Package ‘famish’

December 8, 2025

Type Package

Title Flexibly Tune Families of Probability Distributions

Version 0.2.0

Description Fits probability distributions to data and plugs into the 'probaverse' suite of R packages so distribution objects are ready for further manipulation and evaluation. Supports methods such as maximum likelihood and L-moments, and provides diagnostics including empirical ranking and quantile score.

License MIT + file LICENSE

Encoding UTF-8

Imports checkmate, distionary, fitdistrplus, ismev, lmom, rlang, stats, vctrs

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL `https://famish.probaverse.com/`, `https://github.com/probaverse/famish/`

VignetteBuilder knitr

NeedsCompilation no

Author Vincenzo Coia [aut, cre, cph]

Maintainer Vincenzo Coia <`vincenzo.coia@gmail.com`>

Repository CRAN

Date/Publication 2025-12-08 08:20:11 UTC

Contents

fit_dst *Fit a distribution*

Description

Estimation of probability distributions available in the 'distionary' package. Wraps the 'lmom' package when fitting by L-moments, the 'ismev' package when fitting the GP/GEV/Gumbel by MLE, and the 'fitdistrplus' package for other combinations.

Usage

```
fit_dst(
  family,
  x,
  method = c("mle", "mge", "mme", "lmom", "lmom-log"),
  na_action = c("null", "drop", "fail"),
  on_unres = c("null", "fail")
)
```

Arguments

`family` Name of the target distribution family, such as `"norm"`, `"gev"`, `"pois"`. See details. Character vector of length 1.

`x` Numeric vector containing the observations to fit.

`method` Estimation method to use. Valid choices include `"mle"`, `"mge"`, `"mme"`, `"lmom"`, and `"lmom-log"`. The default is `"mle"`, although beware that not all families support the `"mle"` method yet (pearson3 and lp3).

`na_action` Strategy for dealing with `NA` values in `x`. `"null"` returns a Null distribution (`distionary::dst_null()`); `"drop"` silently removes missing observations before fitting; and `"fail"` aborts with an error.

`on_unres` Behaviour when no distribution can be resolved for the supplied inputs. `"null"` (default) yields a Null distribution (`distionary::dst_null()`) distribution with a warning, whereas `"fail"` propagates an error.

Details

The `fit_dst()` function is currently a lightweight fitting wrapper, with pre-specified behaviour for certain `family` / `method` combinations. A full list of families and their compatible estimation methods is available via specific family wrappers, such as `fit_dst_norm()`, `fit_dst_pois()`, etc.

Here is how fitting is implemented.

- For `method = "lmom"` and distribution families 'gamma', 'gev', 'gp', 'gumbel', 'lnorm', 'norm', 'pearson3', and 'weibull', the 'lmom' package is wrapped by first calling `lmom::samlmu()` on the data vector `x` to calculate the L-moments, then the relevant `lmom::pel*()` function is called to estimate parameters.

- For `method = "lmom"` and distribution families 'exp', 'pois', 'bern', 'geom', 'chisq', and 'unif', the method of L-moments is manually implemented. All of these families except 'unif' have a single parameter for which only the mean is needed (and thus is equivalent to the 'mme' method). The 'unif' family has minimum and maximum parameter values calculated as `l1 - 3 * l2` and `l1 + 3 * l2`, where `l1` and `l2` are the first and second L-moments (see Hosking, 1990, Table 1).
- For `method = "lmom-log"`, only the 'lnorm' and 'lp3' families are supported, otherwise no distribution will be resolved. The method fits the distributions via the 'lmom' method on the log scale. That is, 'norm' and 'pearson3' distributions are fit on the log of the data, for which the respective 'lnorm' or 'lp3' distribution is obtained.
- For `method = "mle"` and distribution families 'gev', 'gp', or 'gumbel', the 'ismev' package is used to fit the distribution by maximum likelihood estimation. This is done by invoking the functions `ismev::gev.fit()`, `ismev::gpd.fit()` (with `threshold = 0`), and `ismev::gum.fit()` (respectively).
- For `method = "mle"` and distribution family 'bern' and 'degenerate', the MLE is calculated manually. For 'bern', the parameter is estimated as the mean of the 0-1 data; for 'degenerate', the unique data value.
- For `method = "mme"` and `"lmom"`, the 'cauchy' family fails to fit because Cauchy distributions don't have finite moments (Feller, 1971).
- For families 'empirical' and 'finite', the empirical distribution is fit to the supplied data.
- For the 'null' family, a Null distribution is returned.
- For any other combination of `family` and `method`, the `fitdistrplus::fitdist()` function is called by inserting the data `x`, the `family` name, and the `method`. Some distributions require starting values for the parameters. For the families 't', 'f', and 'chisq', this is done by moment matching ('mme'). For 'gev', 'gp', and 'gumbel', the MLE is used as starting values (through `method = "mle"`).

