Package 'autopls'

February 24, 2015

Version 1.3

Date 2015-02-24

Title Partial Least Squares Regression with Backward Selection of Predictors

Author Sebastian Schmidtlein, with contributions from C. Oldenburg and H. Feilhauer and with a code snipped borrowed from Bjorn-Helge Mevik

Maintainer Sebastian Schmidtlein <schmidtlein@kit.edu>

Description Some convenience functions for pls regression, including backward variable selection and validation procedures, image based predictions and plotting.

Depends pls

Suggests rgdal, raster

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-02-24 15:30:03

R topics documented:

autopls 2
autoplsVAL 5
extract.autopls 7
murnau.X 8
plot.autopls 9
postprocessing 10
predict.autopls 12
predict.slim 13
prepro 14
reset 15
set.iter 16
set.lv 17
summary.autopls 18

Index **20**

autopls *autopls*

Description

Partial least squares regression with backward selection of predictors

Usage

```
autopls (formula, data, testset = NULL, tselect = "none", prep = "none",
  val = "LOO", scaling = TRUE, stingy = TRUE, verbose = TRUE,
  backselect = "auto", jt.thresh = 0.1, vip.thresh = 0.2, jump = NA,
  lower = NA, method = "oscorespls")
```

Arguments

formula	model formula
data	optional data frame with the data to fit the model
testset	optional vector defining a test set (row indices)
tselect	string specifying the role of the test set in model selection ("none", "passive" or "active", see details)
prep	character. optional preprocessing (only one choice implemented: "bn" (see details)
val	character. Validation used ("CV" or "LOO", see details)
scaling	logical. if TRUE, predictors are scaled by dividing each variable by its standard deviation. This is repeated in all validation steps
stingy	logical. If TRUE, the number of latent vectors is kept low during backward selection
verbose	logical. If TRUE, details about the backward selection processes are reported
backselect	one or more character strings defining the methods used in backwards selection (see details). "no" means no backselection. Defaults to "auto"
jt.thresh	threshold used in predictor selections that are based on jackknife testing (methods based on A1, see details)
vip.thresh	threshold used in predictor selections that are based on VIP (methods based on A2, see details). VIP is scaled to a maximum of 1.
jump	numeric. If a number is given, backward selection starts with a forced reduction of predictors to the given number (see A0 in details). This reduction is based on significance in jackknifing. The argument can be useful in the case of large predictor matrices.
lower	numeric. Backward selection proceeds as long as R2 in validation reaches the given value (experimental, backward selection continues further if models improve in other respects such as decreasing numbers of latent vectors).
method	character string indicating what plsr method to use. autopls works with the orthogonal scores algorithm ("oscorespls") and with the kernel algorithm ("kernelpls").

Details

The autopls function is a wrapper for pls in package **pls** written by Bjørn-Helge Mevik, Ron Wehrens and Kristian Hovde Liland. As for now, the wrapper can be cited as Schmidtlein et al. (2012). autopls works only for single target variables.

If validation = "CV", 10-fold cross-validation is performed. If validation = "LOO", leave-one-out cross-validation is performed. Test set validation takes always place if a test set has been defined. tselect specifies how the test set is used in model selection. "none": just use it for external validation; "passive": use error in external validation for model selection but do not use it for the determination of the number of latent vectors; "active" use the error in external validation for model selection and for the determination of the number of latent vectors. With stingy = TRUE the errors that are used in the selection are measured at a number of latent vectors that depends on the number of observations (1/10 at maximum). Otherwise, the number of latent vectors is chosen where errors approach a first minimum. In order to avoid minor local minima the error values are first smoothed.

Large data matrices: Examine the arguments jump (forced reduction of predictors in the first iteration). Large model objects can be shrinked using the function slim but some functionality (like plotting or change of the number of latent vectors) is lost. Shrinked models can still be used for predictions.

Preprocessing options: The only implemented option is currently "bn", which is a brightness normalization according to Feilhauer et al. (2010).

