

Package: mplot (via r-universe)

October 4, 2026

Type Package

Title Graphical Model Stability and Variable Selection Procedures

Version 1.1.0

Date 2026-10-03

Description Model stability and variable inclusion plots [Mueller and Welsh (2010, <[doi:10.1111/j.1751-5823.2010.00108.x](https://doi.org/10.1111/j.1751-5823.2010.00108.x)>); Murray, Heritier and Mueller (2013, <[doi:10.1002/sim.5855](https://doi.org/10.1002/sim.5855)>)] as well as the adaptive fence [Jiang et al. (2008, <[doi:10.1214/07-AOS517](https://doi.org/10.1214/07-AOS517)>); Jiang et al. (2009, <[doi:10.1016/j.spl.2008.10.014](https://doi.org/10.1016/j.spl.2008.10.014)>)] for linear and generalised linear models.

License GPL (>= 2)

Suggests knitr, mvoutlier, glmulti, rmarkdown, MASS, testthat (>= 3.0.0), RhpcBLASctl

Imports leaps, parallel, bestglm, future, furrr, plyr, shinydashboard, shiny, glmnet, graphics, stats, googleVis, ggplot2, scales, dplyr, tidyr, rlang

URL <https://garhtarr.github.io/mplot/>,
<https://github.com/garhtarr/mplot>

Depends R (>= 4.1.0)

Roxygen list(markdown = TRUE)

LazyData TRUE

Encoding UTF-8

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev zlib1g-dev

Repository <https://garhtarr.r-universe.dev>

Date/Publication 2026-10-04 10:39:15 UTC

RemoteUrl <https://github.com/garhtarr/mplot>

RemoteRef HEAD

RemoteSha 1a4e4341204b1fc3e6ff7aa224a0e75fb9ca2116

Contents

mplot-package	2
af	3
artificialeg	5
bglmnet	6
bodyfat	8
diabetes	9
fev	10
glmfence	11
lmfence	11
mplot	12
plot.af	13
plot.bglmnet	15
plot.vis	17
print.af	21
print.vis	21
summary.af	22
vis	22
wallabies	24
Index	26

mplot-package	<i>mplot: Graphical Model Stability and Variable Selection Procedures</i>
---------------	---

Description

Model stability and variable inclusion plots [Mueller and Welsh (2010, doi:10.1111/j.17515823.2010.00108.x); Murray, Heritier and Mueller (2013, doi:10.1002/sim.5855)] as well as the adaptive fence [Jiang et al. (2008, doi:10.1214/07AOS517); Jiang et al. (2009, doi:10.1016/j.spl.2008.10.014)] for linear and generalised linear models.

Author(s)

- Maintainer:** Garth Tarr <garth.tarr@gmail.com> (ORCID)
- Authors:
- Garth Tarr <garth.tarr@gmail.com> (ORCID)
 - Samuel Mueller <samuel.mueller@sydney.edu.au> (ORCID)
 - Alan H Welsh <alan.welsh@anu.edu.au> (ORCID)

References

Tarr G, Mueller S and Welsh AH (2018). mplot: An R Package for Graphical Model Stability and Variable Selection Procedures. Journal of Statistical Software, 83(9), pp. 1-28. doi: 10.18637/jss.v083.i09

See Also

Useful links:

- <https://garhtarr.github.io/mplot/>
- <https://github.com/garhtarr/mplot>

af

The adaptive fence procedure

Description

This function implements the adaptive fence procedure to first find the optimal cstar value and then finds the corresponding best model as described in Jiang et. al. (2009) with some practical modifications.

Usage

```
af(
  mf,
  B = 60,
  n.c = 20,
  initial.stepwise = FALSE,
  force.in = NULL,
  cores,
  nvmax,
  c.max,
  screen = FALSE,
  seed = NULL,
  ...
)
```

Arguments

mf	a fitted 'full' model, the result of a call to lm or glm (and in the future lme or lmer).
B	number of bootstrap replications at each fence boundary value
n.c	number of boundary values to be considered
initial.stepwise	logical. Performs an initial stepwise procedure to look for the range of model sizes where attention should be focussed. See details for implementation.
force.in	the names of variables that should be forced into all estimated models
cores	number of cores to be used when parallel processing the bootstrap
nvmax	size of the largest model that can still be considered as a viable candidate. Included for performance reasons but if it is an active constraint it could lead to misleading results.

<code>c.max</code>	manually specify the upper boundary limit. Only applies when <code>initial.stepwise=FALSE</code> .
<code>screen</code>	logical, whether or not to perform an initial screen for outliers. Highly experimental, use at own risk. Default = <code>FALSE</code> .
<code>seed</code>	random seed for reproducible results. When set, results are reproducible across runs regardless of <code>cores</code> value. Uses <code>future</code> 's native seeding via <code>furrr_options(seed = TRUE)</code> .
<code>...</code>	further arguments (currently unused)

Details

The initial stepwise procedure performs forward stepwise model selection using the AIC and backward stepwise model selection using BIC. In general the backwise selection via the more conservative BIC will tend to select a smaller model than that of the forward selection AIC approach. The size of these two models is found, and we go two dimensions smaller and larger to estimate a sensible range of `c` values over which to perform a parametric bootstrap.

