Package: genridge (via r-universe)

November 20, 2026

Type Package

Title Generalized Ridge Trace Plots for Ridge Regression

Version 0.8.1

Date 2026-11-20

Maintainer Michael Friendly <friendly@yorku.ca>

Depends R (>= 4.1.0), car

Imports rgl, colorspace, splines

Suggests MASS, bestglm, heplots, knitr, rmarkdown, testthat (>= 3.0.0), vcdExtra, withr

VignetteBuilder knitr

Description The genridge package introduces generalizations of the standard univariate ridge trace plot used in ridge regression and related methods. These graphical methods show both bias (actually, shrinkage) and precision, by plotting the covariance ellipsoids of the estimated coefficients, rather than just the estimates themselves. 2D and 3D plotting methods are provided, both in the space of the predictor variables and in the transformed space of the PCA/SVD of the predictors.

License GPL (>= 2)

LazyLoad yes

LazyData yes

Language en-US

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

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

Encoding UTF-8

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

NeedsCompilation no

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

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

Date/Publication 2026-11-20 16:40:02 UTC

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

RemoteRef HEAD

RemoteSha e230e26b51869f6ef9d6c0046f64f8736dbc1a63

Contents

genridge-package . . . 2
Acetylene . . . 4
as.ridge . . . 5
biplot.pcaridge . . . 6
contourf . . . 8
Detroit . . . 10
diab . . . 12
Manpower . . . 13
mlmridge-methods . . . 14
mlmridge-plots . . . 16
pairs.ridge . . . 19
pca . . . 21
plot.precision . . . 22
plot.ridge . . . 24
plot3d . . . 27
precision . . . 30
prostate . . . 32
ridge . . . 33
traceplot . . . 39
trans.colors . . . 40
vif.ridge . . . 42

Index **45**

`genridge-package` *Generalized ridge trace plots for ridge regression*

Description

The genridge package introduces generalizations of the standard univariate ridge trace plot used in ridge regression and related methods (Friendly, 2012). These graphical methods show both bias (actually, shrinkage) and precision, by plotting the covariance ellipsoids of the estimated coefficients, rather than just the estimates themselves. 2D and 3D plotting methods are provided, both in the space of the predictor variables and in the transformed space of the PCA/SVD of the predictors.

Details

This package provides computational support for the graphical methods described in Friendly (2013). Ridge regression models may be fit using the function `ridge()`, which incorporates features of `MASS::lm.ridge()`. In particular, the shrinkage factors in ridge regression may be specified either in terms of the constant added to the diagonal of $X^T X$ matrix (`lambda`), or the equivalent number of degrees of freedom.

More importantly, the `ridge()` function also calculates and returns the associated covariance matrices of each of the ridge estimates, allowing precision to be studied and displayed graphically.

This provides the support for the main plotting functions in the package:

- `plot.ridge()`: Bivariate ridge trace plots
- `pairs.ridge()`: All pairwise bivariate ridge trace plots
- `plot3d.ridge()`: 3D ridge trace plots
- `traceplot()`: Traditional univariate ridge trace plots

In addition, the function `pca.ridge()` transforms the coefficients and covariance matrices of a `ridge` object from predictor space to the equivalent, but more interesting space of the PCA of $X^T X$ or the SVD of $\mathbf{X}$. The main plotting functions also work for these objects, of class `c("ridge", "pcaridge")`.

Finally, the functions `precision()` and `vif.ridge()` provide other useful measures and plots.

Author(s)

Michael Friendly

Maintainer: Michael Friendly friendly@yorku.ca

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

Arthur E. Hoerl and Robert W. Kennard (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems, *Technometrics*, 12(1), pp. 55-67.

Arthur E. Hoerl and Robert W. Kennard (1970). Ridge Regression: Applications to Nonorthogonal Problems *Technometrics*, 12(1), pp. 69-82.

See Also

`MASS::lm.ridge()`

Examples

```
# see examples for ridge, etc.
```

`Acetylene` *Acetylene Data*

Description

The data consist of measures of `yield` of a chemical manufacturing process for acetylene in relation to numeric parameters.

Format

A data frame with 16 observations on the following 4 variables.

- `yield`: conversion percentage yield of acetylene
- `temp`: reactor temperature (celsius)
- `ratio`: H2 to N-heptone ratio
- `time`: contact time (sec)

Details

Marquardt and Snee (1975) used these data to illustrate ridge regression in a model containing quadratic and interaction terms, particularly the need to center and standardize variables appearing in high-order terms.

Typical models for these data include the interaction of `temp:ratio`, and a squared term in `temp`

Source

SAS documentation example for `PROC REG`, *Ridge Regression for Acetylene Data.*

References

Marquardt, D.W., and Snee, R.D. (1975), "Ridge Regression in Practice," *The American Statistician*, **29**, 3-20.

Marquardt, D.W. (1980), "A Critique of Some Ridge Regression Methods: Comment," *Journal of the American Statistical Association*, Vol. 75, No. 369 (Mar., 1980), pp. 87-91

Examples

```
data(Acetylene)

# naive model, not using centering
amod0 <- lm(yield ~ temp + ratio + time + I(time^2) + temp:time, data=Acetylene)

y <- Acetylene[,"yield"]
X0 <- model.matrix(amod0)[,-1]

lambda <- c(0, 0.0005, 0.001, 0.002, 0.005, 0.01)
aridge0 <- ridge(y, X0, lambda=lambda)
```

```
traceplot(aridge0)
traceplot(aridge0, X="df")
pairs(aridge0, radius=0.2)
```

`as.ridge` *Extract a Univariate Ridge Regression from a Multivariate One*

Description

Experimental. Multivariate ridge regression (class `"mlmridge"`) is new in this version; its functions, arguments and results may change in future releases.

`ridge()` with a matrix response (e.g., `ridge(cbind(Y1, Y2) ~ x1 + x2, data)`) returns an object of class `"mlmridge"`, fitting all responses with a common set of ridge constants. `as.ridge()` extracts the results for one response as an ordinary `"ridge"` object, so that all the methods for `"ridge"` objects (`traceplot()`, `plot.ridge()`, `pairs.ridge()`, `plot3d.ridge()`, `precision()`, `vif.ridge()`, `pca()`, ...) can be applied to it.

The result is the same as fitting `ridge()` to that response alone, with the same `lambda` or `df`.

Usage

```
as.ridge(x, ...)

## S3 method for class 'ridge'
as.ridge(x, ...)

## S3 method for class 'mlmridge'
as.ridge(x, response = 1, ...)
```

Arguments

`x`	An object of class `"mlmridge"`
`...`	Not used
`response`	The response to extract: a name, or a number in `1:q`. The default is the first response.

Value

An object of class `"ridge"`; see `ridge()` for its components.

See Also

`ridge()`

Examples

```
if (requireNamespace("heplots", quietly = TRUE)) {
  data(schooldata, package = "heplots")
  sridge <- ridge(cbind(reading, mathematics, selfesteem) ~
                  education + occupation + visit + counseling + teacher,
                  data = schooldata, lambda = c(0, 1, 2, 5, 10, 20))
  # results for one response
  read.ridge <- as.ridge(sridge, "reading")
  read.ridge
  traceplot(read.ridge)
}
```

`biplot.pcaridge` *Biplot of Ridge Regression Trace Plot in SVD Space*

Description

`biplot.pcaridge` supplements the standard display of the covariance ellipsoids for a ridge regression problem in PCA/SVD space with labeled arrows showing the contributions of the original variables to the dimensions plotted.

Usage

```
## S3 method for class 'pcaridge'
biplot(
  x,
  variables = (p - 1):p,
  labels = NULL,
  asp = 1,
  origin,
  scale,
  var.lab = rownames(V),
  var.lwd = 1,
  var.col = "black",
  var.cex = 1,
  xlab,
  ylab,
  prefix = "Dim ",
  suffix = TRUE,
  ...
)
```

Arguments

`x`	A `pcaridge` object computed by `pca.ridge()` or a `ridge` object.

`variables`	The dimensions or variables to be shown in the the plot. By default, the *last* two dimensions, corresponding to the smallest singular values, are plotted for `class("pcaridge")` objects or the *first* two variables for `class("ridge")` objects.
`labels`	A vector of character strings or expressions used as labels for the ellipses. Use `labels=NULL` to suppress these.
`asp`	Aspect ratio for the plot. The default value, `asp=1` helps ensure that lengths and angles are preserved in these plots. Use `asp=NA` to override this.
`origin`	The origin for the variable vectors in this plot, a vector of length 2. If not specified, the function calculates an origin to make the variable vectors approximately centered in the plot window.
`scale`	The scale factor for variable vectors in this plot. If not specified, the function calculates a scale factor to make the variable vectors approximately fill the plot window.
`var.lab`	Labels for variable vectors. The default is the names of the predictor variables.
`var.lwd`, `var.col`, `var.cex`	Line width, color and character size used to draw and label the arrows representing the variables in this plot.
`xlab`, `ylab`	Labels for the plot dimensions. If not specified, `prefix` and `suffix` are used to construct informative dimension labels.
`prefix`	Prefix for labels of the plot dimensions.
`suffix`	Suffix for labels of the plot dimensions. If `suffix=TRUE` the percent of variance accounted for by each dimension is added to the axis label.
`...`	Other arguments, passed to `plot.pcaridge()`

Details

The biplot view showing the dimensions corresponding to the two *smallest* singular values is particularly useful for understanding how the predictors contribute to shrinkage in ridge regression.

