

# Package: psfmi (via r-universe)

October 9, 2024

**Type** Package

**Depends** R ( $\geq$  4.0.0),

**Imports** ggplot2, norm, survival, mitools, pROC, rms, magrittr,  
rsample, mice, mitml, cvAUC, dplyr, purrr, tidyr, tibble,  
stringr, lme4, car

**Suggests** foreign ( $\geq$  0.8-80), knitr, rmarkdown, testthat ( $\geq$  3.0.0),  
bookdown, readr, gtools, covr

**Title** Prediction Model Pooling, Selection and Performance Evaluation  
Across Multiply Imputed Datasets

**Version** 1.4.0

**Description** Pooling, backward and forward selection of linear, logistic and Cox regression models in multiply imputed datasets. Backward and forward selection can be done from the pooled model using Rubin's Rules (RR), the D1, D2, D3, D4 and the median p-values method. This is also possible for Mixed models. The models can contain continuous, dichotomous, categorical and restricted cubic spline predictors and interaction terms between all these type of predictors. The stability of the models can be evaluated using (cluster) bootstrapping. The package further contains functions to pool model performance measures as ROC/AUC, Reclassification, R-squared, scaled Brier score, H&L test and calibration plots for logistic regression models. Internal validation can be done across multiply imputed datasets with cross-validation or bootstrapping. The adjusted intercept after shrinkage of pooled regression coefficients can be obtained. Backward and forward selection as part of internal validation is possible. A function to externally validate logistic prediction models in multiple imputed datasets is available and a function to compare models. For Cox models a strata variable can be included. Eekhout (2017) <[doi:10.1186/s12874-017-0404-7](https://doi.org/10.1186/s12874-017-0404-7)>. Wiel (2009) <[doi:10.1093/biostatistics/kxp011](https://doi.org/10.1093/biostatistics/kxp011)>. Marshall (2009) <[doi:10.1186/1471-2288-9-57](https://doi.org/10.1186/1471-2288-9-57)>.

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**License** GPL (>= 2)

**URL** <https://mwheymans.github.io/psfmi/>

**BugReports** <https://github.com/mwheymans/psfmi/issues/>

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Repository** <https://mwheymans.r-universe.dev>

**RemoteUrl** <https://github.com/mwheymans/psfmi>

**RemoteRef** HEAD

**RemoteSha** 6afb51f1f1d9d7df11e91e5ffc64b14c3a663e33

## Contents

anderson . . . . .	3
aortadis . . . . .	4
bmd . . . . .	5
bw_single . . . . .	5
chlrform . . . . .	7
chol_long . . . . .	8
chol_wide . . . . .	8
coxph_bw . . . . .	9
coxph_fw . . . . .	11
day2_dataset4_mi . . . . .	13
glm_bw . . . . .	14
glm_fw . . . . .	16
hipstudy . . . . .	18
hipstudy_external . . . . .	19
hoorn_basic . . . . .	20
hoslem_test . . . . .	21
infarct . . . . .	22
ipdna_md . . . . .	22
km_estimates . . . . .	23
km_fit . . . . .	25
lbpmicox . . . . .	26
lbpmilr . . . . .	27
lbpmilr_dev . . . . .	28
lbpmi_extval . . . . .	29
lbp_orig . . . . .	30
lungvolume . . . . .	31
mammaca . . . . .	31
men . . . . .	32
mivalex_lr . . . . .	33
nri_cox . . . . .	35

nri_est . . . . .	37
pool_auc . . . . .	38
pool_compare_models . . . . .	39
pool_D2 . . . . .	41
pool_D4 . . . . .	42
pool_intadj . . . . .	43
pool_performance . . . . .	44
pool_reclassification . . . . .	46
pool_RR . . . . .	46
psfmi_coxr . . . . .	47
psfmi_lm . . . . .	50
psfmi_lr . . . . .	54
psfmi_mm . . . . .	57
psfmi_mm_multiparm . . . . .	59
psfmi_perform . . . . .	61
psfmi_stab . . . . .	64
psfmi_validate . . . . .	66
risk_coxph . . . . .	69
rsq_nagel . . . . .	70
rsq_surv . . . . .	70
sbp_age . . . . .	71
sbp_qas . . . . .	72
scaled_brier . . . . .	72
smoking . . . . .	73
stab_single . . . . .	73
weight . . . . .	75

**Index** **76**

---

anderson	<i>Data from a placebo-controlled RCT with leukemia patients</i>
----------	------------------------------------------------------------------

---

**Description**

Data from a placebo-controlled RCT with leukemia patients

**Usage**

data(anderson)

**Format**

A data frame with 348 observations on the following 5 variables.

remission continuous: remission in weeks

status dichotomous

treatment dichotomous: 0=placebo, 1=verum

sex dichotomous: 0=female, 1=male

log\_wbc continuous: Log (number of white blood cells)

**Examples**

```
data(anderson)
## maybe str(anderson)
```

---

aortadis

*Dataset of patients with a aortadissection*

---

**Description**

Original dataset of patients with a aortadissection

**Usage**

```
data(aortadis)
```

**Format**

A data frame with 226 observations on the following 10 variables.

Gender dichotomous, 1=yes, 0=no

Age continuous

Age\_C categorical: 0 = < 50 years, 1 = 50-59 years, 2 = 60-69 years, 3 = 70-79 years, 4 = 80 years and older

Aortadis dichotomous, 1=yes, 0=no

Acute dichotomous, 1=yes, 0=no

Acute3 categorical: 0 = No, 1 = Little, 2 = Much

Stomach\_Ache dichotomous, 1=yes, 0=no

Hyper dichotomous, Hypertensio, 1=yes, 0=no

Smoking dichotomous, 1=yes, 0=no

Radiation dichotomous, 1=yes, 0=no

**Examples**

```
data(aortadis)
## maybe str(aortadis)
```

---

bmd	<i>Data of a non-experimental study in more than 300 elderly women</i>
-----	------------------------------------------------------------------------

---

**Description**

Data of a non-experimental study in more than 300 elderly women

**Usage**

```
data(bmd)
```

**Format**

A data frame with 348 observations on the following 5 variables.

bmd continuous

age continuous: years

menopaus continuous: age of menopause

weight continuous: weight in kg

walkscor dichotomous: score on a walking test, 0=normal, 1=impaired

**Examples**

```
data(bmd)
## maybe str(bmd)
```

---

bw_single	<i>Predictor selection function for backward selection of Linear and Logistic regression models.</i>
-----------	------------------------------------------------------------------------------------------------------

---

**Description**

bw\_single Backward selection of Linear and Logistic regression models using as selection method the likelihood-ratio Chi-square value.

**Usage**

```
bw_single(  
  data,  
  formula = NULL,  
  Outcome = NULL,  
  predictors = NULL,  
  p.crit = 1,  
  cat.predictors = NULL,  
  spline.predictors = NULL,
```

```

int.predictors = NULL,
keep.predictors = NULL,
nknots = NULL,
model_type = "binomial"
)

```

### Arguments

<code>data</code>	A data frame.
<code>formula</code>	A formula object to specify the model as normally used by <code>glm</code> . See under "Details" and "Examples" how these can be specified.
<code>Outcome</code>	Character vector containing the name of the outcome variable.
<code>predictors</code>	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, <code>age2</code> , <code>gnder10</code> , etc.
<code>p.crit</code>	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
<code>cat.predictors</code>	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
<code>spline.predictors</code>	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
<code>int.predictors</code>	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a ":" symbol.
<code>keep.predictors</code>	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.
<code>nknots</code>	A numerical vector that defines the number of knots for each spline predictor separately.
<code>model_type</code>	A character vector. If "binomial" a logistic regression model is used (default) and for "linear" a linear regression model is used.

### Details

A typical formula object has the form `Outcome ~ terms`. Categorical variables has to be defined as `Outcome ~ factor(variable)`, restricted cubic spline variables as `Outcome ~ rcs(variable, 3)`. Interaction terms can be defined as `Outcome ~ variable1*variable2` or `Outcome ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+".

### Value

An object of class `smods` (single models) from which the following objects can be extracted: original dataset as `data`, final selected model as `RR_model_final`, model at each selection step `RR_model_setp`, p-values at final step according to selection method as `multiparm_final`, and at each step as `multiparm_step`, formula object at final step as `formula_final`, and at each step as `formula_step` and for start model as `formula_initial`, predictors included at each selection step as `predictors_in`, predictors excluded at each step as `predictors_out`, and `Outcome`,

anova\_test, p.crit, call, model\_type, predictors\_final for names of predictors in final selection step and predictors\_initial for names of predictors in start model.

**Author(s)**

Martijn Heymans, 2020

**References**

<http://missingdatasolutions.rbind.io/>

---

chlrform	<i>Data about concentration of <math>\beta</math>2-microglobuline in urine as indicator for possible damage to the kidney</i>
----------	-------------------------------------------------------------------------------------------------------------------------------

---

**Description**

Data about concentration of  $\beta$ 2-microglobuline in urine as indicator for possible damage to the kidney

**Usage**

```
data(chlrform)
```

**Format**

A data frame with 348 observations on the following 5 variables.

pt\_id continuous

sport categorical: 0 = football player, 1 = outdoorswimmer and 2 = indoor swimmer)

gammagt continuous: liver damage

b2 continuous: beta2 microglobuline in mg per mol

age continuous: age in years

**Examples**

```
data(chlrform)
## maybe str(chlrform)
```

---

chol_long	<i>Long dataset of persons from the The Amsterdam Growth and Health Longitudinal Study (AGHLS)</i>
-----------	----------------------------------------------------------------------------------------------------

---

**Description**

Long dataset of persons from the The Amsterdam Growth and Health Longitudinal Study (AGHLS)

**Usage**

```
data(chol_long)
```

**Format**

A data frame with 588 observations on the following 7 variables.

ID continuous

fitness continuous

Smoking dichotomous, 1=yes, 0=no

Sex dichotomous

Time categorical

Cholesterol continuous

SumSkinfolds continuous

**Examples**

```
data(chol_long)
## maybe str(chol_long)
```

---

chol_wide	<i>Wide dataset of persons from the The Amsterdam Growth and Health Longitudinal Study (AGHLS)</i>
-----------	----------------------------------------------------------------------------------------------------

---

**Description**

Wide dataset of persons from the The Amsterdam Growth and Health Longitudinal Study (AGHLS)

**Usage**

```
data(chol_wide)
```



**Format**

A data frame with 147 observations on the following 7 variables.

ID continuous  
Cholesterol1 continuous  
SumSkinfolds1 continuous  
Cholesterol2 continuous  
SumSkinfolds2 continuous  
Cholesterol3 continuous  
SumSkinfolds3 continuous  
Cholesterol4 continuous  
SumSkinfolds4 continuous  
fitness continuous  
Smoking dichotomous  
Sex dichotomous

**Examples**

```
data(chol_wide)
## maybe str(chol_wide)
```

---

coxph\_bw

*Predictor selection function for backward selection of Cox regression models in single complete dataset.*

---

**Description**

coxph\_bw Backward selection of Cox regression models in single complete dataset using as selection method the partial likelihood-ratio statistic.

**Usage**

```
coxph_bw(  
  data,  
  formula = NULL,  
  status = NULL,  
  time = NULL,  
  predictors = NULL,  
  p.crit = 1,  
  cat.predictors = NULL,  
  spline.predictors = NULL,  
  int.predictors = NULL,  
  keep.predictors = NULL,  
  nknots = NULL  
)
```

**Arguments**

<code>data</code>	A data frame.
<code>formula</code>	A formula object to specify the model as normally used by <code>coxph</code> . See under "Details" and "Examples" how these can be specified.
<code>status</code>	The status variable, normally 0=censoring, 1=event.
<code>time</code>	Survival time.
<code>predictors</code>	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, <code>age2</code> , <code>gnder10</code> , etc.
<code>p.crit</code>	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
<code>cat.predictors</code>	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
<code>spline.predictors</code>	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
<code>int.predictors</code>	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a ":" symbol.
<code>keep.predictors</code>	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.
<code>nknots</code>	A numerical vector that defines the number of knots for each spline predictor separately.

**Details**

A typical formula object has the form `Surv(time, status) ~ terms`. Categorical variables has to be defined as `Surv(time, status) ~ factor(variable)`, restricted cubic spline variables as `Surv(time, status) ~ rcs(variable, 3)`. Interaction terms can be defined as `Surv(time, status) ~ variable1*variable2` or `Surv(time, status) ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+".

**Value**

An object of class `smods` (single models) from which the following objects can be extracted: original dataset as `data`, final selected model as `RR_model_final`, model at each selection step `RR_model`, p-values at final step `multiparm_final`, and at each step as `multiparm`, formula object at final step as `formula_final`, and at each step as `formula_step` and for start model as `formula_initial`, predictors included at each selection step as `predictors_in`, predictors excluded at each step as `predictors_out`, and `time`, `status`, `p.crit`, `call`, `model_type`, `predictors_final` for names of predictors in final selection step and `predictors_initial` for names of predictors in start model and `keep.predictors` for variables that are forced in the model during selection.

**Author(s)**

Martijn Heymans, 2021

**References**

<http://missingdatasolutions.rbind.io/>

**Examples**

```
lbpmicox1 <- subset(psfmi::lbpmicox, Impnr==1) # extract first imputed dataset
res_single <- coxph_fw(data=lbpmicox1, p.crit = 0.05, formula=Surv(Time, Status) ~
  Previous + Radiation + Onset + Age + Tampascale +
  Pain + JobControl + factor(Satisfaction),
  spline.predictors = "Function",
  nknots = 3)

res_single$RR_model_final
res_single$multiparm_final
```

---

coxph_fw	<i>Predictor selection function for forward selection of Cox regression models in single complete dataset.</i>
----------	----------------------------------------------------------------------------------------------------------------

---

**Description**

coxph\_fw Forward selection of Cox regression models in single complete dataset using as selection method the partial likelihood-ratio statistic.

