Package 'chiplot'

November 6, 2026

Type Package

Title Chi-Plots for Assessing Bivariate and Multivariate Dependence

Version 0.1.0

Description Graphical methods for assessing dependence. The chi-plot of Fisher and Switzer (1985) <doi:10.1093/biomet/72.2.253> and (2001) <doi:10.1198/000313001317098248> supplements a scatterplot of two variables, producing characteristic patterns for different forms of association. Its multivariate extension (Fisher and Lee, 2026) uses generalized orthants and data depth to explore whether a multivariate distribution factorizes into independent blocks of variables, with permutation tests and guidelines. Functions are provided for single chi-plots, a scatterplot matrix with chi-plots above the diagonal, multivariate chi-plots for a given partition of the variables, and a search over all partitions. Computation-intensive parts are written in C.

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 4.0.0)

Imports graphics, grDevices, stats

Suggests fda.usc

LazyData true

NeedsCompilation yes

Author Nicholas I. Fisher [aut, cre, cph],
Alan J. Lee [aut],
ValueMetrics Australia [cph] (copyright holder of the geochem data)

Maintainer Nicholas I. Fisher <nif@valuemetrics.com.au>

Repository CRAN

Date/Publication 2026-11-06 17:10:11 UTC

Contents

chiplot-package 2

chidepth 3
chimult 4
chipairs 7
chipart 8
chiplot 10
geochem 13

Index **14**

`chiplot-package` *Chi-Plots for Assessing Bivariate and Multivariate Dependence*

Description

The chi-plot (Fisher and Switzer, 1985, 2001) is a graphical aid to interpreting a scatterplot of two variables: under independence the points form a horizontal band, and different forms of association produce characteristically different patterns. Fisher and Lee (2026) extend the idea to multivariate data, to explore whether a multivariate distribution factorizes into independent blocks of variables.

Details

The main functions are:

`chiplot` a chi-plot for two variables;

`chipairs` a scatterplot matrix with the corresponding chi-plots above the diagonal;

`chimult` chi- and lambda-values, permutation tests and the multivariate chi-plot for a given partition of the variables into blocks;

`chipart` a search over all partitions of the variables, keeping those that show a significant departure from independence;

`chidepth` the data depth functions used for the lambda-values of multivariate chi-plots.

The calculations of chi-values are carried out in compiled C code. All other code uses base R only.

Author(s)

Nicholas I. Fisher and Alan J. Lee. Maintainer: Nicholas I. Fisher <nif@valuemetrics.com.au>

References

Fisher, N. I. and Switzer, P. (1985). Chi-plots for assessing dependence. *Biometrika*, **72**, 253–265. doi:10.1093/biomet/72.2.253

Fisher, N. I. and Switzer, P. (2001). Graphical assessment of dependence: is a picture worth a hundred tests? *The American Statistician*, **55**, 233–239. doi:10.1198/000313001317098248

Fisher, N. I. and Lee, A. J. (2026). Chi-plots for assessing multivariate dependence. Submitted to *Journal of Computational and Graphical Statistics*.

chidepth *Data Depth for Multivariate Chi-Plots*

Description

Calculates the data depth of each row of a data matrix with respect to the whole sample. These depths are the basis of the lambda-values in `chimult`.

Usage

```
chidepth(X, depth = c("RP", "TD", "PD", "MhD"), Nsim = 100)
```

Arguments

`X`	a numeric matrix without missing values.
`depth`	the depth function: one of the names below, or a function.
`Nsim`	the number of random directions for `"RP"` and `"PD"`.

Details

All the depth functions are implemented in base R:

`"RP"` random projection depth: the average, over `Nsim` random directions, of the univariate halfspace depth of the projected data. The directions are generated as in `fda.usc::mdepth.RP`, so the results are identical for the same random number seed. This is the default.

`"TD"` the average of the univariate halfspace (Tukey) depths of the coordinates, as `fda.usc::mdepth.TD`. Note that this is not the exact multivariate Tukey depth.

`"PD"` projection depth (Zuo and Serfling, 2000), $1/(1+O(x))$ with outlyingness $O(x) = \max_u |u^T x - \mathrm{med}(u^T X)|/\mathrm{MAD}(u^T X)$, the maximum being taken over `Nsim` random directions.

