Package: DistFamilyFit (via r-universe)

November 25, 2026

Type Package

Title Fitting and Comparing Flexible Continuous Distribution Families

Version 0.1.1

Description Fits and compares four flexible families of continuous distributions, with the Johnson family as the default. Pearson curves described by Pearson (1895) <doi:10.1098/rsta.1895.0010> are fitted by four-moment matching through 'PearsonDS'. The unbounded (SU), lognormal (SL) and bounded (SB) translation systems of Johnson (1949) <doi:10.1093/biomet/36.1-2.149> are fitted by profiled numerical maximum likelihood, and Burr Type III and Type XII distributions of Burr (1942) <doi:10.1214/aoms/1177731607>, see also Tadikamalla (1980) <doi:10.2307/1402945>, by numerical maximum likelihood. The generalized lambda distribution of Ramberg and Schmeiser (1974) <doi:10.1145/360827.360840> is used in the parameterization of Freimer, Kollia, Mudholkar and Lin (1988) <doi:10.1080/03610928808829820> and is fitted by quantile matching with numerical inversion of its quantile function. Candidates are ranked by descriptive distances between fitted and empirical cumulative distribution functions. Johnson marginal transformations to a common normal scale support descriptive dependence analysis without implying joint normality on the original scale. The package also provides numeric data validation with row-level audits, density, distribution and quantile evaluation, and separate plotting functions. Count data are treated as continuous approximations. The methodology and its application to mobile application usage data are described in Karakaya Demir, Purutçuoğlu and Bursalı (2026), accepted at the 3rd International Conference on Advanced Technology in Electronic and Electrical Engineering (ICATEEE 2026).

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports stats, graphics, grDevices, utils

Suggests PearsonDS

NeedsCompilation no

Author Şule Şevval Karakaya Demir [aut, cre] (ORCID: <https://orcid.org/0000-0003-1851-4109>), Vilda Purutçuoğlu [aut] (ORCID: <https://orcid.org/0000-0002-0757-0837>), Ahmet Bursalı [aut] (ORCID: <https://orcid.org/0000-0001-7050-769X>), TÜBİTAK [fnd] (Scientific and Technological Research Council of Türkiye, 1505 research project no. 5240032)

Maintainer Şule Şevval Karakaya Demir <sule.karakaya@metu.edu.tr>

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

Date/Publication 2026-11-25 17:26:06 UTC

RemoteUrl https://github.com/cran/DistFamilyFit

RemoteRef HEAD

RemoteSha cd9ddcce5f5ea0a02fc78b9387f9aeb229493f90

Contents

DistFamilyFit-package 2
cdf_fit 3
DistFamilyFit-methods 4
DistFamilyFit_glossary 5
fit_columns 8
fit_multivariate 9
fit_univariate 11
plot_fit 13
read_fit_data 14

Index **16**

`DistFamilyFit-package` *Fitting and Comparing Flexible Continuous Distribution Families*

Description

Fit and compare the Pearson, Johnson, Burr and generalized lambda families of continuous distributions, and use Johnson marginal transformations for descriptive dependence analysis on a common normal scale. Data preparation is explicit and every omitted observation is recorded.

Details

Start with `read_fit_data` or `prepare_fit_data`. Use `fit_univariate` to fit the Johnson family (the default), Pearson, Burr, generalized lambda or all families. `fit_columns` fits several columns independently. `fit_johnson_multivariate` describes dependence in the Johnson-transformed space. Fitting functions never draw; plots are produced only by the separate plotting functions.

Family	Forms	Estimation
Pearson	Types 0 to VII (via **PearsonDS**)	Four-moment matching
Johnson	SU, SL, SB	Profiled maximum likelihood
Burr	Type III, Type XII	Maximum likelihood
Generalized lambda	FKML parameterization	Quantile matching

Candidates are ranked by the classical Kolmogorov–Smirnov distance between the fitted and empirical cumulative distribution functions. The plotting-position distance `D_CDF`, $\max_i |F(x_{(i)}) - (i - 0.5)/n|$, is also reported. Both are descriptive in-sample discrepancy measures and not formal goodness-of-fit tests.