Several methods for predictor selection are available. In default mode (backselect = "auto") the selection follows an optimization procedure using methods A1 and A3. However, apart from A0 any user-defined combination can be selected using the backselect argument. Note that VIP-based methods (A2, A3, B3 to B6) are meant to be used with the oscorespls method and methods B1 to B6 and C1 do only make sense with sequences of spectral bands or similar sequences of autocorrelated predictors. The methods are coded as follows:

A) Filtering based on thresholds

(A0 and A1) Based on significance, A0 with user-defined threshold (see argument jump); (A2) based on VIP; (A3) based on combined significance and VIP; (A4) removal of 10 % predictors with the lowest significance; (A5) removal of 25 % predictors with the lowest significance.

B) Filtering followed by reduction of autocorrelation

(B1) Filtering based on significance, thinning starting with local maxima in weighted regression coefficients; (B2) filtering based on significance, thinning starting with local maxima in significance; (B3) filtering based on significance, thinning starting with local maxima in VIP; (B4) filtering based on VIP, thinning starting with local maxima in weighted regression coefficients; (B5) filtering based on VIP, thinning starting with local maxima in significance; (B6) filtering based on VIP, thinning starting with local maxima in VIP.

C) Just reduction of autocorrelation

(C1): reduction starting with local maxima in regression coefficients.

Value

An object of class autopls is returned. This equals a pls object and some added objects:

predictors	logical. Vector of predictors that have been or have not been used in the current model

`metapls`	outcomes of the backward selection process
`iterations`	models selected during the backward selection process

The `$metapls` item consists of the following:

`current.iter`	iteration of the backward selection procedure the current model is based upon
`autopls.iter`	iteration of the backward selection procedure originally selected by autopls
`current.lv`	number of latent vectors the current model is based upon
`autopls.lv`	number of latent vectors originally selected by autopls
`lv.history`	sequence of number of latent vectors values selected during iterations in backward selection
`rmse.history`	sequence of root mean squared errors obtained during iterations in backward selection. Errors are reported for calibration and validation. The validation errors are also reported for the number of latent vectors corresponding to `ceiling (nrow (pred) / 10)`.
`r2.history`	sequence of number of r2 values obtained during iterations in backward selection
`X`	original predictors
`Y`	original target variable
`X.testset`	test set: predictors
`Y.testset`	test set: target variable
`preprocessing`	method used for preprocessing
`scaling`	`TRUE` if scaling was requested
`val`	`LOO` or `CV`
`call`	the function call

Author(s)

Sebastian Schmidtlein with contributions from Carsten Oldenburg and Hannes Feilhauer. The code for computing `VIP` is borrowed from Bjørn-Helge Mevik.

References

Feilhauer. H., Asner, G.P., Martin, R.E., Schmidtlein, S. (2010): Brightness-normalized Partial Least Squares regression for hyperspectral data. *Journal of Quantitative Spectroscopy and Radiative Transfer* **111**: 1947–1957.

Schmidtlein, S., Feilhauer, H., Bruelheide, H. (2012): Mapping plant strategy types using remote sensing. *Journal of Vegetation Science* **23**: 395–405. Open Access.

See Also

`pls`, `set.iter`, `set.lv`, `predict.autopls`, `plot.autopls`

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls (murnau.Y ~ murnau.X)

## S3 plot method
## Not run: plot (model)
## Not run: plot (model, type = "rc")

## Loading and score plots
## Not run: plot (model$loadings, main = "Loadings")
## Not run: plot (model$loadings [,c(1,3)], main = "Loadings")
## Not run: plot (model$scores, main = "Scores")
```

---

autoplsVAL *Validate a fitted autopls model*

Description

Functions to extract R2 and RMSEP from autopls objects, for significance testing based on jackknife variance estimates for regression

Usage