`"MhD"` Mahalanobis depth, $1/(1+d^2)$ with d the Mahalanobis distance from the mean, as `fda.usc::mdepth.MhD`. This is very fast but parametric.

The names `"mdepth.RP"`, `"mdepth.TD"`, `"mdepth.MhD"` and `"Projection"` are also accepted.

A user-specified `depth` must be a function of the data matrix that returns one depth per row, either as a numeric vector or as a list with component `dep`. For example, `function(X) fda.usc::mdepth.RP(X, proj = 500)$dep`, or a function calling `depth()` in package **DepthProc**.

Value

A numeric vector of depths, one for each row of `X`.

References

Zuo, Y. and Serfling, R. (2000). General notions of statistical depth function. *The Annals of Statistics*, **28**, 461–482.

See Also

chimult

Examples

```
set.seed(1)
X <- matrix(rnorm(200), 100, 2)
d <- cbind(RP = chidepth(X, "RP"), TD = chidepth(X, "TD"),
           PD = chidepth(X, "PD"), MhD = chidepth(X, "MhD"))
round(cor(d, method = "spearman"), 2)

## a user-supplied depth: inverse distance from the coordinatewise median
chidepth(X[1:5, ], function(X) 1 / (1 + sqrt(rowSums(t(t(X) - apply(X, 2, median))^2))))
```

---

chimult *Multivariate Chi-Plot for a Partition of the Variables*

Description

For a multivariate sample and a partition of the variables into blocks, `chimult` calculates the chi- and lambda-values of Fisher and Lee (2026), permutation tests of independence of the blocks, and the material for the multivariate chi-plot, which is drawn by `plot`.

Usage

```
chimult(Data, D, nperm = 100, Nsim = 100, depth = "RP", jitter = NULL,
        jitter.amount = 0.25, plot = FALSE, verbose = TRUE, ...)

## S3 method for class 'chimult'
plot(x, pc.level = 0.95, main = "", col1 = "red",
     col2 = "blue", pt = 12, box = TRUE, cex = NULL, mar = c(4, 4, 2, 2),
     guide.option = 1, del = 0.2, ...)

## S3 method for class 'chimult'
print(x, ...)
```

Arguments

`Data`	a numeric matrix or data frame ($n \times p$), each row an observation of a p-variate random variable. Ordered factors are allowed.
`D`	the block sizes, in column order: for example, if $p = 5$, `D = c(2, 3)` compares the first two columns with the last three. At least two blocks are needed. Reorder the columns of `Data` first for other partitions.
`nperm`	the number of random permutations used for the p-values and null distributions.
`Nsim`	the number of random directions for the `"RP"` and `"PD"` data depths.
`depth`	the data depth used for the lambda-values: `"RP"` (random projection depth, the default), `"TD"`, `"PD"`, `"MhD"`, or a function. See `chidepth`.

`jitter`, `jitter.amount`
: jittering to break ties, for example for ordered categorical variables: `TRUE`, `FALSE`, a logical vector, or column numbers or names; see `chiplot`. The default `NULL` jitters ordered factors only.

`plot`
: logical: draw the chi-plot?

`verbose`
: logical: report removed observations and jittered variables?

`...`
: for `chimult`, arguments passed to `plot.chimult`; for `plot.chimult`, graphical parameters passed to `plot`.

`x`
: an object of class `"chimult"`.

`pc.level`
: probability level for the guidelines.

`main`
: title of the chi-plot.

`col1`, `col2`
: colours of the horizontal guidelines and of the vertical reference lines.

`pt`
: point size of the text listing the groups.

`box`
: logical: draw a box around the list of groups?

`cex`
: size of the points; by default chosen from the sample size.

`mar`
: plot margins, as in `par`.

`guide.option`
: the horizontal guidelines: `1`, at the normal percentiles for `pc.level`; `2`, at the percentiles of the permutation distribution of the chi-values; `3`, both; `4`, option 1 if the two sets are within `del` of each other, otherwise option 2.

`del`
: tolerance used by `guide.option = 4`.