The package implements the methodology of Karakaya Demir, Purutçuoğlu and Bursalı (2026); please cite this paper when using the package (see `citation("DistFamilyFit")`).

Author(s)

Şule Şevval Karakaya Demir (author and maintainer), Department of Biomedical Engineering, Middle East Technical University, Ankara, Türkiye. <sule.karakaya@metu.edu.tr>. ORCID: 0000-0003-1851-4109.

Vilda Purutçuoğlu (author), Department of Biomedical Engineering and Department of Statistics, Middle East Technical University, Ankara, Türkiye. <vpurutucu@metu.edu.tr>. ORCID: 0000-0002-0757-0837.

Ahmet Bursalı (author), Intelligent Application DC Department, Huawei Turkey Research and Development Center, İstanbul, Türkiye. <ahmet.bursali@huawei.com>. ORCID: 0000-0001-7050-769X.

ORCID stands for Open Researcher and Contributor ID. DC is retained as supplied in the department name; no expansion was provided.

See Also

`fit_univariate`, `fit_johnson_multivariate`, `plot_fit`

`cdf_fit` *Evaluate Fitted Distributions in Original Data Units*

Description

Evaluate CDFs, densities and quantiles of successfully fitted univariate candidates.

Usage

```
cdf_fit(object, x, model = object$best)
density_fit(object, x, model = object$best, log = FALSE)
quantile_fit(object, p, model = object$best)
```

Arguments

object	Univariate fit.
x	Numeric evaluation points.
p	Probabilities in [0,1].
model	Candidate name.
log	Return log density.

Value

Numeric values in original data units.

Examples

```
set.seed(2026)
f <- fit_johnson_univariate(sinh(rnorm(60)), types = "SU")
cdf_fit(f, c(-1, 0, 1))
density_fit(f, c(-1, 0, 1))
quantile_fit(f, c(.1, .5, .9))
```

DistFamilyFit-methods *Inspect Distribution Fit Objects*

Description

Print fit rankings, inspect detailed summaries and plot fitted distributions or transformed variables.

Usage

```
## S3 method for class 'familyfit_univariate'
print(x, ...)
## S3 method for class 'familyfit_univariate'
summary(object, ...)
## S3 method for class 'familyfit_univariate'
plot(x, type = c("cdf", "qq", "density"),
  model = x$best, ...)
## S3 method for class 'familyfit_multivariate'
print(x, ...)
## S3 method for class 'familyfit_multivariate'
summary(object, ...)
## S3 method for class 'familyfit_multivariate'
plot(x, ...)
```

Arguments

x, object	Fit object.
type	Univariate plot type.
model	Univariate candidate name.
...	Additional plotting arguments or unused method arguments.

Value

Print/plot invisibly return the input; summary returns a list.

Examples

```
(function() {
  oldpar <- par(no.readonly = TRUE)
  on.exit({
    par(oldpar)
    par(mfg = oldpar$mfg, new = oldpar$new)
  }, add = TRUE)

set.seed(2026)
u <- fit_johnson_univariate(sinh(rnorm(60)), types = "SU")
print(u)
summary(u)
plot(u)

})()
```

DistFamilyFit_glossary

Abbreviations and diagnostic terminology

Description

Full names and detailed interpretation of abbreviations, identifiers and result fields used by DistFamilyFit.

Details

These definitions describe the implemented methods. A computational status or descriptive distance is not statistical model acceptance.

Abbreviations and identifiers

SU Johnson unbounded form. A Johnson distribution whose support extends over the whole real line. Its normal-scale transformation uses the inverse hyperbolic sine. SU is a family identifier, not a fit-quality score.

SL Johnson lognormal form. A Johnson distribution with a lognormal-type transformation and a finite lower endpoint. The implementation fixes a redundant scale parameter, leaving three identifiable parameters. SL does not mean that the original observations are normally distributed.

SB Johnson bounded form. A Johnson distribution supported between two finite endpoints. Its normal-scale transformation is based on a log ratio of distances to the endpoints. The implemented Johnson labels are SU, SL and SB; SD is not an implemented family label.

GLD Generalized lambda distribution. A flexible continuous distribution specified through its quantile function. In this package it is fitted by matching selected empirical quantiles. Its cumulative probabilities are obtained by numerical inversion; the fitted quantile curve does not guarantee the correct family or accurate extreme tails.

