Package ‘mvinfluence’

September 20, 2022

Type Package

Title Influence Measures and Diagnostic Plots for Multivariate Linear Models

Version 0.9.0

Date 2022-09-10

Maintainer Michael Friendly <friendly@yorku.ca>

Description Computes regression deletion diagnostics for multivariate linear models and provides some associated diagnostic plots. The diagnostic measures include hat-values (leverages), generalized Cook's distance, and generalized squared 'studentized' residuals. Several types of plots to detect influential observations are provided.

Depends car, heplots

Suggests knitr, rmarkdown, ggplot2, tibble, patchwork, rgl, dplyr

LazyData TRUE

VignetteBuilder knitr

Encoding UTF-8

License GPL-2

Language en-US

URL https://github.com/friendly/mvinfluence

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

RoxygenNote 7.2.1

NeedsCompilation no

Author Michael Friendly [aut, cre] (<https://orcid.org/0000-0002-3237-0941>)

Repository CRAN

Date/Publication 2022-09-20 17:10:02 UTC
R topics documented:

as.data.frame.inflmlm ........................................... 2
cooks.distance.mlm ........................................... 3
Fertilizer .......................................................... 4
hatvalues.mlm ..................................................... 5
infIndexPlot.mlm ................................................. 6
influence.mlm ..................................................... 8
influencePlot.mlm ............................................... 10
Jtr ................................................................. 12
lrPlot ............................................................. 14
mlm.influence ...................................................... 16
mpower ............................................................. 18
mvinfluence ......................................................... 19
print.inflmlm ....................................................... 22
tr ................................................................. 22

Index 24

---

as.data.frame.inflmlm  
*Convert an inflmlm object to a data frame*

Description

This function is used internally in the package to convert the result of `mlm.influence()` to a data frame. It is not normally called by the user.

Usage

```r
## S3 method for class 'inflmlm'
as.data.frame(x, ..., FUN = det, funnames = TRUE)
```

Arguments

- **x**: An `inflmlm` object, as returned by `mlm.influence`
- **...**: ignored
- **FUN**: in the case where the subset size, \( m > 1 \), the function used on the \( H, Q, L, R \) to calculate a single statistic. The default is `det`. An alternative is `tr`, for matrix trace.
- **funnames**: logical. Should the `FUN` name be prepended to the statistics when creating a data frame?

Value

A data frame containing the influence statistics
cooks.distance.mlm

## S3 method for class 'mlm'
cooks.distance(model, infl = mlm.influence(model, do.coef = FALSE), ...)

### Arguments
- **model**: A `mlm` object, fit by `lm()`
- **infl**: A `inflmlm` object. The default simply runs `mlm.influence()` on the model, suppressing coefficients.
- **...**: Ignored

### Details
In addition, the functions provide diagnostics for deletion of subsets of observations of size \( m \geq 1 \).

### Value
A vector of Cook’s distances

### References

### Examples
```r
data(Rohwer, package="heplots")
Rohwer2 <- subset(Rohwer, subset=group==2)ownames(Rohwer2) <- 1:nrow(Rohwer2)
Rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ n+s+ns+na+ss, data=Rohwer2)
hatvalues(Rohwer.mod)
```
Description

A small data set on the use of fertilizer (x) in relation to the amount of grain (y1) and straw (y2) produced.

Format

A data frame with 8 observations on the following 3 variables.

- **grain** amount of grain produced
- **straw** amount of straw produced
- **fertilizer** amount of fertilizer applied

Details

The first observation is an obvious outlier and influential observation.

Source


References


Examples

data(Fertilizer)

# simple plots
plot(Fertilizer, col=c('red', rep("blue",7)),
     cex=c(2,rep(1.2,7)),
     pch=as.character(1:8))

# A biplot shows the data in 2D. It gives another view of how case 1 stands out in data space
biplot(prcomp(Fertilizer))

# fit the mlm
mod <- lm(cbind(grain, straw) ~ fertilizer, data=Fertilizer)
Anova(mod)
hatvalues.mlm

## S3 method for class 'mlm'
hatvalues(model, m = 1, infl, ...)

Arguments
- **model**: An object of class `mlm`, as returned by `lm`
- **m**: The size of subsets to be considered
- **infl**: An `inflmlm` object, as returned by `mlm.influence`
- **...**: Other arguments, for compatibility with the generic; ignored.

