Package ‘MVN’

July 21, 2025

Type Package

Title Multivariate Normality Tests

Version 6.1

Maintainer Selcuk Korkmaz <`selcukorkmaz@gmail.com`>

Imports methods, nortest, moments, MASS, boot, car, dplyr, tidyr, purrr, stringr, tibble, ggplot2, viridis, cli, energy, plotly, mice

Collate 'mvn.R' 'mardia.R' 'hz.R' 'hw.R' 'royston.R' 'doornik_hansen.R' 'energy.R' 'descriptives.R' 'test_univariate_normality.R' 'multivariate_diagnostic_plot.R' 'mv_outlier.R' 'univariate_diagnostic_plot.R' 'power_transform.R' 'arw_adjustment.R' 'plot.mvn.R' 'summary.mvn.R' 'impute_missing.R'

Description A comprehensive suite for assessing multivariate normality using six statistical tests (Mardia, Henze–Zirkler, Henze–Wagner, Royston, Doornik–Hansen, Energy). Also includes univariate diagnostics, bivariate density visualization, robust outlier detection, power transformations (e.g., Box–Cox, Yeo–Johnson), and imputation strategies (``mean'', ``median'', ``mice'') for handling missing data. Bootstrap resampling is supported for selected tests to improve p-value accuracy in small samples. Diagnostic plots are available via both 'ggplot2' and interactive 'plotly' visualizations. See Korkmaz et al. (2014) <`https://journal.r-project.org/archive/2014-2/korkmaz-goksuluk-zararsiz.pdf`>.

Encoding UTF-8

License MIT + file LICENSE

URL `https://selcukorkmaz.github.io/mvn-tutorial/`,
`https://github.com/selcukorkmaz/MVN`,
`http://biosoft.erciyes.edu.tr/app/MVN`

BugReports `https://github.com/selcukorkmaz/MVN/issues`

RoxygenNote 7.3.2

NeedsCompilation no

Author Selcuk Korkmaz [aut, cre] (ORCID:
<https://orcid.org/0000-0003-4632-6850>),
Dincer Goksuluk [aut],
Gokmen Zararsiz [aut]

Repository CRAN

Date/Publication 2025-06-10 16:00:06 UTC

Contents

`arw_adjustment` *Atkinson–Riani–Welsh (ARW) Adjusted Cutoff for Robust Mahalanobis Distances*

Description

Implements the ARW procedure to compute an adjusted cutoff for squared Mahalanobis distances, then re-estimates location and scatter excluding points beyond the cutoff.

Usage

```
arw_adjustment(x, m0, c0, alpha, pcrit)
```

Arguments

`x`	A numeric matrix or data frame of observations (rows) by variables (columns), with at least 2 columns.
`m0`	A numeric vector of initial location estimates (length equal to number of columns in `x`).
`c0`	A numeric covariance matrix corresponding to `m0`.
`alpha`	Numeric; significance level for the chi-square threshold. Defaults to 0.05 if not provided.
`pcrit`	Numeric; minimal proportion for the adjusted cutoff. If not provided, it is computed as: $(0.24 - 0.003p)/\sqrt{n}$ if $p \leq 10$, or $(0.252 - 0.0018p)/\sqrt{n}$ if $p > 10$.

Value

A list with the following components: `m`, the updated location vector after excluding outliers; `c`, the updated covariance matrix; `cn`, the adjusted cutoff on Mahalanobis distances; `w`, a logical vector indicating which observations have distance less than or equal to `cn`.

`descriptives` *Descriptive Statistics for Numeric Data*

Description

Computes key descriptive statistics for each numeric variable in a vector, matrix, or data frame.

Usage

```
descriptives(data)
```

Arguments

`data`	A numeric vector, matrix, or data frame with observations in rows and variables in columns.

Value

A data frame where each row corresponds to a variable and each column represents a summary statistic: number of non-missing observations (`n`), arithmetic mean (`Mean`), standard deviation (`Std.Dev`), median (`Median`), minimum (`Min`), maximum (`Max`), first quartile (`25th`), third quartile (`75th`), sample skewness (`Skew`, from `moments::skewness`), and sample kurtosis (`Kurtosis`, from `moments::kurtosis`).

