Package: ebrahim.gof (via r-universe)

June 12, 2026

Type Package

Title Ebrahim-Farrington Goodness-of-Fit Test for Logistic Regression

Version 2.0.0

Date 2026-06-10

Maintainer Ebrahim Khaled Ebrahim <ebrahimkhaled@alexu.edu.eg>

Description Implements the Ebrahim-Farrington goodness-of-fit test for logistic regression models, particularly effective for sparse data and binary outcomes. This test provides an improved alternative to the traditional Hosmer-Lemeshow test by using a modified Pearson chi-square statistic with data-dependent grouping. The test is based on Farrington (1996) theoretical framework but simplified for practical implementation with binary data. Includes functions for both the original Farrington test (for grouped data) and the new Ebrahim-Farrington test (for binary data with automatic grouping), the Directed Ebrahim-Farrington (DEF) test that targets calibration-shape departures, and an ensemble that combines the DEF bases via the Cauchy combination test. For more details see Hosmer (1980) <doi:10.1080/03610928008827941> and Farrington (1996) <doi:10.1111/j.2517-6161.1996.tb02086.x>.

License GPL-3

URL https://github.com/ebrahimkhaled/ebrahim.gof

BugReports https://github.com/ebrahimkhaled/ebrahim.gof/issues

Depends R (>= 3.5.0)

Imports stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, ResourceSelection, ggplot2, CompQuadForm, statmod, mgcv, BAGofT

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Ebrahim Khaled Ebrahim [aut, cre] (ORCID:
<https://orcid.org/0009-0006-7839-8778>)

Repository https://cran.r-universe.dev

Date/Publication 2026-06-12 10:20:02 UTC

RemoteUrl https://github.com/cran/ebrahim.gof

RemoteRef HEAD

RemoteSha 4deb5c499e217d01aa435590adef3ff39c73b211

Contents

`def.ensemble.gof` *Combine Directed GOF Tests into One Decision (Ensemble)*

Description

Combines the three Directed Ebrahim-Farrington (DEF) basis tests (`"poly2"`, `"poly3"`, `"stukel"`) into a single goodness-of-fit decision, so the user does not have to choose a basis. By default the p-values are combined with the Cauchy Combination Test (CCT), which controls the error rate under the strong dependence between tests computed on the same fitted model. The omnibus EF test can optionally be added to the vote.

Usage

```
def.ensemble.gof(
  object,
  predicted_probs = NULL,
  X = NULL,
  components = c("poly2", "poly3", "stukel"),
  add_ef = FALSE,
  combine = c("cct", "minp", "fisher"),
  G = 10,
  extra_pvalues = NULL
)
```

Arguments

object	A fitted binary logistic glm, or a binary (0/1) vector y (then supply predicted_probs).
predicted_probs	Numeric predicted probabilities; required when object is a y vector.
X	Optional design matrix, threaded to def.gof for the exact calibration (only used with the y/predicted_probs form).
components	Character vector, a subset of c("poly2","poly3","stukel"). Default is all three.
add_ef	Logical; if TRUE, the omnibus EF p-value (ef.gof) is appended to the components. Default FALSE.
combine	One of "cct" (default), "minp", "fisher".
G	Integer number of groups passed to def.gof/ef.gof (default 10).
extra_pvalues	Optional named numeric vector of additional p-values to include (e.g. a Tsiatis test computed elsewhere). Default NULL.

Details

Because the component tests are computed on the same fit, their p-values are strongly dependent. The CCT (combine = "cct") has an asymptotic standard-Cauchy null whose tail is robust to this dependence, so it needs no calibration. The "minp" (Sidak) and "fisher" rules assume independence and are offered for comparison only; under positive dependence "minp" is conservative and "fisher" is anti-conservative, so they should be calibrated by simulation before use (not done here).

Value

A one-row data.frame with columns Test, Combiner, Components, k, and p_value.

Author(s)

Ebrahim Khaled Ebrahim <ebrahimkhaled@alexu.edu.eg>

References

Liu, Y. and Xie, J. (2020). Cauchy combination test. *JASA*, 115(529), 393-402.

See Also

def.gof, ef.gof.