This procedure can take some time. It is recommended that you start with a relatively small number of bootstrap samples (`B`) and grid of boundary values (`n.c`) and increase both as required.

If you use `initial.stepwise=TRUE` then in general you will need a smaller grid of boundary values than if you select `initial.stepwise=FALSE`. It can be useful to check `initial.stepwise=FALSE` with a small number of bootstrap replications over a sparse grid to ensure that the `initial.stepwise=TRUE` has landed you in a reasonable region.

The `best.only=FALSE` option when plotting the results of the adaptive fence is a modification to the adaptive fence procedure which considers all models at a particular size that pass the fence hurdle when calculating the p^* values. In particular, for each value of `c` and at each bootstrap replication, if a candidate model is found that passes the fence, then we look to see if there are any other models of the same size that also pass the fence. If no other models of the same size pass the fence, then that model is allocated a weight of 1. If there are two models that pass the fence, then the best model is allocated a weight of 1/2. If three models pass the fence, the best model gets a weight of 1/3, and so on. After `B` bootstrap replications, we aggregate the weights by summing over the various models. The p^* value is the maximum aggregated weight divided by the number of bootstrap replications. This correction penalises the probability associated with the best model if there were other models of the same size that also passed the fence hurdle. The rationale being that if a model has no redundant variables then it will be the only model at that size that passes the fence over a range of values of `c`. The result is more pronounced peaks which can help to determine the location of the correct peak and identify the optimal c^* .

See `?plot.af` or `help("plot.af")` for details of the plot method associated with the result.

The `cores` argument controls parallelization backend: when `cores > 1`, a `future::multisession` plan is registered for the duration of the function. The caller's existing future plan is preserved and restored on exit.

References

- Jiang J., Nguyen T., Sunil Rao J. (2009), A simplified adaptive fence procedure, *Statistics & Probability Letters*, 79(5):625-629. doi: 10.1016/j.spl.2008.10.014
- Jiang J., Sunil Rao J., Gu Z, Nguyen T. (2008), Fence methods for mixed model selection, *Annals of Statistics*, 36(4):1669-1692. doi: 10.1214/07-AOS517

See Also[plot.af](#)Other fence: [glmfence\(\)](#), [lmfence\(\)](#)**Examples**

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = data.frame(y,x1,x2,x3,x4,x5)
lm1 = lm(y ~ ., data = dat)

## Not run:
af1 = af(lm1, initial.stepwise = TRUE, seed = 1)
summary(af1)
plot(af1)

## End(Not run)
```

artificialeg*Artificial example*

Description

An artificial data set which causes stepwise regression procedures to select a non-parsimonious model. The true model is a simple linear regression of y against x_8 .

Usage

```
data(artificialeg)
```

Format

A data frame with 50 observations on 10 variables.

Details

Inspired by the pathoeg data set in the MPV package.

Examples

```
data(artificialeg)
full.mod = lm(y~.,data=artificialeg)
step(full.mod)
# generating model
n=50
set.seed(8) # a seed of 2 also works
x1 = rnorm(n,0.22,2)
x7 = 0.5*x1 + rnorm(n,0,sd=2)
x6 = -0.75*x1 + rnorm(n,0,3)
x3 = -0.5-0.5*x6 + rnorm(n,0,2)
x9 = rnorm(n,0.6,3.5)
x4 = 0.5*x9 + rnorm(n,0,sd=3)
x2 = -0.5 + 0.5*x9 + rnorm(n,0,sd=2)
x5 = -0.5*x2+0.5*x3+0.5*x6-0.5*x9+rnorm(n,0,1.5)
x8 = x1 + x2 -2*x3 - 0.3*x4 + x5 - 1.6*x6 - 1*x7 + x9 +rnorm(n,0,0.5)
y = 0.6*x8 + rnorm(n,0,2)
artificialeg = round(data.frame(x1,x2,x3,x4,x5,x6,x7,x8,x9,y),1)
```

bglmnet

Model stability and variable importance plots for glmnet

Description

Model stability and variable importance plots for glmnet

Usage