Examples

```
## Not run:
data <- iris[1:4]
descriptives(data)

## End(Not run)
```

`doornik_hansen` *Doornik-Hansen Test for Multivariate Normality*

Description

Performs the Doornik–Hansen omnibus test by transforming the data to approximate normality and combining skewness and kurtosis measures to test for multivariate normality.

Usage

```
doornik_hansen(data, bootstrap = FALSE, B = 1000, cores = 1)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and variables in columns.
`bootstrap`	Logical; if `TRUE`, compute p-value via bootstrap resampling. Default is `FALSE`.
`B`	Integer; number of bootstrap replicates used when `bootstrap` = `TRUE`. Default is `1000`.
`cores`	Integer; number of cores for parallel computation when `bootstrap` = `TRUE`. Default is 1.

Value

A data frame with one row containing the following columns: `Test`, the name of the test ("Doornik-Hansen"); `Statistic`, the value of the test statistic; `df`, the degrees of freedom; and `p.value`, the p-value from a chi-square approximation.

Examples

```
## Not run:
data <- iris[1:50, 1:2]
dh_result <- doornik_hansen(data)
dh_result

## End(Not run)
```

`energy` *E-Statistic Test for Multivariate Normality (Energy Test)*

Description

Performs the E-statistic test for multivariate normality using a parametric bootstrap to estimate the null distribution of the test statistic.

Usage

```
energy(data, B = 1000, seed = 123)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and variables in columns.
`B`	Integer; number of bootstrap replicates to estimate the null distribution. Default is 1000.
`seed`	Optional integer to set the random seed for reproducibility.

Value

A data frame with one row containing the following columns: `Test`, the name of the test ("E-Statistic"); `Statistic`, the observed E-statistic; and `p.value`, the p-value obtained from the bootstrap procedure.

Examples

```
## Not run:
data <- iris[1:50, 1:4]
energy_result <- energy(data, B = 500)
energy_result

## End(Not run)
```

`hw` *Henze-Wagner High-Dimensional Test for Multivariate Normality*

Description

Performs the high-dimensional version of the BHEP test for multivariate normality as proposed by Henze and Wagner (1997). When the covariance matrix is singular (e.g., when $p > n$) a Moore-Penrose pseudoinverse is used.

Usage

```
hw(
  data,
  use_population = TRUE,
  tol = 1e-25,
  bootstrap = FALSE,
  B = 1000,
  cores = 1
)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and variables in columns.
`use_population`	Logical; if `TRUE`, uses the population covariance estimator $\frac{n-1}{n} \times \Sigma$; otherwise uses the sample covariance. Default is `TRUE`.
`tol`	Numeric tolerance passed to `solve` when inverting the covariance matrix. Default is `1e-25`.
`bootstrap`	Logical; if `TRUE`, compute p-value via bootstrap resampling. Default is `FALSE`.
`B`	Integer; number of bootstrap replicates used when `bootstrap = TRUE`. Default is `1000`.
`cores`	Integer; number of cores for parallel computation when `bootstrap = TRUE`. Default is 1.

Value

A data frame with one row containing the following columns: `Test` ("Henze-Wagner"), `Statistic` and `p.value`.

Examples

```
## Not run:
data <- iris[1:50, 1:4]
hw_result <- hw(data)
hw_result

## End(Not run)
```

---

hz *Henze-Zirkler Test for Multivariate Normality*

Description

Performs Henze and Zirkler's test to assess multivariate normality based on a log-normal approximation of the test statistic.

Usage

```
hz(
  data,
  use_population = TRUE,
  tol = 1e-25,
  bootstrap = FALSE,
  B = 1000,
  cores = 1
)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and variables in columns.
`use_population`	Logical; if `TRUE`, uses the population covariance estimator $\frac{n-1}{n} \times \Sigma$; otherwise uses the sample covariance. Default is `TRUE`.
`tol`	Numeric tolerance passed to `solve` when inverting the covariance matrix. Default is `1e-25`.
`bootstrap`	Logical; if `TRUE`, compute p-value via bootstrap resampling. Default is `FALSE`.
`B`	Integer; number of bootstrap replicates used when `bootstrap` = `TRUE`. Default is `1000`.
`cores`	Integer; number of cores for parallel computation when `bootstrap` = `TRUE`. Default is 1.