Examples

```
set.seed(1)
n <- 500
x <- runif(n, -3, 3)
y <- rbinom(n, 1, 1 / (1 + exp(-(0.6 * x))))
fit <- glm(y ~ x, family = binomial())
def.ensemble.gof(fit)                # CCT of the three DEF bases
```

```
def.ensemble.gof(fit, add_ef = TRUE)  # add the omnibus EF
```

`def.gof` *Directed Ebrahim-Farrington (DEF) Goodness-of-Fit Test*

Description

Performs the Directed Ebrahim-Farrington (DEF) goodness-of-fit test for a fitted binary logistic regression model. DEF concentrates its power on a small set of calibration-curve "shape" directions by projecting the grouped standardized residuals onto a low-dimensional basis and testing the squared length of that projection.

Usage

```
def.gof(
  object,
  predicted_probs = NULL,
  X = NULL,
  G = 10,
  basis = c("poly3", "poly2", "stukel", "ensemble"),
  method = c("satterthwaite", "imhof")
)
```

Arguments

`object`	A fitted binary logistic `glm`, or a binary (0/1) response vector `y` (then supply `predicted_probs`).
`predicted_probs`	Numeric predicted probabilities; required when `object` is a `y` vector, ignored when it is a glm.
`X`	Optional design matrix, used only with the `y`/`predicted_probs` form: it enables the exact estimation-adjusted (Ω) calibration (logit working weights assumed). Without it the conservative χ^2_k reference is used and a warning is issued. Ignored when `object` is a glm.
`G`	Integer number of equal-frequency groups (default 10; must be >= 3).
`basis`	One of `"poly3"` (default), `"poly2"`, `"stukel"`, or `"ensemble"`.
`method`	One of `"satterthwaite"` (default) or `"imhof"`.

Details

The observations are sorted by predicted probability and split into `G` equal-frequency groups; the standardized grouped residual vector r is projected onto a basis matrix Z of smooth shapes, giving $S = (Z'r)'(Z'Z)^{-1}(Z'r)$. Its null distribution is a weighted sum of χ^2_1 variables with weights equal to the eigenvalues of $(Z'Z)^{-1}Z'\Omega Z$, where $\Omega = I - U(X'WX)^{-1}U'$ is the estimation-adjusted covariance of the grouped residuals. The p-value uses a Satterthwaite scaled-χ^2 approximation (default) or Imhof's method (if the **CompQuadForm** package is installed). Bases: `"poly2"`, `"poly3"` (default), `"stukel"`; `"ensemble"` runs all three and combines them via `def.ensemble.gof`.

Value

A one-row `data.frame` with columns `Test`, `Basis`, `Test_Statistic` (the statistic S), `df`, `Method`, and `p_value`. When `basis = "ensemble"`, the return is that of `def.ensemble.gof`.

Author(s)

Ebrahim Khaled Ebrahim <ebrahimkhaled@alexu.edu.eg>

References

Ebrahim, K. E. and El-Kotory, A. Omnibus versus Directed Goodness-of-Fit Tests for Sparse Data in Binary Logistic Regression (companion paper).

See Also

`ef.gof`, `def.ensemble.gof`.

Examples

```
set.seed(1)
n <- 500
x <- runif(n, -3, 3)
y <- rbinom(n, 1, 1 / (1 + exp(-(0.6 * x))))
fit <- glm(y ~ x, family = binomial())
def.gof(fit)                       # default poly3 basis
def.gof(fit, basis = "stukel")     # tail-shape basis
def.gof(fit, basis = "ensemble")   # combine all three (CCT)
```

`ef.gof` *Ebrahim-Farrington Goodness-of-Fit Test for Logistic Regression*

Description

Performs the Ebrahim-Farrington goodness-of-fit test for logistic regression models. This test is particularly effective for binary data and sparse datasets, providing an improved alternative to the traditional Hosmer-Lemeshow test.

Usage

```
ef.gof(
  y,
  predicted_probs = NULL,
  model = NULL,
  m = NULL,
  G = 10,
  method = c("chisq", "normal")
)
```

Arguments

`y`	A fitted binary logistic `glm` (then `predicted_probs` is taken from it automatically), or a numeric vector of binary responses (0/1) for binary data / counts of successes for grouped data.
`predicted_probs`	Numeric vector of predicted probabilities from the logistic regression model. Must be same length as `y`.
`model`	Optional `glm` object. Required only for the original Farrington test with grouped data (when `m` is provided and `G` is NULL).
`m`	Optional numeric vector of trial counts for each observation (for grouped data). If NULL, data is assumed to be binary.
`G`	Optional integer specifying the number of groups for binary data grouping. Default is 10. If NULL, no grouping is performed and `m` must be provided.
`method`	Reference distribution for the grouped EF statistic: `"chisq"` (default) refers T_{EF} to a χ^2_{G-2} distribution; `"normal"` uses the standardized Z_{EF} (the behaviour of package versions <= 1.0.0).