```
bglmnet(
  mf,
  nlambda = 100,
  lambda = NULL,
  B = 100,
  penalty.factor,
  screen = FALSE,
  redundant = TRUE,
  cores = NULL,
  force.in = NULL,
  seed = NULL
)
```

Arguments

mf	a fitted 'full' model, the result of a call to lm or glm.
nlambda	how many penalty values to consider. Default = 100.
lambda	manually specify the penalty values (optional).
B	number of bootstrap replications

<code>penalty.factor</code>	Separate penalty factors can be applied to each coefficient. This is a number that multiplies lambda to allow differential shrinkage. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is 1 for all variables (and implicitly infinity for variables listed in <code>exclude</code>). Note: the penalty factors are internally rescaled to sum to <code>nvars</code> , and the lambda sequence will reflect this change.
<code>screen</code>	logical, whether or not to perform an initial screen for outliers. Highly experimental, use at own risk. Default = FALSE.
<code>redundant</code>	logical, whether or not to add a redundant variable. Default = TRUE.
<code>cores</code>	number of cores to be used when parallel processing the bootstrap. Defaults to one less than the number of available cores (minimum 1).
<code>force.in</code>	the names of variables that should be forced into all estimated models. (Not yet implemented.)
<code>seed</code>	random seed for reproducible results. When set, results are reproducible across runs regardless of <code>cores</code> value. Uses <code>future</code> 's native seeding via <code>furrr_options(seed = seed)</code> .

Details

The result of this function is essentially just a list. The supplied plot method provides a way to visualise the results.

The `cores` argument controls parallelization backend: when `cores > 1`, a `future::multisession` plan is registered for the duration of the function and the bootstrap loop over `B` replications is run in parallel (the inner loop over `nlambda` penalty values remains sequential within each worker). The caller's existing future plan is preserved and restored on exit.

See Also

[plot.bglmnet](#)

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = data.frame(y, x1, x2, x3, x4, x5)
lm1 = lm(y ~ ., data = dat)

## Not run:
bg1 = bglmnet(lm1, seed = 1)
# plot(bg1, which = "boot_size", interactive = TRUE)
plot(bg1, which = "boot_size", interactive = FALSE)
# plot(bg1, which = "vip", interactive = TRUE)
```

```
plot(bg1, which = "vip", interactive = FALSE)

## End(Not run)
```

bodyfat

Body fat data set

Description

A data frame with 128 observations on 15 variables.

Usage

```
data(bodyfat)
```

Format

A data frame with 128 observations on 15 variables.

Id Identifier

Bodyfat Bodyfat percentage

Age Age (years)

Weight Weight (kg)

Height Height (inches)

Neck Neck circumference (cm)

Chest Chest circumference (cm)

Abdo Abdomen circumference (cm) "at the umbilicus and level with the iliac crest"

Hip Hip circumference (cm)

Thigh Thigh circumference (cm)

Knee Knee circumference (cm)

Ankle Ankle circumference (cm)

Bic Extended biceps circumference (cm)

Fore Forearm circumference (cm)

Wrist Wrist circumference (cm) "distal to the styloid processes"

Details

A subset of the 252 observations available in the mfp package. The selected observations avoid known high leverage points and outliers. The unused points from the data set could be used to validate selected models.

References

Johnson W (1996, Vol 4). Fitting percentage of body fat to simple body measurements. Journal of Statistics Education. Bodyfat data retrieved from <http://www.amstat.org/publications/jse/v4n1/datasets.johnson.html>
An expanded version is included in the mfp R package.

Examples

```
data(bodyfat)
full.mod = lm(Bodyfat~.,data=subset(bodyfat,select=-Id))
```

diabetes

Blood and other measurements in diabetics

Description

The diabetes data frame has 442 rows and 11 columns. These are the data used in Efron et al. (2004).

Usage

```
data(diabetes)
```

Format

A data frame with 442 observations on 11 variables.

age Age

sex Gender

bmi Body mass index

map Mean arterial pressure (average blood pressure)

tc Total cholesterol (mg/dL)? Desirable range: below 200 mg/dL

ldl Low-density lipoprotein ("bad" cholesterol)? Desirable range: below 130 mg/dL

hdl High-density lipoprotein ("good" cholesterol)? Desirable range: above 40 mg/dL

tch Blood serum measurement

ltg Blood serum measurement

glu Blood serum measurement (glucose?)

y A quantitative measure of disease progression one year after baseline

Details

Data sourced from <http://web.stanford.edu/~hastie/Papers/LARS>

References

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., (2004). Least angle regression. The Annals of Statistics 32(2) 407-499. DOI: 10.1214/0090536040000000067

Examples

```
data(diabetes)
full.mod = lm(y~.,data=diabetes)
```

fev

Forced Expiratory Volume

Description

This data set consists of 654 observations on youths aged 3 to 19 from East Boston recorded during the middle to late 1970's. Forced expiratory volume (FEV), a measure of lung capacity, is the variable of interest. Age and height are two continuous predictors. Sex and smoke are two categorical predictors.

Usage

```
data(fev)
```

Format

A data frame with 654 observations on 5 variables.

age Age (years)

fev Forced expiratory volume (liters). Roughly the amount of air an individual can exhale in the first second of a forceful breath.

height Height (inches).

sex Female is 0. Male is 1.

smoke A binary variable indicating whether or not the youth smokes. Nonsmoker is 0. Smoker is 1.

Details

Copies of this data set can also be found in the `coneproj` and `tmle` packages.