Value

A data frame with one row, containing the following columns: `Test`, the name of the test ("Henze-Zirkler"); `HZ`, the test statistic (numeric); and `p.value`, the p-value computed from a log-normal approximation.

Examples

```
## Not run:
data <- iris[1:50, 1:4]
hz_result <- hz(data)
hz_result

## End(Not run)
```

`impute_missing` *Impute Missing Values*

Description

Replace `NA`s in numeric variables using simple methods or `mice`-based imputation.

Usage

```
impute_missing(
  data,
  method = c("mean", "median", "mice"),
  m = 5,
  seed = 123,
  ...
)
```

Arguments

`data`	A numeric matrix or data frame.
`method`	Character; one of `"mean"`, `"median"`, or `"mice"`. Default: `"mean"`.
`m`	Integer; number of multiple imputations when `method = "mice"`. Default: 5.
`seed`	Integer; random seed for `mice` imputation. Default: 123.
`...`	Additional arguments passed to `mice::mice` when `method = "mice"`.

Value

A data frame with missing values imputed.

Examples

```
## Not run:
df <- data.frame(x = c(1, NA, 3), y = c(4, 5, NA))
impute_missing(df, method = "mice")

## End(Not run)
```

`mardia` *Mardia's Test for Multivariate Normality*

Description

Performs Mardia's skewness and kurtosis tests to assess multivariate normality in a multivariate dataset.

Usage

```
mardia(
  data,
  use_population = TRUE,
  tol = 1e-25,
  bootstrap = FALSE,
  B = 1000,
  cores = 1
)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and variables in columns.
`use_population`	Logical; if `TRUE`, uses the population covariance estimator $\frac{n-1}{n} \times \Sigma$; otherwise uses the sample covariance. Default is `TRUE`.
`tol`	Numeric tolerance passed to `solve` when inverting the covariance matrix. Default is `1e-25`.
`bootstrap`	Logical; if `TRUE`, compute p-values via a bootstrap distribution of the test statistics. Default is `FALSE`.
`B`	Integer; number of bootstrap replicates. Only used when `bootstrap = TRUE`. Default is `1000`.
`cores`	Integer; number of cores to use when `bootstrap = TRUE`. Parallelisation is done via `parallel::mclapply`. Default is 1.

Value

A data frame with two rows, one for Mardia's skewness test and one for the kurtosis test. Each row contains the name of the test (`Test`), the test statistic (`Statistic`), and the associated p-value (`p.value`).

Examples

```
## Not run:
data <- iris[1:50, 1:4]
mardia_result <- mardia(data)
mardia_result

## End(Not run)
```

`multivariate_diagnostic_plot`
Plot Multivariate Normal Diagnostics and Bivariate Kernel Density

---

Description

Generates either a Mahalanobis Q-Q plot, an interactive 3D kernel density surface plot, or a 2D kernel density contour plot for exactly two numeric variables. The function is intended for assessing multivariate normality or exploring the bivariate distribution of the input data.

Usage

```
multivariate_diagnostic_plot(
  data,
  type = c("qq", "persp", "contour"),
  tol = 1e-25,
  use_population = TRUE
)
```

Arguments

`data`	A numeric vector, matrix, or data frame. Non-numeric columns are dropped with a warning; incomplete rows are removed. The input must contain exactly two numeric variables.
`type`	Character string specifying the type of plot to generate. Must be one of `"qq"` (Mahalanobis Q-Q plot), `"persp"` (3D KDE surface), or `"contour"` (2D KDE contour). Default is `"qq"`.
`tol`	Numeric tolerance for matrix inversion passed to `solve()`. Default is `1e-25`.
`use_population`	Logical; if `TRUE`, uses the population covariance estimator $\frac{n-1}{n} \times \Sigma$; otherwise uses the sample covariance. Default is `TRUE`.

Value

If `type = "qq"`, returns a `ggplot2` object representing a Mahalanobis Q-Q plot. If `type = "persp"` or `"contour"`, returns an interactive `plotly` widget displaying the KDE surface or contour, respectively.

Examples

