

# Package ‘prinvars’

January 11, 2022

**Type** Package

**Title** Principal Variables

**Version** 0.1.0

**Description** Provides methods for reducing the number of features within a data set. See Bauer JO (2021) <[doi:10.1145/3475827.3475832](https://doi.org/10.1145/3475827.3475832)> and Bauer JO, Dra-bant B (2021) <[doi:10.1016/j.jmva.2021.104754](https://doi.org/10.1016/j.jmva.2021.104754)> for more information on principal load-ing analysis.

**License** MIT + file LICENSE

**Imports** methods, Rdpack

**RdMacros** Rdpack

**Encoding** UTF-8

**URL** <https://github.com/Ronho/prinvars>

**BugReports** <https://github.com/Ronho/prinvars/issues>

**RoxygenNote** 7.1.2

**Collate** 'block.R' 'cor.R' 'explained-variance.R' 'get-blocks.R'  
'thresholding.R' 'scale.R' 'utils.R' 'pla.R'  
'prinvars-package.R'

**Suggests** testthat (>= 3.0.0), AER

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jan Bauer [aut, ctb],  
Ron Holzapfel [aut, cre]

**Maintainer** Ron Holzapfel <[ronholzapfel@outlook.de](mailto:ronholzapfel@outlook.de)>

**Repository** CRAN

**Date/Publication** 2022-01-11 19:22:41 UTC

## R topics documented:

Block-class	2
pla	2
pla.drop_blocks	4
pla.keep_blocks	5
print.pla	6
show,Block-method	7
str,Block-method	7

<b>Index</b>	<b>9</b>
--------------	----------

---

Block-class	<i>Block</i>
-------------	--------------

---

### Description

Class used within the package to keep the structure and information about the generated blocks.

### Slots

`features` a vector of numeric which contains the indices of the block.

`explained_variance` a numeric which contains the variance explained of the blocks variables based on the whole data set.

`is_valid` a logical which indicates if the block structure is valid.

---

pla	<i>Principal Loading Analysis</i>
-----	-----------------------------------

---

### Description

This function performs a principal loading analysis on the given data matrix and returns the results as an object of class `pla`.

### Usage

```
pla(
  x,
  cor = FALSE,
  scaled_ev = FALSE,
  thresholds = 0.33,
  threshold_mode = "cutoff",
  expvar = "approx",
  check = "rnc",
  ...
)
```

## Arguments

x	a numeric matrix or data frame which provides the data for the principal loading analysis.
cor	a logical value indicating whether the calculation should use the correlation or the covariance matrix.
scaled_ev	a logical value indicating whether the eigenvectors should be scaled.
thresholds	a numeric value or list of numeric values used to determine "small" values inside the eigenvectors. If multiple values are given, a list of pla results will be returned.
threshold_mode	a character string indicating how the threshold is determined and used. <code>cutoff</code> indicates the usage of a threshold value. <code>percentage</code> indicates that the cutoff value is determined by the maximum element of each vector multiplied with the threshold value.
expvar	a character string indicating the method used for calculating the explained variance. <code>approx</code> uses the explained variance of each eigenvector i.e. its eigenvalue. <code>exact</code> uses the variance of each variable.
check	a character string indicating if only rows or rows as well as columns are used to detect the underlying block structure. <code>rows</code> checks if the rows fulfill the required structure. <code>rnc</code> checks if rows and columns fulfill the required structure.
...	further arguments passed to or from other methods.

## Value

single or list of pla class containing the following attributes:

x	a numeric matrix or data frame which equals the input of x.
c	a numeric matrix or data frame which is the covariance or correlation matrix based on the input of cov.
loadings	a matrix of variable loadings (i.e. a matrix containing the eigenvectors of the dispersion matrix).
threshold	a numeric value which equals the input of thresholds.
threshold_mode	a character string which equals the input of threshold_mode.
blocks	a list of blocks which are identified by principal loading analysis.

See Bauer JO (2021). "Correlation Based Principal Loading Analysis." In *2021 4th International Conference on Mathematics and Statistics*, 27–34. ISBN 9781450389907, doi: [10.1145/3475827.3475832](https://doi.org/10.1145/3475827.3475832). and Bauer JO, Drabant B (2021). "Principal loading analysis." *Journal of Multivariate Analysis*, **184**, 104754. ISSN 0047259X, doi: [10.1016/j.jmva.2021.104754](https://doi.org/10.1016/j.jmva.2021.104754). for more information.