References

Tager, I. B., Weiss, S. T., Rosner, B., and Speizer, F. E. (1979). Effect of parental cigarette smoking on pulmonary function in children. *American Journal of Epidemiology*, **110**, 15-26.

Rosner, B. (1999). *Fundamentals of Biostatistics*, 5th Ed., Pacific Grove, CA: Duxbury.

Kahn, M.J. (2005). An Exhalent Problem for Teaching Statistics. *Journal of Statistics Education*, **13**(2). <http://www.amstat.org/publications/jse/v13n2/datasets.kahn.html>

Examples

```
data(fev)
full.mod = lm(fev~.,data=fev)
step(full.mod)
```

glmfence

*The fence procedure for generalised linear models***Description**

This function implements the fence procedure to find the best generalised linear model.

Usage

```
glmfence(mf, cstar, nvmax, adaptive = TRUE, trace = TRUE, ...)
```

Arguments

mf	an object of class <code>glm</code> specifying the full model.
cstar	the boundary of the fence, typically found through bootstrapping.
nvmax	the maximum number of variables that will be considered in the model.
adaptive	logical. If TRUE the boundary of the fence is given by cstar. Otherwise, it the original (non-adaptive) fence is performed where the boundary is $c^* \hat{\sigma}_{M, \tilde{M}}$.
trace	logical. If TRUE the function prints out its progress as it iterates up through the dimensions.
...	further arguments (currently unused)

References

Jiming Jiang, Thuan Nguyen, J. Sunil Rao, A simplified adaptive fence procedure, Statistics & Probability Letters, Volume 79, Issue 5, 1 March 2009, Pages 625-629, <http://dx.doi.org/10.1016/j.spl.2008.10.014>.

See Also

`af`, `lmfence`

Other fence: `af()`, `lmfence()`

lmfence

*The fence procedure for linear models***Description**

This function implements the fence procedure to find the best linear model.

Usage

```
lmfence(mf, cstar, nvmax, adaptive = TRUE, trace = TRUE, force.in = NULL, ...)
```

Arguments

<code>mf</code>	an object of class <code>lm</code> specifying the full model.
<code>cstar</code>	the boundary of the fence, typically found through bootstrapping.
<code>nvmax</code>	the maximum number of variables that will be considered in the model.
<code>adaptive</code>	logical. If TRUE the boundary of the fence is given by <code>cstar</code> . Otherwise, it the original (non-adaptive) fence is performed where the boundary is $c^* \hat{\sigma}_{M, \tilde{M}}$.
<code>trace</code>	logical. If TRUE the function prints out its progress as it iterates up through the dimensions.
<code>force.in</code>	the names of variables that should be forced into all estimated models.
<code>...</code>	further arguments (currently unused)

References

Jiming Jiang, Thuan Nguyen, J. Sunil Rao, A simplified adaptive fence procedure, Statistics & Probability Letters, Volume 79, Issue 5, 1 March 2009, Pages 625-629, <http://dx.doi.org/10.1016/j.spl.2008.10.014>.

See Also

`af`, `glmfence`

Other fence: `af()`, `glmfence()`

Examples

```
n = 40 # sample size
beta = c(1,2,3,0,0)
K=length(beta)
set.seed(198)
X = cbind(1,matrix(rnorm(n*(K-1)),ncol=K-1))
e = rnorm(n)
y = X%*%beta + e
dat = data.frame(y,X[, -1])
# Non-adaptive approach (not recommended)
lm1 = lm(y~.,data=dat)
lmfence(lm1,cstar=log(n),adaptive=FALSE)
```

mplot

Model selection and stability curves

Description

Opens a shiny GUI to investigate a range of model selection and stability issues

Usage

```
mplot(mf, ...)
```

Arguments

mf a fitted model.
 ... objects of type vis or af or bglmnet.

References

Tarr G, Mueller S and Welsh AH (2018). mplot: An R Package for Graphical Model Stability and Variable Selection Procedures. Journal of Statistical Software, 83(9), pp. 1-28. doi: 10.18637/jss.v083.i09

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = round(data.frame(y,x1,x2,x3,x4,x5),2)
lm1 = lm(y ~ ., data = dat)
## Not run:
v1 = vis(lm1)
af1 = af(lm1)
bg1 = bglmnet(lm1)
mplot(lm1, v1, af1, bg1)

## End(Not run)
```

plot.af

Plot diagnostics for an af object

Description

Summary plot of the bootstrap results of an af object.

Usage

```
## S3 method for class 'af'
plot(
  x,
  pch,
  interactive = FALSE,
  classic = NULL,
  tag = NULL,
```