To understand what the distribution families are, see the documentation in the 'distionary' package through the `dst_*()` functions. For example, the 'lp3' family can be found at `?distionary::dst_lp3()`. Note that the Gumbel distribution is not available yet in 'distionary', but is simply the 'gev' family with `shape = 0`.

To understand the estimation methods, see the 'lmom' package for the `"lmom"` method. For the `"lmom-log"` method, it is the same as `"lmom"`, but via the log of the data and the corresponding log-transformed distributions. For all others, see the 'fitdistrplus' package documentation.

Handling of missing or unresolvable data:

When `na_action = "drop"`, the function operates on the subset of `x` without missing values (via `x <- x[!is.na(x)]`). This takes priority over behaviour indicated in `on_unres`.

If fitting fails, a Null distribution is output if `on_unres = "null"` (the default), or an error is thrown if `on_unres = "fail"`. Fitting can fail due to not having enough data, not being able to isolate a single distribution, or various other reasons that would typically otherwise result in an error or `NA` parameters in the wrapped fitting method.

Value

A distribution object of class "dst" encapsulating the fitted distribution.

References

Hosking, J. R. M. (1990). L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(1), 105–124.

Feller, W. (1971). *An Introduction to Probability Theory and Its Applications* (Vol. 2, 2nd ed.). Wiley.

See Also

`fit_dst_*()` helpers such as `fit_dst_norm()`.

Examples

```
fit_dst("norm", x = 1:10, method = "mle")
fit_dst("gev", x = c(1, 4, 3, NA, 5), method = "lmom", na_action = "drop")
fit_dst("pois", x = c(1, 4, 3, NA, 5), na_action = "null")

# "lnorm" with "lmom-log" shares parameters with "norm" fit by "lmom".
fit_dst("lnorm", x = 1:10, method = "lmom-log")
fit_dst("norm", x = log(1:10), method = "lmom")
```

---

fit_dst_family_wrappers

Fit Distributions by Family

Description

Convenience wrappers around `fit_dst()` for each distribution family supported by the package.

Usage

```
fit_dst_bern(x, method = c("mle", "lmom"), ...)

fit_dst_beta(x, method = c("mle", "mge", "mme"), ...)

fit_dst_cauchy(x, method = c("mle", "mge"), ...)

fit_dst_chisq(x, method = c("mle", "mge", "lmom"), ...)

fit_dst_degenerate(x, method = "mle", ...)

fit_dst_empirical(x, ...)

fit_dst_exp(x, method = c("mle", "mge", "mme", "lmom"), ...)

fit_dst_f(x, method = c("mle", "mge"), ...)
```

```
fit_dst_finite(x, ...)

fit_dst_gamma(x, method = c("mle", "mge", "mme", "lmom"), ...)

fit_dst_geom(x, method = c("mle", "mme", "lmom"), ...)

fit_dst_gev(x, method = c("mle", "lmom"), ...)

fit_dst_gp(x, method = c("mle", "lmom"), ...)

fit_dst_gumbel(x, method = c("mle", "lmom"), ...)

fit_dst_lnorm(x, method = c("mle", "mge", "mme", "lmom", "lmom-log"), ...)

fit_dst_lp3(x, method = "lmom-log", ...)

fit_dst_nbinom(x, method = c("mle", "mme"), ...)

fit_dst_norm(x, method = c("mle", "mge", "mme", "lmom"), ...)

fit_dst_null(x, ...)

fit_dst_pearson3(x, method = "lmom", ...)

fit_dst_pois(x, method = c("mle", "mme", "lmom"), ...)

fit_dst_t(x, method = c("mle", "mge"), ...)

fit_dst_unif(x, method = c("mle", "mge", "mme", "lmom"), ...)

fit_dst_weibull(x, method = c("mle", "mge", "lmom"), ...)
```

Arguments

`x`	Numeric vector of observations to fit.
`method`	Estimation method to use. Available options depend on the distribution family and are enforced via `rlang::arg_match()`.
`...`	Additional arguments passed on to `fit_dst()`.

Details

Each helper simply forwards to `fit_dst()` with the associated `family` value indicated by the function suffix.

Some families do not have a unique fitting method where it is not applicable. These are the ’finite’ ones (including ’degenerate’ and ’empirical’), and the ’Null’ distribution.

Missing Combinations:

Some combinations of families and methods are not supported, when one might think that they should be. These combinations are:

- ’gp’ fit by ’mge’
- ’lp3’ fit by ’mle’ and ’mge’
- ’pearson3’ fit by ’mle’ and ’mge’
- ’gev’ fit by ’mge’

They are not included because of how the `fitdistrplus::fitdist()` function looks for those distributional representations (i.e., `plp3()`, `dlp3()`, etc.): since we cannot guarantee that it will find the correct ones, these combinations are omitted for safety.