```
  ## S3 method for class 'autopls'
R2(object, estimate, nc = 'inherit', ic = FALSE, ...)
  ## S3 method for class 'autopls'
RMSEP(object, estimate, nc = 'inherit', ic = FALSE, ...)
  jack.test.autopls (object, nc = 'inherit')
  metaval (object, method, estimate, ic)
  repeatedCV (object, k = 100, segments = 4)
  clusterCV (object, valist)
```

Arguments

object	object of class autopls
method	character. Should be or 'R2' or 'RMSEP'
estimate	character vector. Which estimators to use. In metaval this can be "train" or "CV". Additional options in R2 and RMSEP are "all" and "test").
nc	'inherit' returns values corresponding to the number of latent vectors in the current model, 'all' returns values for all numbers of latent vectors. A specific number returns values corresponding to the respective number of latent vectors.

ic	logical. Specifies whether estimates for a model with zero components should be returned
k	number of cross-validations used in repeatedCV
segments	number of cross-validation segments used in repeatedCV
valist	list of segments. The elements are vectors of plots assigned to a cluster of samples
...	Arguments to be passed to methods

Details

Some of these functions are just convenience wrappers for mvrVal functions and for the jack.test function in package **pls**. More details are given here: mvrVal, jack.test. Other functions are specific autopls functions. metaval is used for a summary of validation results during backselection. repeatedCV is a meta cross-validation (repeated ten-fold cross-validation). clusterCV is a leave-one-site-out cross-validation to avoid effects of spatial or other autocorrelation. The elements of the list should be integer vectors specifying the indices of the segments.

Value

see mvrVal and jack.test. The main difference is a reduced selection of functions (see above) and the possibility to inherit a number of latent vectors from the autopls object.

The metaval function provides a matrix overview of model results for all iterations and numbers of latent vectors in an autopls object. repeatedCV provides results and basic statistics for repeated cross-validation runs.

Note

If you want to make full use of the mvrVal functions in the **pls** package assign class mvr to the model object.

Author(s)

Sebastian Schmidtlein, linking to code from package **pls** by Ron Wehrens and Bjørn-Helge Mevik.

See Also

mvrVal, jack.test, autopls, repCV, mvr_dcv

Examples

```
## load predictor and response data to the current environment
data(murnau.X)
data(murnau.Y)

## call autopls with the standard options
model<-autopls (murnau.Y ~ murnau.X)

## Validation
R2 (model)
```

```
R2 (model, nc = 'all')
RMSEP (model)
metaval (model, 'R2', 'CV', ic = FALSE)

## Jackknife test
jack.test.autopls (model)

## Meta cross-validation
repeatedCV (model)
```

`extract.autopls` *Extract information from a fitted autopls model*

Description

Functions to extract information from `autopls` objects: crossvalidation, fitted values, regression coefficients, residuals, scores, loadings, latent vectors used, underlying run.

Usage

```
  predicted (object)
  get.lv (object)
  get.iter (object)
  slim (object)
  ## S3 method for class 'autopls'
scores(object, ...)
  ## S3 method for class 'autopls'
loadings(object, ...)
  ## S3 method for class 'autopls'
fitted(object, ...)
  ## S3 method for class 'autopls'
coef(object, intercept = FALSE, ...)
  ## S3 method for class 'slim'
coef(object, intercept = FALSE, ...)
  ## S3 method for class 'autopls'
residuals(object, ...)
```

Arguments

`object`	object of class `autopls`
`intercept`	logical. Should intercept be given?
`...`	logical. Arguments to be passed to methods

Details

Provides convenience wrappers for `extract` functions in package **pls**. More details are given here: coef.mvr. Other functions extract information specific for `autopls` objects: `get.lv`, `get.iter` or condense the model information to a memory saving object of class `slim` that can be used for predictions with `predict.slim`. This makes sense if large pedictor data sets result in huge `autopls` model objects that are difficult to handle.

Value

see coef.mvr. `get.iter` returns the run in the `autopls` backwards selection procedure that has been used for the current model. `get.lv` returns the number of latent vectors used for the present model. `predicted` returns the predictions in model validation while `fitted` returns the predictions in model calibration. `slim` returns an object of class `slim`.

Note

If you want to make full use of the `extract` functions in the **pls** package assign class `mvr` to the model object.

Reducing a model to an object of class `slim` means loosing plotting options.

Author(s)

Sebastian Schmidtlein, links to code from package **pls** by Ron Wehrens and Bjørn-Helge Mevik.

See Also

`autopls`, `metaval`, `set.iter`, `set.lv`, `predict.slim`

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls (murnau.Y ~ murnau.X)

## get fitted values
fitted(model)
```

---

murnau.X *Hyperspectral reflectance and plant attributes*

Description

This data gives reflectance from raised bogs and fens (`murnau.X`), the corresponding wavelengths (`murnau.W`) and associated vegetation attributes (`murnau.Y`: plant adaptation to stress). More detail is given by Schmidtlein et al. (2012).