**Usage**

```
coxph_fw(
  data,
  formula = NULL,
  status = NULL,
  time = NULL,
  predictors = NULL,
  p.crit = 1,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  nknots = NULL
)
```

**Arguments**

data	A data frame.
formula	A formula object to specify the model as normally used by coxph. See under "Details" and "Examples" how these can be specified.
status	The status variable, normally 0=censoring, 1=event.

<code>time</code>	Survival time.
<code>predictors</code>	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, <code>age2</code> , <code>gnder10</code> , etc.
<code>p.crit</code>	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
<code>cat.predictors</code>	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
<code>spline.predictors</code>	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
<code>int.predictors</code>	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a ":" symbol.
<code>keep.predictors</code>	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.
<code>nknots</code>	A numerical vector that defines the number of knots for each spline predictor separately.

### Details

A typical formula object has the form `Surv(time, status) ~ terms`. Categorical variables has to be defined as `Surv(time, status) ~ factor(variable)`, restricted cubic variables as `Surv(time, status) ~ rcs(variable, 3)`. Interaction terms can be defined as `Surv(time, status) ~ variable1*variable2` or `Surv(time, status) ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+".

### Value

An object of class `smods` (single models) from which the following objects can be extracted: original dataset as `data`, final selected model as `RR_model_final`, model at each selection step `RR_model`, p-values at final step `multiparm_final`, and at each step as `multiparm`, formula object at final step as `formula_final`, and at each step as `formula_step` and for start model as `formula_initial`, predictors included at each selection step as `predictors_in`, predictors excluded at each step as `predictors_out`, and `time`, `status`, `p.crit`, `call`, `model_type`, `predictors_final` for names of predictors in final selection step and `predictors_initial` for names of predictors in start model and `keep.predictors` for variables that are forced in the model during selection.

### Author(s)

Martijn Heymans, 2021

### References

<http://missingdatasolutions.rbind.io/>

**Examples**

```

lbpmicox1 <- subset(psfmi::lbpmicox, Impnr==1) # extract first imputed dataset
res_single <- coxph_bw(data=lbpmicox1, p.crit = 0.05, formula=Surv(Time, Status) ~
  Previous + Radiation + Onset + Age + Tampascale +
  Pain + JobControl + factor(Satisfaction),
  spline.predictors = "Function",
  nknots = 3)

res_single$RR_model_final
res_single$multiparm_final

```

---

day2_dataset4_mi	<i>Dataset of low back pain patients with missing values</i>
------------------	--------------------------------------------------------------

---

**Description**

Dataset of low back pain patients with missing values in 2 variables

**Usage**

```
data(day2_dataset4_mi)
```

**Format**

A data frame with 100 observations on the following 8 variables.

ID continuous: unique patient numbers  
Pain continuous: Pain intensity  
Tampa continuous: Fear of Movement scale  
Function continuous: Functional Status  
JobSocial continuous  
FAB continuous: Fear Avoidance Beliefs  
Gender dichotomous: 1 = male, 0 = female  
Radiation dichotomous: 1 = yes, 0 = no

**Examples**

```

data(day2_dataset4_mi)
## maybe str(day2_dataset4_mi)

```

---

glm_bw	<i>Function for backward selection of Linear and Logistic regression models.</i>
--------	----------------------------------------------------------------------------------

---

### Description

glm\_bw Backward selection of Linear and Logistic regression models in single dataset using as selection method the likelihood-ratio test.

### Usage

```
glm_bw(
  data,
  formula = NULL,
  Outcome = NULL,
  predictors = NULL,
  p.crit = 1,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  nknots = NULL,
  model_type = "binomial"
)
```

### Arguments

data	A data frame.
formula	A formula object to specify the model as normally used by glm. See under "Details" and "Examples" how these can be specified.
Outcome	Character vector containing the name of the outcome variable.
predictors	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, age2, gnder10, etc.
p.crit	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
cat.predictors	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
spline.predictors	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
int.predictors	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a “.” symbol.
keep.predictors	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.

nknots	A numerical vector that defines the number of knots for each spline predictor separately.
model_type	A character vector. If "binomial" a logistic regression model is used (default) and for "linear" a linear regression model is used.

### Details

A typical formula object has the form Outcome ~ terms. Categorical variables has to be defined as Outcome ~ factor(variable), restricted cubic spline variables as Outcome ~ rcs(variable, 3). Interaction terms can be defined as Outcome ~ variable1\*variable2 or Outcome ~ variable1 + variable2 + variable1:variable2. All variables in the terms part have to be separated by a "+".

### Value

An object of class smods (single models) from which the following objects can be extracted: original dataset as data, model at each selection step RR\_model, final selected model as RR\_model\_final, p-values at final step multiparm\_final, and at each step as multiparm, formula object at final step as formula\_final, and at each step as formula\_step and for start model as formula\_initial, predictors included at each selection step as predictors\_in, predictors excluded at each step as predictors\_out, and Outcome, p.crit, call, model\_type, predictors\_final for names of predictors in final selection step and predictors\_initial for names of predictors in start model and keep.predictors for variables that are forced in the model during selection.

### Author(s)

Martijn Heymans, 2021

### References

<http://missingdatasolutions.rbind.io/>

### See Also

[psfmi\\_perform](#)

### Examples

```
data1 <- subset(psfmi::lbpmilr, Impnr==1) # extract first imputed dataset
res_single <- glm_bw(data=data1, p.crit = 0.05, formula=Chronic ~
  Tampascale + Smoking + factor(Satisfaction), model_type="binomial")

res_single$RR_model_final

res_single <- glm_bw(data=data1, p.crit = 0.05, formula=Pain ~
  Tampascale + Smoking + factor(Satisfaction), model_type="linear")

res_single$RR_model_final
```

---

glm_fw	<i>Function for forward selection of Linear and Logistic regression models.</i>
--------	---------------------------------------------------------------------------------

---

### Description

glm\_fw Forward selection of Linear and Logistic regression models in single dataset using as selection method the likelihood-ratio test statistic.

### Usage

```
glm_fw(
  data,
  formula = NULL,
  Outcome = NULL,
  predictors = NULL,
  p.crit = 1,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  nknots = NULL,
  model_type = "binomial"
)
```

### Arguments

data	A data frame.
formula	A formula object to specify the model as normally used by glm. See under "Details" and "Examples" how these can be specified.
Outcome	Character vector containing the name of the outcome variable.
predictors	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, age2, gnder10, etc.
p.crit	A numerical scalar. P-value selection criterium. A value of 1 provides the full model without selection.
cat.predictors	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
spline.predictors	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
int.predictors	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a “.” symbol.
keep.predictors	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.



nknots	A numerical vector that defines the number of knots for each spline predictor separately.
model_type	A character vector. If "binomial" a logistic regression model is used (default) and for "linear" a linear regression model is used.

### Details

A typical formula object has the form Outcome ~ terms. Categorical variables has to be defined as Outcome ~ factor(variable), restricted cubic spline variables as Outcome ~ rcs(variable, 3). Interaction terms can be defined as Outcome ~ variable1\*variable2 or Outcome ~ variable1 + variable2 + variable1:variable2. All variables in the terms part have to be separated by a "+".

### Value

An object of class smods (single models) from which the following objects can be extracted: original dataset as data, model at each selection step RR\_model, final selected model as RR\_model\_final, p-values at final step multiparm\_final, and at each step as multiparm, formula object at final step as formula\_final, and at each step as formula\_step and for start model as formula\_initial, predictors included at each selection step as predictors\_in, predictors excluded at each step as predictors\_out, and Outcome, p.crit, call, model\_type, predictors\_final for names of predictors in final selection step and predictors\_initial for names of predictors in start model and keep.predictors for variables that are forced in the model during selection.

### Author(s)

Martijn Heymans, 2021

### References

<http://missingdatasolutions.rbind.io/>

### See Also

[psfmi\\_perform](#)

### Examples

```
data1 <- subset(psfmi::lbpmilr, Impnr==1) # extract first imputed dataset
res_single <- glm_fw(data=data1, p.crit = 0.05, formula=Chronic ~
  Tampascale + Smoking + factor(Satisfaction), model_type="binomial")

res_single$RR_model_final

res_single <- glm_fw(data=data1, p.crit = 0.05, formula=Pain ~
  Tampascale + Smoking + factor(Satisfaction), model_type="linear")

res_single$RR_model_final
```

---

hipstudy

*Dataset of elderly patients with a hip fracture*

---

**Description**

Original dataset of elderly patients with a hip fracture

**Usage**

```
data(hipstudy)
```

**Format**

A data frame with 426 observations on the following 18 variables.

pat\_id continuous: unique patient numbers

Gender dichotomous: 1 = male, 0 = female

Age continuous: Years

Mobility categorical: 1 = No tools, 2 = Stick / walker, 3 = Wheelchair / bed

Dementia dichotomous: 2=yes, 1=no

Home categorical: 1 = Independent, 2 = Elderly house, 3 = Nursring

Comorbidity continuous: Number of Co\_morbidities (0-4)

ASA continuous: ASA score (1-4)

Hemoglobine continuous: Hemoglobine pre-operative

Leucocytes continuous: Leucocytes preoperative

Thrombocytes continuous: Thrombocytes preoperative

CRP continuous: C-reactive protein (CRP) preoperative

Creatinine continuous: Creatinine preoperative

Urea continuous: Urea preoperative

Albumine continuous: Albumin preoperative

Fracture dichotomous: 1 = per or subtrochanter fracture, 0 = collum fracture

Delay continuous: time till operation in days

Mortality dichotomous: 1 = yes, 0 = no

**Examples**

```
data(hipstudy)
## maybe str(hipstudy)
```

---

hipstudy\_external      *External Dataset of elderly patients with a hip fracture*

---

**Description**

External dataset of elderly patients with a hip fracture

**Usage**

```
data(hipstudy_external)
```

**Format**

A data frame with 381 observations on the following 17 variables.

Gender dichotomous: 1 = male, 0 = female

Age continuous: Years

Mobility categorical: 1 = No tools, 2 = Stick / walker, 3 = Wheelchair / bed

Dementia dichotomous: 2=yes, 1=no

Home categorical: 1 = Independent, 2 = Elderly house, 3 = Nursening

Comorbidity continuous: Number of Co-morbidities

ASA continuous: ASA score

Hemoglobine continuous: Hemoglobine preoperative

Leucocytes continuous: Leucocytes preoperative

Thrombocytes continuous: Thrombocytes preoperative

CRP continuous: Creactive protein (CRP) preoperative

Creatinine continuous: Creatinine preoperative

Urea continuous: Urea preoperative

Albumine continuous: Albumin preoperative

Fracture dichotomous: 1 = per or subtrochanter fracture, 0 = collum fracture

Delay continuous: time till operation in days

Mortality dichotomous: 1 = yes, 0 = no

**Examples**

```
data(hipstudy_external)
## maybe str(hipstudy_external)
```

---

hoorn_basic	<i>Dataset of the Hoorn Study</i>
-------------	-----------------------------------

---

**Description**

Dataset of the Hoorn Study

**Usage**

```
data(hoorn_basic)
```

**Format**

A data frame with 250 observations on the following 12 variables.

patnr continuous

sbldsys1 continuous: Systolic Blood Pressure 1

sbldsys2 continuous: Systolic Blood Pressure 2

sbldds1 continuous: Diastolic Blood Pressure 1

sbldds2 continuous: Diastolic Blood Pressure 2

sex dichotomous: 1=male, 2=female

sfructo continuous: fructosamine level in the blood

sglucn continuous

dmknown dichotomous: 0=no, 1=yes

dmdiet dichotomous: 0=no, 1=yes

infarct dichotomous: 0=no, 1=yes

hypten dichotomous: 0=no, 1=yes

**Examples**

```
data(hoorn_basic)
## maybe str(hoorn_basic)
```

---

hoslem_test	<i>Calculates the Hosmer and Lemeshow goodness of fit test.</i>
-------------	-----------------------------------------------------------------

---

**Description**

hoslem\_test the Hosmer and Lemeshow goodness of fit test.

**Usage**

```
hoslem_test(y, yhat, g = 10)
```

**Arguments**

y	a vector of observations (0/1).
yhat	a vector of predicted probabilities.
g	Number of groups tested. Default is 10. Can not be < 3.

**Value**

The Chi-squared test statistic, the p-value, the observed and expected frequencies.

**Author(s)**

Martijn Heymans, 2021

**References**

Kleinman K and Horton NJ. (2014). SAS and R: Data Management, Statistical Analysis, and Graphics. 2nd Edition. Chapman & Hall/CRC.

**See Also**

[pool\\_performance](#)

**Examples**

```
fit <- glm(Mortality ~ Dementia + factor(Mobility) + ASA +
  Gender + Age, data=hipstudy, family=binomial)
pred <- predict(fit, type = "response")

hoslem_test(fit$y, pred)
```

---

infarct	<i>Data of a patient-control study regarding the relationship between MI and smoking</i>
---------	------------------------------------------------------------------------------------------

---

**Description**

Data of a patient-control study regarding the relationship between MI and smoking

**Usage**

```
data(infarct)
```

**Format**

A data frame with 420 observations on the following 10 variables.

ppnr continuous

infarct dichotomous: 1=yes, 0=no

smoking dichotomous: 1=yes, 0=no

alcohol categorical

active dichotomous: 1=active, 0=inactive

sex dichotomous: 1=male, 0=female

profession categorical: 1=epidemiologist, 2=statistician, 3=other

bmi continuous: body mass index

sys continuous: systolic blood pressure

dias continuous: diastolic blood pressure

**Examples**

```
data(infarct)
## maybe str(infarct)
```

---

ipdna_md	<i>Example dataset for the psfmi_mm function</i>
----------	--------------------------------------------------

---

**Description**

5 imputed datasets of the first 10 centres of the IPDNa dataset in the micemd package.

