Package: VisCollin (via r-universe)

August 27, 2024

Title Visualizing Collinearity Diagnostics

Version 0.1.2

Date 2023-09-05

Description Provides methods to calculate diagnostics for multicollinearity among predictors in a linear or generalized linear model. It also provides methods to visualize those diagnostics following Friendly & Kwan (2009), ``Where's Waldo: Visualizing Collinearity Diagnostics'', <doi:10.1198/tast.2009.0012>. These include better tabular presentation of collinearity diagnostics that highlight the important numbers, a semi-graphic tableplot of the diagnostics to make warning and danger levels more salient, and a ``collinearity biplot'' of the smallest dimensions of predictor space, where collinearity is most apparent.

Depends R (>= 3.5.0)

Suggests car, corrgram, corrplot, dplyr, lmtest, knitr, tidyr

License GPL (>=3)

Language en-US

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL https://github.com/friendly/VisCollin,
https://friendly.github.io/VisCollin/

BugReports https://github.com/friendly/VisCollin/issues

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

RemoteUrl https://github.com/friendly/VisCollin

RemoteRef HEAD

RemoteSha b7965975b0d11872b64a4a671e772fee72dfc928

Contents

`biomass` *Biomass Production in the Cape Fear Estuary*

Description

Data collected by Rick Linthurst (1979) at North Carolina State University for the purpose of identifying the important soil characteristics influencing aerial biomass production of the marsh grass *Spartina alterniflora* in the Cape Fear Estuary of North Carolina. Three types of Spartina vegetation areas (devegetated “dead” areas, “short” Spartina areas, and “tall” Spartina areas) were sampled in each of three locations (Oak Island, Smith Island, and Snows Marsh)

Samples of the soil substrate from 5 random sites within each location–vegetation type (giving 45 total samples) were analyzed for 14 soil physico-chemical characteristics each month for several months.

Format

A data frame with 45 observations on the following 17 variables.

`loc` location, a factor with levels `OI` `SI` `SM`

`type` area type, a factor with levels `DVEG` `SHRT` `TALL`

`biomass` aerial biomass in gm^{-2}, a numeric vector

`H2S` hydrogen sulfide ppm, a numeric vector

`sal` percent salinity, a numeric vector

`Eh7` ester-hydrolase, a numeric vector

`pH` acidity as measured in water, a numeric vector

`buf` a numeric vector

`P` phosphorus ppm, a numeric vector

`K` potassium ppm, a numeric vector

`Ca` calcium ppm, a numeric vector

`Mg` magnesium ppm, a numeric vector

`Na` sodium ppm, a numeric vector

`Mn` manganese ppm, a numeric vector

`Zn` zinc ppm, a numeric vector

`Cu` copper ppm, a numeric vector

`NH4` ammonium ion ppm, a numeric vector

Source

Rawlings, J. O., Pantula, S. G., & Dickey, D. A. (2001). *Applied Regression Analysis: A Research Tool*, 2nd Ed., Springer New York. Table 5.1.

References

R. A. Linthurst. Aeration, nitrogen, pH and salinity as factors affecting *Spartina Alterniflora* growth and dieback. PhD thesis, North Carolina State University, 1979.

Examples

```
data(biomass)
str(biomass)
biomass.mod <- lm (biomass ~ H2S + sal + Eh7 + pH + buf + P + K + Ca + Mg + Na +
                           Mn + Zn + Cu + NH4,
                 data=biomass)
car::vif(biomass.mod)

(cd <- colldiag(biomass.mod, add.intercept=FALSE, center=TRUE))
# simplified display
print(cd, fuzz=.3)
```

`cars` *Cars Data*

Description

Data from the 1983 ASA Data Exposition, held in conjunction with the Annual Meetings in Toronto, August 15-18, 1983, `https://community.amstat.org/jointscsg-section/dataexpo/dataexpobefore1993` The data set was collected by Ernesto Ramos and David Donoho on characteristics of automobiles.