```
## Not run:
library(MASS)
data(iris)

# Mahalanobis Q-Q plot
multivariate_diagnostic_plot(iris[, 1:2], type = "qq")

# 3D KDE surface
multivariate_diagnostic_plot(iris[, 1:2], type = "persp")

# 2D KDE contour
multivariate_diagnostic_plot(iris[, 1:2], type = "contour")

## End(Not run)
```

`mvn` *Comprehensive Multivariate Normality and Diagnostic Function*

Description

Conduct multivariate normality tests, outlier detection, univariate normality tests, descriptive statistics, and Box-Cox or Yeo-Johnson transformation in one wrapper.

Usage

```
mvn(
  data,
  subset = NULL,
  mvn_test = "hz",
  use_population = TRUE,
  tol = 1e-25,
  alpha = 0.05,
  scale = FALSE,
  descriptives = TRUE,
  transform = "none",
  impute = "none",
  bootstrap = FALSE,
  B = 1000,
  cores = 1,
  univariate_test = "AD",
  multivariate_outlier_method = "none",
  power_family = "none",
  power_transform_type = "optimal",
  show_new_data = FALSE,
  tidy = TRUE
)
```

Arguments

`data`	A numeric matrix or data frame where each row represents an observation and each column represents a variable. All variables should be numeric; non-numeric columns will be ignored or cause an error depending on implementation.
`subset`	Optional character string indicating the name of a grouping variable within the data. When provided, analyses will be performed separately for each level of the grouping variable. This is useful for comparing multivariate normality or outlier structure across groups.
`mvn_test`	A character string specifying which multivariate normality test to use. Supported options include "mardia" (Mardia's test), "hz" (Henze-Zirkler's test), "hw" (Henze-Wagner's test), "royston" (Royston's test), "doornik_hansen" (Doornik-Hansen test), and "energy" (Energy-based test). The default is "hz", which provides good power for detecting departures from multivariate normality.
`use_population`	A logical value indicating whether to use the population version of the covariance matrix estimator. If TRUE, scales the covariance matrix by (n - 1)/n to estimate the population covariance. If FALSE, the sample covariance matrix is used instead. The default is TRUE.
`tol`	A small numeric value used as the tolerance parameter for matrix inversion via solve(). This is important when working with nearly singular covariance matrices. The default value is 1e-25, which ensures numerical stability during matrix computations.
`alpha`	A numeric value specifying the significance level used for defining outliers when the multivariate outlier detection method is set to "adj" (adjusted robust

	weights). This threshold controls the false positive rate for identifying multivariate outliers. The default is 0.05.
`scale`	A logical value. If TRUE, the input data will be standardized (zero mean and unit variance) before analysis. This is typically recommended when variables are on different scales. Default is FALSE.
`descriptives`	A logical value indicating whether to compute descriptive statistics (mean, standard deviation, skewness, and kurtosis) for each variable before conducting multivariate normality or outlier analyses. Default is TRUE.
`transform`	A character string specifying a marginal transformation to apply to each variable before analysis. Options are "none" (no transformation), "log" (natural logarithm), "sqrt" (square root), and "square" (square of the values). The default is "none".
`impute`	A character string specifying method for handling missing data. One of `"none"`, `"mean"`, `"median"`, or `"mice"`. Default: `"none"`.
`bootstrap`	Logical; if `TRUE`, p-values for Mardia, Henze-Zirkler and Royston tests are obtained via bootstrap resampling. Default is `FALSE`.
`B`	Integer; number of bootstrap replicates used when `bootstrap = TRUE` or `mvn_test = "energy"`. Default is `1000`.
`cores`	Integer; number of cores to use for bootstrap computation. Default is 1.
`univariate_test`	A character string indicating which univariate normality test to apply to individual variables when such summaries are requested. Options include "SW" (Shapiro-Wilk), "CVM" (Cramér–von Mises), "Lillie" (Lilliefors/Kolmogorov-Smirnov), "SF" (Shapiro–Francia), and "AD" (Anderson–Darling). Default is "AD".
`multivariate_outlier_method`	A character string that specifies the method used for detecting multivariate outliers. Options are "none" (no outlier detection), "quan" (robust Mahalanobis distance based on quantile cutoff), and "adj" (adjusted robust weights with a significance threshold). Default is "none".
`power_family`	A character string specifying the type of power transformation family to apply before analysis. Options include "none" (no transformation), "bcPower" (Box-Cox transformation for positive data), "bcnPower" (Box-Cox transformation that allows for negatives), and "yjPower" (Yeo-Johnson transformation for real-valued data). Default is "none".
`power_transform_type`	A character string indicating whether to use the "optimal" or "rounded" lambda value for the selected power transformation. "optimal" uses the estimated value with maximum likelihood, while "rounded" uses the closest integer value for interpretability. Default is "optimal".
`show_new_data`	A logical value. If TRUE, the cleaned data with identified outliers removed will be included in the output. This is useful for downstream analysis after excluding extreme observations. Default is FALSE.