FKML Freimer-Kollia-Mudholkar-Lin. The names identifying the generalized lambda parameterization implemented here. It has location, scale and two shape parameters. This identifies a parameterization, not the estimation method: DistFamilyFit estimates its parameters by quantile matching, not maximum likelihood.

MLE Maximum likelihood estimation. Parameter estimation by maximizing the product of model densities at the observed values, equivalently minimizing negative log-likelihood. Johnson uses profiled, multistart numerical optimization; Burr uses numerical optimization. Convergence does not establish a global optimum. Pearson moment matching and generalized lambda quantile matching are not maximum likelihood estimates.

CDF Cumulative distribution function. For a value x, the cumulative distribution function is the model probability P(X <= x). It ranges from zero to one. cdf_fit() evaluates it on the original measurement scale. A density value is different: a continuous-variable density is not the probability of observing one exact value.

ECDF Empirical cumulative distribution function. The fraction of retained observations less than or equal to x. It is a step function calculated directly from the sample. The classical distance uses its left and right limits; the separately preserved midpoint distance uses sorted plotting positions instead.

KS Kolmogorov-Smirnov. Here, the largest absolute difference between the fitted cumulative distribution function and the empirical cumulative distribution function, accounting for the left and right limits of its steps. Smaller values indicate closer sample fit. The reported distance is not a p-value, and ks_warning is a descriptive threshold rather than a significance level.

NLL Negative log-likelihood. Minus the sum of log model densities for the retained observations, evaluated in original measurement units. Smaller values mean larger likelihood on the same observations and scale. It is reported for maximum-likelihood candidates. Numerical boundary or subsampling caveats still apply to any information criterion derived from it.

AIC Akaike information criterion. A likelihood criterion combining fit and parameter count: AIC = 2*k - 2*logLik, equivalently 2*k + 2*NLL. The univariate implementation omits it for non-likelihood fits, flagged search-boundary solutions and estimation subsampling. Compare only compatible likelihood fits to the same data. A small value does not establish model acceptance.

BIC Bayesian information criterion. A likelihood criterion with an additional sample-size-dependent penalty: BIC = k*log(n) - 2*logLik. In this package BIC_Z is provided only for the conditional normal model on a fixed transformed data set. It excludes marginal estimation, is omitted for ridge regularization, and must not be compared across different transformations or interpreted as full original-scale joint-model evidence.

RMSE Root mean square error. The square root of the average squared discrepancy. The CDF_quantile_RMSE field compares fitted cumulative probabilities at empirical quantiles with their target probabilities. Its seven probabilities are 0.1, 0.25, 0.5, 0.75, 0.9, 0.95 and 0.99. It is measured on the probability scale, not in the original variable units.

Q-Q Quantile-quantile. A plot comparing theoretical model quantiles with ordered observed values. Proximity to the identity line supports descriptive agreement, while endpoint departures may indicate tail mismatch. The multivariate reference plot compares squared Mahalanobis distances with chi-square quantiles and is descriptive rather than a calibrated test.

L-BFGS-B Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with box constraints. A numerical optimizer that uses limited memory and lower/upper parameter bounds. It optimizes the profiled Johnson location/support parameters. Reaching the finite search guards is a warning, not proof of a distribution support endpoint or a global maximum.

RNG Random number generator. The mechanism producing pseudorandom draws. With seed = NULL, optional estimation subsampling uses and advances the current random number generator state. An explicit seed is applied locally and the previous state is restored. A reproducible seed does not remove sampling uncertainty.

CSV Comma-separated values. A rectangular plain-text data format commonly using commas between fields. One header row names the columns. DistFamilyFit requires explicit separator and decimal-mark choices; semicolon-delimited files are handled by setting sep accordingly.

TSV Tab-separated values. A rectangular plain-text format using a tab character between fields. Use sep="\t" when reading this format. Numeric columns still follow the same observation, missing-value and decimal-mark requirements.

UTF-8 8-bit Unicode Transformation Format. The text encoding expected for input files and used for the package source. It represents non-ASCII characters, including the authors names, without replacing them with unrelated characters. It is an encoding name, not a delimiter or a statistical method.