Details

Observations with missing values are removed. Each column is centred at its median and divided by its interquartile range.

For each observation u_i, the generalized orthant is the set of v with $v_j \le u_{ij}$ if $u_{ij} > 0$ and $v_j > u_{ij}$ otherwise. With H_i the proportion of the sample in this orthant, and $H_{1i}, \ldots, H_{ki}$ the proportions in the corresponding orthants for the variables of each block, the chi-value is

$$\chi_i = \sqrt{n}\,(H_i - P_i)/\{P_i(1 - P_i S_i)\}^{1/2},$$

where $P_i = \prod_l H_{li}$ and $S_i = 1 + \sum_l (1 - H_{li})/H_{li}$. Under independence of the blocks the chi-values are approximately standard normal. The lambda-value of each observation is its data depth, transformed by the empirical distribution function of the depths.

The null distributions are estimated by permuting the rows of blocks $1, \ldots, k-1$, keeping the rows within each block together. `sigP.chi` is the proportion of permutations whose mean squared chi-value is at least that of the data; `sigP.lam` is the p-value of a two-sample Kolmogorov–Smirnov test comparing the data depths with those of the permuted data.

The plot shows the variables in each group, the chi-plot, and, if `sigP.lam <= 1 - pc.level`, a Q-Q plot of the data depths against their permutation distribution.

With the same random number seed, `chimult` gives the same results as the authors' original `chimult.lam` function with `fda.usc::mdepth.RP`.

Value

An object of class `"chimult"`, a list with components

`chi`	the chi-values of the data.
`chisq.actual`	the mean of the squared chi-values.
`chi.values.perm`	the chi-values of the permuted data sets.
`chisq.values.perm`	the mean squared chi-values of the permuted data sets.
`sigP.chi`	the permutation p-value of the mean squared chi-value.
`lambda.actual`	the data depths of the observations.
`lambda`	the lambda-values: the data depths transformed by their empirical distribution function.
`lambda.values.perm`	the data depths of the permuted data sets.
`sigP.lam`	the p-value of the Kolmogorov–Smirnov test for the depths.
`X`	the centred and scaled data.
`D`, `k`	the block sizes and the number of blocks.
`depth`, `nperm`, `Nsim`	the settings used.
`n.removed`, `rows`	the number of observations removed because of missing values, and the rows of `Data` used.
`jittered`	the names of jittered variables.
`call`	the call.

References

Fisher, N. I. and Lee, A. J. (2026). Chi-plots for assessing multivariate dependence. Submitted to *Journal of Computational and Graphical Statistics*.

See Also

`chipart` to search over partitions; `chidepth`; `chiplot`.

Examples

```
## Two independent blocks of two correlated normal variables
set.seed(161225)
n <- 200
z1 <- matrix(rnorm(2 * n), n) %*% chol(matrix(c(1, 0.5, 0.5, 1), 2))
z2 <- matrix(rnorm(2 * n), n) %*% chol(matrix(c(1, 0.7, 0.7, 1), 2))
Data <- data.frame(z1, z2)
colnames(Data) <- paste0("Var.", 1:4)
res <- chimult(Data, D = c(2, 2), nperm = 100)
res
plot(res)
```

```
## Dependence between the blocks
Data$Var.3 <- Data$Var.3 + 0.8 * Data$Var.1
res2 <- chimult(Data, D = c(2, 2), nperm = 100)
plot(res2, guide.option = 3)


## Five variables, mutually uncorrelated but with a hole in the data
set.seed(280926)
z <- matrix(rnorm(20000 * 5), ncol = 5)
z <- z[sqrt(rowSums((z - 1)^2)) >= 2.5, ][1:200, ]
res3 <- chimult(z, D = rep(1, 5), nperm = 500)
plot(res3, guide.option = 4, pt = 10)
```

`chipairs` *Scatterplot Matrix with Chi-Plots*

Description

A scatterplot matrix, like `pairs`, with the scatterplots below the diagonal and the corresponding chi-plots above the diagonal.