This is only a biplot in the loose sense that results are shown in two spaces simultaneously – the transformed PCA/SVD space of the original predictors, and vectors representing the predictors projected into this space.

`biplot.ridge` is a similar extension of `plot.ridge()`, adding vectors showing the relation of the PCA/SVD dimensions to the plotted variables.

`class("ridge")` objects use the transpose of the right singular vectors, `t(x$svd.V)` for the dimension weights plotted as vectors.

Value

None

Author(s)

Michael Friendly, with contributions by Uwe Ligges

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

See Also

`plot.ridge()`, `pca.ridge()`

Examples

```
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(longley.y, longley.X, lambda=lambda)

plridge <- pca(lridge)

plot(plridge, radius=0.5)

# same, with variable vectors
biplot(plridge, radius=0.5)
# add some other options
biplot(plridge, radius=0.5, var.col="brown", var.lwd=2, var.cex=1.2, prefix="Dimension ")

# biplots for ridge objects, showing PCA vectors
plot(lridge, radius=0.5)
biplot(lridge, radius=0.5)
biplot(lridge, radius=0.5, asp=NA)
```

---

contourf *Enhanced Contour Plots*

Description

This is an enhancement to `graphics::contour()`, written as a wrapper for that function. It creates a contour plot, or adds contour lines to an existing plot, allowing the contours to be filled and returning the list of contour lines.

Usage

```
contourf(
  x = seq(0, 1, length.out = nrow(z)),
  y = seq(0, 1, length.out = ncol(z)),
  z,
  nlevels = 10,
  levels = pretty(zlim, nlevels),
```

```
  zlim = range(z, finite = TRUE),
  col = par("fg"),
  color.palette = colorRampPalette(c("white", col)),
  fill.col = color.palette(nlevels + 1),
  fill.alpha = 0.5,
  add = FALSE,
  ...
)
```

Arguments

x, y	locations of grid lines at which the values in z are measured. These must be in ascending order. By default, equally spaced values from 0 to 1 are used. If x is a list, its components x$x and x$y are used for x and y, respectively. If the list has component x$z this is used for z.
z	a matrix containing the values to be plotted (NAs are allowed). Note that x can be used instead of z for convenience.
nlevels	number of contour levels desired **iff** levels is not supplied
levels	numeric vector of levels at which to draw contour lines
zlim	z-limits for the plot. x-limits and y-limits can be passed through ...
col	color for the lines drawn
color.palette	a color palette function to be used to assign fill colors in the plot
fill.col	a call to the color.palette function or an an explicit set of colors to be used in the plot. Use fill.col=NULL to suppress the filled polygons. a vector of fill colors corresponding to levels. By default, a set of possibly transparent colors is calculated ranging from white to col, using transparency given by fill.alpha
fill.alpha	transparency value for fill.col, either a hex character string, or a numeric value between 0 and 1. Use fill.alpha=NA to suppress transparency.
add	logical. If TRUE, add to a current plot.
...	additional arguments passed to graphics::contour(), including all arguments of graphics::contour.default() not mentioned above, as well as additional graphical parameters passed by graphics::contour.default() to more basic functions.

Value

Returns invisibly the list of contours lines, with components levels, x, y. See grDevices::contourLines().

Author(s)

Michael Friendly

See Also

graphics::contour(), grDevices::contourLines()

lattice::contourplot() from package lattice.

Examples

```
x <- 10*1:nrow(volcano)
y <- 10*1:ncol(volcano)
contourf(x,y,volcano, col="blue")
contourf(x,y,volcano, col="blue", nlevels=6)

# return value, unfilled, other graphic parameters
res <- contourf(x,y,volcano, col="blue", fill.col=NULL, lwd=2)
# levels used in the plot
sapply(res, function(x) x[[1]])
```

Detroit *Detroit Homicide Data for 1961-1973*

Description

The data set Detroit was used extensively in the book by Miller (2002) on subset regression. The data are unusual in that a subset of three predictors can be found which gives a very much better fit to the data than the subsets found from the Efroymson stepwise algorithm, or from forward selection or backward elimination. They are also unusual in that, as time series data, the assumption of independence is patently violated, and the data suffer from problems of high collinearity.

As well, ridge regression reveals somewhat paradoxical paths of shrinkage in univariate ridge trace plots, that are more comprehensible in multivariate views.

Format

A data frame with 13 observations on the following 14 variables.

- Police: Full-time police per 100,000 population
- Unemp: Percent unemployed in the population
- MfgWrk: Number of manufacturing workers in thousands
- GunLic: Number of handgun licences per 100,000 population
- GunReg: Number of handgun registrations per 100,000 population
- HClear: Percent of homicides cleared by arrests
- WhMale: Number of white males in the population
- NmfgWrk: Number of non-manufacturing workers in thousands
- GovWrk: Number of government workers in thousands
- HrEarn: Average hourly earnings
- WkEarn: Average weekly earnings
- Accident: Death rate in accidents per 100,000 population
- Assaults: Number of assaults per 100,000 population
- Homicide: Number of homicides per 100,000 of population

Details

The data were originally collected and discussed by Fisher (1976) but the complete dataset first appeared in Gunst and Mason (1980, Appendix A). Miller (2002) discusses this dataset throughout his book, but doesn't state clearly which variables he used as predictors and which is the dependent variable. (`Homicide` was the dependent variable, and the predictors were `Police` ... `WkEarn`.) The data were obtained from StatLib.

A similar version of this data set, with different variable names appears in the `bestglm` package.

Source

https://lib.stat.cmu.edu/datasets/detroit

References

Fisher, J.C. (1976). Homicide in Detroit: The Role of Firearms. *Criminology*, **14**, 387–400.

Gunst, R.F. and Mason, R.L. (1980). *Regression analysis and its application: A data-oriented approach*. Marcel Dekker.

Miller, A. J. (2002). *Subset Selection in Regression*. 2nd Ed. Chapman & Hall/CRC. Boca Raton.

Examples

```
data(Detroit)

# Work with a subset of predictors, from Miller (2002, Table 3.14),
# the "best" 6 variable model
#    Variables: Police, Unemp, GunLic, HClear, WhMale, WkEarn
# Scale these for comparison with other methods

Det <- as.data.frame(scale(Detroit[,c(1,2,4,6,7,11)]))
Det <- cbind(Det, Homicide=Detroit[,"Homicide"])

# use the formula interface; specify ridge constants in terms
# of equivalent degrees of freedom
dridge <- ridge(Homicide ~ ., data=Det, df=seq(6,4,-.5))

# univariate trace plots are seemingly paradoxical in that
# some coefficients "shrink" *away* from 0
traceplot(dridge, X="df")
vif(dridge)
pairs(dridge, radius=0.5)


plot3d(dridge, radius=0.5, labels=dridge$df)

# transform to PCA/SVD space
dpridge <- pca(dridge)

# not so paradoxical in PCA space
traceplot(dpridge, X="df")
biplot(dpridge, radius=0.5, labels=dpridge$df)
```

```
# show PCA vectors in variable space
biplot(dridge, radius=0.5, labels=dridge$df)
```

---

diab *Diabetes Progression*

Description

These data consist of observations on 442 patients, with the response of interest being a quantitative measure of disease progression one year after baseline.

There are ten baseline variables: age, sex, body-mass index (bmi), average blood pressure (map) and six blood serum measurements.

Usage

```
data("diab")
```

Format

A data frame with 442 observations on the following 11 variables.

- prog: disease progression, a numeric vector
- age: age, a numeric vector
- sex: integer, a numeric vector
- bmi: body mass index, a numeric vector
- map: mean arterial blood pressure, a numeric vector
- tc: blood serum TC, a numeric vector
- ldl: blood serum low-density lipoprotein ("bad cholesterol"), a numeric vector
- hdl: blood serum high-density lipoprotein ("good cholesterol"), a numeric vector
- tch: blood serum TCH, a numeric vector
- ltg: blood serum lamotrigine, a numeric vector
- glu: blood serum glucose, a numeric vector

Details

Efron & Hastie describe their analysis using the centered predictor variables standardized to unit L2 norm. ridge() does not (yet) provide this scaling.

Source

The dataset was taken from the web site for Efron & Hastie (2021), *Computer Age Statistical Inference*, https://hastie.su.domains/CASI_files/DATA/diabetes.csv.

References

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least Angle Regression. *The Annals of Statistics*, **32**(2), 407-499. doi:10.1214/009053604000000067

Efron, B., & Hastie, T. (2021). *Computer Age Statistical Inference, Student Edition: Algorithms, Evidence, and Data Science*, Cambridge University Press. doi:10.1017/9781108914062

Examples

```
data(diab)
## maybe str(diab) ; plot(diab) ...
```

`Manpower` *Hospital manpower data*

Description

The hospital manpower data, taken from Myers (1990), table 3.8, are a well-known example of highly collinear data to which ridge regression and various shrinkage and selection methods are often applied.