**Usage**

```
data(ipdna_md)
```

**Format**

A data frame with 13390 observations on the following 13 variables.

```
.imp a numeric vector
.id a numeric vector
centre cluster variable
gender dichotomous
bmi continuous
age continuous
sbp continuous
dbp continuous
hr continuous
lvef dichotomous
bnp categorical
afib continuous
bmi_cat categorical
```

**Examples**

```
data(ipdna_md)
## maybe str(ipdna_md)

#summary per study
by(ipdna_md, ipdna_md$centre, summary)
```

---

km\_estimates

*Kaplan-Meier risk estimates for Net Reclassification Index analysis*

---

**Description**

km\_estimates Kaplan-Meier risk estimates for Net Reclassification Index analysis for Cox Regression Models

**Usage**

```
km_estimates(data, p0, p1, time, status, t_risk, cutoff)
```

**Arguments**

data	Data frame with relevant predictors
p0	risk outcome probabilities for reference model.
p1	risk outcome probabilities for new model.
time	Character vector. Name of time variable.
status	Character vector. Name of status variable.
t_risk	Follow-up value to calculate cases, controls. See details.
cutoff	A numerical vector that defines the outcome probability cutoff values.

**Details**

Follow-up for which cases and controls are determined. For censored cases before this follow-up the expected risk of being a case is calculated by using the Kaplan-Meier value to calculate the expected number of cases. These expected numbers are used to calculate the NRI proportions. (These are not shown by function `nricens`).

**Value**

An object from which the following objects can be extracted:

- `data` dataset.
- `prob_orig` outcome risk probabilities at `t_risk` for reference model.
- `prob_new` outcome risk probabilities at `t_risk` for new model.
- `time` name of time variable.
- `status` name of status variable.
- `cutoff` cutoff value for survival probability.
- `t_risk` follow-up time used to calculate outcome (risk) probabilities.
- `reclass_totals` table with total reclassification numbers.
- `reclass_cases` table with reclassification numbers for cases.
- `reclass_controls` table with reclassification numbers for controls.
- `totals` totals of controls, cases, censored cases.
- `km_est` totals of cases calculated using Kaplan-Meiers risk estimates.
- `nri_est` reclassification measures.

**Author(s)**

Martijn Heymans, 2023

**References**

Cook NR, Ridker PM. Advances in measuring the effect of individual predictors of cardiovascular risk: the role of reclassification measures. *Ann Intern Med.* 2009;150(11):795-802.

Steyerberg EW, Pencina MJ. Reclassification calculations for persons with incomplete follow-up. *Ann Intern Med.* 2010;152(3):195-6 (author reply 196-7).

Pencina MJ, D'Agostino RB Sr, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Stat Med.* 2011;30(1):11-21

Inoue E (2018). `nricens`: NRI for Risk Prediction Models with Time to Event and Binary Response Data. R package version 1.6, <<https://CRAN.R-project.org/package=nricens>>.



**Examples**

```
library(survival)
lbpmicox1 <- subset(psfmi::lbpmicox, Impnr==1) # extract dataset

fit_cox0 <-
  coxph(Surv(Time, Status) ~ Duration + Pain, data=lbpmicox1, x=TRUE)
fit_cox1 <-
  coxph(Surv(Time, Status) ~ Duration + Pain + Function + Radiation,
        data=lbpmicox1, x=TRUE)

p0 <- risk_coxph(fit_cox0, t_risk=80)
p1 <- risk_coxph(fit_cox1, t_risk=80)

res_km <- km_estimates(data=lbpmicox1,
                       p0=p0,
                       p1=p1,
                       time = "Time",
                       status = "Status",
                       cutoff=0.45,
                       t_risk=80)
```

---

km\_fit

*Kaplan-Meier (KM) estimate at specific time point*

---

**Description**

Kaplan-Meier (KM) estimate at specific time point

**Usage**

```
km_fit(time, status, t_risk)
```

**Arguments**

time	Character vector. Name of time variable.
status	Character vector. Name of status variable.
t_risk	Follow-up value to calculate cases, controls. See details.

**Value**

KM estimate at specific time point

**Author(s)**

Martijn Heymans, 2023

**References**

Pencina MJ, D'Agostino RB Sr, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Stat Med.* 2011;30(1):11-21

Inoue E (2018). nricens: NRI for Risk Prediction Models with Time to Event and Binary Response Data. R package version 1.6, <<https://CRAN.R-project.org/package=nricens>>.

**See Also**

[km\\_fit](#)

---

lbpmicox

*Example dataset for psfmi\_coxr function*

---

**Description**

10 imputed datasets

**Usage**

```
data(lbpmicox)
```

**Format**

A data frame with 2650 observations on the following 18 variables.

Impnr a numeric vector

patnr a numeric vector

Status dichotomous event

Time continuous follow up time variable

Duration continuous

Previous dichotomous

Radiation dichotomous

Onset dichotomous

Age continuous

Tampascale continuous

Pain continuous

Function continuous

Satisfaction categorical

JobControl continuous

JobDemand continuous

Social continuous

Expectation a numeric vector

Expect\_cat categorical

**Examples**

```
data(lbpmicox)
## maybe str(lbpmicox)
```

---

lbpmlr

*Example dataset for psfmi\_lr function*

---

**Description**

10 imputed datasets

**Usage**

```
data(lbpmlr)
```

**Format**

A data frame with 1590 observations on the following 17 variables.

Impnr a numeric vector

ID a numeric vector

Chronic dichotomous

Gender dichotomous

Carrying categorical

Pain continuous

Tampascale continuous

Function continuous

Radiation dichotomous

Age continuous

Smoking dichotomous

Satisfaction categorical

JobControl continuous

JobDemands continuous

SocialSupport continuous

Duration continuous

BMI continuous

**Examples**

```
data(lbpmlr)
## maybe str(lbpmlr)
```

---

`lbpmlr_dev`*Example dataset for mivalex\_lr function*

---

**Description**

1 development dataset

**Usage**

```
data(lbpmlr_dev)
```

**Format**

A data frame with 108 observations on the following 16 variables.

ID a numeric vector  
Chronic dichotomous  
Gender dichotomous  
Carrying categorical  
Pain continuous  
Tampascale continuous  
Function continuous  
Radiation dichotomous  
Age continuous  
Smoking dichotomous  
Satisfaction categorical  
JobControl continuous  
JobDemands continuous  
SocialSupport continuous  
Duration continuous  
BMI continuous

**Examples**

```
data(lbpmlr_dev)  
## maybe str(lbpmlr_dev)
```

---

`lbpmi_extval`*Example dataset of Low Back Pain Patients for external validation*

---

**Description**

Five multiply imputed datasets

**Usage**`lbpmi_extval`**Format**

A data frame with 400 rows and 17 variables.

Impnr a numeric vector

ID a numeric vector

Chronic dichotomous

Gender dichotomous

Carrying categorical

Pain continuous

Tampascale continuous

Function continuous

Radiation dichotomous

Age continuous

Smoking dichotomous

Satisfaction categorical

JobControl continuous

JobDemands continuous

SocialSupport continuous

Duration continuous

BMI continuous

**Examples**

```
data(lbpmi_extval)
## maybe str(lbpmi_extval)\
```

---

lbp\_orig

*Example dataset for psfmi\_perform function, method boot\_MI*

---

**Description**

Original dataset with missing values

**Usage**

```
data(lbp_orig)
```

**Format**

A data frame with 159 observations on the following 15 variables.

Chronic dichotomous

Gender dichotomous

Carrying categorical

Pain continuous

Tampascale continuous

Function continuous

Radiation dichotomous

Age continuous

Smoking dichotomous

Satisfaction categorical

JobControl continuous

JobDemands continuous

SocialSupport continuous

Duration continuous

BMI continuous

**Examples**

```
data(lbp_orig)
## maybe str(lbp_orig)
```

---

`lungvolume`*Data of the development of lung and heartvolume of unborn babies*

---

**Description**

Data regarding the development of lung and heartvolume of unborn babies in the 18 till 34 week of pregnancy

**Usage**

```
data(lungvolume)
```

**Format**

A data frame with 152 observations on the following 6 variables.

`pat_id` continuous

`week` continuous: week pregnancy

`weight` continuous: weight in grams

`lungvol` continuous: lung volume

`heartvol` continuous: heart volume

`Nweek` categorical: Percentile Group of week

**Examples**

```
data(lungvolume)
## maybe str(lungvolume)
```

---

`mammaca`*Data of a study among women with breast cancer*

---

**Description**

Data of a study among women with breast cancer

**Usage**

```
data(mammaca)
```

**Format**

A data frame with 1207 observations on the following 10 variables.

id continuous  
 time continuous, Time (months)  
 status dichotomous: 1=yes, 0=no  
 er Estrogen Receptor Status, 1=positive, 0=negative  
 age continuous  
 histgrad categorical  
 ln\_yesno lymph nodes, 0=no, 1=yes  
 pathsd dichotomous: Pathological Tumor Size  
 pr dichotomous: Progesterone Receptor Status, 0=negative, 1=positive

**Examples**

```
data(mammaca)
## maybe str(mammaca)
```

---

men	<i>Data of 613 patients with meningitis</i>
-----	---------------------------------------------

---

**Description**

Data of 613 patients with meningitis

**Usage**

```
data(men)
```

**Format**

A data frame with 420 observations on the following 10 variables.

pt\_id continuous  
 sex dichotomous: 0=male, 1=female  
 predisposition dichotomous: 0=no, 1=yes  
 mensepsi categorical: disease characteristics at admission, 1=meningitis, 2=sepsis, 3=other  
 coma dichotomous: coma at admission, 0=no, 1=coma  
 diastol continuous: diastolic blood pressure at admission  
 course dichotomous: disease course, 0=alive, 1=deceased

**Examples**

```
data(men)
## maybe str(men)
```



---

mivalex_lr	<i>External Validation of logistic prediction models in multiply imputed datasets</i>
------------	---------------------------------------------------------------------------------------

---

## Description

mivalex\_lr External validation of logistic prediction models

## Usage

```
mivalex_lr(
  data.val = NULL,
  data.orig = NULL,
  nimp = 5,
  impvar = NULL,
  formula = NULL,
  lp.orig = NULL,
  cal.plot = FALSE,
  plot.indiv,
  val.check = FALSE,
  g = 10,
  groups_cal = 10,
  plot.method = "mean"
)
```

## Arguments

data.val	Data frame with stacked multiply imputed validation datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
data.orig	A single data frame containing the original dataset that was used to develop the model. Used to estimate the original regression coefficients in case lp.orig is not provided.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
formula	A formula object to specify the model as normally used by glm.
lp.orig	Numeric vector of the original coefficient values that are externally validated.
cal.plot	If TRUE a calibration plot is generated. Default is FALSE.
plot.indiv	This argument is deprecated; please use plot.method instead.
val.check	logical vector. If TRUE the names of the predictors of the LP are provided and can be used as information for the order of the coefficient values as input for lp.orig. If FALSE (default) validation procedure is executed with coefficient values fitted in the order as used under lp.orig.

<code>g</code>	A numerical scalar. Number of groups for the Hosmer and Lemeshow test. Default is 10.
<code>groups_cal</code>	A numerical scalar. Number of groups used on the calibration plot. Default is 10. If the range of predicted probabilities is low, less than 10 groups can be chosen.
<code>plot.method</code>	If "mean" one calibration plot is generated, first taking the mean of the linear predictor values across the multiply imputed datasets (default), if "individual" the calibration plot in each imputed dataset is plotted, if "overlay" calibration plots from each imputed datasets are plotted in one figure.

## Details

The following information of the externally validated model is provided: `calibrate` with information of `pooled_int` and `pooled_slope` that is the pooled linear predictor (LP), after the LP is freely estimated in each external imputed dataset  $\text{Outcome} \sim a + \text{LP}$  (provides information about miscalibration in intercept and slope), `pooled_offset_int` as  $\text{Outcome} \sim a + \text{offset}(\text{LP})$  and `pooled_offset_slope` as  $\text{Outcome} \sim a + \text{LP} + \text{offset}(\text{LP})$  with information about miscalibration in intercept and slope separately by using an offset procedure (see Steyerberg, p. 300), `coef_pooled` with the pooled coefficients when the model is freely estimated in imputed datasets, `ROC_pooled` ROC curve (back transformed after pooling log transformed ROC curves), `R2_pooled` Nagelkerke R-Square value (back transformed after pooling Fisher transformed values), `HLtest_pooled` Hosmer and Lemeshow Test (using function `pool_D2`). In addition information is provided about `nimp`, `impvar`, `formula`, `val_ckeck`, `g` and `coef_check`. When the external validation is very poor, the `R2` can become negative due to the poor fit of the model in the external dataset (in that case you may report a `R2` of zero).

## Value

A `mivalex_lr` object from which the following objects can be extracted: `calibrate` with information about mis-calibration in intercept and slope with and without offset procedure, `coef_pooled`, coefficients pooled, ROC results as `ROC`, R squared results as `R2`, Hosmer and Lemeshow test as `HL_test`, `nimp`, `formula`, `impvar`, `val.check`, `g`, `coef.check` and `groups_cal`.