Usage

```
chipairs(Data, labels = NULL, ctrl.lines = 2, jitter = NULL,
         jitter.amount = 0.25, cex.points = NULL, pch.type = 19,
         dot.col = "black", cex.label = NULL, cex.axis = 0.7,
         plgap = 0.2, axes = TRUE, main = NULL, ...)
```

Arguments

`Data`	a numeric matrix or data frame ($n \times p$), each column a variable. Ordered factors are allowed.
`labels`	variable names for the diagonal; by default the column names.
`ctrl.lines`	control lines on the chi-plots, as in `chiplot`.
`jitter`, `jitter.amount`	jittering of variables to break ties, as in `chiplot`; by default ordered factors are jittered.
`cex.points`	size of the points. By default it is chosen from the number of points and the number of panels.
`pch.type`, `dot.col`	plotting symbol and colour of the points.
`cex.label`	size of the variable names on the diagonal.
`cex.axis`	size of the axis annotation.
`plgap`	space between panels, in lines of text.

axes	logical: draw axes on the outer panels? The scatterplots are annotated on the bottom and left, the chi-plots (on the scale -1 to 1) on the top and right.
main	an overall title.
...	further graphical parameters passed to plot.

Details

As in pairs, the panel in row i and column j has variable j on the horizontal axis and variable i on the vertical axis. Each pair of variables uses all the observations for which both are present. The chi-plot is symmetric in the two variables.

Value

NULL, invisibly.

See Also

chiplot, pairs

Examples

```
chipairs(geochem[, 1:5], main = "geochem")

## a smaller sample, with the points emphasised
chipairs(geochem[1:300, c("Cr2O3", "MgO", "CaO", "Ni")], cex.points = 0.5)
```

---

chipart *Search All Partitions of the Variables with Multivariate Chi-Plots*

Description

chipart enumerates the partitions of the p variables into two or more blocks, runs chimult for each, and keeps the results for partitions where either test statistic has p-value less than SigProb. chi.partitions lists the partitions examined.

Usage

```
chipart(Data, kfix = 0, SigProb = 0.05, nperm = 100, Nsim = 100,
        depth = "RP", nblocks = NULL, nperm.screen = NULL,
        screen.level = 0.2, jitter = NULL, jitter.amount = 0.25,
        verbose = TRUE)

chi.partitions(p, kfix = 0, nblocks = NULL)

## S3 method for class 'chipart'
plot(x, which = NULL, ask = NULL, ...)

## S3 method for class 'chipart'
print(x, ...)
```

Arguments

`Data`	a numeric matrix or data frame ($n \times p$).
`kfix`	if positive, the first `kfix` variables are always kept together as one block, and only the remaining variables are partitioned. This reduces the number of partitions from $B_p - 1$ to B_{p-kfix}, where B_m is the mth Bell number.
`SigProb`	the significance level used to select interesting partitions.
`nperm`, `Nsim`, `depth`	as in `chimult`.
`nblocks`	if given, only partitions with these numbers of blocks are examined; for example, `nblocks = 2` compares each subset of the variables with its complement.
`nperm.screen`	if given, a smaller number of permutations used to screen all partitions; partitions with either p-value at most `screen.level` are then re-run with `nperm` permutations.
`screen.level`	the screening threshold used with `nperm.screen`.
`jitter`, `jitter.amount`	jittering to break ties, applied once to the data before the search, as in `chimult`.
`verbose`	logical: report progress?
`p`	the number of variables.
`x`	an object of class `"chipart"`.
`which`	partition numbers to plot; by default all interesting partitions.
`ask`	logical: pause before each plot? By default, on interactive devices when there is more than one plot.
`...`	arguments passed to `plot.chimult`.

Details

The number of partitions grows quickly with p: the Bell numbers are 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, ... The options `kfix`, `nblocks`, `nperm.screen` and a fast depth such as `depth = "MhD"` reduce the computing time.

For each partition the variables are reordered so that the blocks are contiguous, keeping the original order within each block, and `chimult` is called. A partition is "interesting" if `sigP.chi` or `sigP.lam` is less than `SigProb`. No allowance is made for multiple testing.

Value

`chipart` returns an object of class `"chipart"`, a list with components

`summary`	a data frame with one row per partition: the partition number, the groups, the number of blocks, `sigP.chi`, `sigP.lam`, the number of permutations used, and whether the partition is interesting.
`results`	a list of `"chimult"` objects for the interesting partitions, named by partition number.
`partitions`	the partitions: a matrix with one column per partition giving the block number of each variable.