```

    shiny = FALSE,
    best.only = FALSE,
    width = 800,
    height = 400,
    fontSize = 12,
    left = 50,
    top = 30,
    chartWidth = "60%",
    chartHeight = "80%",
    backgroundColor = "transparent",
    legend.position = "top",
    model.wrap = NULL,
    legend.space = NULL,
    options = NULL,
    ...
)

```

Arguments

x	af object, the result of af
pch	plotting character, i.e., symbol to use
interactive	logical. If interactive=TRUE a googleVis plot is provided instead of the base graphics plot. Default is interactive=FALSE.
classic	logical. Deprecated. If classic=TRUE a base graphics plot is provided instead of a googleVis plot. For now specifying classic will overwrite the default interactive behaviour, though this is likely to be removed in the future.
tag	Default NULL. Name tag of the objects to be extracted from a gvis (googleVis) object. The default tag for is NULL, which will result in R opening a browser window. Setting tag='chart' or setting options(gvis.plot.tag='chart') is useful when googleVis is used in scripts, like knitr or rmarkdown.
shiny	Default FALSE. Set to TRUE when using in a shiny interface.
best.only	logical determining whether the output used the standard fence approach of only considering the best models that pass the fence (TRUE) or if it should take into account all models that pass the fence at each boundary value (FALSE).
width	Width of the googleVis chart canvas area, in pixels. Default: 800.
height	Height of the googleVis chart canvas area, in pixels. Default: 400.
fontSize	font size used in googleVis chart. Default: 12.
left	space at left of chart (pixels?). Default: "50".
top	space at top of chart (pixels?). Default: "30".
chartWidth	googleVis chart area width. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "60%"
chartHeight	googleVis chart area height. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "80%"

backgroundColor	The background colour for the main area of the chart. A simple HTML color string, for example: 'red' or '#00cc00'. Default: 'transparent'
legend.position	legend position, e.g. "topleft" or "bottomright"
model.wrap	Optional parameter to split the legend names if they are too long for classic plots. model.wrap=2 means that there will be two variables per line, model.wrap=2 gives three variables per line and model.wrap=4 gives 4 variables per line.
legend.space	Optional parameter to add additional space between the legend items for the classic plot.
options	If you want to specify the full set of googleVis options.
...	further arguments (currently unused)

Details

For each value of c a parametric bootstrap is performed under the full model. For each bootstrap sample we identify the smallest model inside the fence, $\hat{\alpha}(c)$. We calculate the empirical probability of selecting model α for a given value of c as

$$p^*(c, \alpha) = P^*\{\hat{\alpha}(c) = \alpha\}.$$

Hence, if B bootstrap replications are performed, $p^*(c, \alpha)$ is the proportion of times that model α is selected. Finally, define an overall selection probability,

$$p^*(c) = \max_{\alpha \in \mathcal{A}} p^*(c, \alpha)$$

and we plot $p^*(c)$ against c . The points on the scatter plot are colour coded by the model that yielded the highest inclusion probability.

plot.bglmnet	<i>Plot diagnostics for a bglmnet object</i>
--------------	--

Description

A plot method to visualise the results of a bglmnet object.

Usage

```
## S3 method for class 'bglmnet'
plot(
  x,
  highlight,
  interactive = FALSE,
  classic = NULL,
  tag = NULL,
  shiny = FALSE,
```

```

which = c("vip", "boot", "boot_size"),
width = 800,
height = 400,
fontSize = 12,
left = 50,
top = 30,
chartWidth = "60%",
chartHeight = "80%",
axisTitlesPosition = "out",
dataOpacity = 0.5,
options = NULL,
hAxis.logScale = TRUE,
ylim,
text = FALSE,
backgroundColor = "transparent",
legend.position = "right",
jitterk = 0.1,
srt = 45,
max.circle = 15,
min.prob = 0.1,
...
)

```

Arguments

x	bglmnet object, the result of bglmnet
highlight	the name of a variable that will be highlighted.
interactive	logical. If interactive=TRUE a googleVis plot is provided instead of the base graphics plot. Default is interactive=FALSE.
classic	logical. Deprecated. If classic=TRUE a base graphics plot is provided instead of a googleVis plot. For now specifying classic will overwrite the default interactive behaviour, though this is likely to be removed in the future.
tag	Default NULL. Name tag of the objects to be extracted from a gvis (googleVis) object. The default tag for is NULL, which will result in R opening a browser window. Setting tag='chart' or setting options(gvis.plot.tag='chart') is useful when googleVis is used in scripts, like knitr or rmarkdown.
shiny	Default FALSE. Set to TRUE when using in a shiny interface.
which	a vector specifying the plots to be output. Variable inclusion type plots which = "vip" or plots where the size of the point representing each model is proportional to selection probabilities by model size which = "boot_size" or by penalty paramter which = "boot".
width	Width of the googleVis chart canvas area, in pixels. Default: 800.
height	Height of the googleVis chart canvas area, in pixels. Default: 400.
fontSize	font size used in googleVis chart. Default: 12.
left	space at left of chart (pixels?). Default: "50".

top	space at top of chart (pixels?). Default: "30".
chartWidth	googleVis chart area width. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "60%"
chartHeight	googleVis chart area height. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "80%"
axisTitlesPosition	Where to place the googleVis axis titles, compared to the chart area. Supported values: "in" - Draw the axis titles inside the the chart area. "out" - Draw the axis titles outside the chart area. "none" - Omit the axis titles.
dataOpacity	The transparency of googleVis data points, with 1.0 being completely opaque and 0.0 fully transparent.
options	a list to be passed to the googleVis function giving complete control over the output. Specifying a value for options overwrites all other plotting variables.
hAxis.logScale	logical, whether or not to use a log scale on the horizontal axis. Default = TRUE.
ylim	the y limits of the which="boot" plots.
text	logical, whether or not to add text labels to classic boot plot. Default = FALSE.
backgroundColor	The background colour for the main area of the chart. A simple HTML color string, for example: 'red' or '#00cc00'. Default: 'transparent'
legend.position	the postion of the legend for classic plots. Default legend.position="right" alternatives include legend.position="top" and legend.position="bottom"
jitterk	amount of jittering of the model size in the lvk and boot plots. Default = 0.1.
srt	when text=TRUE, the angle of rotation for the text labels. Default = 45.
max.circle	determines the maximum circle size. Default = 15.
min.prob	lower bound on the probability of a model being selected. If a model has a selection probability lower than min.prob it will not be plotted.
...	further arguments (currently unused)

See Also[bglnet](#)

plot.vis

*Plot diagnostics for a vis object***Description**

A plot method to visualise the results of a vis object.

Usage