`tidy`	A logical value. If TRUE, the output will be returned as a tidy data frame, making it easier to use with packages from the tidyverse. A "Group" column will be included when subset analysis is performed. Default is TRUE.

Details

If `mvn_test = "mardia"`, it calculates the Mardia's multivariate skewness and kurtosis coefficients as well as their corresponding statistical significance. It can also calculate corrected version of skewness coefficient for small sample size (n< 20). For multivariate normality, both p-values of skewness and kurtosis statistics should be greater than 0.05. If sample size less than 20 then p.value.small should be used as significance value of skewness instead of p.value.skew. If there are missing values in the data, a listwise deletion will be applied and a complete-case analysis will be performed.

If `mvn_test = "hz"`, it calculates the Henze-Zirkler's multivariate normality test. The Henze-Zirkler test is based on a non-negative functional distance that measures the distance between two distribution functions. If the data is multivariate normal, the test statistic HZ is approximately lognormally distributed. It proceeds to calculate the mean, variance and smoothness parameter. Then, mean and variance are lognormalized and the p-value is estimated.

If `mvn_test = "hw"`, it calculates the Henze-Wagner's multivariate normality test. The Henze-Wagner test is based on a class of weighted L2-statistics that quantify the deviation of the empirical characteristic function from that of the multivariate normal distribution. It uses a weight function involving a smoothness parameter to control the influence of differences in the tails. The test statistic is computed and its null distribution is approximated to obtain the p-value.

If `mvn_test = "royston"`, it calculates the Royston's multivariate normality test. A function to generate the Shapiro-Wilk's W statistic needed to feed the Royston's H test for multivariate normality However, if kurtosis of the data greater than 3 then Shapiro-Francia test is used for leptokurtic samples else Shapiro-Wilk test is used for platykurtic samples.

If `mvn_test = "doornik_hansen"`, it calculates the Doornik-Hansen's multivariate normality test. The code is adapted from asbio package (Aho, 2017).

If `mvn_test = "energy"`, it calculates the Energy multivariate normality test. The code is adapted from energy package (Rizzo and Szekely, 2017).

Value

A named list containing:

multivariate_normality A data frame of the selected multivariate normality (MVN) test results.

univariate_normality A data frame of univariate normality test results.

descriptives (Optional) A data frame of descriptive statistics if `descriptives = TRUE`.

multivariate_outliers (Optional) A data frame of flagged multivariate outliers if `multivariate_outlier_method != "none"`.

new_data (Optional) Original data with multivariate outliers removed if `show_new_data = TRUE`.

powerTransformLambda (Optional) Estimated power transform lambda values if `power_family = "bcPower"`.

data The processed data matrix used in the analysis (transformed and/or cleaned).

subset (Optional) The grouping variable used for subset analysis, if applicable.

Author(s)

Selcuk Korkmaz, `<selcukorkmaz@gmail.com>`

References

Korkmaz S, Goksuluk D, Zararsiz G. MVN: An R Package for Assessing Multivariate Normality. The R Journal. 2014 6(2):151-162. URL `https://journal.r-project.org/archive/2014-2/korkmaz-goksuluk-zararsiz.pdf`

Mardia, K. V. (1970), Measures of multivariate skewness and kurtosis with applications. Biometrika, 57(3):519-530.

Mardia, K. V. (1974), Applications of some measures of multivariate skewness and kurtosis for testing normality and robustness studies. Sankhy A, 36:115-128.

Henze, N. and Zirkler, B. (1990), A Class of Invariant Consistent Tests for Multivariate Normality. Commun. Statist.-Theor. Meth., 19(10): 35953618.

Henze, N. and Wagner, Th. (1997), A New Approach to the BHEP tests for multivariate normality. Journal of Multivariate Analysis, 62:1-23.

Royston, J.P. (1982). An Extension of Shapiro and Wilks W Test for Normality to Large Samples. Applied Statistics, 31(2):115124.

Royston, J.P. (1983). Some Techniques for Assessing Multivariate Normality Based on the Shapiro-Wilk W. Applied Statistics, 32(2).

Royston, J.P. (1992). Approximating the Shapiro-Wilk W-Test for non-normality. Statistics and Computing, 2:117-119.121133.

Royston, J.P. (1995). Remark AS R94: A remark on Algorithm AS 181: The W test for normality. Applied Statistics, 44:547-551.

Shapiro, S. and Wilk, M. (1965). An analysis of variance test for normality. Biometrika, 52:591611.

Doornik, J.A. and Hansen, H. (2008). An Omnibus test for univariate and multivariate normality. Oxford Bulletin of Economics and Statistics 70, 927-939.

G. J. Szekely and M. L. Rizzo (2013). Energy statistics: A class of statistics based on distances, Journal of Statistical Planning and Inference, http://dx.doi.org/10.1016/j.jspi.2013.03.018

M. L. Rizzo and G. J. Szekely (2016). Energy Distance, WIRES Computational Statistics, Wiley, Volume 8 Issue 1, 27-38. Available online Dec., 2015, http://dx.doi.org/10.1002/wics.1375.

G. J. Szekely and M. L. Rizzo (2017). The Energy of Data. The Annual Review of Statistics and Its Application 4:447-79. 10.1146/annurev-statistics-060116-054026

Examples