Usage

```
data(murnau.X)
```

Format

Matrix containing 40 observations and 105 spectral bands (`murnau.X`), a vector with 105 wavelengths corresponding to bands (`murnau.W`) and a vector with 40 observations of plant attributes (`murnau.Y`).

Source

Schmidtlein, S., Feilhauer, H., Bruelheide, H. (2012): Mapping plant strategy types using remote sensing. *Journal of Vegetation Science* **23**: 395–405. Open Access.

---

plot.autopls *Plotting function for autopls objects*

Description

Produces plots illustrating the outcomes of `autopls`: predicted vs. observed values, errors vs. numbers of latent vectors, regression coefficients, influences of observations regarding X and Y, latent vectors and R2 in backward selection

Usage

```
  ## S3 method for class 'autopls'
plot(x,type="all",wl=NULL,rcxlab = "Predictors",
  plab=FALSE, bw = FALSE, ...)
```

Arguments

`x`	object of class `autopls`
`type`	specifying the type of plot. (`"all"`: all plot; `"ovp"`: observed vs. predicted values; `"ovp.test"`: test set: observed vs. predicted values; `"rmse"`: internal validation errors vs. numbers of latent vectors; `"rmse.test"`: test set errors vs. numbers of latent vectors; `"rc"`: regression coefficients; `"x.inf"`: influence plot (X-variance); `"y.inf"`: influence plot (Y-variance); `"meta"`: latent vectors and R2 in backward selection)
`wl`	denotes an optional vector of numerical values describing the position of predictors along the x axis in the `rc` plot. The values should refer to all bands (before backward selection) or to the bands that are actually used.
`rcxlab`	Label for x axis in rc plot.
`plab`	logical. Whether observations are labeled.
`bw`	logical. Whether plots are given in grey-scales (partly realized).
`...`	Arguments to be passed to methods

Details

Red dots in the influence plots indicate potentially dangerous outliers

Value

Apart from the plots the function returns the underlying values

Author(s)

Sebastian Schmidtlein, Carsten Oldenburg

The placement of observation labels if `plab = TRUE` is done using code borrowed from the `pointLabel` in **maptools**. The author of this function is Tom Short (EPRI).

See Also

`autopls`

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)
data (murnau.W)

## call autopls with the standard options
model<-autopls (murnau.Y ~ murnau.X)

## plot results
## Not run: plot (model)

## use wavelengths in rc plot
## Not run: plot (model, type = "rc", wl = murnau.W, rcxlab = "Wavelength (nm)")

## predicted vs. observed
## Not run: x <- plot (model, type = "ovp")
## Not run: x
```

---

postprocessing *Test for model extrapolations or interpolations and removal of bold predictions in autopls*

Description

Departures of values predicted by `autopls` from the original data space and removal of exceedingly extrapolated predictions.

Usage

```
liability (object, prediction)
confine (object, prediction, tolerance)
```

Arguments

object	object of class autopls
prediction	predicted values as single vector or single layer raster image (RasterLayer from package **raster**).
tolerance	maximum departure of preserved prediction values

Details

Takes its time with large images and many objects used in calibration.

Value

Vector or raster image depending on the type of prediction Uncertainties (liability function) are given in original units. After confine, values exceeding tolerance are replaced by NA.

Author(s)

Sebastian Schmidtlein

See Also

autopls, predict.autopls

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls (murnau.Y ~ murnau.X)

## new data
new <- murnau.X + 500

## prediction
pred <- predict (model, new)

## check uncertainty
liability (model, pred)

## remove predictions with uncertainty value > 5
confine (model, pred, 5)
```

predict.autopls *Prediction using a fitted autopls model*

Description

Applies a model from a autopls object to a vector, matrix or to a stack or brick from package **raster**.