The data consist of measures taken at 17 U.S. Naval Hospitals and the goal is to predict the required monthly man hours for staffing purposes.

Format

A data frame with 17 observations on the following 6 variables.

- `Hours`: monthly man hours (response variable)
- `Load`: average daily patient load
- `Xray`: monthly X-ray exposures
- `BedDays`: monthly occupied bed days
- `AreaPop`: eligible population in the area in thousands
- `Stay`: average length of patient's stay in days

Details

Myers (1990) indicates his source was "Procedures and Analysis for Staffing Standards Development: Data/Regression Analysis Handbook", Navy Manpower and Material Analysis Center, San Diego, 1979.

Source

Raymond H. Myers (1990). *Classical and Modern Regression with Applications*, 2nd ed., PWS-Kent, pp. 130-133.

References

Donald R. Jensen and Donald E. Ramirez (2012). Variations on Ridge Traces in Regression, *Communications in Statistics - Simulation and Computation*, 41 (2), 265-278.

See Also

bestglm::manpower for the same data, and other analyses

Examples

```
data(Manpower)
mmod <- lm(Hours ~ ., data=Manpower)
vif(mmod)
# ridge regression models, specified in terms of equivalent df
mridge <- ridge(Hours ~ ., data=Manpower, df=seq(5, 3.75, -.25))
vif(mridge)

# univariate ridge trace plots
traceplot(mridge)
traceplot(mridge, X="df")

# bivariate ridge trace plots
plot(mridge, radius=0.25, labels=mridge$df)
pairs(mridge, radius=0.25)


# 3D views
# ellipsoids for Load, Xray & BedDays are nearly 2D
plot3d(mridge, radius=0.2, labels=mridge$df)
# variables in model selected by AIC & BIC
plot3d(mridge, variables=c(2,3,5), radius=0.2, labels=mridge$df)

# plots in PCA/SVD space
mpridge <- pca(mridge)
traceplot(mpridge, X="df")
biplot(mpridge, radius=0.25)
```

---

mlmridge-methods *Methods for Multivariate Ridge Regression Objects*

Description

Experimental. Multivariate ridge regression (class "mlmridge") is new in this version; its functions, arguments and results may change in future releases.

Basic methods for objects of class "mlmridge", returned by ridge() when the response is a matrix.

- `coef()` extracts the coefficients: by default a list with one $p \times q$ matrix for each ridge constant; with `type = "array"`, a 3-d array indexed by `[lambda, predictor, response]`; with `response`, the $\lambda \times p$ matrix for one response, as for a `"ridge"` object.
- `vcov()` extracts the covariance matrices of the coefficients: by default a list with, for each ridge constant, the $pq \times pq$ covariance matrix of the stacked coefficients $\mathrm{vec}(\mathbf{B})$, which is `kronecker(Sigma, cov[[i]])`; with `response`, the list of $p \times p$ matrices for one response.
- `print()` shows the coefficients for each response and the optimal ridge constants given by the HKB, LW and GCV criteria, for each response and pooled over responses.
- `best()` returns the optimal ridge constants as a data frame, with one row for each response and criterion (HKB, LW, GCV), followed by a row for GCV pooled over responses (`response = "(pooled)"`; the pooling method is `object$pool`, see `ridge()`); with `response`, the univariate result for one response, as from `best()` for a `"ridge"` object.

As for a `"ridge"` object, the HKB and LW values are closed-form estimates, while the GCV values (per response and pooled) are the best among the ridge constants supplied in `lambda` (or `df`).

Usage

```
## S3 method for class 'mlmridge'
coef(object, response = NULL, type = c("list", "array"), ...)

## S3 method for class 'mlmridge'
vcov(object, response = NULL, ...)

## S3 method for class 'mlmridge'
best(object, response = NULL, ...)

## S3 method for class 'mlmridge'
print(x, digits = max(5, getOption("digits") - 5), ...)
```

Arguments

`object`, `x`	An object of class `"mlmridge"`
`response`	If not `NULL`, a response (name or number) for which to return the results, in the same form as for a univariate `"ridge"` object; see `as.ridge()`.
`type`	For `coef()` with `response = NULL`: `"list"` (the default) or `"array"`.
`...`	Other arguments, not used.
`digits`	Number of significant digits to print.

Value

`coef()` and `vcov()` return a list, array or matrix, as described above; `print()` returns `x`, invisibly.

`best()` returns a data frame with columns `response` and `criterion` (factors), `k` (the optimal ridge constant) and, if the object contains the singular values of the predictors (`svd = TRUE`, the default), `df` (the equivalent degrees of freedom).

See Also

ridge(), as.ridge()

Examples

```
if (requireNamespace("heplots", quietly = TRUE)) {
  data(schooldata, package = "heplots")
  sridge <- ridge(cbind(reading, mathematics, selfesteem) ~
                  education + occupation + visit + counseling + teacher,
                  data = schooldata, lambda = c(0, 0.1, 0.5, 1, 2, 5))
  sridge
  coef(sridge)[[4]]                       # coefficients for lambda = 1
  coef(sridge, response = "reading")      # lambda x p, for one response
  dim(coef(sridge, type = "array"))
  dim(vcov(sridge)[[1]])                  # pq x pq
  best(sridge)                            # optimal ridge constants, all responses
  best(sridge, response = "selfesteem")   # one response
}
```

---

mlmridge-plots *Plot and Summary Methods for Multivariate Ridge Regression*

Description

Experimental. Multivariate ridge regression (class "mlmridge") is new in this version; its functions, arguments and results may change in future releases.

These methods apply the plots and summaries for univariate ridge regression to one response of a multivariate ridge regression (class "mlmridge"). Each extracts the results for that response with as.ridge() and calls the corresponding method for "ridge" objects:

- traceplot(): univariate ridge trace plots (traceplot()), by default for all responses, one panel each
- plot(): bivariate ridge trace plot of coefficients with covariance ellipses (plot.ridge()); with predictor =, coefficient ellipses for one or more predictors in the space of two responses (see **Details**)
- pairs(): scatterplot matrix of bivariate ridge trace plots (pairs.ridge())
- plot3d(): 3D ridge trace plot (plot3d.ridge())
- biplot(): ridge trace plot with variable vectors (biplot.ridge())
- pca(): transformation to PCA/SVD space (pca.ridge()), returning a "pcaridge" object
- precision(): measures of precision and shrinkage (precision())
- vif(): variance inflation factors (vif.ridge())

Because all responses share the same ridge constants and predictors, the shrinkage factors, effective degrees of freedom, PCA/SVD of the predictors and the variance inflation factors are the same for every response; the coefficients, their covariance matrices (scaled by each response's residual variance) and the HKB, LW and GCV criteria differ.

Usage

```
## S3 method for class 'mlmridge'
traceplot(x, response = NULL, main, layout = TRUE, GCV = TRUE, ...)

## S3 method for class 'mlmridge'
plot(x, response = 1, predictor = NULL, responses = 1:2, main, ...)

## S3 method for class 'mlmridge'
pairs(x, response = 1, ...)

## S3 method for class 'mlmridge'
plot3d(x, response = 1, ...)

## S3 method for class 'mlmridge'
biplot(x, response = 1, main, ...)

## S3 method for class 'mlmridge'
pca(x, response = 1, ...)

## S3 method for class 'mlmridge'
precision(
  object,
  det.fun = c("log", "root"),
  normalize = TRUE,
  response = 1,
  ...
)

## S3 method for class 'mlmridge'
vif(mod, response = 1, ...)
```

Arguments

x, object, mod	An object of class "mlmridge"
response	The response to use: a name, or a number in 1:q. If not given, the first response is used, with a message. For traceplot(), response can name several responses, and the default (NULL) plots all of them, one panel per response.
main	Title for the plot. The default is the name of the response. For traceplot() with several responses, recycled over the panels.
layout	For traceplot() with several responses: if TRUE (the default), set up a multi-panel layout with one panel per response, side by side where possible, and restore the previous layout afterwards. Use layout = FALSE to draw into a layout you have set up yourself, e.g., with par(mfrow =).
GCV	For traceplot(): if TRUE (the default), mark the ridge constant chosen by GCV pooled over responses (x$kGCV.pooled) with a vertical line labeled "GCV" above each panel, in addition to the HKB and LW lines for each response (labeled at the bottom). As for those, this is drawn only when plotting against

lambda, not df.

`...` Other arguments, passed to the method for "ridge" objects.

`predictor` For plot(): if given, plot coefficient ellipses across responses for this predictor, or several predictors (names or numbers), instead of the within-response plot; see **Details**. Further arguments to control the plot (radius, which.lambda, labels, col, fill, fill.alpha, lwd, lty, center.pch, center.cex, ref, ref.col, xlim, ylim, xlab, ylab) have the same meaning as in plot.ridge(). By default, the centers are labeled with the values of λ for one predictor, and with the predictor names for several.