`variables`, `SigProb`, `kfix`, `nperm`, `nperm.screen`, `Nsim`, `depth`, `n`, `n.removed`, `call`

details of the search.

`chi.partitions` returns the matrix of partitions.

References

Fisher, N. I. and Lee, A. J. (2026). Chi-plots for assessing multivariate dependence. Submitted to *Journal of Computational and Graphical Statistics*.

See Also

`chimult`

Examples

```
## the partitions examined for 4 variables, and with the first kept fixed
chi.partitions(4)
chi.partitions(4, kfix = 1)

## four variables, of which the second and fourth are dependent
set.seed(5)
W <- matrix(rnorm(100 * 4), 100, 4)
W[, 4] <- W[, 4] + W[, 2]
colnames(W) <- paste0("V", 1:4)
res <- chipart(W, nperm = 20, verbose = FALSE)
res
plot(res, which = 3)


## screening with fewer permutations
res2 <- chipart(W, nperm = 200, nperm.screen = 20)
res2$summary
```

`chiplot` *Chi-Plot for Assessing Dependence of Two Variables*

Description

`chivals` calculates the chi- and lambda-values of Fisher and Switzer (1985) for a bivariate sample; `chiplot` calculates them and draws the chi-plot.

Usage

```
chiplot(x, y = NULL, ctrl.lines = 2, jitter = NULL, jitter.amount = 0.25,
        cex.points = NULL, pch.type = 19, dot.col = "black", main = "",
        cex.axis = 1, cex.lam = 1.2, plot = TRUE, ...)
```

```
chivals(x, y = NULL, jitter = NULL, jitter.amount = 0.25)

## S3 method for class 'chiplot'
plot(x, ctrl.lines = 2, cex.points = NULL, pch.type = 19,
     dot.col = "black", main = "", cex.axis = 1, cex.lam = 1.2,
     xlab = expression(lambda), ylab = expression(chi),
     col.ctrl = "red", lty.ctrl = 1, ...)

## S3 method for class 'chiplot'
print(x, ...)
```

Arguments

`x`	a numeric vector or ordered factor; or, if `y` is `NULL`, a matrix or data frame whose first two columns are used. For the methods, an object of class `"chiplot"`.
`y`	a numeric vector or ordered factor of the same length as `x`.
`ctrl.lines`	control lines enclosing, on average, the given percentage of points under independence: `0` (none), `1` (90%), `2` (95%) or `3` (99%). The values `0.90`, `0.95` and `0.99` are also accepted.
`jitter`	which variables to jitter, to break ties (for example, for ordered categorical variables): `TRUE` (both), `FALSE` (neither), a logical vector, or column numbers or names. The default `NULL` jitters ordered factors only.
`jitter.amount`	size of the jitter, as a fraction (less than 0.5) of the smallest gap between distinct values, so that the order of distinct values is preserved.
`cex.points`	size of the plotted points. The default `NULL` chooses a size, as large as possible without crowding, from the number of points.
`pch.type`, `dot.col`	plotting symbol and colour of the points.
`main`	title of the plot.
`cex.axis`	size of the axis annotation.
`cex.lam`	size of the axis labels λ and χ.
`plot`	logical: if `FALSE` the values are calculated but not plotted.
`xlab`, `ylab`	axis labels.
`col.ctrl`, `lty.ctrl`	colour and line type of the control lines.
`...`	further graphical parameters passed to `plot`.

Details

Observations with a missing value in either variable are removed. For each point (x_i, y_i), let F_i, G_i and H_i be the proportions of the other $n-1$ points with $x_j \le x_i$, $y_j \le y_i$, and both, respectively. Then

$$\chi_i = (H_i - F_i G_i)/\{F_i(1-F_i)G_i(1-G_i)\}^{1/2},$$

$$\lambda_i = 4\,\mathrm{sign}(\tilde{F}_i\tilde{G}_i)\ \max(\tilde{F}_i^2, \tilde{G}_i^2),$$

where $\tilde{F}_i = F_i - 1/2$ and $\tilde{G}_i = G_i - 1/2$. Points with $|\lambda_i| \geq 4\{1/(n-1) - 1/2\}^2$, at the edges of the data, are not plotted. Under independence the points lie in a horizontal band about $\chi = 0$, approximately the stated percentage of them between the control lines at $\pm c_p/\sqrt{n}$, with c_p = 1.54, 1.78 and 2.18 for 90%, 95% and 99%.

