 rows:

- **ID** individual identifier
- **LFP** labor force participation
- **KID1** # of kids aged between 0 and 2
- **KID2** # of kids aged between 3 and 5
- **KID3** # of kids aged between 6 and 17
- **INCH** income husband
- **AGE** age of woman
- **TIME** time identifier

References


See Also

bife
Summary statistics for objects of class "bife".

## S3 method for class 'bife'

```r
summary(object, type = c("sp", "fe"), corrected = NULL, fixed = NULL, ...)
```

### Arguments

- **object**: an object of class "bife".
- **type**: the type of parameter estimates the summary statistics are related to: structural parameters or fixed effects. Default is "sp" referring to the structural parameters.
- **corrected, fixed**: deprecated.
- **...**: other arguments.

### Value

Returns an object of class "summary.bife" which is a list of summary statistics of object.

## See Also

- `bifeAPEs`
Value

Returns an object of class "summary.bifeAPEs" which is a list of summary statistics of object.

See Also

get_APEs

---

table

| vcov.bife          | Extract estimates of the covariance matrix |

Description

vcov.bife computes an estimate of the covariance matrix of the estimator of the structural parameters from objects returned by bife. The estimate is obtained using the inverse of the negative Hessian after convergence.

Usage

```r
## S3 method for class 'bife'
vcov(object, ...)
```

Arguments

- `object`: an object of class "bife".
- `...`: other arguments.

Value

The function vcov.bife returns a named matrix of covariance estimates.

See Also

bife
vcov.bifeAPEs

Extract estimates of the covariance matrix

Description

vcov.bifeAPEs computes an estimate of the covariance matrix of the estimator of the average partial parameters from objects returned by get_APEs.

Usage

```r
## S3 method for class 'bifeAPEs'
vcov(object, ...)
```

Arguments

- `object` an object of class "bifeAPEs".
- `...` other arguments.

Value

The function vcov.bifeAPEs returns a named matrix of covariance estimates.

See Also

get_APEs
Index

* datasets
  psid, 14

  apeff_bife, 8, 9
  apeff_bife (get_APEs), 8

  bias_corr, 2, 2, 3, 4, 9
  bife, 3, 3, 4–16
  bife_control, 4, 5, 6

  coef.bife, 6, 6
  coef.bifeAPEs, 7, 7

  fitted.bife, 7, 7

  get_APEs, 7, 8, 8, 9, 12, 14, 16, 17
  glm, 4

  logLik.bife, 10, 10

  predict.bife, 10, 10
  print.bife, 11, 11
  print.bifeAPEs, 12, 12
  print.summary.bife, 13, 13
  print.summary.bifeAPEs, 13, 13
  psid, 14

  summary.bife, 13, 15
  summary.bifeAPEs, 13, 15

  vcov.bife, 16, 16
  vcov.bifeAPEs, 17, 17