Details
Hat values are a component of influence diagnostics, measuring the leverage or outlyingness of observations in the space of the predictor variables.

The usual case considers observations one at a time (\(m=1\)), where the hatvalue is proportional to the squared Mahalanobis distance, \(D^2\) of each observation from the centroid of all observations. This function extends that definition to calculate a comparable quantity for subsets of size \(m\geq 1\).

Value
A vector of hatvalues

References
See Also

cooks.distance.mlm

Examples

data(Rohwer, package="heplots")
Rohwer2 <- subset(Rohwer, subset=group==2)
rownames(Rohwer2)<- 1:nrow(Rohwer2)
Rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ n+s+ns+na+ss, data=Rohwer2)

options(digits=3)
hatvalues(Rohwer.mod)
cooks.distance(Rohwer.mod)

infIndexPlot.mlm

Influence Index Plots for Multivariate Linear Models

Description

Provides index plots of some diagnostic measures for a multivariate linear model: Cook’s distance, a generalized (squared) studentized residual, hat-values (leverages), and Mahalanobis squared distances of the residuals.

Usage

## S3 method for class 'mlm'
infIndexPlot(
  model,
  inf1 = mlm.influence(model, do.coef = FALSE),
  FUN = det,
  vars = c("Cook", "Studentized", "hat", "DSQ"),
  main = paste("Diagnostic Plots for", deparse(substitute(model))),
  pch = 19,
  labels,
  id.method = "y",
  id.n = if (id.method[1] == "identify") Inf else 0,
  id.cex = 1,
  id.col = palette()[1],
  id.location = "lr",
  grid = TRUE,
  ...
)
Arguments

model
A multivariate linear model object of class `mlm`.

infl
influence measure structure as returned by `mlm.influence`

FUN
For m>1, the function to be applied to the H and Q matrices returning a scalar value. `FUN=det` and `FUN=tr` are possible choices, returning the |H| and tr(H) respectively.

vars
All the quantities listed in this argument are plotted. Use "Cook" for generalized Cook’s distances, "Studentized" for generalized Studentized residuals, "hat" for hat-values (or leverages), and DSQ for the squared Mahalanobis distances of the model residuals. Capitalization is optional. All may be abbreviated by the first one or more letters.

main
main title for graph

pch
Plotting character for points

id.method, labels, id.n, id.cex, id.col, id.location
Arguments for the labeling of points. The default is `id.n=0` for labeling no points. See `showLabels` for details of these arguments.

grid
If TRUE, the default, a light-gray background grid is put on the graph

... Arguments passed to `plot`

Details

This function produces index plots of the various influence measures calculated by `influence.mlm`, and in addition, the measure based on the Mahalanobis squared distances of the residuals from the origin.

Value

None. Used for its side effect of producing a graph.

Author(s)

Michael Friendly; borrows code from `car::infIndexPlot`

References


See Also

`influencePlot.mlm`, `Mahalanobis`, `infIndexPlot`
Examples

```r
# iris data
data(iris)
iris.mod <- lm(as.matrix(iris[,1:4]) ~ Species, data=iris)
inflIndexPlot(iris.mod, col=iris$Species, id.n=3)

# Sake data
data(Sake, package="heplots")
Sake.mod <- lm(cbind(taste,smell) ~ ., data=Sake)
inflIndexPlot(Sake.mod, id.n=3)

# Rohwer data
data(Rohwer, package="heplots")
Rohwer2 <- subset(Rohwer, subset=group==2)
rownames(Rohwer2)<- 1:nrow(Rohwer2)
rohwer.mlm <- lm(cbind(SAT, PPVT, Raven) ~ n + s + ns + na + ss, data=Rohwer2)
inflIndexPlot(rohwer.mlm, id.n=3)
```

influence.mlm

**Regression Deletion Diagnostics for Multivariate Linear Models**

**Description**

This collection of functions is designed to compute regression deletion diagnostics for multivariate linear models following Barrett & Ling (1992) that are close analogs of methods for univariate and generalized linear models handled by the `influence.measures` in the `stats` package.