```
result = mvn(data = iris[-4], subset = "Species", mvn_test = "hz",
             univariate_test = "AD",
             multivariate_outlier_method = "adj",
             show_new_data = TRUE)

### Multivariate Normality Result
summary(result, select = "mvn")

### Univariate Normality Result
summary(result, select = "univariate")

### Descriptives
summary(result, select = "descriptives")
```

```
### Multivariate Outliers
summary(result, select = "outliers")

### New data without multivariate outliers
summary(result, select = "new_data")
```

`mv_outlier` *Identify Multivariate Outliers via Robust Mahalanobis Distances*

Description

Computes robust Mahalanobis distances for multivariate data using the Minimum Covariance Determinant (MCD) estimator, flags outliers based on either a chi-square quantile cutoff or an adjusted cutoff using the Atkinson–Riani–Welsh (ARW) method, and optionally generates a Mahalanobis Q–Q plot.

Usage

```
mv_outlier(
  data,
  outlier = TRUE,
  qqplot = TRUE,
  alpha = 0.05,
  method = c("quan", "adj"),
  label = TRUE,
  title = "Chi-Square Q-Q Plot"
)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and at least two numeric columns.
`outlier`	Logical; if `TRUE`, includes the Mahalanobis distance values and outlier classification in the output. If `FALSE`, suppresses this component. Default is `TRUE`.
`qqplot`	Logical; if `TRUE`, a Chi-Square Q–Q plot is generated to visualize outlier detection. Default is `TRUE`.
`alpha`	Numeric; significance level used for the adjusted cutoff method (only applies if `method = "adj"`). Default is `0.05`.
`method`	Character string specifying the outlier detection method. Must be either `"quan"` (quantile-based cutoff) or `"adj"` (adjusted cutoff via ARW). Default is `"quan"`.
`label`	Logical; if `TRUE` and `qqplot = TRUE`, labels the detected outliers in the plot. Default is `TRUE`.
`title`	Optional character string specifying the title for the Q–Q plot. Default is `"Chi-Square Q-Q Plot"`.