Details

The Ebrahim-Farrington test is based on Farrington's (1996) theoretical framework but simplified for practical implementation with binary data. The test uses a modified Pearson chi-square statistic with data-dependent grouping, where observations are grouped by their predicted probabilities.

For binary data (when `G` is specified), the test automatically groups observations into `G` groups based on predicted probabilities and applies the simplified Ebrahim-Farrington statistic:

$$Z_{EF} = \frac{T_{EF} - (G-2)}{\sqrt{2(G-2)}}$$

where T_{EF} is the modified Pearson chi-square statistic, and G is the number of groups.

For grouped data (when `m` is provided), the test applies the original Farrington test with full variance calculations.

Value

A data frame with the following columns:

`Test`	Character string identifying the test performed
`Test_Statistic`	Numeric value of the standardized test statistic
`p_value`	Numeric p-value for the test

Note

- For binary data with automatic grouping (`G` specified): Use the Ebrahim-Farrington test which is computationally efficient and doesn't require the model specification.
- For grouped data (`m` provided): Use the original Farrington test which requires the fitted model object.

- The test statistic follows a standard normal distribution under the null hypothesis of adequate model fit.
- For binary data with `m=1` for all observations and no grouping, the test is not applicable and will return a p-value of 1.

Author(s)

Ebrahim Khaled Ebrahim <ebrahimkhaled@alexu.edu.eg>

References

Farrington, C. P. (1996). On Assessing Goodness of Fit of Generalized Linear Models to Sparse Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(2), 349-360. Ebrahim, K. E. (2025). Goodness-of-Fits Tests and Calibration Machine Learning Algorithms for Logistic Regression Model with Sparse Data. *Master's Thesis*, Alexandria University. Hosmer, D. W., & Lemeshow, S. (1980). A goodness-of-fit test for the multiple logistic regression model. *Communications in Statistics - Theory and Methods*, 9(10), 1043–1069. https://doi.org/10.1080/03610928008827941

See Also

`hoslem.test` for the Hosmer-Lemeshow test

Examples

```
# Example 1: Binary data with automatic grouping (Ebrahim-Farrington test)
set.seed(123)
n <- 500
x <- rnorm(n)
linpred <- 0.5 + 1.2 * x
prob <- 1 / (1 + exp(-linpred))
y <- rbinom(n, 1, prob)

# Fit logistic regression
model <- glm(y ~ x, family = binomial())
predicted_probs <- fitted(model)

# Perform Ebrahim-Farrington test with 10 groups
result <- ef.gof(y, predicted_probs, G = 10)
print(result)

# Example 2: Compare with different number of groups
result_4 <- ef.gof(y, predicted_probs, G = 4)
result_20 <- ef.gof(y, predicted_probs, G = 20)

# Example 3: Grouped data (original Farrington test)
# Note: This requires actual grouped data with trials > 1
## Not run:
# Simulated grouped data
n_groups <- 50
m_trials <- sample(5:20, n_groups, replace = TRUE)
x_grouped <- rnorm(n_groups)
```

```
linpred_grouped <- -0.5 + 1.0 * x_grouped
prob_grouped <- 1 / (1 + exp(-linpred_grouped))
y_grouped <- rbinom(n_groups, m_trials, prob_grouped)

# Fit model for grouped data
data_grouped <- data.frame(successes = y_grouped, trials = m_trials, x = x_grouped)
model_grouped <- glm(cbind(successes, trials - successes) ~ x,
                     data = data_grouped, family = binomial())
predicted_probs_grouped <- fitted(model_grouped)

# Original Farrington test
result_grouped <- ef.gof(y_grouped, predicted_probs_grouped,
                         model = model_grouped, m = m_trials)
print(result_grouped)

## End(Not run)
```

`run.all.gof` *Run a Battery of Goodness-of-Fit Tests at Once*

Description

Runs several goodness-of-fit tests for a binary logistic regression in one call and returns one tidy `data.frame`, one row per test. Pass a fitted `glm` to run the whole battery; pass `(y, predicted_probs)` to run the tests that need only predictions. Each test is wrapped so that a failure of one test never aborts the whole run.