**Usage**

```r
## S3 method for class 'mlm'
influence(model, do.coef = TRUE, m = 1, ...)
```

**Arguments**

- **model**: An `mlm` object, as returned by `lm`
- **do.coef**: logical. Should the coefficients be returned in the `inflmlm` object?
- **m**: Size of the subsets for deletion diagnostics
- **...**: Other arguments passed to methods

**Details**

In addition, the functions provide diagnostics for deletion of subsets of observations of size \( m \geq 1 \).

`influence.mlm` is a simple wrapper for the computational function, `mlm.influence` designed to provide an S3 method for class "mlm" objects.

There are still infelicities in the methods for the \( m > 1 \) case in the current implementation. In particular, for \( m > 1 \), you must call `influence.mlm` directly, rather than using the S3 generic `influence()`. 
influence.mlm

Value

influence.mlm returns an S3 object of class inflmlm, a list with the following components:

- **m**: Deletion subset size
- **H**: Hat values, \(H_i\). If \(m=1\), a vector of diagonal entries of the ‘hat’ matrix. Otherwise, a list of \(m \times m\) matrices corresponding to the subsets.
- **Q**: Residuals, \(Q_i\).
- **CookD**: Cook’s distance values
- **L**: Leverage components
- **R**: Residual components
- **subsets**: Indices of the observations in the subsets of size \(m\)
- **labels**: Observation labels
- **call**: Model call for the mlm object
- **Beta**: Deletion regression coefficients— included if `do.coef=TRUE`

Author(s)

Michael Friendly

References


See Also

influencePlot.mlm, mlm.influence

Examples

# Rohwer data
data(Rohwer, package="heplots")
Rohwer2 <- subset(Rohwer, subset=group==2)ownames(Rohwer2)<-1:nrow(Rohwer2)
Rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ n+s+ns+na+ss, data=Rohwer2)

# m=1 diagnostics
influence(Rohwer.mod) |> head()

# try an m=2 case
## res2 <- influence.mlm(Rohwer.mod, m=2, do.coef=FALSE)
## res2.df <- as.data.frame(res2)
## head(res2.df)
## scatterplotMatrix(log(res2.df))

influencePlot(Rohwer.mod, id.n=4, type="cookd")
```r
# Sake data
data(Sake, package="heplots")
Sake.mod <- lm(cbind(taste,smell) ~ ., data=Sake)
influence(Sake.mod)
influencePlot(Sake.mod, id.n=3, type="cookd")
```

---

**influencePlot.mlm**

**Influence Plots for Multivariate Linear Models**

**Description**

This function creates various types of “bubble” plots of influence measures with the areas of the circles representing the observations proportional to generalized Cook’s distances.

**Usage**

```r
## S3 method for class 'mlm'
influencePlot(
  model, 
  scale = 12, 
  type = c("stres", "LR", "cookd"), 
  infl = mlm.influence(model, do.coef = FALSE), 
  FUN = det, 
  fill = TRUE, 
  fill.col = "red", 
  fill.alpha.max = 0.5, 
  labels, 
  id.method = "noteworthy", 
  id.n = if (id.method[1] == "identify") Inf else 0, 
  id.cex = 1, 
  id.col = palette()[1], 
  ref.col = "gray", 
  ref.lty = 2, 
  ref.lab = TRUE, 
  ...
)
```

**Arguments**

- `model` An `mlm` object, as returned by `lm` with a multivariate response.
- `scale` a factor to adjust the radii of the circles, in relation to `sqrt(CookD)`
- `type` Type of plot: one of c("stres", "cookd", "LR"). See Details.
- `infl` influence measure structure as returned by `mlm.influence`
FUN
For \( m > 1 \), the function to be applied to the \( H \) and \( Q \) matrices returning a scalar value. \( \text{FUN} = \text{det} \) and \( \text{FUN} = \text{tr} \) are possible choices, returning the \( |H| \) and \( tr(H) \) respectively.