Format

A data frame with 406 observations on the following 10 variables:

`make` make of car, a factor with levels `amc` `audi` `bmw` `buick` `cadillac` `chev` `chrysler` `citroen` `datsun` `dodge` `fiat` `ford` `hi` `honda` `mazda` `mercedes` `mercury` `nissan` `oldsmobile` `opel` `peugeot` `plymouth` `pontiac` `renault` `saab` `subaru` `toyota` `triumph` `volvo` `vw`

`model` model of car, a character vector

`mpg` miles per gallon, a numeric vector

`cylinder` number of cylinders, a numeric vector `engine` engine displacement (cu. inches), a numeric vector

`horse` horsepower, a numeric vector

`weight` vehicle weight (lbs.), a numeric vector

`accel` time to accelerate from O to 60 mph (sec.), a numeric vector

`year` model year (modulo 100), a numeric vector ranging from 70 – 82

`origin` region of origin, a factor with levels `Amer Eur Japan`

Source

The data was provided for the ASA Data Exposition in a "shar" file, http://lib.stat.cmu.edu/datasets/cars.data. It is a version of that used by Donoho and Ramos (1982) to illustrate PRIM-H.

References

Donoho, David and Ramos, Ernesto (1982), “PRIMDATA: Data Sets for Use With PRIM-H” (Draft).

Examples

```
data(cars)
cars.mod <- lm (mpg ~ cylinder + engine + horse + weight + accel + year,
                data=cars)
car::vif(cars.mod)

(cd <- colldiag(cars.mod, center=TRUE))

# simplified display
print(cd, fuzz=.3)
```

`cellgram` *Draw one cell in a tableplot*

Description

Draws a graphic representing one or more values for one cell in a tableplot, using shapes whose size is proportional to the cell values and other visual attributes (outline color, fill color, outline line type, ...). Several values can be shown in a cell, using different proportional shapes.

Usage

```
cellgram(
  cell,
  shape = 0,
  shape.col = "black",
  shape.lty = 1,
  cell.fill = "white",
  back.fill = "white",
  label = 0,
  label.size = 0.7,
  ref.col = "grey80",
  ref.grid = FALSE,
  scale.max = 1,
  shape.name = ""
)
```

Arguments

`cell` Numeric value(s) to be depicted in the table cell

`shape` Integer(s) or character string(s) specifying the shape(s) used to encode the numerical value of `cell`. Any of `0="circle"`, `1="diamond"`, `2="square"`. Recycled to match the number of values in the cell.

`shape.col` Outline color(s) for the shape(s). Recycled to match the number of values in the cell.

`shape.lty` Outline line type(s) for the shape(s). Recycled to match the number of values in the cell.

`cell.fill` Inside color of |smallest| shape in a cell

`back.fill` Background color of cell

`label` Number of cell values to be printed in the corners of the cell; max is 4

`label.size` Character size of cell label(s)

`ref.col` color of reference lines

`ref.grid` whether to draw ref lines in the cells or not

`scale.max` scale values to this maximum

`shape.name` character string to uniquely identify shapes to help fill in smallest one

Value

None. Used for its graphic side effect

Examples

```
# None
```

colldiag *Collinearity Diagnostics*

Description

Calculates condition indexes and variance decomposition proportions in order to test for collinearity among the independent variables of a regression model and identifies the sources of collinearity if present.

Usage

```
colldiag(mod, scale = TRUE, center = FALSE, add.intercept = FALSE)

## S3 method for class 'colldiag'
print(x, dec.places = 3, fuzz = NULL, fuzzchar = ".", ...)
```

Arguments

`mod`	A model object, such as computed by `lm` or `glm`, or a data-frame to be used as predictors in such a model.
`scale`	If `FALSE`, the data are left unscaled. If `TRUE`, the data are scaled, typically to mean 0 and variance 1 using `scale`. Default is `TRUE`.
`center`	If TRUE, data are centered. Default is `FALSE`.
`add.intercept`	if `TRUE`, an intercept is added. Default is `FALSE`.
`x`	A `colldiag` object
`dec.places`	Number of decimal places to use when printing
`fuzz`	Variance decomposition proportions less than *fuzz* are printed as *fuzzchar*
`fuzzchar`	Character for small variance decomposition proportion values
`...`	arguments to be passed on to or from other methods (unused)

Details

`colldiag` is an implementation of the regression collinearity diagnostic procedures found in Belsley, Kuh, and Welsch (1980). These procedures examine the "conditioning" of the matrix of independent variables.