`responses` For plot() with predictor: the two responses (names or numbers) giving the horizontal and vertical axes. Default: the first two.

`det.fun, normalize`
See precision().

Details

Two kinds of plot from plot(). plot() for an "mlmridge" object draws one of two different displays, depending on whether predictor is given:

- **Within one response** (the default, response =): as for a univariate ridge regression (plot.ridge()), the axes are two *predictors*, and each ellipse shows the joint variability of their coefficients for that response, at one value of λ.
- **Across responses** (predictor =): the axes are two *responses* (responses =), and each ellipse shows the joint variability of one predictor's coefficients for those two responses, at one value of λ. The path of the centers traces how the predictor's effects on both responses change together as λ increases. With several predictors, each has its own path, labeled with its name.

For predictor j, the coefficients for all responses (row j of $\mathbf{B}(\lambda)$) have covariance matrix $[\mathbf{V}_\lambda]_{jj}\,\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is the residual covariance matrix of the responses (x$Sigma) and $\mathbf{V}_\lambda$ depends only on the predictors (x$cov). So in the across-responses plot, all the ellipses have the **same shape and orientation**, that of the residual covariance of the two responses: only their centers and sizes change with λ and between predictors.

The across-responses display is the ridge analogue of heplots::coefplot() for a multivariate linear model, which draws confidence ellipses for the coefficients of each predictor in the space of two responses. There are two differences in meaning:

- coefplot() shows each predictor once, for the OLS fit; here each predictor is traced over the values of λ.
- coefplot() scales its ellipses as confidence regions (by level, using an F or χ^2 quantile), so whether an ellipse covers the origin is a test of the coefficients. Here, as in plot.ridge(), the ellipses are scaled by radius (1 = standard "unit" ellipse) and describe the sampling variability around a *biased* estimate; they are not confidence regions for the true coefficients, and covering the origin is not a test.

Value

As for the corresponding method for "ridge" objects.

See Also

ridge(), as.ridge(), mlmridge-methods

Examples

```
if (requireNamespace("heplots", quietly = TRUE)) {
  data(schooldata, package = "heplots")
  lambda <- c(0, 0.1, 0.5, 1, 2, 5)
  sridge <- ridge(cbind(reading, mathematics, selfesteem) ~
                  education + occupation + visit + counseling + teacher,
                  data = schooldata, lambda = lambda)

  # all responses, one panel each
  traceplot(sridge)
  traceplot(sridge, X = "df")
  # one response
  traceplot(sridge, response = "selfesteem")

  # bivariate ridge trace plot for two predictors
  plot(sridge, response = "mathematics", variables = c("occupation", "visit"),
       radius = 0.5)

  # coefficient ellipses across two responses, for one predictor ...
  plot(sridge, predictor = "visit", responses = c("reading", "mathematics"), fill = TRUE)
  # ... and for all predictors, at lambda = 0 and 5
  plot(sridge, predictor = 1:5, which.lambda = c(1, 6), col = c("black", "red"))

  pairs(sridge, response = "reading", radius = 0.5)

  vif(sridge)
  precision(sridge, response = "reading")
}
```

---

pairs.ridge *Scatterplot Matrix of Bivariate Ridge Trace Plots*

Description

Displays all possible pairs of bivariate ridge trace plots for a given set of predictors.

Usage

```
## S3 method for class 'ridge'
pairs(
  x,
  variables,
  radius = 1,
  lwd = 1,
  lty = 1,
```

```
  col = c("black", "red", "darkgreen", "blue", "darkcyan", "magenta", "brown",
    "darkgray"),
  center.pch = 16,
  center.cex = 1.25,
  digits = getOption("digits") - 3,
  diag.cex = 2,
  diag.panel = panel.label,
  fill = FALSE,
  fill.alpha = 0.3,
  ...
)
```

Arguments

x	A ridge object, as fit by ridge()
variables	Predictors in the model to be displayed in the plot: an integer or character vector, giving the indices or names of the variables.
radius	Radius of the ellipse-generating circle for the covariance ellipsoids.
lwd, lty	Line width and line type for the covariance ellipsoids. Recycled as necessary.
col	A numeric or character vector giving the colors used to plot the covariance ellipsoids. Recycled as necessary.
center.pch	Plotting character used to show the bivariate ridge estimates. Recycled as necessary.
center.cex	Size of the plotting character for the bivariate ridge estimates
digits	Number of digits to be displayed as the (min, max) values in the diagonal panels
diag.cex	Character size for predictor labels in diagonal panels
diag.panel	Function to draw diagonal panels. Not yet implemented: just uses internal panel.label to write the variable name and ranges.
fill	Logical vector: Should the covariance ellipsoids be filled? Recycled as necessary.
fill.alpha	Numeric vector: alpha transparency value(s) for filled ellipsoids. Recycled as necessary.
...	Other arguments passed down

Value

None. Used for its side effect of plotting.

Author(s)

Michael Friendly

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

See Also

ridge() for details on ridge regression as implemented here

plot.ridge(), traceplot() for other plotting methods

Examples

```
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(longley.y, longley.X, lambda=lambda)

pairs(lridge, radius=0.5, diag.cex=1.75)

data(prostate)
py <- prostate[, "lpsa"]
pX <- data.matrix(prostate[, 1:8])
pridge <- ridge(py, pX, df=8:1)

pairs(pridge)
```

---

pca *Transform Ridge Estimates to PCA Space*

Description

The function pca.ridge transforms a ridge object from parameter space, where the estimated coefficients are β_k with covariance matrices Σ_k, to the principal component space defined by the right singular vectors, V, of the singular value decomposition of the scaled predictor matrix, X.

In this space, the transformed coefficients are $V\beta_k$, with covariance matrices

$$V\Sigma_k V^T$$

.

This transformation provides alternative views of ridge estimates in low-rank approximations. In particular, it allows one to see where the effects of collinearity typically reside — in the smallest PCA dimensions.

Usage

```
pca(x, ...)
```

Arguments

x	A ridge object, as fit by ridge()
...	Other arguments passed down. Not presently used in this implementation.

Value

An object of class `c("ridge", "pcaridge")`, with the same components as the original `ridge` object.

Author(s)

Michael Friendly

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

See Also

`ridge()`

Examples

```
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(longley.y, longley.X, lambda=lambda)

plridge <- pca(lridge)
traceplot(plridge)
pairs(plridge)
# view in space of smallest singular values
plot(plridge, variables=5:6)
```

---

plot.precision *Plot Shrinkage vs. Variance for Ridge Precision*

Description

This function uses the results of `precision()` to plot a measure of shrinkage of the coefficients in ridge regression against a selected measure of their estimated sampling variance, so as to provide a direct visualization of the tradeoff between bias and precision.

Usage

```
## S3 method for class 'precision'
plot(
  x,
  xvar = "norm.beta",
```

```
  yvar = c("det", "trace", "max.eig"),
  labels = c("lambda", "df"),
  label.cex = 1.25,
  label.prefix,
  criteria = NULL,
  pch = 16,
  cex = 1.5,
  col,
  main = NULL,
  xlab,
  ylab,
  ...
)
```

Arguments

x	A data frame of class "precision" resulting from precision() called on a "ridge" object. Named x only to conform with the plot() generic.
xvar	The character name of the column to be used for the horizontal axis. Typically, this is the normalized sum of squares of the coefficients ("norm.beta") used as a measure of shrinkage / bias.
yvar	The character name of the column to be used for the vertical axis. One of c("det", "trace", "max.eig"). See precision() for definitions of these measures.
labels	The character name of the column to be used for point labels. One of c("lambda", "df").
label.cex	Character size for point labels.
label.prefix	Character or expression prefix for the point labels.
criteria	The vector of optimal shrinkage criteria from the ridge() call to be added as points in the plot.
pch	Plotting character for points
cex	Character size for points
col	Point colors
main	Plot title
xlab	Label for horizontal axis
ylab	Label for vertical axis
...	Other arguments passed to plot().

Value

Returns nothing. Used for the side effect of plotting.

Author(s)

Michael Friendly

See Also

ridge() for details on ridge regression as implemented here. precision() for definitions of the measures

Examples

```
lambda <- c(0, 0.001, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(Employed ~ GNP + Unemployed + Armed.Forces +
                  Population + Year + GNP.deflator,
                data=longley, lambda=lambda)

criteria <- lridge$criteria |> print()

pridge <- precision(lridge) |> print()

plot(pridge)
# also show optimal criteria
plot(pridge, criteria = criteria)

# use degrees of freedom as point labels
plot(pridge, labels = "df")
plot(pridge, labels = "df", label.prefix="df:")
# show the trace measure
plot(pridge, yvar="trace")
```

---

plot.ridge *Bivariate Ridge Trace Plots*

Description

The bivariate ridge trace plot displays 2D projections of the covariance ellipsoids for a set of ridge regression estimates indexed by a ridge tuning constant.

The centers of these ellipses show the bias induced for each parameter, and also how the change in the ridge estimate for one parameter is related to changes for other parameters.