fill, fill.col, fill.alpha.max
fill: logical, specifying whether the circles should be filled. When fill=TRUE, fill.col gives the base fill color to which transparency specified by fill.alpha.max is applied.

labels, id.method, id.n, id.cex, id.col
settings for labeling points; see showLabels for details. To omit point labeling, set id.n=0, the default. The default id.method="noteworthy" is used in this function to indicate setting labels for points with large Studentized residuals, hat-values or Cook's distances. See Details below. Set id.method="identify" for interactive point identification.

ref.col, ref.lty, ref.lab
arguments for reference lines. Incompletely implemented in this version

Details

type="stres" plots squared (internally) Studentized residuals against hat values; type="cookd" plots Cook's distance against hat values; type="LR" plots residual components against leverage components, with the attractive property that contours of constant Cook's distance fall on diagonal lines with slope = -1. Adjacent reference lines represent multiples of influence.

The id.method="noteworthy" setting also requires setting id.n>0 to have any effect. Using id.method="noteworthy", and id.n>0, the number of points labeled is the union of the largest id.n values on each of L, R, and CookD.

Value
If points are identified, returns a data frame with the hat values, Studentized residuals and Cook's distance of the identified points. If no points are identified, nothing is returned. This function is primarily used for its side-effect of drawing a plot.

Author(s)
Michael Friendly

References


See Also

`mlm.influence`, `lrPlot`

`influencePlot` in the car package

Examples

```r
data(Rohwer, package="heplots")
Rohwer2 <- subset(Rohwer, subset=group==2)
Rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ n+s+ns+na+ss, data=Rohwer2)

influencePlot(Rohwer.mod, id.n=4, type="stres")
influencePlot(Rohwer.mod, id.n=4, type="LR")
influencePlot(Rohwer.mod, id.n=4, type="cookd")

# Sake data
data(Sake, package="heplots")
Sake.mod <- lm(cbind(taste,smell) ~ ., data=Sake)
influencePlot(Sake.mod, id.n=3, type="stres")
influencePlot(Sake.mod, id.n=3, type="LR")
influencePlot(Sake.mod, id.n=3, type="cookd")

# Adopted data
data(Adopted, package="heplots")
Adopted.mod <- lm(cbind(Age2IQ, Age4IQ, Age8IQ, Age13IQ) ~ AMED + BMIQ, data=Adopted)
influencePlot(Adopted.mod, id.n=3)
influencePlot(Adopted.mod, id.n=3, type="LR", ylim=c(-4,-1.5))
```

Description

These functions implement the general classes of influence measures for multivariate regression models defined in Barrett and Ling (1992), Eqn 2.3, 2.4, as shown in their Table 1.

Usage

- `Jtr(H, Q, a, b, f)`
- `Jdet(H, Q, a, b, f)`
- `COOKD(H, Q, n, p, r, m)`
- `DFFITS(H, Q, n, p, r, m)`
- `COVRATIO(H, Q, n, p, r, m)`
Arguments

- \( H \) a scalar or \( m \times m \) matrix giving the hat values for subset \( I \)
- \( Q \) a scalar or \( m \times m \) matrix giving the residual values for subset \( I \)
- \( a \) the \( a \) parameter for the \( J^{det} \) and \( J^{tr} \) classes
- \( b \) the \( b \) parameter for the \( J^{det} \) and \( J^{tr} \) classes
- \( f \) scaling factor for the \( J^{det} \) and \( J^{tr} \) classes
- \( n \) sample size
- \( p \) number of predictor variables
- \( r \) number of response variables
- \( m \) deletion subset size

Details

There are two classes of functions, denoted \( J^{det} \) and \( J^{tr} \), with parameters \( n, p, q \) of the data, \( m \) of the subset size and \( a \) and \( b \) which define powers of terms in the formulas, typically in the set \(-2, -1, 0\).