## References

- F. Harrell. Regression Modeling Strategies. With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. 2nd Edition. Springer, New York, NY, 2015.
- EW. Steyerberg (2019). Clinical Prediction Models. A Practical Approach to Development, Validation, and Updating (2nd edition). Springer Nature Switzerland AG.
- Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.
- <http://missingdatasolutions.rbind.io/>

## Examples

```
mivalex_lr(data.val=lbpmlr, nimp=5, impvar="Impnr",
  formula = Chronic ~ Gender + factor(Carrying) + Function +
  Tampascale + Age, lp.orig=c(-10, -0.35, 1.00, 1.00, -0.04, 0.26, -0.01),
```

```
cal.plot=TRUE, val.check = FALSE)
```

---

nri\_cox

*Net Reclassification Index for Cox Regression Models*


---

## Description

nri\_cox Net Reclassification Index for Cox Regression Models

## Usage

```
nri_cox(data, formula0, formula1, t_risk, cutoff, B = FALSE, nboot = 10)
```

## Arguments

data	Data frame with relevant predictors
formula0	A formula object to specify the reference model as normally used by glm. See under "Details" and "Examples" how these can be specified.
formula1	A formula object to specify the new model as normally used by glm.
t_risk	Follow-up value to calculate cases, controls. See details.
cutoff	A numerical vector that defines the outcome probability cutoff values.
B	A logical scalar. If TRUE bootstrap confidence intervals are calculated, if FALSE only the NRI estimates are reported.
nboot	A numerical scalar. Number of bootstrap samples to derive the percentile bootstrap confidence intervals. Default is 10.

## Details

A typical formula object has the form Outcome ~ terms. Categorical variables has to be defined as Outcome ~ factor(variable), restricted cubic spline variables as Outcome ~ rcs(variable, 3). Interaction terms can be defined as Outcome ~ variable1\*variable2 or Outcome ~ variable1 + variable2 + variable1:variable2. All variables in the terms part have to be separated by a "+". If a formula object is used set predictors, cat.predictors, spline.predictors or int.predictors at the default value of NULL.

Follow-up for which cases and controls are determined. For censored cases before this follow-up the expected risk of being a case is calculated by using the Kaplan-Meier value to calculate the expected number of cases. These expected numbers are used to calculate the NRI proportions but are not shown by function nricens.

**Value**

An object from which the following objects can be extracted:

- data dataset.
- prob\_orig outcome risk probabilities at t\_risk for reference model.
- prob\_new outcome risk probabilities at t\_risk for new model.
- time name of time variable.
- status name of status variable.
- cutoff cutoff value for survival probability.
- t\_risk follow-up time used to calculate outcome (risk) probabilities.
- reclass\_totals table with total reclassification numbers.
- reclass\_cases table with reclassification numbers for cases.
- reclass\_controls table with reclassification numbers for controls.
- totals totals of controls, cases, censored cases.
- km\_est totals of cases calculated using Kaplan-Meiers risk estimates.
- nri\_est reclassification measures.

**Author(s)**

Martijn Heymans, 2023

**References**

Cook NR, Ridker PM. Advances in measuring the effect of individual predictors of cardiovascular risk: the role of reclassification measures. *Ann Intern Med.* 2009;150(11):795-802.

Steyerberg EW, Pencina MJ. Reclassification calculations for persons with incomplete follow-up. *Ann Intern Med.* 2010;152(3):195-6; author reply 196-7.

Pencina MJ, D'Agostino RB Sr, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Stat Med.* 2011;30(1):11-21

Inoue E (2018). nricens: NRI for Risk Prediction Models with Time to Event and Binary Response Data. R package version 1.6, <<https://CRAN.R-project.org/package=nricens>>.

**Examples**

```
library(survival)
lbpmicox1 <- subset(psfmi::lbpmicox, Impnr==1) # extract one dataset
risk_est <- nri_cox(data=lbpmicox1, formula0 = Surv(Time, Status) ~ Duration + Pain,
  formula1 = Surv(Time, Status) ~ Duration + Pain + Function + Radiation,
  t_risk = 80, cutoff=c(0.45), B=TRUE, nboot=10)
```

---

nri\_est

*Calculation of Net Reclassification Index measures*


---

**Description**

nri\_est Calculation of proportion of Reclassified persons and NRI for Cox Regression Models

**Usage**

```
nri_est(data, p0, p1, time, status, t_risk, cutoff)
```

**Arguments**

data	Data frame with relevant predictors
p0	risk outcome probabilities for reference model.
p1	risk outcome probabilities for new model.
time	Character vector. Name of time variable.
status	Character vector. Name of status variable.
t_risk	Follow-up value to calculate cases, controls. See details.
cutoff	A numerical vector that defines the outcome probability cutoff values.

**Details**

Follow-up for which cases and controls are determined. For censored cases before this follow-up the expected risk of being a case is calculated by using the Kaplan-Meier value to calculate the expected number of cases. These expected numbers are used to calculate the NRI proportions but are not shown by function nri\_cens.

**Value**

An object from which the following objects can be extracted:

- prop\_up\_case proportion of cases reclassified upwards.
- prop\_down\_case proportion of cases reclassified downwards.
- prop\_up\_ctr proportion of controls reclassified upwards.
- prop\_down\_ctr proportion of controls reclassified downwards.
- nri\_plus proportion reclassified for events.
- nri\_min proportion reclassified for nonevents.
- nri net reclassification improvement.

**Author(s)**

Martijn Heymans, 2023

## References

Cook NR, Ridker PM. Advances in measuring the effect of individual predictors of cardiovascular risk: the role of reclassification measures. *Ann Intern Med.* 2009;150(11):795-802.

Steyerberg EW, Pencina MJ. Reclassification calculations for persons with incomplete follow-up. *Ann Intern Med.* 2010;152(3):195-6; author reply 196-7.

Pencina MJ, D'Agostino RB Sr, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Stat Med.* 2011;30(1):11-21

Inoue E (2018). nricsens: NRI for Risk Prediction Models with Time to Event and Binary Response Data. R package version 1.6, <<https://CRAN.R-project.org/package=nricesens>>.

## Examples

```
library(survival)
lbpmicox1 <- subset(psfmi::lbpmicox, Impnr==1) # extract dataset

fit_cox0 <-
  coxph(Surv(Time, Status) ~ Duration + Pain, data=lbpmicox1, x=TRUE)
fit_cox1 <-
  coxph(Surv(Time, Status) ~ Duration + Pain + Function + Radiation,
        data=lbpmicox1, x=TRUE)

p0 <- risk_coxph(fit_cox0, t_risk=80)
p1 <- risk_coxph(fit_cox1, t_risk=80)

nri <- nri_est(data=lbpmicox1,
               p0=p0,
               p1=p1,
               time = "Time",
               status = "Status",
               cutoff=0.45,
               t_risk=80)
```

---

pool\_auc

*Calculates the pooled C-statistic (Area Under the ROC Curve) across Multiply Imputed datasets*

---

## Description

pool\_auc Calculates the pooled C-statistic and 95 by using Rubin's Rules. The C-statistic values are log transformed before pooling.

## Usage

```
pool_auc(est_auc, est_se, nimp = 5, log_auc = TRUE)
```

**Arguments**

est_auc	A list of C-statistic (AUC/ROC) values estimated in Multiply Imputed datasets.
est_se	A list of standard errors of C-statistic values estimated in Multiply Imputed datasets.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
log_auc	If TRUE natural logarithmic transformation is applied before pooling and finally back transformed. If FALSE the raw values are pooled.

**Value**

The pooled C-statistic value and the 95

**Author(s)**

Martijn Heymans, 2021

**See Also**

[psfmi\\_perform](#), [pool\\_performance](#)

---

pool\_compare\_models *Compare the fit and performance of prediction models across Multiply Imputed data*

---

**Description**

pool\_compare\_model Compares the fit and performance of prediction models in multiply imputed data sets by using clinical important performance measures

**Usage**

```
pool_compare_models(  
  pobj,  
  compare.predictors = NULL,  
  compare.group = NULL,  
  cutoff = 0.5,  
  boot_auc = FALSE,  
  nboot = 1000  
)
```

**Arguments**

<code>pobj</code>	An object of class <code>pmods</code> (pooled models), produced by a previous call to <code>psfmi_lr.compare.predictors</code>
<code>compare.predictors</code>	Character vector with the names of the predictors that are compared. See details.
<code>compare.group</code>	Character vector with the names of the group of predictors that are compared. See details.
<code>cutoff</code>	A numerical scalar. Cutoff used for the categorical NRI value. More than one cutoff value can be used.
<code>boot_auc</code>	If TRUE the standard error of the AUC is calculated with stratified bootstrapping. If FALSE (is default), the standard error is calculated with De Long's method.
<code>nboot</code>	A numerical scalar. The number of bootstrap samples for the AUC standard error, used when <code>boot_auc</code> is TRUE. Default is 1000.

**Details**

The fit of the models are compared by using the D3 method for pooling Likelihood ratio statistics (method of Meng and Rubin). The pooled AIC difference is calculated according to the formula  $AIC = D - 2 * p$ , where  $D$  is the pooled likelihood ratio tests of constrained models (numerator in D3 statistic) and  $p$  is the difference in number of parameters between the full and restricted models that are compared. The pooled AUC difference is calculated, after the standard error is obtained in each imputed data set by method DeLong or bootstrapping. The NRI categorical and continuous and IDI are calculated in each imputed data set and pooled.

**Value**

An object from which the following objects can be extracted:

- `DR_stats` p-value of the D3 statistic, the D3 statistic, LRT fixed is the likelihood Ratio test value of the constrained models.
- `stats_compare` Mean of `LogLik0`, `LogLik1`, `AIC0`, `AIC1`, `AIC_diff` values of the restricted (containing a 0) and full models (containing a 1).
- `NRI` pooled values for the categorical and continuous Net Reclassification improvement values and the Integrated Discrimination improvement.
- `AUC_stats` Pooled Area Under the Curve of restricted and full models.
- `AUC_diff` Pooled difference in AUC.
- `formula_test` regression formula of full model.
- `cutoff` Cutoff value used for reclassification values.
- `formula_null` regression formula of null model
- `compare_predictors` Predictors used in full model.
- `compare_group` group of predictors used in full model.



## References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. *BMC Med Res Methodol.* 2017;17(1):129.

Consentino F, Claeskens G. Order Selection tests with multiply imputed data *Computational Statistics and Data Analysis.*2010;54:2284-2295.

## Examples

```
pool_lr <- psfmi_lr(data=lbpmlr, p.crit = 1, direction="FW", nimp=10, impvar="Impnr",
  Outcome="Chronic", predictors=c("Radiation"), cat.predictors = ("Satisfaction"),
  int.predictors = NULL, spline.predictors="Tampascale", nknots=3, method="D1")

res_compare <- pool_compare_models(pool_lr, compare.predictors = c("Pain", "Duration",
  "Function"), cutoff = 0.4)
res_compare
```

---

pool\_D2

*Combines the Chi Square statistics across Multiply Imputed datasets*

---

## Description

pool\_D2 The D2 statistic to combine the Chi square values across Multiply Imputed datasets.

## Usage

```
pool_D2(dw, v)
```

## Arguments

dw a vector of Chi square values obtained after multiple imputation.  
v single value for the degrees of freedom of the Chi square statistic.

## Value

The pooled chi square values as the D2 statistic, the p-value, the numerator, df1 and denominator, df2 degrees of freedom for the F-test.

## Author(s)

Martijn Heymans, 2021

## References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. *BMC Med Res Methodol.* 2017;17(1):129.

Van Buuren S. (2018). *Flexible Imputation of Missing Data*. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

## Examples

```
pool_D2(c(2.25, 3.95, 6.24, 5.27, 2.81), 4)
```

---

pool_D4	<i>Pools the Likelihood Ratio tests across Multiply Imputed datasets (method D4)</i>
---------	--------------------------------------------------------------------------------------

---

## Description

pool\_D4 The D4 statistic to combine the likelihood ratio tests (LRT) across Multiply Imputed datasets according method D4.

## Usage

```
pool_D4(data, nimp, impvar, fm0, fm1, robust = TRUE, model_type = "binomial")
```

## Arguments

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
fm0	the null model.
fm1	the (nested) model to compare. Must be larger than the null model.
robust	if TRUE a robust LRT is used (algorithm 1 in Chan and Meng), otherwise algorithm 2 is used.
model_type	if TRUE (default) a logistic regression model is fitted, otherwise a linear regression model is used

## Value

The D4 statistic, the numerator, df1 and denominator, df2 degrees of freedom for the F-test.

**Author(s)**

Martijn Heymans, 2021

**References**

Chan, K. W., & Meng, X.-L. (2019). Multiple improvements of multiple imputation likelihood ratio tests. ArXiv:1711.08822 [Math, Stat]. <https://arxiv.org/abs/1711.08822>

Grund, Simon, Oliver Lüdtke, and Alexander Robitzsch. 2021. "Pooling Methods for Likelihood Ratio Tests in Multiply Imputed Data Sets." PsyArXiv. January 29. doi:10.31234/osf.io/d459g.

**Examples**

```
fm0 <- Chronic ~ BMI + factor(Carrying) +
  Satisfaction + SocialSupport + Smoking
fm1 <- Chronic ~ BMI + factor(Carrying) +
  Satisfaction + SocialSupport + Smoking +
  Radiation

psfmi::pool_D4(data=lbpmlr, nimp=10, impvar="Impnr",
  fm0=fm0, fm1=fm1, robust = TRUE)
```

---

pool_intadj	<i>Provides pooled adjusted intercept after shrinkage of pooled coefficients in multiply imputed datasets</i>
-------------	---------------------------------------------------------------------------------------------------------------

---

**Description**

pool\_intadj Provides pooled adjusted intercept after shrinkage of the pooled coefficients in multiply imputed datasets for models selected with the psfmi\_lr function and internally validated with the psfmi\_perform function.