ID Identifier. A row, subject, application or other label used to identify records. An identifier column is not automatically an analysis variable. Select numeric analysis columns explicitly rather than fitting a distribution to numeric-looking identifiers.

NA Not available. The standard R marker for a missing value. Blank input fields and the text NA denote missing values by default. Missing or nonfinite selected observations cause an error unless na_action="omit" is explicitly requested; omission and original row indices are recorded.

CRAN Comprehensive R Archive Network. The repository network used to distribute R and contributed packages. A successful local package check does not imply acceptance by its reviewers. Release and development-version checks also do not replace native operating-system tests.

MIT The MIT License is named after the Massachusetts Institute of Technology. DESCRIPTION records MIT + file LICENSE; LICENSE supplies the copyright year and holders. The license permits use, modification and redistribution, including commercial use, subject to retaining the copyright and permission notices.

Rd R documentation. The source format of the function and package help pages in the man directory. R uses these files to generate help views and the reference manual. These files are maintained directly in this package.

S3 R S3 method-dispatch system. The name of the R system used to dispatch print(), summary() and plot() to methods appropriate for the fitted object class. It is a system name, not a distribution family or a fit statistic.

PDF Portable Document Format. The rendered format of the worked vignette and generated reference manual. The editable vignette source is the Sweave file; the rendered document must be rebuilt when that source changes.

HTML HyperText Markup Language. The markup format used for browser-based R help pages. Package checks can validate the generated help in this format as well as the rendered reference manual.

MATLAB Matrix Laboratory. The software name associated with the original implementation and the separately retained midpoint-distance convention. The software name does not denote a distribution or an accuracy measure.

OK Okay; computational checks passed. A result-status label, not a calibrated statistical decision. Warning candidates may also be selected; Unstable and Failed candidates are excluded from automatic selection. Inspect diagnostic notes before interpreting the selected model.

Other result fields

`Z` denotes Johnson-transformed observations. It is a mathematical symbol, not an acronym. `logLik_Z`, `AIC_Z` and `BIC_Z` are conditional normal-model quantities on a fixed transformed data set, excluding marginal estimation.

`MatchRate` is 100 times one minus the midpoint distance evaluated at plotting positions (i - 0.5)/n; it is not accuracy, a probability or a p-value. `TailError` is the root mean square difference of base-10 logarithms of fitted and target survival probabilities at 0.9, 0.95 and 0.99, with a machine- precision offset before taking logarithms. It is not a tail confidence bound.

Burr III and XII denote types three and twelve. **PearsonDS** is an optional software package name, not a statistical quantity. TRUE, FALSE, NULL and Inf are R constants or special values.

See Also

`fit_univariate`, `fit_multivariate`, `plot_fit`

`fit_columns` *Column-Wise Fitting and Johnson Convenience Functions*

Description

Fit independent columns, fit Johnson-only models or evaluate a Johnson normal-scale transformation.

Usage

```
fit_columns(data, columns = NULL, families = "Johnson",
  na_action = c("fail", "omit"), ...)
fit_johnson_univariate(x, types = c("SU", "SL", "SB"), ...)
fit_johnson_multivariate(data, columns = NULL, types = c("SU", "SL", "SB"), ...)
transform_johnson(object, x, model = object$best)
```

Arguments

data	Numeric matrix, data.frame, vector or prepared data (multivariate requires at least two columns).
columns	Analysis column names or indices.
families	Johnson by default; Pearson, Burr, GLD, a subset or all.
na_action	Fail or omit complete rows explicitly.
x	Numeric vector or one-column prepared dataset for fitting; numeric original-scale values for transformation.
types	Subset of unbounded (SU), lognormal (SL) and bounded (SB) Johnson forms.
object	A univariate fit containing a Johnson candidate.
model	Johnson candidate name.
...	Arguments forwarded to fit_univariate or fit_multivariate, except arguments already supplied by the wrapper.

Value

fit_columns returns a named list of independent univariate fits and an input_audit attribute; it is not a joint model. Complete-case preprocessing is shared across selected columns, and individual fits preserve original-row audit identifiers. Johnson wrappers return their corresponding fit object. transform_johnson returns a numeric vector.