```
## S3 method for class 'vis'
plot(
  x,
  highlight,
  interactive = FALSE,
  classic = NULL,
  tag = NULL,
  shiny = FALSE,
  nbest = "all",
  which = c("vip", "lvk", "boot"),
  width = 800,
  height = 400,
  fontSize = 12,
  left = 50,
  top = 30,
  chartWidth = "60%",
  chartHeight = "80%",
  axisTitlesPosition = "out",
  dataOpacity = 0.5,
  options = NULL,
  ylim,
  legend.position = "right",
  backgroundColor = "transparent",
  text = FALSE,
  min.prob = 0.4,
  srt = 45,
  max.circle = 15,
  print.full.model = FALSE,
  jitterk = 0.1,
  seed = NULL,
  ...
)
```

Arguments

<code>x</code>	vis object, the result of <code>vis</code>
<code>highlight</code>	the name of a variable that will be highlighted
<code>interactive</code>	logical. If <code>interactive=TRUE</code> a googleVis plot is provided instead of the base graphics plot. Default is <code>interactive=FALSE</code> .
<code>classic</code>	logical. Deprecated. If <code>classic=TRUE</code> a base graphics plot is provided instead of a googleVis plot. For now specifying <code>classic</code> will overwrite the default interactive behaviour, though this is likely to be removed in the future.
<code>tag</code>	Default NULL. Name tag of the objects to be extracted from a gvis (googleVis) object. The default tag for is NULL, which will result in R opening a browser window. Setting <code>tag='chart'</code> or setting <code>options(gvis.plot.tag='chart')</code> is useful

	when googleVis is used in scripts, like knitr or rmarkdown.
shiny	Default FALSE. Set to TRUE when using in a shiny interface.
nbest	maximum number of models at each model size that will be considered for the lvk plot. Can also take a value of "all" which displays all models (default).
which	a vector specifying the plots to be output. Variable inclusion plots which="vip"; description loss against model size which="lvk"; bootstrapped description loss against model size which="boot".
width	Width of the googleVis chart canvas area, in pixels. Default: 800.
height	Height of the googleVis chart canvas area, in pixels. Default: 400.
fontSize	font size used in googleVis chart. Default: 12.
left	space at left of chart (pixels?). Default: "50".
top	space at top of chart (pixels?). Default: "30".
chartWidth	googleVis chart area width. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "60%"
chartHeight	googleVis chart area height. A simple number is a value in pixels; a string containing a number followed by % is a percentage. Default: "80%"
axisTitlesPosition	Where to place the googleVis axis titles, compared to the chart area. Supported values: "in" - Draw the axis titles inside the the chart area. "out" - Draw the axis titles outside the chart area. "none" - Omit the axis titles.
dataOpacity	The transparency of googleVis data points, with 1.0 being completely opaque and 0.0 fully transparent.
options	a list to be passed to the googleVis function giving complete control over the output. Specifying a value for options overwrites all other plotting variables.
ylim	the y limits of the lvk and boot plots.
legend.position	the postion of the legend for classic plots. Default legend.position="right" alternatives include legend.position="top" and legend.position="bottom"
backgroundColor	The background colour for the main area of the chart. A simple HTML color string, for example: 'red' or '#00cc00'. Default: 'null' (there is an issue with GoogleCharts when setting 'transparent' related to the zoom window sticking - once that's sorted out, the default will change back to 'transparent')
text	logical, whether or not to add text labels to classic boot plot. Default = FALSE.
min.prob	when text=TRUE, a lower bound on the probability of selection before a text label is shown.
srt	when text=TRUE, the angle of rotation for the text labels. Default = 45.
max.circle	determines the maximum circle size. Default = 15.
print.full.model	logical, when text=TRUE this determines if the full model gets a label or not. Default=FALSE.
jitterk	amount of jittering of the model size in the lvk and boot plots. Default = 0.1.
seed	random seed for reproducible results
...	further arguments (currently unused)

Details

Specifying `which = "lvk"` generates a scatter plot where the points correspond to description loss is plot against model size for each model considered. The `highlight` argument is used to differentiate models that contain a particular variable from those that do not.

Specifying `which = "boot"` generates a scatter plot where each circle represents a model with a non-zero bootstrap probability, that is, each model that was selected as the best model of a particular dimension in at least one bootstrap replication. The area of each circle is proportional to the corresponding model's bootstrapped selection probability.

References

Mueller, S. and Welsh, A. H. (2010), On model selection curves. *International Statistical Review*, 78:240-256. doi: 10.1111/j.1751-5823.2010.00108.x

Murray, K., Heritier, S. and Mueller, S. (2013), Graphical tools for model selection in generalized linear models. *Statistics in Medicine*, 32:4438-4451. doi: 10.1002/sim.5855

Tarr G, Mueller S and Welsh AH (2018). `mplot`: An R Package for Graphical Model Stability and Variable Selection Procedures. *Journal of Statistical Software*, 83(9), pp. 1-28. doi: 10.18637/jss.v083.i09

See Also

[vis](#)

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = data.frame(y,x1,x2,x3,x4,x5)
lm1 = lm(y~.,data=dat)

## Not run:
v1 = vis(lm1, seed = 1)
plot(v1, highlight = "x1", which = "lvk")
plot(v1, which = "boot")
plot(v1, which = "vip")

## End(Not run)
```