**Usage**

```
pool_intadj(pobj, shrinkage_factor)
```

**Arguments**

pobj	An object of class smodsmi (selected models in multiply imputed datasets), produced by a previous call to psfmi_lr.
shrinkage_factor	A numerical scalar. Shrinkage factor value as a result of internal validation with the psfmi_perform function.

**Details**

The function provides the pooled adjusted intercept after shrinkage of pooled regression coefficients in multiply imputed datasets. The function is only available for logistic regression models without random effects.

**Value**

A `pool_intadj` object from which the following objects can be extracted: `int_adj`, the adjusted intercept value, `coef_shrink_pooled`, the pooled regression coefficients after shrinkage, `coef_orig_pooled`, the (original) pooled regression coefficients before shrinkage and `nimp`, the number of imputed datasets.

**References**

F. Harrell. *Regression Modeling Strategies. With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis* (2nd edition). Springer, New York, NY, 2015.

EW. Steyerberg (2019). *Clinical Prediction MOdels. A Practical Approach to Development, Validation, and Updating* (2nd edition). Springer Nature Switzerland AG.

<http://missingdatasolutions.rbind.io/>

**Examples**

```
res_psfmi <- psfmi_lr(data=lbpmlr, nimp=5, impvar="Impnr", Outcome="Chronic",
  predictors=c("Gender", "Pain", "Tampascale", "Smoking", "Function",
    "Radiation", "Age"), p.crit = 1, method="D1", direction="BW")
res_psfmi$RR_Model
```

```
## Not run:
set.seed(100)
res_val <- psfmi_perform(res_psfmi, method = "MI_boot", nboot=10,
  int_val = TRUE, p.crit=1, cal.plot=FALSE, plot.indiv=FALSE)
res_val$intval
```

```
res <- pool_intadj(res_psfmi, shrinkage_factor = 0.9774058)
res$int_adj
res$coef_shrink_pooled
```

```
## End(Not run)
```

---

pool\_performance

*Pooling performance measures across multiply imputed datasets*

---

**Description**

`pool_performance` Pooling performance measures for logistic and Cox regression models.

**Usage**

```
pool_performance(
  data,
  formula,
  nimp,
  impvar,
```

```

    plot.indiv,
    model_type = "binomial",
    cal.plot = TRUE,
    plot.method = "mean",
    groups_cal = 10
  )

```

## Arguments

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset.
formula	A formula object to specify the model as normally used by glm or coxph. See details.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
plot.indiv	This argument is deprecated; please use plot.method instead.
model_type	If "binomial" (default), performance measures are calculated for logistic regression models, if "survival" for Cox regression models. See details.
cal.plot	If TRUE a calibration plot is generated. Default is TRUE. model_type must be "binomial".
plot.method	If "mean" one calibration plot is generated, first taking the mean of the linear predictor across the multiply imputed datasets (default), if "individual" the calibration plot of each imputed dataset is plotted, if "overlay" calibration plots from each imputed datasets are plotted in one figure.
groups_cal	A numerical scalar. Number of groups used on the calibration plot and. for the Hosmer and Lemeshow test. Default is 10. If the range of predicted probabilities. is low, less than 10 groups can be chosen, but not < 3.

## Details

A typical formula object for logistic regression models has the form `formula = Outcome ~ terms`. For Cox regression models the formula object must be defined as `Surv(time, status) ~ terms`. For Cox models calibration curves can not be generated.

## Examples

```

perf <- pool_performance(data=lbpmlr, nimp=5, impvar="Impnr",
  formula = Chronic ~ Gender + Pain + Tampascale +
  Smoking + Function + Radiation + Age + factor(Carrying),
  cal.plot=TRUE, plot.method="mean",
  groups_cal=10, model_type="binomial")

perf$ROC_pooled
perf$R2_pooled

```

---

pool\_reclassification *Function to pool NRI measures over Multiply Imputed datasets*

---

### Description

pool\_reclassification Function to pool categorical and continuous NRI and IDI over Multiply Imputed datasets

### Usage

```
pool_reclassification(datasets, cutoff = cutoff)
```

### Arguments

datasets	a list of data frames corresponding to the multiply imputed datasets, within each dataset in the first column the predicted probabilities of model 1, in the second column those of model 2 and in the third column the observed outcomes coded as '0' and '1'.
cutoff	cutoff value for the categorical NRI, must lie between 0 and 1.

### Details

This function is called by the function pool\_compare\_model

### Author(s)

Martijn Heymans, 2020

---

pool\_RR *Function to combine estimates by using Rubin's Rules*

---

### Description

pool\_RR Rubin's Rules

### Usage

```
pool_RR(est, se, conf.level = 0.95, n, k)
```

### Arguments

est	A vector of multiple parameter estimates
se	A vector of multiple standard error estimates
conf.level	desired confidence limits
n	sample size in completed dataset
k	number of parameters to pool

**Author(s)**

Martijn Heymans, 2021

---

psfmi_coxr	<i>Pooling and Predictor selection function for backward or forward selection of Cox regression models across multiply imputed data.</i>
------------	------------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

psfmi\_coxr Pooling and backward or forward selection of Cox regression prediction models in multiply imputed data using selection methods D1, D2 and MPR.

**Usage**

```
psfmi_coxr(
  data,
  formula = NULL,
  nimp = 5,
  impvar = NULL,
  time,
  status,
  predictors = NULL,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  strata.variable = NULL,
  nknots = NULL,
  p.crit = 1,
  method = "RR",
  direction = NULL
)
```

**Arguments**

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
formula	A formula object to specify the model as normally used by coxph. See under "Details" and "Examples" how these can be specified. If a formula object is used set predictors, cat.predictors, spline.predictors or int.predictors at the default value of NULL.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.

<code>time</code>	Survival time.
<code>status</code>	The status variable, normally 0=censoring, 1=event.
<code>predictors</code>	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, <code>age2</code> , <code>gnder10</code> , etc.
<code>cat.predictors</code>	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
<code>spline.predictors</code>	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
<code>int.predictors</code>	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a ":" symbol.
<code>keep.predictors</code>	A single string or a vector of strings including the variables that are forced in the model during predictor selection. Categorical and interaction variables are allowed.
<code>strata.variable</code>	A single string including the strata variable. See under "Details" and "Examples" how such a variable can be specified.
<code>nknots</code>	A numerical vector that defines the number of knots for each spline predictor separately.
<code>p.crit</code>	A numerical scalar. P-value selection criterion. A value of 1 provides the pooled model without selection.
<code>method</code>	A character vector to indicate the pooling method for p-values to pool the total model or used during predictor selection. This can be "RR", "D1", "D2", or "MPR". See details for more information. Default is "RR".
<code>direction</code>	The direction of predictor selection, "BW" means backward selection and "FW" means forward selection.

## Details

The basic pooling procedure to derive pooled coefficients, standard errors, 95 confidence intervals and p-values is Rubin's Rules (RR). However, RR is only possible when the model included continuous or dichotomous variables. Specific procedures are available when the model also included categorical (> 2 categories) or restricted cubic spline variables. These pooling methods are: "D1" is pooling of the total covariance matrix, "D2" is pooling of Chi-square values and "MPR" is pooling of median p-values (MPR rule). Spline regression coefficients are defined by using the `rcs` function for restricted cubic splines of the `rms` package. A minimum number of 3 knots as defined under `knots` is required.

A typical formula object has the form `Surv(time, status) ~ terms`. Categorical variables has to be defined as `Surv(time, status) ~ factor(variable)`, restricted cubic spline variables as `Surv(time, status) ~ rcs(variable, 3)`. Interaction terms can be defined as `Surv(time, status) ~ variable1*variable2` or `Surv(time, status) ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+". If a formula object is used set `predictors`, `cat.predictors`, `spline.predictors` or `int.predictors` at the default value of NULL. For Cox models also a strata variable is allowed to include in the formula as `Surv(time, status) ~ strata(variable) + terms`.



**Value**

An object of class `pmods` (multiply imputed models) from which the following objects can be extracted:

- `data` imputed datasets
- `RR_model` pooled model at each selection step
- `RR_model_final` final selected pooled model
- `multiparm` pooled p-values at each step according to pooling method
- `multiparm_final` pooled p-values at final step according to pooling method
- `multiparm_out` (only when `direction = "FW"`) pooled p-values of removed predictors
- `formula_step` formula object at each step
- `formula_final` formula object at final step
- `formula_initial` formula object at final step
- `predictors_in` predictors included at each selection step
- `predictors_out` predictors excluded at each step
- `impvar` name of variable used to distinguish imputed datasets
- `nimp` number of imputed datasets
- `status` name of the status variable
- `time` name of the time variable
- `method` selection method
- `p.crit` p-value selection criterium
- `call` function call
- `model_type` type of regression model used
- `direction` direction of predictor selection
- `predictors_final` names of predictors in final selection step
- `predictors_initial` names of predictors in start model
- `keep.predictors` names of predictors that were forced in the model
- `strata.variable` names of the strata variable in the model

**Vignettes**

[https://mwheymans.github.io/psfmi/articles/psfmi\\_CoxModels.html](https://mwheymans.github.io/psfmi/articles/psfmi_CoxModels.html)

**Author(s)**

Martijn Heymans, 2020

## References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. *BMC Med Res Methodol.* 2017;17(1):129.

Enders CK (2010). *Applied missing data analysis*. New York: The Guilford Press.

van de Wiel MA, Berkhof J, van Wieringen WN. Testing the prediction error difference between 2 predictors. *Biostatistics.* 2009;10:550-60.

Marshall A, Altman DG, Holder RL, Royston P. Combining estimates of interest in prognostic modelling studies after multiple imputation: current practice and guidelines. *BMC Med Res Methodol.* 2009;9:57.

Van Buuren S. (2018). *Flexible Imputation of Missing Data*. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

EW. Steyerberg (2019). *Clinical Prediction MOdels. A Practical Approach to Development, Validation, and Updating* (2nd edition). Springer Nature Switzerland AG.

<http://missingdatasolutions.rbind.io/>

## Examples

```
pool_coxr <- psfmi_coxr(formula = Surv(Time, Status) ~ Pain + Tampascale +
  Radiation + Radiation*Pain + Age + Duration + Previous,
  data=lbpmicox, p.crit = 0.05, direction="BW", nimp=5, impvar="Impnr",
  keep.predictors = "Radiation*Pain", method="D1")
```

```
pool_coxr$RR_model_final
```

```
pool_coxr <- psfmi_coxr(formula = Surv(Time, Status) ~ Pain + Tampascale +
  Previous + strata(Radiation), data=lbpmicox, p.crit = 0.05,
  direction="BW", nimp=5, impvar="Impnr", method="D1")
```

```
pool_coxr$RR_model_final
```

---

psfmi\_lm

*Pooling and Predictor selection function for backward or forward selection of Linear regression models across multiply imputed data.*

---

## Description

psfmi\_lm Pooling and backward or forward selection of Linear regression models in multiply imputed data using selection methods RR, D1, D2 and MPR.

**Usage**

```
psfmi_lm(
  data,
  formula = NULL,
  nimp = 5,
  impvar = NULL,
  Outcome = NULL,
  predictors = NULL,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  nknots = NULL,
  p.crit = 1,
  method = "RR",
  direction = NULL
)
```

**Arguments**

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
formula	A formula object to specify the model as normally used by glm. See under "Details" and "Examples" how these can be specified. If a formula object is used set predictors, cat.predictors, spline.predictors or int.predictors at the default value of NULL.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
Outcome	Character vector containing the name of the continuous outcome variable.
predictors	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, age2, gender10, etc.
cat.predictors	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
spline.predictors	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
int.predictors	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a ":" symbol.
keep.predictors	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.
nknots	A numerical vector that defines the number of knots for each spline predictor separately.

<code>p.crit</code>	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
<code>method</code>	A character vector to indicate the pooling method for p-values to pool the total model or used during predictor selection. This can be "RR", "D1", "D2", "D3" or "MPR". See details for more information. Default is "RR".
<code>direction</code>	The direction of predictor selection, "BW" means backward selection and "FW" means forward selection.

### Details

The basic pooling procedure to derive pooled coefficients, standard errors, 95 confidence intervals and p-values is Rubin's Rules (RR). However, RR is only possible when the model included continuous or dichotomous variables. Specific procedures are available when the model also included categorical (> 2 categories) or restricted cubic spline variables. These pooling methods are: "D1" is pooling of the total covariance matrix, "D2" is pooling of Chi-square values and "MPR" is pooling of median p-values (MPR rule). Spline regression coefficients are defined by using the `rcs` function for restricted cubic splines of the `rms` package. A minimum number of 3 knots as defined under `knots` is required.

A typical formula object has the form `Outcome ~ terms`. Categorical variables has to be defined as `Outcome ~ factor(variable)`, restricted cubic spline variables as `Outcome ~ rcs(variable, 3)`. Interaction terms can be defined as `Outcome ~ variable1*variable2` or `Outcome ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+". If a formula object is used set `predictors`, `cat.predictors`, `spline.predictors` or `int.predictors` at the default value of `NULL`.

### Value

An object of class `pmods` (multiply imputed models) from which the following objects can be extracted:

- `data` imputed datasets
- `RR_model` pooled model at each selection step
- `RR_model_final` final selected pooled model
- `multiarm` pooled p-values at each step according to pooling method
- `multiarm_final` pooled p-values at final step according to pooling method
- `multiarm_out` (only when `direction = "FW"`) pooled p-values of removed predictors
- `formula_step` formula object at each step
- `formula_final` formula object at final step
- `formula_initial` formula object at final step
- `predictors_in` predictors included at each selection step
- `predictors_out` predictors excluded at each step
- `impvar` name of variable used to distinguish imputed datasets
- `nimp` number of imputed datasets
- `Outcome` name of the outcome variable

- method selection method
- p.crit p-value selection criterium
- call function call
- model\_type type of regression model used
- direction direction of predictor selection
- predictors\_final names of predictors in final selection step
- predictors\_initial names of predictors in start model
- keep.predictors names of predictors that were forced in the model

### Author(s)

Martijn Heymans, 2021

### References

Enders CK (2010). Applied missing data analysis. New York: The Guilford Press.

van de Wiel MA, Berkhof J, van Wieringen WN. Testing the prediction error difference between 2 predictors. *Biostatistics*. 2009;10:550-60.