Examples

```
set.seed(2026)
x <- sinh(rnorm(60))
u <- fit_johnson_univariate(x, types = "SU")
transform_johnson(u, c(-1, 0, 1))
d <- data.frame(a = x, b = sinh(rnorm(60)))
separate <- fit_columns(d, families = "Johnson", johnson_types = "SU")
joint <- fit_johnson_multivariate(d, types = "SU")
```

---

fit_multivariate *Johnson-Transformed Multivariate Normal Model*

Description

Fit Johnson marginals, transform observations and estimate dependence in normal space.

Usage

```
fit_multivariate(data, columns = NULL, na_action = c("fail", "omit"),
  ridge = 0, johnson_types = c("SU", "SL", "SB"), ks_warning = 0.2, ...)
```

Arguments

data	Numeric matrix or data frame with at least two variables.
columns	Optional column selection.
na_action	Fail or use complete-case observations.
ridge	Explicit diagonal regularization, zero by default.
johnson_types	Johnson marginal types.
ks_warning	Descriptive marginal warning threshold.
...	Other arguments to fit_univariate; not families, na_action or component.

Details

Covariance and correlation describe transformed Z variables. The normal maximum likelihood estimation (MLE) covariance uses divisor n. Information criteria describe the Z-space normal model conditional on fitted marginals; they do not describe the original-scale joint model and must not be used to compare models with different fitted transformations. With regularization AIC/BIC are unavailable. Singular covariance is reported rather than silently regularized. Each marginal is mapped to the normal scale by its fitted Johnson (1949) transformation; the resulting dependence summaries are descriptive.

Value

A `familyfit_multivariate` object with marginals, Z, covariance/correlation and data audit.

References

Karakaya Demir, Ş. Ş., Purutçuoğlu, V. and Bursalı, A. (2026). Modeling Mobile Application Usage with Distribution Families. Accepted at the 3rd International Conference on Advanced Technology in Electronic and Electrical Engineering (ICATEEE 2026), Hammamet, Tunisia.

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, 36(1–2), 149–176. doi:10.1093/biomet/36.12.149

Examples

```
set.seed(2026)
z <- rnorm(60)
d <- data.frame(first = sinh(z), second = sinh(.5 * z + rnorm(60)))
f <- fit_multivariate(d, johnson_types = "SU")
f$correlation_Z
```

`fit_univariate` *Fit and Compare Continuous Distribution Families*

Description

Fit and rank Pearson, Johnson, Burr and Freimer-Kollia-Mudholkar-Lin (FKML) generalized lambda candidates for one numeric variable.

Usage

```
fit_univariate(x, families = "Johnson",
  na_action = c("fail", "omit"), component = c("all", "positive"),
  min_n = 20L, max_fit_n = Inf, seed = NULL,
  control = list(maxit = 8000, reltol = 1e-9),
  johnson_types = c("SU", "SL", "SB"), ks_warning = 0.2)
```

Arguments

`x`	Numeric vector or one-column prepared dataset.
`families`	Nonempty selection of `"Pearson"`, `"Johnson"`, `"Burr"` and `"GLD"`, or `"all"`. Johnson is the default.
`na_action`	Fail or explicitly omit nonfinite values.
`component`	All data, or a conditional fit to the positive observations.
`min_n`	Minimum retained sample size, at least five.
`max_fit_n`	Maximum estimation sample size. When the retained sample is larger, parameters are estimated on a random subsample of this size; all retained observations are still used for evaluation. `Inf` (default) uses all observations and involves no random numbers.
`seed`	Only used when `max_fit_n` triggers subsampling. `NULL` (default) draws the subsample from the current random number stream, so results follow `set.seed` called beforehand. A nonnegative integer is applied locally for a reproducible subsample, after which the caller's random number state is restored.
`control`	Named control options: maxit, reltol, factr, pgtol, trace, REPORT, ndeps, alpha, beta, gamma. Solver-specific options are filtered; ndeps is a positive scalar. Objective direction and scaling cannot be overridden. Johnson uses profile likelihood and the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with box constraints (L-BFGS-B).
`johnson_types`	Subset of unbounded (SU), lognormal (SL) and bounded (SB) Johnson forms.
`ks_warning`	Descriptive warning threshold, not a significance level.