Value

A list containing the following components: `outlier`, a data frame of Mahalanobis distances with observation IDs and outlier flags (if `outlier = TRUE`); `qq_outlier_plot`, a ggplot object of the Mahalanobis Q–Q plot (if `qqplot = TRUE`); and `newData`, a data frame of non-outlier observations.

Examples

```
## Not run:
data <- iris[, 1:4]
res <- mv_outlier(data, method = "adj", alpha = 0.025)
res$outlier
res$qq_outlier_plot
head(res$newData)

## End(Not run)
```

`plot.mvn` *Plot Diagnostics for Multivariate Normality Analysis*

Description

Generates diagnostic plots for objects of class `mvn`, including multivariate Q-Q plots, 3D or contour kernel density plots, univariate plots (e.g., Q-Q, histograms, boxplots), and multivariate outlier detection plots. If a grouping variable (subset) was used in the `mvn` function, plots will be generated separately for each group.

Usage

```
## S3 method for class 'mvn'
plot(x, ...)
```

Arguments

`x`	An object of class `mvn`, as returned by the `mvn` function.
`...`	Additional arguments passed to internal plotting functions: `diagnostic` (`"multivariate"`, `"univariate"`, `"outlier"`), `type` (e.g., `"qq"`, `"boxplot"`, `"persp"`), `interactive` (logical; use **plotly**), and

Value

This function is called for its side effect of producing plots. It does not return a value.

Examples

```
## Not run:
data <- iris[1:4]
result <- mvn(data)

plot(result, diagnostic = "multivariate", type = "qq")
plot(result, diagnostic = "univariate", type = "boxplot")
plot(result, diagnostic = "outlier")

## End(Not run)
```

`power_transform` *Apply Power Transformation to Numeric Data*

Description

Applies a power transformation to numeric input data using the `car::powerTransform` function. Supported transformation families include Box-Cox (`"bcPower"`), Box-Cox with negative values (`"bcnPower"`), and Yeo-Johnson (`"yjPower"`). The function estimates either optimal or rounded lambda values for each numeric variable and transforms the data accordingly.

Usage

```
power_transform(
  data,
  family = c("bcPower", "bcnPower", "yjPower"),
  type = c("optimal", "rounded")
)
```

Arguments

`data`	A numeric vector, matrix, or data frame. Only numeric columns will be transformed. Non-numeric columns are dropped with a warning.
`family`	A character string specifying the transformation family. Must be one of `"bcPower"`, `"bcnPower"`, or `"yjPower"`.
`type`	A character string specifying whether to use the estimated optimal lambda values (`"optimal"`) or the rounded values (`"rounded"`).

Details

Rows with missing values are removed prior to estimating lambda parameters. A warning is issued if any non-numeric columns are dropped or if any rows are excluded due to missingness. The same estimated lambda values are then applied to the original data (excluding dropped rows or columns).

Value

A list containing two elements. The first is a data frame of transformed numeric columns. The second is a named numeric vector of the lambda values used for the transformation.

Examples

```
if (requireNamespace("car", quietly = TRUE)) {
  x <- rnorm(100, mean = 10, sd = 2)
  y <- rexp(100, rate = 0.2)
  df <- data.frame(x = x, y = y)
  result <- power_transform(df, family = "bcPower", type = "optimal")
  head(result$data)
  result$lambda
}
```

`royston` *Royston's Multivariate Normality Test*

Description

Performs Royston's test for multivariate normality by combining univariate W-statistics (Shapiro–Wilk or Shapiro–Francia) across variables and adjusting for the correlation structure.