Marshall A, Altman DG, Holder RL, Royston P. Combining estimates of interest in prognostic modelling studies after multiple imputation: current practice and guidelines. *BMC Med Res Methodol*. 2009;9:57.

Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

EW. Steyerberg (2019). Clinical Prediction MOdels. A Practical Approach to Development, Validation, and Updating (2nd edition). Springer Nature Switzerland AG.

<http://missingdatasolutions.rbind.io/>

### Examples

```
pool_lm <- psfmi_lm(data=lbpmlr, formula = Pain ~ factor(Satisfaction) +
  rcs(Tampascale,3) + Radiation +
  Radiation*factor(Satisfaction) + Age + Duration + BMI,
  p.crit = 0.05, direction="FW", nimp=5, impvar="Impnr",
  keep.predictors = c("Radiation*factor(Satisfaction)", "Age"), method="D1")

pool_lm$RR_model_final
```

---

psfmi_lr	<i>Pooling and Predictor selection function for backward or forward selection of Logistic regression models across multiply imputed data.</i>
----------	-----------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

psfmi\_lr Pooling and backward or forward selection of Logistic regression models across multiply imputed data using selection methods RR, D1, D2, D3, D4 and MPR.

### Usage

```
psfmi_lr(
  data,
  formula = NULL,
  nimp = 5,
  impvar = NULL,
  Outcome = NULL,
  predictors = NULL,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  nknots = NULL,
  p.crit = 1,
  method = "RR",
  direction = NULL
)
```

### Arguments

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
formula	A formula object to specify the model as normally used by glm. See under "Details" and "Examples" how these can be specified. If a formula object is used set predictors, cat.predictors, spline.predictors or int.predictors at the default value of NULL.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
Outcome	Character vector containing the name of the outcome variable.
predictors	Character vector with the names of the predictor variables. At least one predictor variable has to be defined. Give predictors unique names and do not use predictor name combinations with numbers as, age2, gender10, etc.

<code>cat.predictors</code>	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
<code>spline.predictors</code>	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
<code>int.predictors</code>	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a “:” symbol.
<code>keep.predictors</code>	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.
<code>nknots</code>	A numerical vector that defines the number of knots for each spline predictor separately.
<code>p.crit</code>	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
<code>method</code>	A character vector to indicate the pooling method for p-values to pool the total model or used during predictor selection. This can be "RR", "D1", "D2", "D3", "D4", or "MPR". See details for more information. Default is "RR".
<code>direction</code>	The direction of predictor selection, "BW" means backward selection and "FW" means forward selection.

## Details

The basic pooling procedure to derive pooled coefficients, standard errors, 95 confidence intervals and p-values is Rubin’s Rules (RR). However, RR is only possible when the model included continuous or dichotomous variables. Specific procedures are available when the model also included categorical (> 2 categories) or restricted cubic spline variables. These pooling methods are: “D1” is pooling of the total covariance matrix, “D2” is pooling of Chi-square values, “D3” and “D4” is pooling Likelihood ratio statistics (method of Meng and Rubin) and “MPR” is pooling of median p-values (MPR rule). Spline regression coefficients are defined by using the `rcs` function for restricted cubic splines of the `rms` package. A minimum number of 3 knots as defined under `knots` is required.

A typical formula object has the form `Outcome ~ terms`. Categorical variables has to be defined as `Outcome ~ factor(variable)`, restricted cubic spline variables as `Outcome ~ rcs(variable, 3)`. Interaction terms can be defined as `Outcome ~ variable1*variable2` or `Outcome ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+". If a formula object is used set predictors, `cat.predictors`, `spline.predictors` or `int.predictors` at the default value of NULL.

## Value

An object of class `pmods` (multiply imputed models) from which the following objects can be extracted:

- `data` imputed datasets
- `RR_model` pooled model at each selection step
- `RR_model_final` final selected pooled model
- `multiparm` pooled p-values at each step according to pooling method

- `multiparm_final` pooled p-values at final step according to pooling method
- `multiparm_out` (only when `direction = "FW"`) pooled p-values of removed predictors
- `formula_step` formula object at each step
- `formula_final` formula object at final step
- `formula_initial` formula object at final step
- `predictors_in` predictors included at each selection step
- `predictors_out` predictors excluded at each step
- `impvar` name of variable used to distinguish imputed datasets
- `nimp` number of imputed datasets
- Outcome name of the outcome variable
- `method` selection method
- `p.crit` p-value selection criterium
- `call` function call
- `model_type` type of regression model used
- `direction` direction of predictor selection
- `predictors_final` names of predictors in final selection step
- `predictors_initial` names of predictors in start model
- `keep.predictors` names of predictors that were forced in the model

### Vignettes

[https://mwheymans.github.io/psfmi/articles/psfmi\\_LogisticModels.html](https://mwheymans.github.io/psfmi/articles/psfmi_LogisticModels.html)

### Author(s)

Martijn Heymans, 2020

### References

- Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. *BMC Med Res Methodol.* 2017;17(1):129.
- Enders CK (2010). *Applied missing data analysis*. New York: The Guilford Press.
- Meng X-L, Rubin DB. Performing likelihood ratio tests with multiply-imputed data sets. *Biometrika.* 1992;79:103-11.
- van de Wiel MA, Berkhof J, van Wieringen WN. Testing the prediction error difference between 2 predictors. *Biostatistics.* 2009;10:550-60.
- Marshall A, Altman DG, Holder RL, Royston P. Combining estimates of interest in prognostic modelling studies after multiple imputation: current practice and guidelines. *BMC Med Res Methodol.* 2009;9:57.
- Van Buuren S. (2018). *Flexible Imputation of Missing Data*. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.
- EW. Steyerberg (2019). *Clinical Prediction MOdels. A Practical Approach to Development, Validation, and Updating* (2nd edition). Springer Nature Switzerland AG.
- <http://missingdatasolutions.rbind.io/>



**Examples**

```
pool_lr <- psfmi_lr(data=lbpmlr, formula = Chronic ~ Pain +
  factor(Satisfaction) + rcs(Tampascale,3) + Radiation +
  Radiation*factor(Satisfaction) + Age + Duration + BMI,
  p.crit = 0.05, direction="FW", nimp=5, impvar="Impnr",
  keep.predictors = c("Radiation*factor(Satisfaction)", "Age"), method="D1")

pool_lr$RR_model_final
```

psfmi\_mm

*Pooling and Predictor selection function for multilevel models in multiply imputed datasets*

**Description**

psfmi\_mm Pooling and backward selection for 2 level (generalized) linear mixed models in multiply imputed datasets using different selection methods.

**Usage**

```
psfmi_mm(
  data,
  nimp = 5,
  impvar = NULL,
  clusvar = NULL,
  Outcome,
  predictors = NULL,
  random.eff = NULL,
  family = "linear",
  p.crit = 1,
  cat.predictors = NULL,
  spline.predictors = NULL,
  int.predictors = NULL,
  keep.predictors = NULL,
  nknots = NULL,
  method = "RR",
  print.method = FALSE
)
```

**Arguments**

**data** Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under `impvar`, and starting by 1 and the clusters should be distinguished by a cluster variable, specified under `clusvar`.

nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
clusvar	A character vector. Name of the variable that distinguishes the clusters.
Outcome	Character vector containing the name of the outcome variable.
predictors	Character vector with the names of the predictor variables. At least one predictor variable has to be defined.
random.eff	Character vector to specify the random effects as used by the lmer and glmer functions of the lme4 package.
family	Character vector to specify the type of model, "linear" is used to call the lmer function and "binomial" is used to call the glmer function of the lme4 package. See details for more information.
p.crit	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
cat.predictors	A single string or a vector of strings to define the categorical variables. Default is NULL categorical predictors.
spline.predictors	A single string or a vector of strings to define the (restricted cubic) spline variables. Default is NULL spline predictors. See details.
int.predictors	A single string or a vector of strings with the names of the variables that form an interaction pair, separated by a ":" symbol.
keep.predictors	A single string or a vector of strings including the variables that are forced in the model during predictor selection. Categorical and interaction variables are allowed.
nknots	A numerical vector that defines the number of knots for each spline predictor separately.
method	A character vector to indicate the pooling method for p-values to pool the total model or used during predictor selection. This can be "D1", "D2", "D3" or "MPR". See details for more information.
print.method	logical vector. If TRUE full matrix with p-values of all variables according to chosen method (under method) is shown. If FALSE (default) p-value for categorical variables according to method are shown and for continuous and dichotomous predictors Rubin's Rules are used.

## Details

The basic pooling procedure to derive pooled coefficients, standard errors, 95 confidence intervals and p-values is Rubin's Rules (RR). Specific procedures are available to derive pooled p-values for categorical (> 2 categories) and spline variables. print.method allows to choose between the pooling methods: D1, D2 and D3 and MPR for pooling of median p-values (MPR rule). The D1, D2 and D3 methods are called from the package `mi tm1`. For Logistic multilevel models (that are estimated using the `glmer` function), the D3 method is not yet available. Spline regression coefficients are defined by using the `rcs` function for restricted cubic splines of the `rms` package. A minimum number of 3 knots as defined under `knots` is required.

**Value**

An object of class `smodsmi` (selected models in multiply imputed datasets) from which the following objects can be extracted: imputed datasets as `data`, selected pooled model as `RR_model`, pooled p-values according to pooling method as `multiparm`, random effects as `random.eff`, predictors included at each selection step as `predictors_in`, predictors excluded at each step as `predictors_out`, and family, `impvar`, `clusvar`, `nimp`, `Outcome`, `method`, `p.crit`, `predictors`, `cat.predictors`, `keep.predictors`, `int.predictors`, `spline.predictors`, `knots`, `print.method`, `model_type`, `call`, `predictors_final` for names of predictors in final step and `fit.formula` is the regression formula of start model.

**References**

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. *BMC Med Res Methodol*. 2017;17(1):129.

Enders CK (2010). *Applied missing data analysis*. New York: The Guilford Press.

Meng X-L, Rubin DB. Performing likelihood ratio tests with multiply-imputed data sets. *Biometrika*.1992;79:103-11.

van de Wiel MA, Berkhof J, van Wieringen WN. Testing the prediction error difference between 2 predictors. *Biostatistics*. 2009;10:550-60.

mitml package <https://cran.r-project.org/web/packages/mitml/index.html>

Van Buuren S. (2018). *Flexible Imputation of Missing Data*. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

<http://missingdatasolutions.rbind.io/>

**Examples**

```
## Not run:
pool_mm <- psfmi_mm(data=ipdna_md, nimp=5, impvar=".imp", family="linear",
predictors=c("gender", "afib", "sbp"), clusvar = "centre",
random.eff="( 1 | centre)", Outcome="dbp", cat.predictors = "bmi_cat",
p.crit=0.15, method="D1", print.method = FALSE)
pool_mm$RR_Model
pool_mm$multiparm

## End(Not run)
```

---

psfmi\_mm\_multiparm      *Multiparameter pooling methods called by psfmi\_mm*

---

**Description**

`psfmi_mm_multiparm` Function to pool according to D1, D2 and D3 methods

**Usage**

```
psfmi_mm_multiparm(
  data,
  nimp,
  impvar,
  Outcome,
  P,
  p.crit,
  family,
  random.eff,
  method,
  print.method
)
```

**Arguments**

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under <code>impvar</code> , and starting by 1 and the clusters should be distinguished by a cluster variable, specified under <code>clusvar</code> .
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
Outcome	Character vector containing the name of the outcome variable.
P	Character vector with the names of the predictor variables. At least one predictor variable has to be defined.
p.crit	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
family	Character vector to specify the type of model, "linear" is used to call the <code>lmer</code> function and "binomial" is used to call the <code>glmer</code> function of the <code>lme4</code> package. See details for more information.
random.eff	Character vector to specify the random effects as used by the <code>lmer</code> and <code>glmer</code> functions of the <code>lme4</code> package.
method	A character vector to indicate the pooling method for p-values to pool the total model or used during predictor selection. This can be "D1", "D2", "D3" or "MPR". See details for more information.
print.method	logical vector. If TRUE full matrix with p-values of all variables according to chosen method (under <code>method</code> ) is shown. If FALSE (default) p-value for categorical variables according to <code>method</code> are shown and for continuous and dichotomous predictors Rubin's Rules are used.

**Examples**

```
## Not run:
psfmi_mm_multiparm(data=ipdna_md, nimp=5, impvar=".imp", family="linear",
P=c("gender", "bnp", "dbp", "lvef", "bmi_cat"),
```

```

random.eff="( 1 | centre)", Outcome="sbp",
p.crit=0.05, method="D1", print.method = FALSE)

## End(Not run)

```

---

psfmi_perform	<i>Internal validation and performance of logistic prediction models across Multiply Imputed datasets</i>
---------------	-----------------------------------------------------------------------------------------------------------

---

## Description

psfmi\_perform Evaluate Performance of logistic regression models selected with the psfmi\_lr function of the psfmi package by using cross-validation or bootstrapping.