Details

Pearson curves are fitted by matching the first four empirical moments with **PearsonDS** (Pearson, 1895). Johnson unbounded (SU), lognormal (SL) and bounded (SB) forms (Johnson, 1949) are fitted by maximum likelihood with the normal-scale mean and variance profiled out analytically. Burr Type III and Type XII (Burr, 1942; Tadikamalla, 1980) are fitted by numerical maximum likelihood and require strictly positive data. The generalized lambda distribution (Ramberg and Schmeiser, 1974) uses the FKML parameterization (Freimer et al., 1988; Chalabi et al., 2012) and is fitted by matching empirical quantiles; its CDF is evaluated by numerical inversion of the quantile function.

These are continuous models. Discrete data and point masses are not modeled. `KS` is the classical ECDF distance and `D_CDF` is the plotting-position distance $\max_i |F(x_{(i)}) - (i - 0.5)/n|$. `MatchRate` is $100(1 - D_{CDF})$, not a probability or p-value. Original-row audit identifiers survive preprocessing; `n_preprocessing_excluded` counts previous omissions. Pearson moment targets use the empirical divisor-n convention. SL has three identifiable parameters. Pearson and GLD do not receive AIC, and AIC is omitted when estimation uses a subsample. Status OK denotes computational diagnostics, not statistical acceptance. Warning candidates may be selected; Unstable and Failed candidates are excluded. Numerical Johnson search bounds on standardized log scales are [-12, 12]; reaching them is reported.

Value

A `familyfit_univariate` object containing the ranking table, fitted models, best candidate, retained data, settings and data audit.

References

Karakaya Demir, Ş. Ş., Purutçuoğlu, V. and Bursalı, A. (2026). Modeling Mobile Application Usage with Distribution Families. Accepted at the 3rd International Conference on Advanced Technology in Electronic and Electrical Engineering (ICATEEE 2026), Hammamet, Tunisia.

Pearson, K. (1895) doi:10.1098/rsta.1895.0010. Johnson, N. L. (1949) doi:10.1093/biomet/36.1-2.149. Burr, I. W. (1942) doi:10.1214/aoms/1177731607. Tadikamalla, P. R. (1980) doi:10.2307/1402945. Ramberg, J. S. and Schmeiser, B. W. (1974) doi:10.1145/360827.360840. Freimer, M., Kollia, G., Mudholkar, G. S. and Lin, C. T. (1988) doi:10.1080/03610928808829820. Chalabi, Y., Scott, D. J. and Wuertz, D. (2012) MPRA Paper No. 43333. See `DistFamilyFit-package` for full citations.

See Also

`fit_johnson_univariate`, `cdf_fit`, `plot_fit`

Examples

```
set.seed(1)
f <- fit_univariate(rnorm(80), families = "Johnson")
print(f)

# Estimation on a subsample: reproducible either through set.seed() or seed.
x <- sinh(rnorm(300))
a <- fit_univariate(x, johnson_types = "SU", max_fit_n = 100, seed = 42)
```

```
b <- fit_univariate(x, johnson_types = "SU", max_fit_n = 100, seed = 42)
identical(a$ranking, b$ranking)
```

---

plot_fit *Distribution and Dependence Visualizations*

Description

Plot distributions after estimation. Fitting functions do not open graphics devices.

Usage

```
plot_fit(object, type = c("cdf", "qq", "density"), model = object$best, ...)
plot_comparison(object, models = NULL, ...)
plot_dependence(object, type = c("correlation", "covariance"))
plot_transformed(object, type = c("pairs", "mahalanobis"), ...)
```

Arguments

object	Corresponding univariate or multivariate fit.
type	Requested plot type.
model	Univariate candidate name.
models	Candidates to compare. NULL uses computationally usable candidates.
...	Distinct named arguments to the base plotting function; main and axis labels override defaults. Evaluation data cannot be overridden. Density overlays require probability=TRUE/freq=FALSE.

Details

Dependence is displayed in transformed Z space. The Mahalanobis quantile-quantile (Q-Q) plot uses a chi-square reference and is descriptive because parameters and transformations were estimated. It is not a calibrated goodness-of-fit test.