To elaborate, the current version of the wrapped `fitdistrplus::fitdist()` function looks for these representations starting from the `namespace:fitdistrplus` environment. `famish` cannot control what’s found in that search path until the global environment, but that’s too late because that’s where behaviour becomes conditional on the user’s actions. This is not a problem for other distributions because these are found in the `namespace:base` environment before the global environment is reached.

Value

A distribution object made by the ’distionary’ package.

See Also

`fit_dst()`

Examples

```
# Calls can be quite simple.
fit_dst_norm(1:10)
fit_dst_gumbel(2:6)

# Still have access to the functionality available through `fit_dst()`
x <- c(1, 4, 3, NA, 5)
fit_dst_lnorm(x, method = "lmom", na_action = "null")
fit_dst_lnorm(x, method = "lmom", na_action = "drop")

# Fitting by l-moments on the log-scale not the same as original scale.
fit_dst_lnorm(x, method = "lmom-log", na_action = "drop")
```

quantile_score *Quantile score (pinball loss)*

Description

Computes the asymmetric absolute loss commonly used to assess quantile forecasts. Lower scores indicate a better match between the estimated quantile and the observed value at level `tau`.

Usage

```
quantile_score(x, xhat, tau)
```

Arguments

`x`	Numeric vector of observed values.
`xhat`	Numeric vector of estimated quantiles.
`tau`	Numeric vector of quantile levels in `(0, 1)`.

Details

The score minimises to zero when the observation equals the estimated quantile, so that smaller scores indicate a better fitting model. Positive residuals are penalised by a factor of `tau`, and negative residuals by `tau - 1`. This loss is also known as the stick function, check loss, asymmetric absolute deviation, or pinball loss.

For observation `x`, estimate `x_hat`, and level `tau`, the score (c.f. Gneiting, 2011) is

$$S_\tau(x, \hat{x}) = \begin{cases} \tau|x - \hat{x})|, & x \geq \hat{x}, \\ (1 - \tau)|x - \hat{x}|, & x < \hat{x}. \end{cases}$$

Vector recycling of all three arguments follows the rules in `vctrs::vec_recycle_common()`.

Value

Numeric vector of quantile scores corresponding to each element of the recycled inputs.

References

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494), 746–762.

Examples

```
quantile_score(c(5, 15, 10), xhat = 7, tau = 0.8)
quantile_score(c(5, 15, 10), xhat = c(6, 19, 12), tau = c(0.2, 0.9, 0.5))
```

`uscore` *Rank-based Scores*

Description

Converts a numeric vector to its rank-based scores. For `uscore()` (uniform scores), values become roughly equally spaced between 0 and 1, keeping their order. `nscore()` calculates normal scores by spacing the uniform scores along a standard normal distribution; `rpscore()` calculates empirical return periods by spacing the uniform scores `u` by `1 / (1 - u)`.

Usage

```
uscore(x, pos = "Hazen", na.rm = FALSE)

nscore(x, pos = "Hazen", na.rm = FALSE)

rpscore(x, pos = "Hazen", na.rm = FALSE)
```

Arguments

`x`	Numeric vector.
`pos`	Positional adjustment for uniform scores. See Details. Can be a single numeric, or could be named after one of the proponents behind a choice of the numeric: "Weibull", "Beard", "Gringorten", or "Hazen".
`na.rm`	Logical indicating whether `NA` and `NaN` values should be removed from the output.

Details

Uniform scores are calculated by `(rank+a)/(n+1+2*a)`, where `rank` is the ranked `x` values, and `a` is the positional adjustment `pos`. Alternatively, could be named after an individual associated with a choice of `a`:

- Weibull (1939) proposed `a` = `0`.
- Beard (1943) proposed `a` = `-0.31`.
- Gringorten (1963) proposed `a` = `-0.44`.
- Hazen (1914) proposed `a` = `-0.5`.

Value

Vector of uniform scores corresponding to values in `x`.

Author(s)

Thanks to Dr. Harry Joe for providing a starting framework for the `uscore()` function.

References

Beard, L. R. (1943). Statistical analysis in hydrology. *Transactions of the American Society of Civil Engineers*, 108, 1110–1160.

Gringorten, I. I. (1963). A plotting rule for extreme probability paper. *Journal of Geophysical Research*, 68(3), 813–814.

Hazen, A. (1914). Storage to be provided in impounding reservoirs for municipal water supply. *Transactions of the American Society of Civil Engineers*, 77, 1539–1640.

Weibull, W. (1939). A statistical theory of the strength of materials. *IVB-Handl.*, 151.

Examples

```
x <- c(0.3, 0.56, NA, 0.1, -12)
uscore(x)
uscore(x, pos = "Gringorten")
nscore(x, pos = -0.4)
rpscore(x)
```

Index