It computes the condition indexes of the model matrix. If the largest condition index (the condition number) is *large* (Belsley et al suggest 30 or higher), then there may be collinearity problems. All *large* condition indexes may be worth investigating.

`colldiag` also provides further information that may help to identify the source of these problems, the *variance decomposition proportions* associated with each condition index. If a large condition index is associated two or more variables with *large* variance decomposition proportions, these variables may be causing collinearity problems. Belsley et al suggest that a *large* proportion is 50 percent or more.

Note that such collinearity diagnostics are often provided by other software for the model matrix including the constant term for the intercept (e.g., SAS PROC REG, with the option COLLIN). However, these are generally useless and misleading unless the intercept has some real interpretation and the origin of the regressors is contained within the prediction space, as explained by Fox (1997, p. 351). The default values for `scale`, `center` and `add.intercept` exclude the constant term, and correspond to the SAS option COLLINNOINT.

Value

A `"colldiag"` object, containing:

`condindx` A one-column matrix of condition indexes

`pi` A square matrix of variance decomposition proportions. The rows refer to the principal component dimensions, the columns to the predictor variables.

`print.colldiag` prints the condition indexes as the first column of a table with the variance decomposition proportions beside them. `print.colldiag` has a `fuzz` option to suppress printing of small numbers. If fuzz is used, small values are replaces by a period ".". `Fuzzchar` can be used to specify an alternative character.

Note

Missing data is silently omitted in these calculations

Author(s)

John Hendrickx

Source

These functions were taken from the (now defunct) `perturb` package by John Hendrickx. He credits the Stata program `coldiag` by Joseph Harkness `<joe.harkness@jhu.edu>`, Johns Hopkins University.

References

Belsley, D.A., Kuh, E. and Welsch, R. (1980). *Regression Diagnostics*, New York: John Wiley & Sons.

Belsley, D.A. (1991). *Conditioning diagnostics, collinearity and weak data in regression*. New York: John Wiley & Sons.

Fox, J. (1997). *Applied Regression Analysis, Linear Models, and Related Methods*. thousand Oaks, CA: Sage Publications.

Friendly, M., & Kwan, E. (2009). Where's Waldo: Visualizing Collinearity Diagnostics. *The American Statistician*, **63**, 56–65.

See Also

`lm`, `scale`, `svd`, `[car]vif`, `[rms]vif`

Examples

```
data(cars)
cars.mod <- lm (mpg ~ cylinder + engine + horse + weight + accel + year,
                data=cars)
car::vif(cars.mod)

# SAS PROC REG / COLLIN option, including the intercept
colldiag(cars.mod, add.intercept = TRUE)

# Default settings: scaled, not centered, no intercept, like SAS PROC REG / COLLINNOINT
colldiag(cars.mod)

(cd <- colldiag(cars.mod, center=TRUE))

# fuzz small values
print(cd, fuzz = 0.5)

# Biomass data
data(biomass)

biomass.mod <- lm (biomass ~ H2S + sal + Eh7 + pH + buf + P + K +
                             Ca + Mg + Na + Mn + Zn + Cu + NH4,
                   data=biomass)
car::vif(biomass.mod)

cd <- colldiag(biomass.mod, center=TRUE)
# simplified display
print(colldiag(biomass.mod, center=TRUE), fuzz=.3)

# None yet
```

`consumption` *Consumption Function Dataset*

Description

Example from pp 149-154 of Belsley (1991), Conditioning Diagnostics

Format

A data frame with 28 observations on the following 5 variables.

year 1947 to 1974

cons total consumption, 1958 dollars

rate the interest rate (Moody's Aaa)

dpi disposable income, 1958 dollars

d_dpi annual change in disposable income

References

Belsley, D.A. (1991). *Conditioning diagnostics, collinearity and weak data in regression.* New York: John Wiley & Sons.