Value

Invisibly the input object, except plot_dependence, which invisibly returns the plotted matrix.

Examples

```
(function() {
  oldpar <- par(no.readonly = TRUE)
  on.exit({
    par(oldpar)
    par(mfg = oldpar$mfg, new = oldpar$new)
  }, add = TRUE)

set.seed(2026)
```

```
x <- sinh(rnorm(60))
u <- fit_johnson_univariate(x, types = "SU")
plot_fit(u, type = "qq")
plot_comparison(u)
m <- fit_johnson_multivariate(data.frame(a = x, b = sinh(rnorm(60))), types = "SU")
plot_dependence(m)
plot_transformed(m, type = "mahalanobis")

})()
```

`read_fit_data` *Read and Validate Rectangular Numeric Data*

Description

Select explicit numeric columns, validate observations and record every omitted row.

Usage

```
read_fit_data(file, sep = ",", dec = ".", columns = NULL,
  na_strings = c("", "NA"), na_action = c("fail", "omit"), min_n = 20L)
prepare_fit_data(data, columns = NULL, na_action = c("fail", "omit"),
  convert_numeric = FALSE, min_n = 20L)
```

Arguments

`file`	Existing UTF-8 CSV or TSV text file with a unique header.
`data`	Numeric vector, matrix, data.frame or familyfit_data object.
`sep`	Single-character separator.
`dec`	Decimal mark, dot or comma; distinct from sep.
`columns`	Distinct existing column names or positive indices. NULL means all columns.
`na_strings`	Strings representing missing values.
`na_action`	Fail or explicitly omit rows with nonfinite selected values.
`min_n`	Minimum retained sample size, at least five.
`convert_numeric`	Explicitly allow safe character-to-numeric conversion in memory.

Details

One row represents one observation. For multivariate analysis all values in a row must correspond to the same observation. ID, category and date columns must be excluded with columns. Matrix-valued data.frame columns and time-valued analysis vectors are rejected. No negative-value, zero or outlier deletion is performed. Omission applies only to selected columns and is complete-case across them.

Value

A familyfit_data object containing a numeric data.frame and an audit.

Examples

```
set.seed(1)
d <- data.frame(id = letters[1:20], value = rnorm(20))
p <- prepare_fit_data(d, columns = "value")
p$audit
path <- system.file("extdata", "input_template.csv", package = "DistFamilyFit")
read_fit_data(path, columns = "variable_1")$audit$n_retained
```

Index

∗ **distribution**
    DistFamilyFit-package, 2
∗ **package**
    DistFamilyFit-package, 2

cdf_fit, 3, *12*

density_fit *(*cdf_fit*)*, 3
DistFamilyFit *(*DistFamilyFit-package*)*, 2
DistFamilyFit-glossary
        *(*DistFamilyFit_glossary*)*, 5
DistFamilyFit-methods, 4
DistFamilyFit-package, 2
DistFamilyFit_glossary, 5

fit_columns, *3*, 8
fit_johnson_multivariate, *3*
fit_johnson_multivariate *(*fit_columns*)*,
        8
fit_johnson_univariate, *12*
fit_johnson_univariate *(*fit_columns*)*, 8
fit_multivariate, *8*, 9
fit_univariate, *3*, *8*, 11

plot.familyfit_multivariate
        *(*DistFamilyFit-methods*)*, 4
plot.familyfit_univariate
        *(*DistFamilyFit-methods*)*, 4
plot_comparison *(*plot_fit*)*, 13
plot_dependence *(*plot_fit*)*, 13
plot_fit, *3*, *8*, *12*, 13
plot_transformed *(*plot_fit*)*, 13
prepare_fit_data, *3*
prepare_fit_data *(*read_fit_data*)*, 14
print.familyfit_multivariate
        *(*DistFamilyFit-methods*)*, 4
print.familyfit_univariate
        *(*DistFamilyFit-methods*)*, 4

quantile_fit *(*cdf_fit*)*, 3

read_fit_data, *3*, 14

set.seed, *11*
summary.familyfit_multivariate
        *(*DistFamilyFit-methods*)*, 4
summary.familyfit_univariate
        *(*DistFamilyFit-methods*)*, 4

transform_johnson *(*fit_columns*)*, 8