Usage

```
royston(data, tol = 1e-25, bootstrap = FALSE, B = 1000, cores = 1)
```

Arguments

`data`	A numeric matrix or data frame with observations in rows and variables in columns.
`tol`	Numeric tolerance passed to `solve` when inverting the covariance matrix. Default is `1e-25`.
`bootstrap`	Logical; if `TRUE`, compute p-value via bootstrap resampling. Default is `FALSE`.
`B`	Integer; number of bootstrap replicates used when `bootstrap` = `TRUE`. Default is `1000`.
`cores`	Integer; number of cores for parallel computation when `bootstrap` = `TRUE`. Default is 1.

Value

A data frame with one row containing the test name (`Test`), the Royston test statistic (`Statistic`), and the associated p-value (`p.value`) from a chi-square approximation.

Examples

```
## Not run:
data <- iris[1:50, 1:4]
royston_result <- royston(data)
royston_result

## End(Not run)
```

`summary.mvn` *Summarize Multivariate Normality Analysis Results*

Description

Provides a structured summary of the results from an object of class `mvn`, including multivariate and univariate normality tests, descriptive statistics, and multivariate outlier detection (if applicable).

Usage

```
## S3 method for class 'mvn'
summary(
  object,
  select = c("mvn", "univariate", "descriptives", "outliers", "new_data"),
  ...
)
```

Arguments

`object`	An object of class `mvn`, as returned by the `mvn` function.
`select`	A character vector specifying which components to display. Must be one or more of `"mvn"`, `"univariate"`, `"descriptives"`, `"outliers"`, or `"new_data"`. Defaults to showing all available sections.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object.

Examples

```
## Not run:
data <- iris[1:4]
result <- mvn(data)

summary(result)  # Show all sections
summary(result, select = c("mvn", "outliers"))  # Show selected sections only

## End(Not run)
```

`test_univariate_normality` *Univariate Normality Tests*

Description

Performs one of several common univariate normality tests on each numeric variable in a vector, matrix, or data frame.

Usage

```
test_univariate_normality(data, test = c("SW", "CVM", "Lillie", "SF", "AD"))
```

Arguments

`data`	A numeric vector, matrix, or data frame with observations in rows and variables in columns. Non-numeric columns are dropped with a warning. Each column is tested individually.
`test`	A character string specifying the normality test to use. Choices are: `"SW"` (Shapiro–Wilk), `"SF"` (Shapiro–Francia), `"AD"` (Anderson–Darling), `"CVM"` (Cramér–von Mises), and `"Lillie"` (Lilliefors test). Default is the first match from this list.

Value

A data frame with one row per variable and the following columns: `Test`, the name of the test used; `Variable`, the name of the tested variable; `Statistic`, the test statistic; and `p.value`, the associated p-value.

Examples

```
## Not run:
data(iris)
test_univariate_normality(iris[, 1:4], test = "AD")

## End(Not run)
```

`univariate_diagnostic_plot` *Diagnostic Plots for Univariate and Multivariate Data*

Description

Generates QQ plots, histograms with density overlays, boxplots, or scatterplot matrices for numeric data (vector, matrix, or data frame).

Usage

```
univariate_diagnostic_plot(
  data,
  type = c("qq", "histogram", "boxplot", "scatter"),
  title = NULL,
  interactive = FALSE
)
```

Arguments

`data`	A numeric vector, matrix, or data frame with observations in rows and variables in columns.
`type`	Character; type of plot. One of: "qq", "histogram", "boxplot", "scatter". Default selects the first.
`title`	Character; plot title.
`interactive`	Logical; if TRUE, renders the plot interactively using plotly.

Examples

```
## Not run:
data <- iris[1:50, 1:3]
univariate_diagnostic_plot(data, type = "histogram")
univariate_diagnostic_plot(data, type = "qq")
univariate_diagnostic_plot(data, type = "boxplot")
univariate_diagnostic_plot(data, type = "scatter", interactive = TRUE)

## End(Not run)
```

Index