print.af	<i>Print method for an af object</i>
----------	--------------------------------------

Description

Prints basic output of the bootstrap results of an af object.

Usage

```
## S3 method for class 'af'  
print(x, best.only = TRUE, ...)
```

Arguments

x	an af object, the result of af
best.only	logical determining whether the output used the standard fence approach of only considering the best models that pass the fence (TRUE) or if it should take into account all models that pass the fence at each boundary value (FALSE).
...	further arguments (currently unused)

print.vis	<i>Print method for a vis object</i>
-----------	--------------------------------------

Description

Prints basic output of the bootstrap results of an vis object.

Usage

```
## S3 method for class 'vis'  
print(x, min.prob = 0.3, print.full.model = FALSE, ...)
```

Arguments

x	a vis object, the result of vis
min.prob	a lower bound on the probability of selection before the result is printed
print.full.model	logical, determines if the full model gets printed or not. Default=FALSE.
...	further arguments (currently unused)

summary.af	<i>Summary method for an af object</i>
------------	--

Description

Provides comprehensive output of the bootstrap results of an af object.

Usage

```
## S3 method for class 'af'
summary(object, best.only = TRUE, ...)
```

Arguments

object	af object, the result of af
best.only	logical determining whether the output used the standard fence approach of only considering the best models that pass the fence (TRUE) or if it should take into account all models that pass the fence at each boundary value (FALSE).
...	further arguments (currently unused)

vis	<i>Model stability and variable inclusion plots</i>
-----	---

Description

Calculates and provides the plot methods for standard and bootstrap enhanced model stability plots (lvk and boot) as well as variable inclusion plots (vip).

Usage

```
vis(
  mf,
  nvmax,
  B = 100,
  lambda.max,
  nbest = "all",
  use.glmulti = FALSE,
  cores,
  force.in = NULL,
  screen = FALSE,
  redundant = TRUE,
  seed = NULL,
  ...
)
```

Arguments

<code>mf</code>	a fitted 'full' model, the result of a call to <code>lm</code> or <code>glm</code> (and in the future <code>lme</code> or <code>lmer</code>)
<code>nvmax</code>	size of the largest model that can still be considered as a viable candidate
<code>B</code>	number of bootstrap replications
<code>lambda.max</code>	maximum penalty value for the vip plot, defaults to $2 \cdot \log(n)$
<code>nbest</code>	maximum number of models at each model size that will be considered for the <code>lvk</code> plot. Can also take a value of "all" which displays all models.
<code>use.glmulti</code>	logical. Whether to use the <code>glmulti</code> package instead of <code>bestglm</code> . Default <code>use.glmulti=FALSE</code> .
<code>cores</code>	number of cores to be used when parallel processing the bootstrap
<code>force.in</code>	the names of variables that should be forced into all estimated models. (Not yet implemented.)
<code>screen</code>	logical, whether or not to perform an initial screen for outliers. Highly experimental, use at own risk. Default = <code>FALSE</code> .
<code>redundant</code>	logical, whether or not to add a redundant variable. Default = <code>TRUE</code> .
<code>seed</code>	random seed for reproducible results. When set, results are reproducible across runs regardless of <code>cores</code> value. Uses <code>future</code> 's native seeding via <code>furrr_options(seed = TRUE)</code> .
<code>...</code>	further arguments (currently unused)

Details

The result of this function is essentially just a list. The supplied plot method provides a way to visualise the results.