The size and shapes of the covariance ellipses show directly the effect on precision of the estimates as a function of the ridge tuning constant.

plot.pcaridge does these bivariate ridge trace plots for "pcaridge" objects, defaulting to plotting the two smallest components.

Usage

```
## S3 method for class 'ridge'
plot(
  x,
  variables = 1:2,
  radius = 1,
  which.lambda = 1:length(x$lambda),
```

```
  labels = lambda,
  pos = 3,
  cex = 1.2,
  lwd = 2,
  lty = 1,
  xlim,
  ylim,
  col = c("black", "red", "darkgreen", "blue", "darkcyan", "magenta", "brown",
    "darkgray"),
  center.pch = 16,
  center.cex = 1.5,
  fill = FALSE,
  fill.alpha = 0.3,
  ref = TRUE,
  ref.col = gray(0.7),
  ...
)

## S3 method for class 'pcaridge'
plot(x, variables = (p - 1):p, labels = NULL, ...)
```

Arguments

x	A ridge object, as fit by ridge()
variables	Predictors in the model to be displayed in the plot: an integer or character vector of length 2, giving the indices or names of the variables. Defaults to the first two predictors for ridge objects or the *last* two dimensions for pcaridge objects.
radius	Radius of the ellipse-generating circle for the covariance ellipsoids. The default, radius=1 gives a standard "unit" ellipsoid. Typically, values radius<1 gives less cluttered displays.
which.lambda	A vector of indices used to select the values of lambda for which ellipses are plotted. The default is to plot ellipses for all values of lambda in the ridge object.
labels	A vector of character strings or expressions used as labels for the ellipses. Use labels=NULL to suppress these.
pos, cex	Scalars or vectors of positions (relative to the ellipse centers) and character size used to label the ellipses
lwd, lty	Line width and line type for the covariance ellipsoids. Recycled as necessary.
xlim, ylim	X, Y limits for the plot, each a vector of length 2. If missing, the range of the covariance ellipsoids is used.
col	A numeric or character vector giving the colors used to plot the covariance ellipsoids. Recycled as necessary.
center.pch	Plotting character used to show the bivariate ridge estimates. Recycled as necessary.
center.cex	Size of the plotting character for the bivariate ridge estimates

`fill`	Logical vector: Should the covariance ellipsoids be filled? Recycled as necessary.
`fill.alpha`	Numeric vector: alpha transparency value(s) in the range (0, 1) for filled ellipsoids. Recycled as necessary.
`ref`	Logical: whether to draw horizontal and vertical reference lines at 0.
`ref.col`	Color of reference lines.
`...`	Other arguments passed down to `graphics::plot.default()`, e.g., `xlab`, `ylab`, and other graphic parameters.

Value

None. Used for its side effect of plotting.

Author(s)

Michael Friendly

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

See Also

`ridge()` for details on ridge regression as implemented here; `pairs.ridge()`, `traceplot()`, for basic plots.

`pca.ridge()` for transformation of ridge regression estimates to PCA space. `biplot.pcaridge()` and `plot3d.ridge()` for other plotting methods

Examples

```
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lambdaf <- c("", ".005", ".01", ".02", ".04", ".08")
lridge <- ridge(longley.y, longley.X, lambda=lambda)

op <- par(mfrow=c(2,2), mar=c(4, 4, 1, 1)+ 0.1)
for (i in 2:5) {
 plot(lridge, variables=c(1,i), radius=0.5, cex.lab=1.5)
 text(lridge$coef[1,1], lridge$coef[1,i], expression(~widehat(beta)^OLS),
      cex=1.5, pos=4, offset=.1)
 if (i==2) text(lridge$coef[-1,1:2], lambdaf[-1], pos=3, cex=1.25)
}
par(op)

data(prostate)
py <- prostate[, "lpsa"]
pX <- data.matrix(prostate[, 1:8])
```

```
pridge <- ridge(py, pX, df=8:1)

plot(pridge)
plot(pridge, fill=c(TRUE, rep(FALSE,7)))
```

---

plot3d *3D Ridge Trace Plots*

Description

The 3D ridge trace plot displays 3D projections of the covariance ellipsoids for a set of ridge regression estimates indexed by a ridge tuning constant.

The centers of these ellipses show the bias induced for each parameter, and also how the change in the ridge estimate for one parameter is related to changes for other parameters.

The size and shapes of the covariance ellipsoids show directly the effect on precision of the estimates as a function of the ridge tuning constant.

`plot3d.ridge` and `plot3d.pcaridge` differ only in the defaults for the variables plotted.

Usage

```
plot3d(x, ...)

## S3 method for class 'pcaridge'
plot3d(x, variables = (p - 2):p, ...)

## S3 method for class 'ridge'
plot3d(
  x,
  variables = 1:3,
  radius = 1,
  which.lambda = 1:length(x$lambda),
  lwd = 1,
  lty = 1,
  xlim,
  ylim,
  zlim,
  xlab,
  ylab,
  zlab,
  col = c("black", "red", "darkgreen", "blue", "darkcyan", "magenta", "brown",
    "darkgray"),
  labels = lambda,
  ref = TRUE,
  ref.col = gray(0.7),
```

```
  segments = 40,
  shade = TRUE,
  shade.alpha = 0.1,
  wire = FALSE,
  aspect = 1,
  add = FALSE,
  ...
)
```

Arguments

x	A ridge object, as fit by ridge() or a pcaridge object as transformed by pca.ridge()
...	Other arguments passed down
variables	Predictors in the model to be displayed in the plot: an integer or character vector of length 3, giving the indices or names of the variables. Defaults to the first three predictors for ridge objects or the *last* three dimensions for pcaridge objects.
radius	Radius of the ellipse-generating circle for the covariance ellipsoids. The default, radius=1 gives a standard "unit" ellipsoid. Typically, radius<1 gives less cluttered displays.
which.lambda	A vector of indices used to select the values of lambda for which ellipsoids are plotted. The default is to plot ellipsoids for all values of lambda in the ridge object.
lwd, lty	Line width and line type for the covariance ellipsoids. Recycled as necessary.
xlim, ylim, zlim	X, Y, Z limits for the plot, each a vector of length 2. If missing, the range of the covariance ellipsoids is used.
xlab, ylab, zlab	Labels for the X, Y, Z variables in the plot. If missing, the names of the predictors given in variables is used.
col	A numeric or character vector giving the colors used to plot the covariance ellipsoids. Recycled as necessary.
labels	A numeric or character vector giving the labels to be drawn at the centers of the covariance ellipsoids.
ref	Logical: whether to draw horizontal and vertical reference lines at 0. This is not yet implemented.
ref.col	Color of reference lines.
segments	Number of line segments used in drawing each dimension of a covariance ellipsoid.
shade	a logical scalar or vector, indicating whether the ellipsoids should be rendered with rgl::shade3d(). Recycled as necessary.
shade.alpha	a numeric value in the range [0, 1], or a vector of such values, giving the alpha transparency for ellipsoids rendered with shade=TRUE.
wire	a logical scalar or vector, indicating whether the ellipsoids should be rendered with rgl::wire3d(). Recycled as necessary.

aspect	a scalar or vector of length 3, or the character string "iso", indicating the ratios of the x, y, and z axes of the bounding box. The default, aspect=1 makes the bounding box display as a cube approximately filling the display. See rgl::aspect3d() for details.
add	if TRUE, add to the current rgl plot; the default is FALSE.

Value

None. Used for its side-effect of plotting

Note

This is an initial implementation. The details and arguments are subject to change.

Author(s)

Michael Friendly

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

See Also

plot.ridge(), pairs.ridge(), pca.ridge()

Examples

```
lmod <- lm(Employed ~ GNP + Unemployed + Armed.Forces + Population +
                      Year + GNP.deflator, data=longley)
longley.y <- longley[, "Employed"]
longley.X <- model.matrix(lmod)[,-1]

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lambdaf <- c("0", ".005", ".01", ".02", ".04", ".08")
lridge <- ridge(longley.y, longley.X, lambda=lambda)


plot3d(lridge, var=c(1,4,5), radius=0.5)

# view in SVD/PCA space
plridge <- pca(lridge)
plot3d(plridge, radius=0.5)
```

precision *Measures of Precision and Shrinkage for Ridge Regression*

Description

The goal of `precision` is to allow you to study the relationship between shrinkage of ridge regression coefficients and their precision directly by calculating measures of each.

Three measures of (inverse) precision based on the "size" of the covariance matrix of the parameters are calculated. Let $V_k \equiv \text{Var}(\boldsymbol{\beta}_k)$ be the covariance matrix for a given ridge constant, and let $\lambda_i, i = 1, \ldots p$ be its eigenvalues. Then the variance (= 1/precision) measures are:

1. `"det"`: $\log |V_k| = \log \prod \lambda$ (with `det.fun = "log"`, the default) or $|V_k|^{1/p} = (\prod \lambda)^{1/p}$ (with `det.fun = "root"`) measures the linearized volume of the covariance ellipsoid and corresponds conceptually to Wilks' Lambda criterion
2. `"trace"`: $\text{trace}(V_k) = \sum \lambda$ corresponds conceptually to Pillai's trace criterion
3. `"max.eig"`: $\lambda_1 = \max(\lambda)$ corresponds to Roy's largest root criterion.