## Usage

```

psfmi_perform(
  pobj,
  val_method = NULL,
  data_orig = NULL,
  int_val = TRUE,
  nboot = 10,
  folds = 3,
  nimp_cv = 5,
  nimp_mice = 5,
  p.crit = 1,
  BW = FALSE,
  direction = NULL,
  cv_naive_appt = FALSE,
  cal.plot = FALSE,
  plot.method = "mean",
  groups_cal = 5,
  miceImp,
  ...
)

```

## Arguments

pobj	An object of class pmods (pooled models), produced by a previous call to psfmi_lr.
val_method	Method for internal validation. MI_boot for first Multiple Imputation and then bootstrapping in each imputed dataset and boot_MI for first bootstrapping and then multiple imputation in each bootstrap sample, and cv_MI, cv_MI_RR and MI_cv_naive for the combinations of cross-validation and multiple imputation. To use cv_MI, cv_MI_RR and boot_MI, data_orig has to be specified. See details for more information.

<code>data_orig</code>	dataframe of original dataset that contains missing data for methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>boot_MI</code> .
<code>int_val</code>	If TRUE internal validation is conducted using bootstrapping or cross-validation. Default is TRUE. If FALSE only apparent performance measures are calculated.
<code>nboot</code>	The number of bootstrap resamples, default is 10. Used for methods <code>boot_MI</code> and <code>MI_boot</code> .
<code>fold</code>	The number of folds, default is 3. Used for methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>MI_cv_naive</code> .
<code>nimp_cv</code>	Numerical scalar. Number of (multiple) imputation runs for method <code>cv_MI</code> .
<code>nimp_mice</code>	Numerical scalar. Number of imputed datasets for method <code>cv_MI_RR</code> and <code>boot_MI</code> . When not defined, the number of multiply imputed datasets is used of the previous call to the function <code>psfmi_lr</code> .
<code>p.crit</code>	A numerical scalar. P-value selection criterium used for backward or forward selection during validation. When set at 1, pooling and internal validation is done without backward selection.
<code>BW</code>	Only used for methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>MI_cv_naive</code> . If TRUE backward selection is conducted within cross-validation. Default is FALSE.
<code>direction</code>	Can be used together with <code>val_methods</code> <code>boot_MI</code> and <code>MI_boot</code> . The direction of predictor selection, "BW" is for backward selection and "FW" for forward selection.
<code>cv_naive_appt</code>	Can be used in combination with <code>val_method</code> <code>MI_cv_naive</code> . Default is TRUE for showing the cross-validation apparent (train) and test results. Set to FALSE to only give test results.
<code>cal.plot</code>	If TRUE a calibration plot is generated. Default is FALSE. Can be used in combination with <code>int_val = FALSE</code> .
<code>plot.method</code>	If "mean" one calibration plot is generated, first taking the mean of the linear predictor across the multiply imputed datasets (default), if "individual" the calibration plot of each imputed dataset is plotted, if "overlay" calibration plots from each imputed datasets are plotted in one figure.
<code>groups_cal</code>	A numerical scalar. Number of groups used on the calibration plot and. for the Hosmer and Lemeshow test. Default is 10. If the range of predicted probabilities. is low, less than 10 groups can be chosen, but not < 3.
<code>miceImp</code>	Wrapper function around the <code>mice</code> function.
<code>...</code>	Arguments as <code>predictorMatrix</code> , <code>seed</code> , <code>maxit</code> , etc that can be adjusted for the <code>mice</code> function. To be used in combination with validation methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>MI_boot</code> . For method <code>cv_MI</code> the number of imputed datasets is fixed at 1 and cannot be changed.

## Details

For internal validation five methods can be used, `cv_MI`, `cv_MI_RR`, `MI_cv_naive`, `MI_boot` and `boot_MI`. Method `cv_MI` uses imputation within each cross-validation fold definition. By repeating this in several imputation runs, multiply imputed datasets are generated. Method `cv_MI_RR` uses multiple imputation within the cross-validation definition. `MI_cv_naive`, applies cross-validation

within each imputed dataset. MI\_boot draws for each bootstrap step the same cases in all imputed datasets. With boot\_MI first bootstrap samples are drawn from the original dataset with missing values and then multiple imputation is applied. For multiple imputation the mice function from the mice package is used. It is recommended to use a minimum of 100 imputation runs for method cv\_MI or 100 bootstrap samples for method boot\_MI or MI\_boot. Methods cv\_MI, cv\_MI\_RR and MI\_cv\_naive can be combined with backward selection during cross-validation and with methods boot\_MI and MI\_boot, backward and forward selection can be used. For methods cv\_MI and cv\_MI\_RR the outcome in the original dataset has to be complete.

### Value

A psfmi\_perform object from which the following objects can be extracted: res\_boot, result of pooled performance (in multiply imputed datasets) at each bootstrap step of ROC app (pooled ROC), ROC test (pooled ROC after bootstrap model is applied in original multiply imputed datasets), same for R2 app (Nagelkerke's R2), R2 test, Scaled Brier app and Scaled Brier test. Information is also provided about testing the Calibration slope at each bootstrap step as interc test and Slope test. The performance measures are pooled by a call to the function pool\_performance. Another object that can be extracted is intval, with information of the AUC, R2, Scaled Brier score and Calibration slope averaged over the bootstrap samples, in terms of: Orig (original datasets), Apparent (models applied in bootstrap samples), Test (bootstrap models are applied in original datasets), Optimism (difference between apparent and test) and Corrected (original corrected for optimism).

### Author(s)

Martijn Heymans, 2020

### References

- Heymans MW, van Buuren S, Knol DL, van Mechelen W, de Vet HC. Variable selection under multiple imputation using the bootstrap in a prognostic study. *BMC Med Res Methodol.* 2007(13);7:33.
- F. Harrell. *Regression Modeling Strategies. With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis* (2nd edition). Springer, New York, NY, 2015.
- Van Buuren S. (2018). *Flexible Imputation of Missing Data*. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.
- Harel, O. (2009). The estimation of R2 and adjusted R2 in incomplete data sets using multiple imputation. *Journal of Applied Statistics*, 36(10), 1109-1118.
- Musoro JZ, Zwinderman AH, Puhan MA, ter Riet G, Geskus RB. Validation of prediction models based on lasso regression with multiply imputed data. *BMC Med Res Methodol.* 2014;14:116.
- Wahl S, Boulesteix AL, Zierer A, Thorand B, van de Wiel MA. Assessment of predictive performance in incomplete data by combining internal validation and multiple imputation. *BMC Med Res Methodol.* 2016;16(1):144.
- EW. Steyerberg (2019). *Clinical Prediction MOdels. A Practical Approach to Development, Validation, and Updating* (2nd edition). Springer Nature Switzerland AG.
- <http://missingdatasolutions.rbind.io/>

---

psfmi_stab	<i>Function to evaluate bootstrap predictor and model stability in multiply imputed datasets.</i>
------------	---------------------------------------------------------------------------------------------------

---

### Description

psfmi\_stab Stability analysis of predictors and prediction models selected with the psfmi\_lr, psfmi\_coxr or psfmi\_mm functions of the psfmi package.

### Usage

```
psfmi_stab(
  pobj,
  boot_method = NULL,
  nboot = 20,
  p.crit = 0.05,
  start_model = TRUE,
  direction = NULL
)
```

### Arguments

pobj	An object of class pmods (pooled models), produced by a previous call to psfmi_lr, psfmi_coxr or psfmi_mm.
boot_method	A single string to define the bootstrap method. Use "single" after a call to psfmi_lr and psfmi_coxr and "cluster" after a call to psfmi_mm.
nboot	A numerical scalar. Number of bootstrap samples to evaluate the stability. Default is 20.
p.crit	A numerical scalar. Used as P-value selection criterium during bootstrap model selection.
start_model	If TRUE the bootstrap evaluation takes place from the start model of object pobj, if FALSE the final model is used for the evaluation.
direction	The direction of predictor selection, "BW" for backward selection and "FW" for forward selection. #'

### Details

The function evaluates predictor selection frequency in stratified or cluster bootstrap samples. The stratification factor is the variable that separates the imputed datasets. The same bootstrap cases are drawn in each bootstrap sample. It uses as input an object of class pmods as a result of a previous call to the psfmi\_lr, psfmi\_coxr or psfmi\_mm functions. In combination with the psfmi\_mm function a cluster bootstrap method is used where bootstrapping is used on the level of the clusters only (and not also within the clusters).



**Value**

A `psfmi_stab` object from which the following objects can be extracted: bootstrap inclusion (selection) frequency of each predictor `bif`, total number each predictor is included in the bootstrap samples as `bif_total`, percentage a predictor is selected in each bootstrap sample as `bif_perc` and number of times a prediction model is selected in the bootstrap samples as `model_stab`.

**Vignettes**

[https://mwheymans.github.io/psfmi/articles/psfmi\\_StabilityAnalysis.html](https://mwheymans.github.io/psfmi/articles/psfmi_StabilityAnalysis.html)

**References**

Heymans MW, van Buuren S. et al. Variable selection under multiple imputation using the bootstrap in a prognostic study. *BMC Med Res Methodol.* 2007;13:7-33.

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. *BMC Med Res Methodol.* 2017;17(1):129.

Sauerbrei W, Schumacher M. A bootstrap resampling procedure for model building: application to the Cox regression model. *Stat Med.* 1992;11:2093–109.

Royston P, Sauerbrei W (2008) *Multivariable model-building – a pragmatic approach to regression analysis based on fractional polynomials for modelling continuous variables.* (2008). Chapter 8, *Model Stability.* Wiley, Chichester

Heinze G, Wallisch C, Dunkler D. Variable selection - A review and recommendations for the practicing statistician. *Biom J.* 2018;60(3):431-449.

<http://missingdatasolutions.rbind.io/>

**Examples**

```
pool_lr <- psfmi_coxr(formula = Surv(Time, Status) ~ Pain + factor(Satisfaction) +
  rcs(Tampascale,3) + Radiation + Radiation*factor(Satisfaction) + Age + Duration +
  Previous + Radiation*rcs(Tampascale, 3), data=lbpmicox, p.crit = 0.157, direction="FW",
  nimp=5, impvar="Impnr", keep.predictors = NULL, method="D1")
```

```
pool_lr$RR_Model
pool_lr$multiparm
```

```
## Not run:
```

```
stab_res <- psfmi_stab(pool_lr, direction="FW", start_model = TRUE,
  boot_method = "single", nboot=20, p.crit=0.05)
stab_res$bif
stab_res$bif_perc
stab_res$model_stab
```

```
## End(Not run)
```

---

psfmi_validate	<i>Internal validation and performance of logistic prediction models across Multiply Imputed datasets</i>
----------------	-----------------------------------------------------------------------------------------------------------

---

### Description

psfmi\_validate Evaluate Performance of logistic regression models selected with the psfmi\_lr function of the psfmi package by using cross-validation or bootstrapping.

### Usage

```
psfmi_validate(
  pobj,
  val_method = NULL,
  data_orig = NULL,
  int_val = TRUE,
  nboot = 10,
  folds = 3,
  nimp_cv = 5,
  nimp_mice = 5,
  p.crit = 1,
  BW = FALSE,
  direction = NULL,
  cv_naive_appt = FALSE,
  cal.plot = FALSE,
  plot.method = "mean",
  groups_cal = 5,
  miceImp,
  ...
)
```

### Arguments

pobj	An object of class pmods (pooled models), produced by a previous call to psfmi_lr.
val_method	Method for internal validation. MI_boot for first Multiple Imputation and then bootstrapping in each imputed dataset and boot_MI for first bootstrapping and then multiple imputation in each bootstrap sample, and cv_MI, cv_MI_RR and MI_cv_naive for the combinations of cross-validation and multiple imputation. To use cv_MI, cv_MI_RR and boot_MI, data_orig has to be specified. See details for more information.
data_orig	dataframe of original dataset that contains missing data for methods cv_MI, cv_MI_RR and boot_MI.
int_val	If TRUE internal validation is conducted using bootstrapping or cross-validation. Default is TRUE. If FALSE only apparent performance measures are calculated.
nboot	The number of bootstrap resamples, default is 10. Used for methods boot_MI and MI_boot.

<code>folds</code>	The number of folds, default is 3. Used for methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>MI_cv_naive</code> .
<code>nimp_cv</code>	Numerical scalar. Number of (multiple) imputation runs for method <code>cv_MI</code> .
<code>nimp_mice</code>	Numerical scalar. Number of imputed datasets for method <code>cv_MI_RR</code> and <code>boot_MI</code> . When not defined, the number of multiply imputed datasets is used of the previous call to the function <code>psfmi_1r</code> .
<code>p.crit</code>	A numerical scalar. P-value selection criterium used for backward or forward selection during validation. When set at 1, pooling and internal validation is done without backward selection.
<code>BW</code>	Only used for methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>MI_cv_naive</code> . If TRUE backward selection is conducted within cross-validation. Default is FALSE.
<code>direction</code>	Can be used together with <code>val_methods</code> <code>boot_MI</code> and <code>MI_boot</code> . The direction of predictor selection, "BW" is for backward selection and "FW" for forward selection.
<code>cv_naive_appt</code>	Can be used in combination with <code>val_method</code> <code>MI_cv_naive</code> . Default is TRUE for showing the cross-validation apparent (train) and test results. Set to FALSE to only give test results.
<code>cal.plot</code>	If TRUE a calibration plot is generated. Default is FALSE. Can be used in combination with <code>int_val = FALSE</code> .
<code>plot.method</code>	If "mean" one calibration plot is generated, first taking the mean of the linear predictor across the multiply imputed datasets (default), if "individual" the calibration plot of each imputed dataset is plotted, if "overlay" calibration plots from each imputed datasets are plotted in one figure.
<code>groups_cal</code>	A numerical scalar. Number of groups used on the calibration plot and. for the Hosmer and Lemeshow test. Default is 10. If the range of predicted probabilities. is low, less than 10 groups can be chosen, but not < 3.
<code>miceImp</code>	Wrapper function around the <code>mice</code> function.
<code>...</code>	Arguments as <code>predictorMatrix</code> , <code>seed</code> , <code>maxit</code> , etc that can be adjusted for the <code>mice</code> function. To be used in combination with validation methods <code>cv_MI</code> , <code>cv_MI_RR</code> and <code>MI_boot</code> . For method <code>cv_MI</code> the number of imputed datasets is fixed at 1 and cannot be changed.