Examples

```
data(consumption)

ct1 <- with(consumption, c(NA,cons[-length(cons)]))
# compare (5.3)
m1 <- lm(cons ~ ct1 + dpi + rate + d_dpi, data = consumption)
anova(m1)

# compare exhibit 5.11
with(consumption, cor(cbind(ct1, dpi, rate, d_dpi), use="complete.obs"))
# compare exhibit 5.12
cd<-colldiag(m1)
cd
print(cd,fuzz=.3)
```

---

make.patterns *Construct collection of pattern specifications for tableplot*

Description

Construct collection of pattern specifications for tableplot

Usage

```
make.patterns(
  n = NULL,
  shape = 0,
  shape.col = "black",
  shape.lty = 1,
  cell.fill = "white",
  back.fill = "white",
  label = 0,
  label.size = 0.7,
  ref.col = "gray80",
  ref.grid = FALSE,
  scale.max = 1,
  as.data.frame = FALSE
)
```

Arguments

`n` Number of patterns

`shape` Shape(s) used to encode the numerical value of `cell`. Any of `0="circle"`, `1="diamond"`, `2="square"`. Recycled to match the number of values in the cell.

`shape.col` Outline color(s) for the shape(s)

`shape.lty` Outline line type(s) for the shape(s)

`cell.fill` inside color of |smallest| shape in a cell

`back.fill` background color of cell

`label` how many cell values will be labeled in the cell; max is 4

`label.size` size of cell label(s)

`ref.col` color of reference lines

`ref.grid` whether to draw ref lines in the cells or not

`scale.max` scale values to this maximum

`as.data.frame` whether to return a data.frame or a list.

Value

Returns either a data.frame of a list. If a data.frame, the pattern specifications appear as columns

Examples

```
# None
```

`tableplot` *Tableplot: A Semi-graphic Display of a Table*

Description

A tableplot (Kwan, 2008) is designed as a semi-graphic display in the form of a table with numeric values, but supplemented by symbols with size proportional to cell value(s), and with other visual attributes (shape, color fill, background fill, etc.) that can be used to encode other information essential to direct visual understanding. Three-way arrays, where the last dimension corresponds to levels of a factor for which the first two dimensions are to be compared are handled by superimposing symbols.

The specifications for each cell are given by the `types` argument, whose elements refer to the attributes specified in `patterns`.

Usage

```
tableplot(values, ...)

## Default S3 method:
tableplot(
  values,
  types,
  patterns = list(list(0, "black", 1, "white", "white", 0, 0.5, "grey80", FALSE, 1)),
  title = "Tableplot",
  side.label = "row",
  top.label = "col",
  table.label = TRUE,
  label.size = 1,
  side.rot = 0,
  gap = 2,
  v.parts = 0,
  h.parts = 0,
  cor.matrix = FALSE,
  var.names = "var",
  ...
)
```

Arguments

`values` A matrix or 3-dimensional array of values to be displayed in a tableplot

`...` Arguments passed down to `tableplot.default`

`types` Matrix of specification assignments, of the same size as the first two dimensions of `values`. Entries refer to the sub-lists of `patterns`. Defaults to a matrix of all 1s, `matrix(1, dim(values)[1], dim(values[2]))`, indicating that all cells use the same pattern specification.

`patterns` List of lists; each list is one specification for the arguments to `cellgram`.