Two measures of shrinkage are also calculated:

- `norm.beta`: the root mean square of the coefficient vector $\|\boldsymbol{\beta}_k\|$, normalized to a maximum of 1.0 if `normalize == TRUE` (the default).
- `norm.diff`: the root mean square of the difference from the OLS estimate $\|\boldsymbol{\beta}_{\text{OLS}} - \boldsymbol{\beta}_k\|$. This measure is inversely related to `norm.beta`

A plot method, `plot.precision()` facilitates making graphs of these quantities.

Usage

```
precision(object, det.fun, normalize, ...)
```

Arguments

`object`	An object of class `ridge` or `lm`
`det.fun`	Function to be applied to the determinants of the covariance matrices, one of `c("log","root")`.
`normalize`	If `TRUE` the length of the coefficient vector $\boldsymbol{\beta}_k$ is normalized to a maximum of 1.0.
`...`	Other arguments (currently unused)

Value

An object of class `c("precision", "data.frame")` with the following columns:

- `lambda`: The ridge constant
- `df`: The equivalent effective degrees of freedom
- `det`: The `det.fun` function of the determinant of the covariance matrix

- `trace`: The trace of the covariance matrix
- `max.eig`: Maximum eigen value of the covariance matrix
- `norm.beta`: The root mean square of the estimated coefficients, possibly normalized
- `norm.diff`: The root mean square of the difference between the OLS solution (`lambda = 0`) and ridge solutions

Note

Models fit by `lm` and `ridge` use a different scaling for the predictors, so the results of `precision` for an `lm` model will not correspond to those for `ridge` with ridge constant = 0.

Author(s)

Michael Friendly

See Also

`ridge()`, `plot.precision()`

Examples

```
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(longley.y, longley.X, lambda=lambda)

# same, using formula interface
lridge <- ridge(Employed ~ GNP + Unemployed + Armed.Forces + Population + Year + GNP.deflator,
  data=longley, lambda=lambda)

clr <- c("black", rainbow(length(lambda)-1, start=.6, end=.1))
coef(lridge)

(pdat <- precision(lridge))
# plot log |Var(b)| vs. length(beta)
with(pdat, {
 plot(norm.beta, det, type="b",
 cex.lab=1.25, pch=16, cex=1.5, col=clr, lwd=2,
 xlab='shrinkage: ||b|| / max(||b||)',
 ylab='variance: log |Var(b)|')
 text(norm.beta, det, lambda, cex=1.25, pos=c(rep(2,length(lambda)-1),4))
 text(min(norm.beta), max(det), "Variance vs. Shrinkage", cex=1.5, pos=4)
 })

# plot trace[Var(b)] vs. length(beta)
with(pdat, {
 plot(norm.beta, trace, type="b",
 cex.lab=1.25, pch=16, cex=1.5, col=clr, lwd=2,
 xlab='shrinkage: ||b|| / max(||b||)',
 ylab='variance: trace [Var(b)]')
```

```
 text(norm.beta, trace, lambda, cex=1.25, pos=c(2, rep(4,length(lambda)-1)))
# text(min(norm.beta), max(det), "Variance vs. Shrinkage", cex=1.5, pos=4)
 })
```

`prostate` *Prostate Cancer Data*

Description

Data to examine the correlation between the level of prostate-specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy.

Format

A data frame with 97 observations on the following 10 variables.

- `lcavol`: log cancer volume
- `lweight`: log prostate weight
- `age`: in years
- `lbph`: log of the amount of benign prostatic hyperplasia
- `svi`: seminal vesicle invasion
- `lcp`: log of capsular penetration
- `gleason`: a numeric vector
- `pgg45`: percent of Gleason score 4 or 5
- `lpsa`: response
- `train`: a logical vector

Details

This data set came originally from the (now defunct) ElemStatLearn package.

The last column indicates which 67 observations were used as the "training set" and which 30 as the test set, as described on page 48 in the book.

Note

There was an error in this dataset in earlier versions of the package, as indicated in a footnote on page 3 of the second edition of the book. As of version 2012.04-0 this was corrected.

Source

Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E. and Yang, N (1989) Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate II. Radical prostatectomy treated patients, *Journal of Urology*, **16**: 1076–1083.

Examples

```
data(prostate)
str( prostate )
cor( prostate[,1:8] )
prostate <- prostate[, -10]

prostate.mod <- lm(lpsa ~ ., data=prostate)
vif(prostate.mod)

py <- prostate[, "lpsa"]
pX <- data.matrix(prostate[, 1:8])
pridge <- ridge(py, pX, df=8:1)
pridge

# univariate ridge trace plots
traceplot(pridge)
traceplot(pridge, X="df")

# bivariate ridge trace plots
plot(pridge)
pairs(pridge)
```

`ridge` *Ridge Regression Estimates*

Description

The function `ridge` fits linear models by ridge regression, returning an object of class `ridge` designed to be used with the plotting methods in this package.

It is also designed to facilitate an alternative representation of the effects of shrinkage in the space of uncorrelated (PCA/SVD) components of the predictors.

The standard formulation of ridge regression is that it regularizes the estimates of coefficients by adding small positive constants λ to the diagonal elements of $\mathbf{X}^\top\mathbf{X}$ in the least squares solution to achieve a more favorable tradeoff between bias and variance (inverse of precision) of the coefficients.

$$\widehat{\boldsymbol{\beta}}_k^{\text{RR}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

Ridge regression shrinkage can be parameterized in several ways.

- If a vector of `lambda` values is supplied, these are used directly in the ridge regression computations.
- Otherwise, if a vector `df` can be supplied the equivalent values for effective degrees of freedom corresponding to shrinkage, going down from the number of predictors in the model.

In either case, both `lambda` and `df` are returned in the `ridge` object, but the `rownames` of the coefficients are given in terms of `lambda`.

`coef` extracts the estimated coefficients for each value of the shrinkage factor

`vcov` extracts the estimated $p \times p$ covariance matrices of the coefficients for each value of the shrinkage factor.

`best` extracts the optimal shrinkage values according to several criteria: HKB: Hoerl et al. (1975); LW: Lawless & Wang (1976); GCV: Golub et al. (1979)

Usage

```
ridge(y, ...)

## S3 method for class 'formula'
ridge(formula, data, lambda = 0, df, svd = TRUE, contrasts = NULL, ...)

## S3 method for class 'lm'
ridge(y, lambda = 0, df, svd = TRUE, ...)

## Default S3 method:
ridge(
  y,
  X,
  lambda = 0,
  df,
  svd = TRUE,
  pool = c("scaled", "raw", "mahalanobis"),
  ...
)

## S3 method for class 'ridge'
coef(object, ...)

## S3 method for class 'ridge'
print(x, digits = max(5, getOption("digits") - 5), ...)

## S3 method for class 'ridge'
vcov(object, ...)

best(object, ...)

## S3 method for class 'ridge'
best(object, ...)
```

Arguments

`y`	For the default method, a numeric vector containing the response variable, or a matrix of several response variables (see **Multivariate response** below). NAs not allowed. For the `lm` method, a linear model fitted by `stats::lm()`, whose

	response, predictors (including any factor contrasts and transformed terms) and offset are used.
`...`	Other arguments, passed down to methods
`formula`	For the `formula` method, a two-sided formula.
`data`	For the `formula` method, data frame within which to evaluate the formula.
`lambda`	A scalar or vector of ridge constants. A value of 0 corresponds to ordinary least squares.
`df`	A scalar or vector of effective degrees of freedom corresponding to `lambda`
`svd`	If `TRUE` the SVD of the centered and scaled `X` matrix is returned in the `ridge` object.
`contrasts`	a list of contrasts to be used for some or all of factor terms in the formula. See the `contrasts.arg` of `stats::model.matrix.default()`.
`X`	A matrix of predictor variables. NA's not allowed. Should not include a column of 1's for the intercept.
`pool`	For a multivariate response only: how the GCV criterion is pooled over responses to choose a common ridge constant. `"scaled"` (the default) sums each response's residual sum of squares divided by its OLS residual variance, so the result does not depend on the units of the responses; `"raw"` sums the residual sums of squares, as in `lm_multiv_ridge()` in the **VARshrink** package (Lee & Kim); `"mahalanobis"` uses $\mathrm{tr}(\hat{\Sigma}^{-1} R^\top R)$ for the matrix R of residuals.
`x, object`	An object of class `ridge`
`digits`	For the `print` method, the number of digits to print.

Details

If an intercept is present in the model, its coefficient is not penalized. (If you want to penalize an intercept, put in your own constant term and remove the intercept.)

The predictors are centered and scaled to unit mean square (dividing by n), as in `MASS::lm.ridge()`, so the coefficients are on the scale of the standardized predictors.