They are defined in terms of the submatrices for a deleted index subset \( I \),

\[
H_I = X_I (X^T X)^{-1} X_I
\]
\[
Q_I = E_I (E^T E)^{-1} E_I
\]

corresponding to the hat and residual matrices in univariate models.

For subset size \( m = 1 \) these evaluate to scalar equivalents of hat values and studentized residuals.

For subset size \( m > 1 \) these are \( m \times m \) matrices and functions in the \( J^{det} \) class use \(|H_I|\) and \(|Q_I|\), while those in the \( J^{tr} \) class use \( tr(H_I) \) and \( tr(Q_I) \).

The functions \texttt{COOKD}, \texttt{COVRATIO}, and \texttt{DFFITS} implement some of the standard influence measures in these terms for the general cases of multivariate linear models and deletion of subsets of size \( m > 1 \), but they have not yet been incorporated into our main functions \texttt{mlm.influence} and \texttt{influence.mlm}.

Value

The scalar result of the computation.

Author(s)

Michael Friendly

References

lrPlot  

Regression LR Influence Plot

Description

This function creates a “bubble” plot of functions, \( R = \log(\text{Studentized residuals}^2) \) by \( L = \log(H/p*(1-H)) \) of the hat values, with the areas of the circles representing the observations proportional to Cook’s distances.

Usage

```r
lrPlot(model, ...) 
## S3 method for class lm
lrPlot(
  model, 
  scale = 12,
  xlab = "log Leverage factor \[\log H/p*(1-H)\]",
  ylab = "log (Studentized Residual^2)",
  xlim = NULL, 
  ylim, 
  labels, 
  id.method = "noteworthy",
  id.n = if (id.method[1] == "identify") Inf else 0,
  id.cex = 1,
  id.col = palette()[1],
  ref = c("h", "v", "d", "c"),
  ref.col = "gray",
  ref.lty = 2,
  ref.lab = TRUE,
  ...
)
```

Arguments

- `model`  
  a model object fit by `lm`
- `...`  
  arguments to pass to the `plot` and `points` functions.
- `scale`  
  a factor to adjust the radii of the circles, in relation to \( \sqrt{\text{CookD}} \)
- `xlab, ylab`  
  axis labels.
- `xlim, ylim`  
  Limits for x and y axes. In the space of (L, R) very small residuals typically extend the y axis enough to swamp the large residuals, so the default for `ylim` is set to a range of 6 log units starting at the maximum value.
- `labels, id.method, id.n, id.cex, id.col`  
  settings for labeling points; see `link{showLabels}` for details. To omit point labeling, set `id.n`=0, the default. The default `id.method="noteworthy"` is used
in this function to indicate setting labels for points with large Studentized residuals, hat-values or Cook’s distances. See Details below. Set `id.method="identify"` for interactive point identification.

**ref**
Options to draw reference lines, any one or more of `c("h", "v", "d", "c")`. "h" and "v" draw horizontal and vertical reference lines at noteworthy values of R and L respectively. "d" draws equally spaced diagonal reference lines for contours of equal CookD. "c" draws diagonal reference lines corresponding to approximate 0.95 and 0.99 contours of CookD.

**ref.col, ref.lty**
Color and line type for reference lines. Reference lines for `"c" %in% ref` are handled separately.

**ref.lab**
A logical, indicating whether the reference lines should be labeled.

**Details**

This plot, suggested by McCulloch & Meeter (1983) has the attractive property that contours of equal Cook’s distance are diagonal lines with slope = -1. Various reference lines are drawn on the plot corresponding to twice and three times the average hat value, a “large” squared studentized residual and contours of Cook’s distance.

The `id.method="noteworthy"` setting also requires setting `id.n>0` to have any effect. Using `id.method="noteworthy", and id.n>0`, the number of points labeled is the union of the largest `id.n` values on each of L, R, and CookD.