Usage

```
  ## S3 method for class 'autopls'
predict(object, dat, ...)
```

Arguments

object	object of class autopls
dat	vector, matrix, dataframe or imagery (the latter as stack or brick from package **raster**.
...	logical. Arguments to be passed to method

Details

Elements, columns or layers must have the same number and order as the input predictors for autopls. The predictors resulting from autopls are selected silently. In case of large image files the function is based on tile processing.

Value

A new vector matrix or image depending on the type of newdata

Author(s)

Sebastian Schmidtlein

See Also

autopls

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model<-autopls (murnau.Y ~ murnau.X)

## new data
```

```
new <- murnau.X + 500

## prediction
predict (model, new)
```

---

predict.slim *Prediction using a condensed autopls model*

Description

Applies a model object of class slim originating from autopls to a vector, matrix or to a stack or brick from package **raster**.

Usage

```
  ## S3 method for class 'slim'
predict(object, dat, ...)
```

Arguments

object	object of class slim
dat	vector, matrix, dataframe or imagery (the latter as stack or brick from package **raster**.
...	logical. Arguments to be passed to method

Details

Elements, columns or layers must have the same number and order as the input predictors for autopls. In case of large image files the function is based on tile processing.

Value

A new vector matrix or image depending on the type of newdata

Author(s)

Sebastian Schmidtlein

See Also

autopls, slim

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls (murnau.Y ~ murnau.X)

## condensed model object
new.model <- slim (model)

## new data
new <- murnau.X + 500

## prediction
predict (new.model, new)
```

`prepro` *Preprocessing in autopls*

Description

Used for preprocessing predictor data in functions `autopls` and `predict.autopls`

Usage

```
prepro(X, method = 'bn')
```

Arguments

`X`	predictors as vector, matrix, raster brick or raster stack
`method`	type of preprocessing (currently only brightness normalization coded as "bn")

Details

The function is called within `autopls` and `predict.autopls`. The only implemented option is currently "bn", which is a brightness normalization according to Feilhauer et al. (2010). Raster brick and raster stack are objects of package **raster**.

Value

Returns the transformed matrix or raster object.

Author(s)

Hannes Feilhauer

References

Feilhauer. H., Asner, G.P., Martin, R.E., Schmidtlein, S. (2010): Brightness-normalized Partial Least Squares regression for hyperspectral data. *Journal of Quantitative Spectroscopy and Radiative Transfer* **111**: 1947–1957.

See Also

autopls, predict.autopls

`reset` *Resets the number of latent vectors and the iteration used in autopls objects*

Description

Resets the number of latent vectors and the iteration used in autopls objects to the values originally selected by the autopls procedure

Usage

```
reset(object, verbose = TRUE)
```

Arguments

`object`	object of class `autopls`
`verbose`	logical. If a summary of the resulting object should be printed on the screen

Value

Returns an object of class `autopls`

Author(s)

Sebastian Schmidtlein

See Also

autopls, set.iter, set.lv

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls (murnau.Y ~ murnau.X)
```

```
## select another iteration
newmodel <- set.iter (model,3)

## set another number of latent vectors
evennewermodel <- set.lv (newmodel,2)

## return to the original values
firstmodel <- reset (evennewermodel)
```

`set.iter` *Sets the run of an autopls backwards selection to be used*

Description

Changes the run of an autopls backwards selection to be used

Usage

```
set.iter (object, iteration, verbose = TRUE)
```

Arguments

`object`	object of class `autopls`
`iteration`	new value for the iteration used
`verbose`	logical. If a summary of the resulting object should be printed on the screen

Details

The number of latent vectors is set to the orinal number for this run.

Value

Returns an object of class `autopls`

Author(s)

Sebastian Schmidtlein

See Also

`autopls`, `set.lv`

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls (murnau.Y ~ murnau.X)

## set another number of latent vectors
newmodel <- set.iter (model,3)
```