## Details

For internal validation five methods can be used, `cv_MI`, `cv_MI_RR`, `MI_cv_naive`, `MI_boot` and `boot_MI`. Method `cv_MI` uses imputation within each cross-validation fold definition. By repeating this in several imputation runs, multiply imputed datasets are generated. Method `cv_MI_RR` uses multiple imputation within the cross-validation definition. `MI_cv_naive`, applies cross-validation within each imputed dataset. `MI_boot` draws for each bootstrap step the same cases in all imputed datasets. With `boot_MI` first bootstrap samples are drawn from the original dataset with missing values and than multiple imputation is applied. For multiple imputation the `mice` function from the `mice` package is used. It is recommended to use a minimum of 100 imputation runs for method `cv_MI` or 100 bootstrap samples for method `boot_MI` or `MI_boot`. Methods `cv_MI`, `cv_MI_RR` and `MI_cv_naive` can be combined with backward selection during cross-validation and with methods `boot_MI` and `MI_boot`, backward and forward selection can be used. For methods `cv_MI` and `cv_MI_RR` the outcome in the original dataset has to be complete.

**Value**

A `psfmi_perform` object from which the following objects can be extracted: `res_boot`, result of pooled performance (in multiply imputed datasets) at each bootstrap step of ROC app (pooled ROC), ROC test (pooled ROC after bootstrap model is applied in original multiply imputed datasets), same for R2 app (Nagelkerke's R2), R2 test, Scaled Brier app and Scaled Brier test. Information is also provided about testing the Calibration slope at each bootstrap step as `interc` test and `Slope` test. The performance measures are pooled by a call to the function `pool_performance`. Another object that can be extracted is `intval`, with information of the AUC, R2, Scaled Brier score and Calibration slope averaged over the bootstrap samples, in terms of: `Orig` (original datasets), `Apparent` (models applied in bootstrap samples), `Test` (bootstrap models are applied in original datasets), `Optimism` (difference between apparent and test) and `Corrected` (original corrected for optimism).

**Author(s)**

Martijn Heymans, 2020

**References**

Heymans MW, van Buuren S, Knol DL, van Mechelen W, de Vet HC. Variable selection under multiple imputation using the bootstrap in a prognostic study. *BMC Med Res Methodol*. 2007(13);7:33.

F. Harrell. *Regression Modeling Strategies. With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis* (2nd edition). Springer, New York, NY, 2015.

Van Buuren S. (2018). *Flexible Imputation of Missing Data*. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

Harel, O. (2009). The estimation of R2 and adjusted R2 in incomplete data sets using multiple imputation. *Journal of Applied Statistics*, 36(10), 1109-1118.

Musoro JZ, Zwinderman AH, Puhan MA, ter Riet G, Geskus RB. Validation of prediction models based on lasso regression with multiply imputed data. *BMC Med Res Methodol*. 2014;14:116.

Wahl S, Boulesteix AL, Zierer A, Thorand B, van de Wiel MA. Assessment of predictive performance in incomplete data by combining internal validation and multiple imputation. *BMC Med Res Methodol*. 2016;16(1):144.

EW. Steyerberg (2019). *Clinical Prediction Models. A Practical Approach to Development, Validation, and Updating* (2nd edition). Springer Nature Switzerland AG.

<http://missingdatasolutions.rbind.io/>

**Examples**

```
pool_lr <- psfmi_lr(data=lbpmlr, formula = Chronic ~ Pain + JobDemands + rcs(Tampascale, 3) +
  factor(Satisfaction) + Smoking, p.crit = 1, direction="FW",
  nimp=5, impvar="Impnr", method="D1")
```

```
pool_lr$RR_model
```

```
res_perf <- psfmi_validate(pool_lr, val_method = "cv_MI", data_orig = lbp_orig, folds=3,
  nimp_cv = 2, p.crit=0.05, BW=TRUE, miceImp = miceImp, printFlag = FALSE)
```

```
res_perf
```

```
## Not run:
set.seed(200)
res_val <- psfmi_validate(pobj, val_method = "boot_MI", data_orig = lbp_orig, nboot = 5,
p.crit=0.05, BW=TRUE, miceImp = miceImp, nimp_mice = 5, printFlag = FALSE, direction = "FW")

res_val$stats_val

## End(Not run)
```

---

risk\_coxph

*Risk calculation at specific time point for Cox model*

---

### Description

Risk calculation at specific time point for Cox model

### Usage

```
risk_coxph(mod, t_risk)
```

### Arguments

mod                    a Cox regression model object.  
t\_risk                 Follow-up value to calculate cases, controls. See details.

### Value

Cox regression Risk estimates at specific time point.

### Author(s)

Martijn Heymans, 2023

### References

Pencina MJ, D'Agostino RB Sr, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Stat Med.* 2011;30(1):11-21

Inoue E (2018). nricens: NRI for Risk Prediction Models with Time to Event and Binary Response Data. R package version 1.6, <<https://CRAN.R-project.org/package=nricens>>.

### See Also

[nri\\_cox](#)

---

`rsq_nagel`*Nagelkerke's R-square calculation for logistic regression / glm models*

---

**Description**

Nagelkerke's R-square calculation for logistic regression / glm models

**Usage**

```
rsq_nagel(fitobj)
```

**Arguments**

`fitobj` a logistic regression model object of "glm"

**Value**

The value for the explained variance.

**Author(s)**

Martijn Heymans, 2020

**See Also**

[psfmi\\_perform](#), [pool\\_performance](#)

---

`rsq_surv`*R-square calculation for Cox regression models*

---

**Description**

R-square calculation for Cox regression models

**Usage**

```
rsq_surv(fitobj)
```

**Arguments**

`fitobj` a Cox regression model object of "coxph"

**Value**

The value for the explained variance.

**Author(s)**

Martijn Heymans, 2021

**References**

F. Harrell. Regression Modeling Strategies. With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. 2nd Edition. Springer, New York, NY, 2015.

**See Also**

[pool\\_performance](#)

---

sbp\_age

*Dataset with blood pressure measurements*

---

**Description**

Dataset with blood pressure measurements

**Usage**

```
data(sbp_age)
```

**Format**

A data frame with 30 observations on the following 3 variables.

pat\_id continuous

sbp continuous: systolic blood pressure

age continuous: age (years)

**Examples**

```
data(sbp_age)
## maybe str(sbp_age)
```

---

sbp_qas	<i>Dataset with blood pressure measurements</i>
---------	-------------------------------------------------

---

**Description**

Dataset with blood pressure measurements

**Usage**

```
data(sbp_qas)
```

**Format**

A data frame with 32 observations on the following 5 variables.

pat\_id continuous

sbp continuous: systolic blood pressure

bmi continuous: body mass index

age continuous: age (years)

smk dichotomous: 0 = no, 1 = yes

**Examples**

```
data(sbp_qas)
## maybe str(sbp_qas)
```

---

scaled_brier	<i>Calculates the scaled Brier score</i>
--------------	------------------------------------------

---

**Description**

Calculates the scaled Brier score

**Usage**

```
scaled_brier(obs, pred)
```

**Arguments**

obs Observed outcomes.

pred Predicted outcomes in the form of probabilities.

**Value**

The value for the scaled Brier score.



**Author(s)**

Martijn Heymans, 2020

**See Also**

[psfmi\\_perform](#), [pool\\_performance](#)

---

smoking

*Survival data about smoking*

---

**Description**

Survival data about smoking

**Usage**

```
data(smoking)
```

**Format**

A data frame with 20 observations on the following 3 variables.

smoking dichotomous: 1=yes, 0=no

time continuous: Survival time in years

death dichotomous: Status at end of study

**Examples**

```
data(smoking)
## maybe str(smoking)
```

---

stab\_single

*Function to evaluate bootstrap predictor and model stability.*

---

**Description**

stab\_single Stability analysis of predictors and prediction models selected with the glm\_bw.

**Usage**

```
stab_single(pobj, nboot = 20, p.crit = 0.05, start_model = TRUE)
```

**Arguments**

<code>pobj</code>	An object of class <code>smods</code> (single models), produced by a previous call to <code>glm_bw</code> .
<code>nboot</code>	A numerical scalar. Number of bootstrap samples to evaluate the stability. Default is 20.
<code>p.crit</code>	A numerical scalar. Used as P-value selection criterium during bootstrap model selection.
<code>start_model</code>	If TRUE the bootstrap evaluation takes place from the start model of object <code>pobj</code> , if FALSE the final model is used for the evaluation.

**Details**

The function evaluates predictor selection frequency in bootstrap samples. It uses as input an object of class `smods` as a result of a previous call to the `glm_bw`.

**Value**

A `psfmi_stab` object from which the following objects can be extracted: bootstrap inclusion (selection) frequency of each predictor `bif`, total number each predictor is included in the bootstrap samples as `bif_total`, percentage a predictor is selected in each bootstrap sample as `bif_perc` and number of times a prediction model is selected in the bootstrap samples as `model_stab`.

**References**

Heymans MW, van Buuren S. et al. Variable selection under multiple imputation using the bootstrap in a prognostic study. *BMC Med Res Methodol.* 2007;13:7-33.

Sauerbrei W, Schumacher M. A bootstrap resampling procedure for model building: application to the Cox regression model. *Stat Med.* 1992;11:2093–109.

Royston P, Sauerbrei W (2008) Multivariable model-building – a pragmatic approach to regression analysis based on fractional polynomials for modelling continuous variables. (2008). Chapter 8, *Model Stability*. Wiley, Chichester.

Heinze G, Wallisch C, Dunkler D. Variable selection - A review and recommendations for the practicing statistician. *Biom J.* 2018;60(3):431-449.

<http://missingdatasolutions.rbind.io/>

**Examples**

```

model_lr <- glm_bw(formula = Radiation ~ Pain + factor(Satisfaction) +
  rcs(Tampascale,3) + Age + Duration + JobControl + JobDemands + SocialSupport,
  data=lbpmlr_dev, p.crit = 0.05)

## Not run:
stab_res <- stab_single(model_lr, start_model = TRUE, nboot=20, p.crit=0.05)
stab_res$bif
stab_res$bif_perc
stab_res$model_stab

## End(Not run)

```

---

weight

*Dataset of persons from the The Amsterdam Growth and Health Longitudinal Study (AGHLS)*

---

**Description**

Dataset of persons from the The Amsterdam Growth and Health Longitudinal Study (AGHLS)

**Usage**

```
data(weight)
```

**Format**

A data frame with 450 observations on the following 7 variables.

ID continuous

SBP continuous: Systolic Blood Pressure

LDL continuous: Cholesterol

Glucose continuous

HDL continuous: Cholesterol

Gender dichotomous: 1=male, 0=female

Weight continuous: bodyweight

**Examples**

```
data(weight)
```

```
## maybe str(weight)
```

# Index

## \* datasets

- anderson, 3
- aortadis, 4
- bmd, 5
- chlrform, 7
- chol\_long, 8
- chol\_wide, 8
- day2\_dataset4\_mi, 13
- hipstudy, 18
- hipstudy\_external, 19
- hoorn\_basic, 20
- infarct, 22
- ipdna\_md, 22
- lbp\_orig, 30
- lbpmicox, 26
- lbpmlr, 27
- lbpmlr\_dev, 28
- lungvolume, 31
- mammaca, 31
- men, 32
- sbp\_age, 71
- sbp\_qas, 72
- smoking, 73
- weight, 75

## \* dataset

- lbpmi\_extval, 29

- anderson, 3
- aortadis, 4

- bmd, 5
- bw\_single, 5

- chlrform, 7
- chol\_long, 8
- chol\_wide, 8
- coxph\_bw, 9
- coxph\_fw, 11

- day2\_dataset4\_mi, 13

- glm\_bw, 14
- glm\_fw, 16

- hipstudy, 18
- hipstudy\_external, 19
- hoorn\_basic, 20
- hoslem\_test, 21

- infarct, 22
- ipdna\_md, 22

- km\_estimates, 23
- km\_fit, 25, 26

- lbp\_orig, 30
- lbpmi\_extval, 29
- lbpmicox, 26
- lbpmlr, 27
- lbpmlr\_dev, 28
- lungvolume, 31

- mammaca, 31
- men, 32
- mivalex\_lr, 33

- nri\_cox, 35, 69
- nri\_est, 37

- pool\_auc, 38
- pool\_compare\_models, 39
- pool\_D2, 41
- pool\_D4, 42
- pool\_intadj, 43
- pool\_performance, 21, 39, 44, 70, 71, 73
- pool\_reclassification, 46
- pool\_RR, 46
- psfmi\_coxr, 47
- psfmi\_lm, 50
- psfmi\_lr, 54
- psfmi\_mm, 57
- psfmi\_mm\_multiparm, 59

psfmi\_perform, [15](#), [17](#), [39](#), [61](#), [70](#), [73](#)

psfmi\_stab, [64](#)

psfmi\_validate, [66](#)

risk\_coxph, [69](#)

rsq\_nagel, [70](#)

rsq\_surv, [70](#)

sbp\_age, [71](#)

sbp\_qas, [72](#)

scaled\_brier, [72](#)

smoking, [73](#)

stab\_single, [73](#)

weight, [75](#)