A number of the methods in the package assume that `lambda` is a vector of shrinkage constants increasing from `lambda[1] = 0`, or equivalently, a vector of `df` decreasing from p.

Choosing the ridge constant. Three criteria are computed. The HKB (Hoerl et al., 1975) and LW (Lawless & Wang, 1976) values are closed-form estimates from the OLS fit, $k_{HKB} = (p-2)\, s^2 / \hat{\boldsymbol{\beta}}^\top \hat{\boldsymbol{\beta}}$ and $k_{LW} = (p-2)\, s^2\, n / \hat{\mathbf{y}}^\top \hat{\mathbf{y}}$, where s^2 is the residual mean square, so they need not be among the values of `lambda`. The GCV value (Golub et al., 1979) is the value of `lambda` (among those supplied) that minimizes the generalized cross-validation criterion.

Because of the factor $p-2$, the HKB and LW estimates are defined only for $p \geq 3$ predictors (they would be 0 when $p = 2$ and negative when $p = 1$). For $p < 3$, `kHKB` and `kLW` are `NA`, and the plot methods do not mark them.

Value

A list with the following components:

- `lambda`: The vector of ridge constants

- df: The vector of effective degrees of freedom corresponding to lambda
- coef: The matrix of estimated ridge regression coefficients
- scales: scalings used on the X matrix
- kHKB: HKB estimate of the ridge constant
- kLW: L-W estimate of the ridge constant
- GCV: vector of GCV values
- kGCV: value of lambda with the minimum GCV
- criteria: Collects the criteria kHKB, kLW, and kGCV in a named vector

If svd==TRUE (the default), the following are also included:

- svd.D: Singular values of the svd of the scaled X matrix
- svd.U: Left singular vectors of the svd of the scaled X matrix. Rows correspond to observations and columns to dimensions.
- svd.V: Right singular vectors of the svd of the scaled X matrix. Rows correspond to variables and columns to dimensions.

Multivariate response (experimental). If the response is a matrix with more than one column (e.g., cbind(Y1, Y2) ~ x1 + x2 in the formula, a multivariate lm() fit, or a matrix y), all responses are fitted with the same ridge constants and the result is an object of class "mlmridge". Its components follow those above, but:

- coef is a list, with one $p \times q$ matrix of coefficients for each value of lambda.
- cov is a list of the $p \times p$ matrices that depend only on X; the covariance matrix of the coefficients for response k is Sigma[k, k] * cov[[i]].
- Sigma is the residual covariance matrix of the responses, from the OLS fit.
- mse and GCV are matrices, with one column per response; kHKB, kLW and kGCV are vectors, and criteria is a matrix, with one row per response.
- GCV.pooled and kGCV.pooled give the GCV criterion pooled over responses (see pool) and the value of lambda minimizing it.

Use as.ridge() to extract the results for one response as a "ridge" object. The estimator, $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{Y}$, is that of Breiman & Friedman (1997, eq. 6.3); the implementation was adapted from lm_multiv_ridge() in the **VARshrink** package (Lee & Kim). For the same scaling of X, lambda here is n times the lambda used there. This extension is experimental: its interface and results may change in future releases.

A data.frame with one row for each of the HKB, LW, and GCV criteria

Author(s)

Michael Friendly

References

Hoerl, A. E., Kennard, R. W., and Baldwin, K. F. (1975), "Ridge Regression: Some Simulations," *Communications in Statistics*, 4, 105-123.

Lawless, J.F., and Wang, P. (1976), "A Simulation Study of Ridge and Other Regression Estimators," *Communications in Statistics*, 5, 307-323.

Golub G.H., Heath M., Wahba G. (1979) Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21**:215–223. doi:10.2307/1268518

Breiman, L. and Friedman, J. H. (1997). Predicting Multivariate Responses in Multiple Linear Regression. *Journal of the Royal Statistical Society, Series B*, **59**(1), 3-54. doi:10.1111/1467-9868.00054

Lee, N. and Kim, S.-H. VARshrink: Shrinkage Estimation Methods for Vector Autoregressive Models. R package. doi:10.32614/CRAN.package.VARshrink

Imdad Ullah, M., Aslam, M., and Altaf, S. (2018). lmridge: A Comprehensive R Package for Ridge Regression. *The R Journal*, **10**(2), 326–343. doi:10.32614/RJ2018060

See Also

`MASS::lm.ridge()` for other implementations of ridge regression. The **lmridge** package (Imdad Ullah et al., 2018) provides many estimators of the ridge constant and other ridge statistics.

`traceplot()`, `plot.ridge()`, `pairs.ridge()`, `plot3d.ridge()`, for 1D, 2D, 3D plotting methods

`pca.ridge()`, `biplot.ridge()`, `biplot.pcaridge()` for views in PCA/SVD space

`precision.ridge()` for measures of shrinkage and precision

Examples

```
#\donttest{
# Longley data, using number Employed as response
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(longley.y, longley.X, lambda=lambda)

# same, using formula interface
lridge <- ridge(Employed ~ GNP + Unemployed + Armed.Forces + Population + Year + GNP.deflator,
  data=longley, lambda=lambda)

# same, from a fitted lm() model
longley.lm <- lm(Employed ~ GNP + Unemployed + Armed.Forces + Population + Year + GNP.deflator,
  data=longley)
lridge <- ridge(longley.lm, lambda=lambda)

coef(lridge)

# standard trace plot
traceplot(lridge)
```

```
# plot vs. equivalent df
traceplot(lridge, X="df")
pairs(lridge, radius=0.5)
#}


data(prostate)
py <- prostate[, "lpsa"]
pX <- data.matrix(prostate[, 1:8])
pridge <- ridge(py, pX, df=8:1)
pridge

plot(pridge)
pairs(pridge)
traceplot(pridge)
traceplot(pridge, X="df")


# Hospital manpower data from Table 3.8 of Myers (1990)
data(Manpower)
str(Manpower)

mmod <- lm(Hours ~ ., data=Manpower)
vif(mmod)
# ridge regression models, specified in terms of equivalent df
mridge <- ridge(Hours ~ ., data=Manpower, df=seq(5, 3.75, -.25))
vif(mridge)

# univariate ridge trace plots
traceplot(mridge)
traceplot(mridge, X="df")


# bivariate ridge trace plots
plot(mridge, radius=0.25, labels=mridge$df)
pairs(mridge, radius=0.25)

# 3D views
# ellipsoids for Load, Xray & BedDays are nearly 2D
plot3d(mridge, radius=0.2, labels=mridge$df)
# variables in model selected by AIC & BIC
plot3d(mridge, variables=c(2,3,5), radius=0.2, labels=mridge$df)

# plots in PCA/SVD space
mpridge <- pca(mridge)
traceplot(mpridge, X="df")
biplot(mpridge, radius=0.25)
```

`traceplot` *Univariate Ridge Trace Plots*

Description

The `traceplot` function extends and simplifies the univariate ridge trace plots for ridge regression provided in the `plot` method for `MASS::lm.ridge()`

Usage

```
traceplot(x, ...)

## S3 method for class 'ridge'
traceplot(
  x,
  X = c("lambda", "df"),
  col = c("black", "red", "darkgreen", "blue", "darkcyan", "magenta", "brown",
    "darkgray"),
  pch = c(15:18, 7, 9, 12, 13),
  xlab,
  ylab = "Coefficient",
  xlim,
  ylim,
  ...
)
```

Arguments

`x`	A `ridge` object, as fit by `ridge()`
`...`	Other arguments passed to `graphics::matplot()`
`X`	What to plot as the horizontal coordinate, one of `c("lambda", "df")`
`col`	A numeric or character vector giving the colors used to plot the ridge trace curves. Recycled as necessary.
`pch`	Vector of plotting characters used to plot the ridge trace curves. Recycled as necessary.
`xlab`	Label for horizontal axis
`ylab`	Label for vertical axis
`xlim`, `ylim`	x, y limits for the plot. You may need to adjust these to allow for the variable labels.

Details

For ease of interpretation, the variables are labeled at the side of the plot (left, right) where the coefficient estimates are expected to be most widely spread. If `xlim` is not specified, the range of the `X` variable is extended slightly to accommodate the variable names.

Value

None. Used for its side effect of plotting.

Author(s)

Michael Friendly

References

Friendly, M. (2013). The Generalized Ridge Trace Plot: Visualizing Bias *and* Precision. *Journal of Computational and Graphical Statistics*, **22**(1), 50-68, doi:10.1080/10618600.2012.681237.

Hoerl, A. E. and Kennard R. W. (1970). "Ridge Regression: Applications to Nonorthogonal Problems", *Technometrics*, 12(1), 69-82.

See Also

`ridge()` for details on ridge regression as implemented here

`plot.ridge()`, `pairs.ridge()` for other plotting methods

Examples

```
longley.y <- longley[, "Employed"]
longley.X <- data.matrix(longley[, c(2:6,1)])

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(longley.y, longley.X, lambda=lambda)

traceplot(lridge)
#abline(v=lridge$kLW, lty=3)
#abline(v=lridge$kHKB, lty=3)
#text(lridge$kLW, -3, "LW")
#text(lridge$kHKB, -3, "HKB")

traceplot(lridge, X="df")
```

---

trans.colors *Make Colors Transparent*

Description

Takes a vector of colors (as color names or rgb hex values) and adds a specified alpha transparency to each.