**Value**

If points are identified, returns a data frame with the hat values, Studentized residuals and Cook’s distance of the identified points. If no points are identified, nothing is returned. This function is primarily used for its side-effect of drawing a plot.

**Author(s)**

Michael Friendly

**References**


**See Also**

`influencePlot.mlm` `influencePlot` in the `car` package for other methods
Examples

# artificial example from Lawrence (1995)
x <- c(0, 0, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 18, 18)
y <- c(0, 6, 6, 7, 6, 7, 6, 7, 6, 7, 6, 7, 7, 18)
DF <- data.frame(x, y, row.names=LETTERS[1:length(x)])
DF

with(DF, {
    plot(x, y, pch=16, cex=1.3)
    abline(lm(y~x), col="red", lwd=2)
    NB <- c(1,2,13,14)
    text(x[NB], y[NB], LETTERS[NB], pos=c(4,4,2,2))
}
)

mod <- lm(y~x, data=DF)
# standard influence plot from car
influencePlot(mod, id.n=4)

# lrPlot version
lrPlot(mod, id.n=4)

library(car)
dmod <- lm(prestige ~ income + education, data = Duncan)
influencePlot(dmod, id.n=3)
lrPlot(dmod, id.n=3)

---

### mlm.influence

**Calculate Regression Deletion Diagnostics for Multivariate Linear Models**

Description

mlm.influence is the main computational function in this package. It is usually not called directly, but rather via its alias, influence.mlm, the S3 method for a mlm object.

Usage

```r
mlm.influence(model, do.coef = TRUE, m = 1, ...)
```

Arguments

- **model**: An mlm object, as returned by lm with a multivariate response.
- **do.coef**: logical. Should the coefficients be returned in the inflmlm object?
- **m**: Size of the subsets for deletion diagnostics
- **...**: Further arguments passed to other methods
Details

The computations and methods for the $m=1$ case are straightforward, as are the computations for the $m>1$ case. Associated methods for $m>1$ are still under development.

Value

`mlm.influence` returns an S3 object of class `inflmlm`, a list with the following components:

- **m**: Deletion subset size
- **H**: Hat values, $H_I$. If $m=1$, a vector of diagonal entries of the ‘hat’ matrix. Otherwise, a list of $m \times m$ matrices corresponding to the subsets.
- **Q**: Residuals, $Q_I$.
- **CookD**: Cook’s distance values
- **L**: Leverage components
- **R**: Residual components
- **subsets**: Indices of the subsets
- **CookD**: Cook’s distance values
- **L**: Leverage components
- **R**: Residual components
- **subsets**: Indices of the observations in the subsets of size $m$
- **labels**: Observation labels
- **call**: Model call for the `mlm` object
- **Beta**: Deletion regression coefficients– included if `do.coef=TRUE`

Author(s)

Michael Friendly

References


See Also

`influencePlot.mlm`
mpower

General Matrix Power

Description

Calculates the \( n \)-th power of a square matrix, where \( n \) can be a positive or negative integer or a fractional power.

Usage

\[
\text{mpower}(A, n)
\]

\( A \%^{\%} n \)

Arguments

- \( A \) : A square matrix. Must also be symmetric for non-integer powers.
- \( n \) : matrix power

Details

If \( n<0 \), the method is applied to \( A^{-1} \). When \( n \) is an integer, the function uses the Russian peasant method, or repeated squaring for efficiency. Otherwise, it uses the spectral decomposition of \( A \), \( A^n = V D^n V^T \) requiring a symmetric matrix.
Value

Returns the matrix $A^n$

Author(s)

Michael Friendly

References

https://en.wikipedia.org/wiki/Exponentiation_by_squaring

See Also

Packages corpcor and expm define similar functions.

Examples

```r
M <- matrix(sample(1:9), 3,3)
mpower(M,2)
mpower(M,4)

# make a symmetric matrix
MM <- crossprod(M)
mpower(MM, -1)
Mhalf <- mpower(MM, 1/2)
all.equal(MM, Mhalf %*% Mhalf)
```

---

**mv influence**  
Influence Measures and Diagnostic Plots for Multivariate Linear Models

Description

Functions in this package compute regression deletion diagnostics for multivariate linear models following methods proposed by Barrett & Ling (1992) and provide some associated diagnostic plots.