`set.lv` *Sets the number of latent vectors in autopls objects*

Description

Sets the number of latent vectors in a `autopls` object

Usage

```
set.lv(object, lv, verbose = TRUE)
```

Arguments

`object`	object of class `autopls`
`lv`	new value for the number of latent vectors used
`verbose`	logical. If a summary of the resulting object should be printed on the screen

Value

Returns an object of class `autopls`

Author(s)

Sebastian Schmidtlein

See Also

`autopls`, `set.iter`

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model<-autopls (murnau.Y ~ murnau.X)

## set another number of latent vectors
newmodel <- set.lv (model,2)
```

`summary.autopls` *Summary and print functions for autopls objects*

Description

Summary and print methods for `autopls` and `slim` objects.

Usage

```
  ## S3 method for class 'autopls'
summary(object, ...)
  ## S3 method for class 'slim'
summary(object, ...)
  ## S3 method for class 'autopls'
print(x, ...)
  ## S3 method for class 'slim'
print(x, ...)
```

Arguments

`object`	object of class `autopls` or `slim` respectively
`x`	object of class `autopls` or `slim`
`...`	Arguments to be passed to methods

Value

`predictors`	number of predictors used in the final model
`lv`	number of latent vectors used in the final model
`rmse.cal`	root mean squared errors in calibration
`rmse.val`	root mean squared errors in validation
`r2.cal`	R2 in calibration
`r2.val`	R2 in validation

Print returns a screen output and an unvisible object with the same content

Author(s)

Sebastian Schmidtlein

See Also

autopls, slim

Examples

```
## load predictor and response data to the current environment
data (murnau.X)
data (murnau.Y)

## call autopls with the standard options
model <- autopls(murnau.Y ~ murnau.X)

## print and plot results
print (model)
```

Index

∗Topic **datasets**
murnau.X, 8
∗Topic **hplot**
plot.autopls, 9
∗Topic **multivariate**
autopls, 2
autoplsVAL, 5
plot.autopls, 9
postprocessing, 10
predict.autopls, 12
predict.slim, 13
reset, 15
set.iter, 16
set.lv, 17
summary.autopls, 18
∗Topic **regression**
autopls, 2
autoplsVAL, 5
plot.autopls, 9
postprocessing, 10
predict.autopls, 12
predict.slim, 13
reset, 15
set.iter, 16
set.lv, 17
summary.autopls, 18

autopls, 2, *6*, *8–13*, *15–17*, *19*
autoplsVAL, 5

clusterCV *(*autoplsVAL*)*, 5
coef.autopls *(*extract.autopls*)*, 7
coef.mvr, *8*
coef.slim *(*extract.autopls*)*, 7
confine *(*postprocessing*)*, 10

extract.autopls, 7

fitted.autopls *(*extract.autopls*)*, 7

get.iter *(*extract.autopls*)*, 7
get.lv *(*extract.autopls*)*, 7

jack.test, *6*
jack.test.autopls *(*autoplsVAL*)*, 5

liability *(*postprocessing*)*, 10
loadings.autopls *(*extract.autopls*)*, 7

metaval, *8*
metaval *(*autoplsVAL*)*, 5
murnau.W *(*murnau.X*)*, 8
murnau.X, 8
murnau.Y *(*murnau.X*)*, 8
mvr_dcv, *6*
mvrVal, *6*

plot.autopls, *4*, 9
pls, *3*, *4*
postprocessing, 10
predict.autopls, *4*, *11*, 12, *15*
predict.slim, *8*, 13
predicted *(*extract.autopls*)*, 7
prepro, 14
print.autopls *(*summary.autopls*)*, 18
print.slim *(*summary.autopls*)*, 18

R2.autopls *(*autoplsVAL*)*, 5
repCV, *6*
repeatedCV *(*autoplsVAL*)*, 5
reset, 15
residuals.autopls *(*extract.autopls*)*, 7
RMSEP.autopls *(*autoplsVAL*)*, 5

scores.autopls *(*extract.autopls*)*, 7
set.iter, *4*, *8*, *15*, 16, *17*
set.lv, *4*, *8*, *15*, *16*, 17
slim, *3*, *13*, *19*
slim *(*extract.autopls*)*, 7
summary.autopls, 18
summary.slim *(*summary.autopls*)*, 18