## Examples

```
if(requireNamespace("AER")){
  require(AER)
  data("OECDGrowth")
}
```

```

## the scales in OECDGrowth differ hence using the
## correlation matrix is highly recommended

pla(OECDGrowth,thresholds = 0.5) ## not recommended
pla(OECDGrowth,cor=TRUE,thresholds = 0.5)

## we obtain three blocks: (randd), (gdp85,gdp60) and
## (invest, school, popgrowth). Block 1, i.e. the 1x1 block
## (randd), explains only 5.76% of the overall variance.
## Hence, discarding this block seems appropriate.

pla_obj = pla(OECDGrowth,cor=TRUE,thresholds = 0.5)
pla.drop_blocks(pla_obj, c(1)) ## drop block 1

## Sometimes, considering the blocks we keep rather than
## the blocks we want to discard might be more convenient.

pla.keep_blocks(pla_obj, c(2,3)) ## keep block 2 and block 3
}

```

---

pla.drop\_blocks      *Drop Blocks*

---

### Description

Used to pass the indices of the blocks we want to discard.

### Usage

```
pla.drop_blocks(object, blocks, ...)
```

### Arguments

object	a pla object.
blocks	a list of numeric values indicating the indices of the blocks that should be removed.
...	further arguments passed to or from other methods.

### Value

list of the following attributes:

x	a numeric matrix or data frame containing the reduced set of original variables.
cc_matrix	a numeric matrix or data frame which contains the conditional dispersion matrix. Depending on the pla procedure, this is either the conditional covariance matrix or the conditional correlation matrix.

**Examples**

```

if(requireNamespace("AER")){
  require(AER)
  data("OECDGrowth")

  pla(OECDGrowth,cor=TRUE,thresholds = 0.5)

  ## we obtain three blocks: (randd), (gdp85,gdp60) and
  ## (invest, school, popgrowth). Block 1, i.e. the 1x1 block
  ## (randd), explains only 5.76% of the overall variance.
  ## Hence, discarding this block seems appropriate.

  pla_obj = pla(OECDGrowth,cor=TRUE,thresholds = 0.5)
  pla.drop_blocks(pla_obj, c(1)) ## drop block 1
}

```

---

pla.keep_blocks	<i>Keep Blocks</i>
-----------------	--------------------

---

**Description**

Used to pass the indices of the blocks we want to keep (i.e. which we do not want to be discarded).

**Usage**

```
pla.keep_blocks(object, blocks, ...)
```

**Arguments**

object	a pla object.
blocks	a list of numeric values indicating the indices of the blocks that should be kept.
...	further arguments passed to or from other methods.

**Value**

list of the following attributes:

x	a numeric matrix or data frame containing the reduced set of original variables.
cc_matrix	a numeric matrix or data frame which contains the conditional dispersion matrix. Depending on the pla procedure, this is either the conditional covariance matrix or the conditional correlation matrix.

**Examples**

```

if(requireNamespace("AER")){
  require(AER)
  data("OECDGrowth")

  pla(OECDGrowth,cor=TRUE,thresholds = 0.5)

  ## we obtain three blocks: (randd), (gdp85,gdp60) and
  ## (invest, school, popgrowth). Block 1, i.e. the 1x1 block
  ## (randd), explains only 5.76% of the overall variance.
  ## Hence, discarding this block seems appropriate. Therefore,
  ## we keep block 2 and block 3

  pla_obj = pla(OECDGrowth,cor=TRUE,thresholds = 0.5)
  pla_obj$keep_blocks(pla_obj, c(2,3)) ## keep block 2 and block 3
}

```

---

print.pla

*Print Function for pla S3*


---

**Description**

Prints the blocks, threshold, threshold\_mode and the loadings.

**Usage**

```

## S3 method for class 'pla'
print(x, ...)

```

**Arguments**

x                    a pla object.  
...                    further arguments passed to or from other methods.

**Value**

A pla object which equals the input of x.

**Examples**

```

if(requireNamespace("AER")){
  require(AER)
  data("OECDGrowth")

  pla_obj = pla(OECDGrowth,cor=TRUE,thresholds = 0.5)
  print(pla_obj)
}

```

---

show,Block-method	<i>Block - Show Prints the blocks structure.</i>
-------------------	--

---

**Description**

Block - Show  
Prints the blocks structure.

**Usage**

```
## S4 method for signature 'Block'  
show(object)
```

**Arguments**

object            block.

**Value**

No return value.

**Examples**

```
block <- new("Block", features = c(2, 5), explained_variance = 0.03)  
print(block)
```

---

str,Block-method	<i>Block - str Generic function to create a string out of the blocks structure.</i>
------------------	---

---

**Description**

Block - str  
Generic function to create a string out of the blocks structure.

**Usage**

```
## S4 method for signature 'Block'  
str(object)
```

**Arguments**

object            block.

**Value**

A string representing the Block.

**Examples**

```
block <- new("Block", features = c(2, 5), explained_variance = 0.03)
str(block)
```

# Index

Block-class, [2](#)

pla, [2](#)

pla.drop\_blocks, [4](#)

pla.keep\_blocks, [5](#)

print.pla, [6](#)

show,Block-method, [7](#)

str,Block-method, [7](#)