Details

The design goal for this package is that, as an extension of standard methods for univariate linear models, you should be able to fit a linear model with a multivariate response,

```r
mymlm <- lm( cbind(y1, y2, y3) ~ x1 + x2 + x3, data=mydata)
```

and then get useful diagnostics and plots with
The diagnostic measures include hat-values (leverages), generalized Cook’s distance and generalized squared ‘studentized’ residuals. Several types of plots to detect influential observations are provided.

In addition, the functions provide diagnostics for deletion of subsets of observations of size \( m \geq 1 \). This case is theoretically interesting because sometimes pairs \( (m=2) \) of influential observations can mask each other, sometimes they can have joint influence far exceeding their individual effects, as well as other interesting phenomena described by Lawrence (1995). Associated methods for the case \( m > 1 \) are still under development in this package.

The main function in the package is the S3 method, `influence.mlm`, a simple wrapper for `mlm.influence`, which does the actual computations. This design was dictated by that used in the `stats` package, which provides the generic method `influence` and methods `influence.lm` and `influence.glm`. The `car` package extends this to include `influence.lme` for models fit by `lme`.

The following sections describe the notation and measures used in the calculations.

**Notation**

Let \( X \) be the model matrix in the multivariate linear model, \( Y_{n \times p} = X_{n \times r} \beta_{r \times p} + E_{n \times p} \). The usual least squares estimate of \( \beta \) is given by \( \hat{B} = (X^T X)^{-1} X^T Y \).

Then let

- \( X_I \) be the submatrix of \( X \) whose \( m \) rows are indexed by \( I \),
- \( X_{(I)} \) is the complement, the submatrix of \( X \) with the \( m \) rows in \( I \) deleted,

Matrices \( Y_I, Y_{(I)} \) are defined similarly.

In the calculation of regression coefficients, \( \hat{B(I)} = (X_I^T X_I)^{-1} X_I^T Y_I \) are the estimated coefficients when the cases indexed by \( I \) have been removed. The corresponding residuals are \( E_I = Y_I - X_I \hat{B(I)} \).

**Measures**

The influence measures defined by Barrett & Ling (1992) are functions of two matrices \( H_I \) and \( Q_I \) defined as follows:

- For the full data set, the ‘hat matrix’, \( H \), is given by \( H = X(X^T X)^{-1} X^T \).
- \( H_I \) is \( m \times m \) the submatrix of \( H \) corresponding to the index set \( I \), \( H_I = X(X_I^T X_I)^{-1} X_I^T \),
- \( Q \) is the analog of \( H \) defined for the residual matrix \( E \), that is, \( Q = E(E^T E)^{-1} E^T \), with corresponding submatrix \( Q_I = E(E_I^T E_I)^{-1} E_I^T \),

**Cook’s distance**

In these terms, Cook’s distance is defined for a univariate response by

\[
D_I = (b - b_{(I)})^T (X^T X)(b - b_{(I)}) / ps^2 ,
\]
a measure of the squared distance between the coefficients \( b \) for the full data set and those \( b_{(I)} \) obtained when the cases in \( I \) are deleted.

In the multivariate case, Cook’s distance is obtained by replacing the vector of coefficients \( b \) by \( \text{vec}(B) \), the result of stringing out the coefficients for all responses in a single \( n \times p \)-length vector.

\[
D_I = \frac{1}{p} [\text{vec}(B - B_{(I)})]^T (S_1 \otimes X^T X) \text{vec}(B - B_{(I)}) ,
\]

where \( \otimes \) is the Kronecker (direct) product and \( S = E^T E / (n - p) \) is the covariance matrix of the residuals.