Usage

```
trans.colors(col, alpha = 0.5, names = NULL)
```

Arguments

col	A character vector of colors, either as color names or rgb hex values
alpha	alpha transparency value(s) to apply to each color (0 means fully transparent and 1 means opaque)
names	optional character vector of names for the colors

Details

Colors (col) and alpha need not be of the same length. The shorter one is replicated to make them of the same length.

Value

A vector of color values of the form "#rrggbbaa"

Author(s)

Michael Friendly

See Also

grDevices::col2rgb(), grDevices::rgb(),

Examples

```
trans.colors(palette(), alpha=0.5)

# alpha can be vectorized
trans.colors(palette(), alpha=seq(0, 1, length=length(palette())))

# lengths need not match: shorter one is repeated as necessary
trans.colors(palette(), alpha=c(.1, .2))

trans.colors(colors()[1:20])

# single color, with various alphas
trans.colors("red", alpha=seq(0,1, length=5))
# assign names
trans.colors("red", alpha=seq(0,1, length=5), names=paste("red", 1:5, sep=""))
```

`vif.ridge` *Variance Inflation Factors for Ridge Regression*

Description

The function `vif.ridge` calculates variance inflation factors for the predictors in a set of ridge regression models indexed by the tuning/shrinkage factor, returning one row for each value of the λ parameter.

Variance inflation factors are calculated using the simplified formulation in Fox & Monette (1992).

The `plot.vif.ridge` method plots variance inflation factors for a `"vif.ridge"` object in a similar style to what is provided by `traceplot()`. That is, it plots the VIF for each coefficient in the model against either the ridge λ tuning constant or it's equivalent effective degrees of freedom.

Usage

```
## S3 method for class 'ridge'
vif(mod, ...)

## S3 method for class 'vif.ridge'
print(x, digits = max(4, getOption("digits") - 5), ...)

## S3 method for class 'vif.ridge'
plot(
  x,
  X = c("lambda", "df"),
  Y = c("vif", "sqrt"),
  col = c("black", "red", "darkgreen", "blue", "darkcyan", "magenta", "brown",
    "darkgray"),
  pch = c(15:18, 7, 9, 12, 13),
  xlab,
  ylab,
  xlim,
  ylim,
  ...
)
```

Arguments

`mod`	A `"ridge"` object computed by `ridge()`
`...`	Other arguments passed to methods
`x`	A `ridge` object, as fit by `ridge()`
`digits`	Number of digits to display in the `print` method
`X`	What to plot as the horizontal coordinate, one of `c("lambda", "df")`
`Y`	What to plot as the vertical coordinate, one of `c("vif", "sqrt")`, where the latter plots $\sqrt{VIF}$.

col	A numeric or character vector giving the colors used to plot the ridge trace curves. Recycled as necessary.
pch	Vector of plotting characters used to plot the ridge trace curves. Recycled as necessary.
xlab	Label for horizontal axis
ylab	Label for vertical axis
xlim, ylim	x, y limits for the plot. You may need to adjust these to allow for the variable labels.

Value

`vif` returns a `"vif.ridge"` object, which is a list of four components

- `vif`: a data frame of the same size and shape as `coef(mod)`. The columns correspond to the predictors in the model and the rows correspond to the values of `lambda` in ridge estimation.
- `lambda`: the vector of ridge constants from the original call to `ridge()`
- `df`: the vector of effective degrees of freedom corresponding to `lambda`
- `criteria`: the optimal values of `lambda`

Author(s)

Michael Friendly

References

Fox, J. and Monette, G. (1992). Generalized collinearity diagnostics. *JASA*, **87**, 178-183, doi:10.1080/01621459.1992.10475190.

See Also

`car::vif()`, `precision()`

Examples

```
data(longley)
lmod <- lm(Employed ~ GNP + Unemployed + Armed.Forces + Population +
                      Year + GNP.deflator, data=longley)
vif(lmod)

lambda <- c(0, 0.005, 0.01, 0.02, 0.04, 0.08)
lridge <- ridge(Employed ~ GNP + Unemployed + Armed.Forces +
                           Population + Year + GNP.deflator,
               data=longley, lambda=lambda)

coef(lridge)

# get VIFs for the shrunk estimates
vridge <- vif(lridge)
vridge
```

```
names(vridge)



# plot VIFs
pch <- c(15:18, 7, 9)
clr <- c("black", rainbow(5, start=.6, end=.1))

plot(vridge,
     col=clr, pch=pch, cex = 1.2,
     xlim = c(-0.02, 0.08))

plot(vridge, X = "df",
     col=clr, pch=pch, cex = 1.2,
     xlim = c(4, 6.5))

# Better to plot sqrt(VIF). Plot against degrees of freedom
plot(vridge, X = "df", Y="sqrt",
     col=clr, pch=pch, cex = 1.2,
     xlim = c(4, 6.5))
```

Index

∗ **color**
trans.colors, 40
∗ **datasets**
Acetylene, 4
Detroit, 10
diab, 12
Manpower, 13
prostate, 32
∗ **dplot**
pca, 21
∗ **hplot**
biplot.pcaridge, 6
contourf, 8
pairs.ridge, 19
plot.ridge, 24
plot3d, 27
traceplot, 39
∗ **models**
precision, 30
ridge, 33
vif.ridge, 42
∗ **multivariate**
pca, 21
∗ **package**
genridge-package, 2
∗ **regression**
precision, 30
ridge, 33
vif.ridge, 42
∗ **ridge**
diab, 12
_PACKAGE *(*genridge-package*)*, 2

Acetylene, 4
as.ridge, 5
as.ridge(), *15, 16*, *19*, *36*

best *(*ridge*)*, 33
best(), *15*
best.mlmridge *(*mlmridge-methods*)*, 14

bestglm::manpower, *14*
biplot.mlmridge *(*mlmridge-plots*)*, 16
biplot.pcaridge, 6
biplot.pcaridge(), *26*, *37*
biplot.ridge *(*biplot.pcaridge*)*, 6
biplot.ridge(), *16*, *37*

car::vif(), *43*
coef.mlmridge *(*mlmridge-methods*)*, 14
coef.ridge *(*ridge*)*, 33
contourf, 8

Detroit, 10
diab, 12

genridge *(*genridge-package*)*, 2
genridge-package, 2
graphics::contour(), *8, 9*
graphics::contour.default(), *9*
graphics::matplot(), *39*
graphics::plot.default(), *26*
grDevices::col2rgb(), *41*
grDevices::contourLines(), *9*
grDevices::rgb(), *41*

heplots::coefplot(), *18*

lattice::contourplot(), *9*

Manpower, 13
MASS::lm.ridge(), *3*, *35*, *37*, *39*
mlmridge-methods, 14, *19*
mlmridge-plots, 16

pairs.mlmridge *(*mlmridge-plots*)*, 16
pairs.ridge, 19
pairs.ridge(), *3*, *5*, *16*, *26*, *29*, *37*, *40*
pca, 21
pca(), *5*
pca.mlmridge *(*mlmridge-plots*)*, 16
pca.ridge(), *3*, *6*, *8*, *16*, *26*, *28, 29*, *37*

plot(), *23*
plot.mlmridge *(*mlmridge-plots*)*, 16
plot.pcaridge *(*plot.ridge*)*, 24
plot.pcaridge(), *7*
plot.precision, 22
plot.precision(), *30, 31*
plot.ridge, 24
plot.ridge(), *3*, *5*, *7, 8*, *16*, *18*, *21*, *29*, *37*, *40*
plot.vif.ridge *(*vif.ridge*)*, 42
plot3d, 27
plot3d.mlmridge *(*mlmridge-plots*)*, 16
plot3d.ridge(), *3*, *5*, *16*, *26*, *37*
precision, 30
precision(), *3*, *5*, *16*, *18*, *22–24*, *43*
precision.mlmridge *(*mlmridge-plots*)*, 16
precision.ridge(), *37*
print.mlmridge *(*mlmridge-methods*)*, 14
print.ridge *(*ridge*)*, 33
print.vif.ridge *(*vif.ridge*)*, 42
prostate, 32

rgl::aspect3d(), *29*
rgl::shade3d(), *28*
rgl::wire3d(), *28*
ridge, 33
ridge(), *3*, *5*, *12*, *14–16*, *19–26*, *28*, *31*, *39, 40*, *42, 43*

stats::lm(), *34*
stats::model.matrix.default(), *35*

traceplot, 39
traceplot(), *3*, *5*, *16*, *21*, *26*, *37*, *42*
traceplot.mlmridge *(*mlmridge-plots*)*, 16
trans.colors, 40

vcov.mlmridge *(*mlmridge-methods*)*, 14
vcov.ridge *(*ridge*)*, 33
vif.mlmridge *(*mlmridge-plots*)*, 16
vif.ridge, 42
vif.ridge(), *3*, *5*, *16*