As in the original program, the sine transformation of the chi-values mentioned by Fisher and Switzer (1985) is not used. The chi- and lambda-values depend only on the ranks of the data.

Value

An object of class `"chiplot"` (returned invisibly by `chiplot`): a list with components

`lambda, chi`	the lambda- and chi-values of the plotted points;
`index`	the row numbers, in the original data, of the plotted points;
`n`	the number of complete pairs;
`npts`	the number of points plotted;
`n.missing`	the number of pairs removed because of missing values;
`names`	the names of the two variables.

References

Fisher, N. I. and Switzer, P. (1985). Chi-plots for assessing dependence. *Biometrika*, **72**, 253–265. doi:10.1093/biomet/72.2.253

Fisher, N. I. and Switzer, P. (2001). Graphical assessment of dependence: is a picture worth a hundred tests? *The American Statistician*, **55**, 233–239. doi:10.1198/000313001317098248

See Also

`chipairs` for all pairs of variables; `chimult` for blocks of variables.

Examples

```
set.seed(1)
op <- par(mfrow = c(2, 2))
## Independent normal variables
x <- rnorm(100); y <- rnorm(100)
chiplot(x, y, main = "Independence")
## Bivariate normal, correlation 0.7
y <- 0.7 * x + sqrt(1 - 0.7^2) * rnorm(100)
chiplot(x, y, main = "Correlation 0.7")
## Non-monotone association: points near a half circle
x <- rnorm(2000); y <- rnorm(2000)
keep <- y >= 0 & abs(x^2 + y^2 - 1) <= 0.5
chiplot(x[keep][1:100], y[keep][1:100], main = "Half ring")
## Ordered categorical variable: jittered automatically
u <- rnorm(100)
grade <- cut(u + rnorm(100), c(-Inf, -1, 0, 1, Inf), ordered_result = TRUE)
res <- chiplot(u, grade, main = "Ordered categories")
par(op)
res
```

geochem *Geochemical Measurements*

Description

Measurements of 11 geochemical constituents on 1000 samples, used to illustrate chipairs. The values are the ranks of the measurements within each variable, with tied values given their average rank.

Usage

```
geochem
```

Format

A data frame with 1000 rows and 11 numeric columns: TiO2, Cr2O3, FeO, MnO, MgO, CaO, Ni, Zn, Ga, Y and Zr.

Source

ValueMetrics Australia. Included with the permission of the copyright holder.

Examples

```
summary(geochem[, 1:4])
chipairs(geochem[, c("Cr2O3", "MgO", "Ni")])
```

Index

∗ **datasets**
  geochem, 13
∗ **hplot**
  chimult, 4
  chipairs, 7
  chipart, 8
  chiplot, 10
  chiplot-package, 2
∗ **multivariate**
  chidepth, 3
  chimult, 4
  chipart, 8
∗ **nonparametric**
  chimult, 4
  chiplot, 10
∗ **package**
  chiplot-package, 2

chi.partitions *(*chipart*)*, 8
chidepth, *2*, 3, *4*, *6*
chimult, *2–4*, 4, *8–10*, *12*
chipairs, *2*, 7, *12, 13*
chipart, *2*, *6*, 8
chiplot, *2*, *5–8*, 10
chiplot-package, 2
chivals *(*chiplot*)*, 10

geochem, 13

pairs, *7, 8*
par, *5*
plot, *5*, *8*, *11*
plot.chimult, *9*
plot.chimult *(*chimult*)*, 4
plot.chipart *(*chipart*)*, 8
plot.chiplot *(*chiplot*)*, 10
print.chimult *(*chimult*)*, 4
print.chipart *(*chipart*)*, 8
print.chiplot *(*chiplot*)*, 10