**Leverage and residual components**

For a univariate response, and when \( m = 1 \), Cook’s distance can be re-written as a product of leverage and residual components as

\[
D_i = \left( \frac{n - p}{p} \right) \frac{h_{ii}}{(1 - h_{ii})^2 q_{ii}} .
\]

Then we can define a leverage component \( L_i \) and residual component \( R_i \) as

\[
L_i = \frac{h_{ii}}{1 - h_{ii}} \quad R_i = \frac{q_{ii}}{1 - h_{ii}} ,
\]

\( R_i \) is the studentized residual, and \( D_i \propto L_i \times R_i \).

In the general, multivariate case there are analogous matrix expressions for \( L \) and \( R \). When \( m > 1 \), the quantities \( H_I, Q_I, L_I, \) and \( R_I \) are \( m \times m \) matrices. Where scalar quantities are needed, the package functions apply a function, \( \text{FUN} \), either \( \text{det}() \) or \( \text{tr}() \) to calculate a measure of “size”, as in

\[
H <- \text{sapply}(x$H, \text{FUN})
Q <- \text{sapply}(x$Q, \text{FUN})
L <- \text{sapply}(x$L, \text{FUN})
R <- \text{sapply}(x$R, \text{FUN})
\]

**References**


print.inflmlm  

*Print an inflmlm object*

**Description**

Print an `inflmlm` object

**Usage**

```r
## S3 method for class 'inflmlm'
print(x, digits = max(3, getOption("digits") - 4), FUN = det, ...)
```

**Arguments**

- `x` An `inflmlm` object
- `digits` Number of digits to print
- `FUN` Function to combine diagnostics when \(m>1\), one of `det` or `tr`
- `...` passed to `print()`

**Value**

Invisibly returns the object

**Examples**

```r
# none
```

---

*tr*  

*Matrix trace*

**Description**

Calculates the trace of a matrix

**Usage**

```r
tr(M)
```

**Arguments**

- `M` a matrix

**Details**

For square, symmetric matrices, such as covariance matrices, the trace is sometimes used as a measure of size, e.g., in Pillai’s trace criterion for a MLM.
**tr**

**Value**

returns the sum of the diagonal elements of the matrix

**Author(s)**

Michael Friendly

**Examples**

```r
M <- matrix(sample(1:9), 3,3)
tr(M)
```
Index

* J
  Jtr, 12
* array
  Jtr, 12
  mpower, 18
  tr, 22
* datasets
  Fertilizer, 4
* function
  Jtr, 12
* hplot
  infIndexPlot.mlm, 6
* models
  cooks.distance.mlm, 3
  hatvalues.mlm, 5
  influence.mlm, 8
  influencePlot.mlm, 10
  mlm.influence, 16
* multivariate
  influencePlot.mlm, 10
  mlm.influence, 16
* regression
  cooks.distance.mlm, 3
  hatvalues.mlm, 5
  influence.mlm, 8
  influencePlot.mlm, 10
  lrPlot, 14
  mlm.influence, 16
* trace
  Jtr, 12
  %*% (mpower), 18
  as.data.frame.inflmlm, 2
  COOKD (Jtr), 12
  cooks.distance.mlm, 3, 6
  COVRATIO (Jtr), 12
  DFFITS (Jtr), 12
  Fertilizer, 4

  hatvalues.mlm, 5
  infIndexPlot, 7
  infIndexPlot.mlm, 6
  influence, 20
  influence.glm, 20
  influence.lm, 20
  influence.lme, 20
  influence.measures, 3, 5, 8
  influence.mlm, 7, 8, 13, 16, 20
  influenceIndexPlot (infIndexPlot.mlm), 6
  influencePlot, 12, 15
  influencePlot.mlm, 7, 9, 10, 15, 17
  Jdet (Jtr), 12
  Jtr, 12
  lm, 5, 8, 10, 16
  lme, 20
  lrPlot, 12, 14
  Mahalanobis, 7
  mlm.influence, 7–10, 12, 13, 16, 20
  mpower, 18
  mv Influence, 19
  mv Influence-package (mv Influence), 19
  print.inflmlm, 22
  showLabels, 7, 11
  tr, 22