Usage

```
run.all.gof(
  object,
  predicted_probs = NULL,
  X = NULL,
  tests = "all",
  G = 10,
  include_slow = FALSE,
  control = list()
)
```

Arguments

`object`	A fitted binary logistic `glm`, or a binary (0/1) response vector `y` (then supply `predicted_probs`).
`predicted_probs`	Numeric predicted probabilities; required when `object` is a `y` vector.
`X`	Optional design matrix; lets the directed (DEF) tests run from the `(y, predicted_probs)` form.

tests	Either "all" (default) or a character vector of test names to run (e.g. c("EF","DEF.poly3","HL")).
G	Integer number of groups passed to the grouping tests (default 10).
include_slow	Logical; when TRUE, also run the opt-in slow tests (currently the le Cessie-van Houwelingen smoothing test, which builds an n-by-n kernel matrix and is O(n^2)-O(n^3)). Default FALSE.
control	Optional named list of per-test options (reserved).

Details

The currently bundled tests are: Pearson, Deviance, Osius-Rojek, Copas-RSS, and Information-Matrix (the White/Orme test) (global / standardized); HL (Hosmer-Lemeshow deciles), HL-equalwidth, and Pigeon-Heyse (partition); EF and EF-normal (the omnibus Ebrahim-Farrington test with the chi-square and normal references; the normal form reproduces the thesis simulation); DEF.poly2/poly3/stukel and Stukel (directed); Tsiatis, Xie, and Pulkstenis-Robinson (covariate-space); the two ensemble rows (Ensemble.Vote(3DEF) and Ensemble.Univ(3DEF+EF)) from the Cauchy combination test; and, when include_slow = TRUE, the opt-in slow tests: le-Cessie-van Houwelingen smoothing, the GAM-based tests HL-GAM, PR-GAM, Xie-GAM (need **mgcv**; fit an overfit GAM for grouping), Stute-Zhu (a cumulative-residual parametric-bootstrap test; set the number of reps with control = list("Stute-Zhu" = list(B = ...))), eHL (the e-value Hosmer-Lemeshow test, reported as p = min(1, 1/e)), and BAGofT (the binary-adaptive GOF test; needs the **BAGofT** package, control = list(BAGofT = list(nsim = ...))), and Lai-Liu-HL (Lai & Liu's standardized-power procedure for the Hosmer-Lemeshow test, which has no p-value: it reports the standardized power as the statistic and a randomized accept/reject decision in the Note; target size via control = list("Lai-Liu-HL" = list(n0 = ..., k = ...))).

Notes: Tsiatis and Xie cluster the covariate space with k-means (a fixed internal seed, so results are reproducible and the caller's RNG is left untouched). Xie uses the corrected degrees of freedom $G-k/2-1$ with k the number of predictors. Pulkstenis-Robinson auto-detects the categorical covariate (any factor/character/logical, or a numeric with at most getOption("ebrahim.gof.pr.maxlev", 6) distinct values); it returns NA with a note when none is present.

Every bundled test reproduces the implementation used in the original thesis simulation: Osius-Rojek and Stukel follow **LogisticDx**'s gof.glm (Stukel via statmod::glm.scoretest when **statmod** is installed), Copas-RSS follows **rms**'s gof residual, HL follows ResourceSelection::hoslem.test, and the others match their standalone reference functions; all were checked to agree numerically.

Value

A data.frame with columns Test, Family, Statistic, df, p_value, and Note.

Author(s)

Ebrahim Khaled Ebrahim <ebrahimkhaled@alexu.edu.eg>

See Also

ef.gof, def.gof, def.ensemble.gof.

Examples

```
set.seed(1)
n <- 500
x <- runif(n, -3, 3)
y <- rbinom(n, 1, 1 / (1 + exp(-(0.6 * x))))
fit <- glm(y ~ x, family = binomial())
run.all.gof(fit)                       # the whole battery + ensemble rows
run.all.gof(fit, tests = c("EF", "DEF.poly3", "HL"))
run.all.gof(y, fitted(fit))            # prediction-only tests
```

Index