The `cores` argument controls parallelization backend: when `cores > 1`, a `future::multisession` plan is registered for the duration of the function. The caller's existing future plan is preserved and restored on exit.

See `?plot.vis` or `help("plot.vis")` for details of the plot method associated with the result.

References

- Mueller, S. and Welsh, A. H. (2010), On model selection curves. *International Statistical Review*, 78:240-256. doi: 10.1111/j.1751-5823.2010.00108.x
- Murray, K., Heritier, S. and Mueller, S. (2013), Graphical tools for model selection in generalized linear models. *Statistics in Medicine*, 32:4438-4451. doi: 10.1002/sim.5855
- Tarr G, Mueller S and Welsh AH (2018). `mplot`: An R Package for Graphical Model Stability and Variable Selection Procedures. *Journal of Statistical Software*, 83(9), pp. 1-28. doi: 10.18637/jss.v083.i09

See Also

[plot.vis](#)

Examples

```
n = 100
set.seed(11)
e = rnorm(n)
x1 = rnorm(n)
x2 = rnorm(n)
x3 = x1^2
x4 = x2^2
x5 = x1*x2
y = 1 + x1 + x2 + e
dat = data.frame(y, x1, x2, x3, x4, x5)
lm1 = lm(y ~ ., data = dat)

## Not run:
v1 = vis(lm1, seed = 1)
plot(v1, highlight = "x1", which = "lvk")
plot(v1, which = "boot")
plot(v1, which = "vip")

## End(Not run)
```

wallabies

Rock-wallabies data set

Description

On Chalkers Top in the Warrumbungles (NSW, Australia) 200 evenly distributed one metre squared plots were surveyed. Plots were placed at a density of 7-13 per hectare. The presence or absence of fresh (<1 month old) scats of rock-wallabies was recorded for each plot along with location and a selection of predictor variables.

Usage

```
data(wallabies)
```

Format

A data frame with 200 observations on 9 variables.

rw Presence of rock-wallaby scat

edible Percentage cover of edible vegetation

inedible Percentage cover of inedible vegetation

canopy Percentage canopy cover

distance Distance from diurnal refuge

shelter Whether or not a plot occurred within a shelter point (large rock or boulder pile)

lat Latitude of the plot location

long Longitude of the plot location

Details

Macropods defaecate randomly as they forage and scat (faecal pellet) surveys are a reliable method for detecting the presence of rock-wallabies and other macropods. Scats are used as an indication of spatial foraging patterns of rock-wallabies and sympatric macropods. Scats deposited while foraging were not confused with scats deposited while resting because the daytime refuge areas of rock-wallabies were known in detail for each colony and no samples were taken from those areas. Each of the 200 sites were examined separately to account for the different levels of predation risk and the abundance of rock-wallabies.

References

Tuft KD, Crowther MS, Connell K, Mueller S and McArthur C (2011), Predation risk and competitive interactions affect foraging of an endangered refuge-dependent herbivore. *Animal Conservation*, 14: 447-457. doi: 10.1111/j.1469-1795.2011.00446.x

Examples

```
data(wallabies)
wdat = data.frame(subset(wallabies,select=-c(lat,long)),
  EaD = wallabies$edible*wallabies$distance,
  EaS = wallabies$edible*wallabies$shelter,
  DaS = wallabies$distance*wallabies$shelter)
M1 = glm(rw~., family = binomial(link = "logit"), data = wdat)
```

Index

* Internal

glmfence, [11](#)

lmfence, [11](#)

* datasets

artificialeg, [5](#)

bodyfat, [8](#)

diabetes, [9](#)

fev, [10](#)

wallabies, [24](#)

* fence

af, [3](#)

glmfence, [11](#)

lmfence, [11](#)

af, [3](#), [11](#), [12](#), [14](#), [21](#), [22](#)

af(), [11](#), [12](#)

artificialeg, [5](#)

bglmnet, [6](#), [16](#), [17](#)

bodyfat, [8](#)

diabetes, [9](#)

fev, [10](#)

glm, [11](#)

glmfence, [11](#), [12](#)

glmfence(), [5](#), [12](#)

lm, [12](#)

lmfence, [11](#), [11](#)

lmfence(), [5](#), [11](#)

mplot, [12](#)

mplot-package, [2](#)

plot.af, [5](#), [13](#)

plot.bglmnet, [7](#), [15](#)

plot.vis, [17](#), [23](#)

print.af, [21](#)

print.vis, [21](#)

summary.af, [22](#)

vis, [18](#), [20](#), [21](#), [22](#)

wallabies, [24](#)