`title` Main title

`side.label` a character vector providing labels for the rows of the tableplot

`top.label` a character vector providing labels for the columns of the tableplot

`table.label` Whether to print row/column labels

`label.size` Character size for labels

`side.rot` Degree of rotation (positive for counter-clockwise)

`gap` Width of the gap in each partition, if partitions are requested by `v.parts` and/or `h.parts`

`v.parts` An integer vector giving the number of columns in two or more partitions of the table. If provided, sum must equal number of columns.

`h.parts` An integer vector giving the number of rows in two or more partitions of the table. If provided, sum must equal number of rows.

`cor.matrix` Logical. `TRUE` for a correlation matrix

`var.names` a list of variable names

Value

None. Used for its graphic side effect

Note

The original version of tableplots was in the now-defunct tableplot package `https://cran.r-project.org/package=tableplot`. The current implementation is a modest re-design focused on its use for collinearity diagnostics, but usable in more general contexts.

Author(s)

Ernest Kwan and Michael Friendly

References

Kwan, E. (2008). Improving Factor Analysis in Psychology: Innovations Based on the Null Hypothesis Significance Testing Controversy. Ph. D. thesis, York University.

See Also

`cellgram`

Examples

```
data(cars)
cars.mod <- lm (mpg ~ cylinder + engine + horse + weight + accel + year,
                data=cars)
car::vif(cars.mod)

(cd <- colldiag(cars.mod, center=TRUE))
tableplot(cd, title = "Tableplot of cars data", cond.max = 30 )

data(baseball, package = "corrgram")

baseball$Years7 <- pmin(baseball$Years,7)

base.mod <- lm(logSal ~ Years7 + Atbatc + Hitsc + Homerc + Runsc + RBIc + Walksc,
               data=baseball)
car::vif(base.mod)

cd <- colldiag(base.mod, center=TRUE)
tableplot(cd)
```

tableplot.colldig *Tableplot for Collinearity Diagnostics*

Description

These methods produce a tableplot of collinearity diagnostics, showing the condition indices and variance proportions for predictors in a linear or generalized linear regression model. This encodes the condition indices using *squares* whose background color is red for condition indices > 10, green for values > 5 and green otherwise, reflecting danger, warning and OK respectively. The value of the condition index is encoded within this using a white square proportional to the value (up to some maximum value, `cond.max`),

Variance decomposition proportions are shown by filled *circles* whose radius is proportional to those values and are filled (by default) with shades ranging from white through pink to red. Rounded values of those diagnostics are printed in the cells.

Usage

```
## S3 method for class 'lm'
tableplot(values, ...)

## S3 method for class 'glm'
tableplot(values, ...)

## S3 method for class 'colldiag'
tableplot(
  values,
  prop.col = c("white", "pink", "red"),
  cond.col = c("#A8F48D", "#DDAB3E", "red"),
  cond.max = 100,
  prop.breaks = c(0, 20, 50, 100),
  cond.breaks = c(0, 5, 10, 1000),
  show.rows = nvar:1,
  title = "",
  patterns,
  ...
)
```

Arguments

`values`	A `"colldiag"`, `"lm"` or `"glm"` object
`...`	other arguments, for consistency with generic
`prop.col`	A vector of colors used for the variance proportions. The default is `c("white", "pink", "red")`.
`cond.col`	A vector of colors used for the condition indices
`cond.max`	Maximum value to scale the white squares for the condition indices

`prop.breaks` Scale breaks for the variance proportions

`cond.breaks` Scale breaks for the condition indices

`show.rows` Rows of the eigenvalue decompositon of the model matrix to show in the display. The default `nvar:1` puts the smallest dimensions at the top of the display.

`title` title used for the resulting graphic

`patterns` pattern matrix used for table plot.

Value

None. Used for its graphic side-effect

Author(s)

Michael Friendly

References

Friendly, M., & Kwan, E. (2009). "Where's Waldo: Visualizing Collinearity Diagnostics." *The American Statistician*, **63**, 56–65. Online: https://www.datavis.ca/papers/viscollin-tast.pdf.

Examples

```
# None yet
```

Index