# Package: sMSROC (via r-universe)

October 24, 2026

**Type** Package

**Title** Assessment of Diagnostic and Prognostic Markers

**Version** 0.1.3

**Description** Provides estimations of the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) based on the two-stages mixed-subjects ROC curve estimator (Diaz-Coto et al. (2020) <doi:10.1515/ijb-2019-0097> and Diaz-Coto et al. (2020) <doi:10.1080/00949655.2020.1736071>).

**License** GPL

**Encoding** UTF-8

**Imports** survival, rms, foreach, ggplot2, flextable, plotROC, icenReg, thregI

**LazyData** true

**Depends** R (>= 3.5.0)

**NeedsCompilation** no

**Author** Susana Diaz-Coto [aut], Susana Diaz-Coto [cre]

**Maintainer** Susana Diaz-Coto <susanacoto@gmail.com>

**Config/pak/sysreqs** libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev zlib1g-dev

**Repository** https://susanadc2009.r-universe.dev

**Date/Publication** 2025-10-20 02:10:02 UTC

**RemoteUrl** https://github.com/cran/sMSROC

**RemoteRef** HEAD

**RemoteSha** 93d5bda405f1f98a229bf7f679b8876ebc69bbf9

## Contents


auc_ci_boot-internal . . . . . 2
auc_ci_empr-internal . . . . . 3

auc_ci_nvar-internal . . . . . 4
check_ci_cl-internal . . . . . 5
check_conf_int-internal . . . . . 6
check_grid-internal . . . . . 7
check_marke_timeic-internal . . . . . 8
check_marker_binout-internal . . . . . 9
check_marker_timerc-internal . . . . . 10
check_meth-internal . . . . . 11
check_nboots-internal . . . . . 12
check_ncpus-internal . . . . . 12
check_tim-internal . . . . . 13
check_type_outcome-internal . . . . . 13
compute_ROC-internal . . . . . 14
conf_int_print . . . . . 15
diabet . . . . . 16
evol_auc . . . . . 17
explore_plot . . . . . 18
explore_table . . . . . 19
fibrosis . . . . . 20
ktfs . . . . . 21
pred_model_binout-internal . . . . . 22
pred_model_emp-internal . . . . . 23
pred_model_timeic-internal . . . . . 24
pred_model_timerc-internal . . . . . 25
print.sMSROC . . . . . 26
probs_pred . . . . . 27
sMS_binout-internal . . . . . 28
sMS_timeic-internal . . . . . 29
sMS_timerc . . . . . 31
sMSROC . . . . . 32
sMSROC_plot . . . . . 36
variance_probs-internal . . . . . 37

**Index** **39**

---

`auc_ci_boot-internal` *Confidence intervals for the AUC (bootstrap)*

---

## Description

Computation of confidence intervals for the AUC based on Bootstrap Percentile.

## Usage

```
auc_ci_boot(marker, outcome, status, observed.time, left, right, time,
                  data_type, meth, grid, probs, ci.cl, ci.nboots, parallel,
                  ncpus, all)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| outcome | vector with the condition of the subjects as positive, negative or unknown at the considered time time. |
| status | response vector. |
| observed.time | vector with the observed times for each subject. |
| left | vector with the lower edges of the observed intervals. |
| right | vector with the upper edges of the observed intervals. |
| time | point of time at which the sMS ROC curve estimator will be computed. |
| data_type | scenario handled. |
| meth | method for approximating the predictive model $P(D\|X = x)$. |
| grid | grid size. |
| probs | vector containing the probabilities estimated through the predictive model. |
| ci.cl | confidence level at which the confidence intervals will be computed. |
| ci.nboots | number of bootstrap samples. |
| parallel | indicates whether parallel computing will be performed or not. |
| ncpus | number of CPUs to use if parallel computing is performed. |
| all | indicates whether the probabilities from the predictive model will be considered for all individuals, or only for those whose outcome value (condition) is unknown. |

## Value

List with two components:

| | |
|---|---|
| ic.l | lower edge of the confidence interval. |
| ic.u | upper edge of the confidence interval. |

---

# auc_ci_empr-internal *Confidence intervals for the AUC (empirical variance estimation)*

---

## Description

Computation of confidence intervals for the AUC by implementing the empirical procedure for estimating the variance of the AUC, as described in doi:10.1515/ijb20190097.

## Usage

```
auc_ci_empr(SE, SP, auc, probs, controls, cases, ci.cl)
```

### Arguments

| | |
|---|---|
| SE | vector containing the values of the sensitivity returned from the sMSROC function. |
| SP | vector containing the values of the specificity. |
| auc | value with the AUC estimate. |
| probs | vector containing the probabilities estimated through the predictive model. |
| controls | number of negative individuals. |
| cases | number of positive individuals. |
| ci.cl | confidence level at which confidence intervals will be computed. |

### Value

List with two components:

| | |
|---|---|
| ic.l | lower edge of the confidence interval. |
| ic.u | upper edge of the confidence interval. |

---

## auc_ci_nvar-internal *Confidence intervals for the AUC (theoretical variance estimation)*

---

### Description

Computation of confidence intervals for the AUC by implementing the theoretical procedure for estimating the variance of the AUC, as described in doi:10.1515/ijb20190097.

### Usage

```
auc_ci_nvar(marker, outcome, status, observed.time, left, right, time,
                    meth, data_type, grid, probs, sd.probs, ci.cl, nboots,
                    SE, SP, auc, parallel, ncpus, all)
```

### Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| outcome | vector with the condition of the subjects as positive, negative or unknown at the considered time time. |
| status | response vector. |
| observed.time | vector with the observed times for each subject. |
| left | vector with the lower edges of the observed intervals. |
| right | vector with the upper edges of the observed intervals. |
| time | point of time at which the sMS ROC curve estimator will be computed. |
| meth | method for approximating the predictive model $P(D\|X = x)$. |
| data_type | scenario handled. |

| | |
|---|---|
| grid | grid size. |
| probs | vector containing the probabilities estimated through the predictive model. |
| sd.probs | vector containing the standard deviation of the probabilities of the predictive model. |
| ci.cl | confidence levet at which the confidence intervals will be computed. |
| nboots | number of bootstrap samples. |
| SE | vector containing the values of the sensitivity returned from sMSROC function. |
| SP | vector containing the values of the specificity. |
| auc | value with the AUC estimate. |
| parallel | indicates whether parallel computing will be performed or not. |
| ncpus | number of CPUs to use if parallel computing is performed. |
| all | parameter indicating whether all probabilities given by the predictive model should be considered (value **"T"**) or just those corresponding to individuals whose condition as positive or negative is unknown (**"F"**). The default value is (**"T"**). |

## Value

List with two components:

| | |
|---|---|
| ic.l | lower edge of the confidence interval. |
| ic.u | upper edge of the confidence interval. |

---

# check_ci_cl-internal *Check confidence level for AUC's confidence intervals*

---

## Description

Checks the validity of the value entered as confidence level for computing the confidence intervals for the AUC.

## Usage

```
check_ci_cl(ci.cl)
```

## Arguments

| | |
|---|---|
| ci.cl | confidence level at which the confidence intervals for the AUC will be computed. |

## Details

Verifies that the value entered as confidence level ranges between 0 and 1. The 0.95 confidence level is taken as default.

## Value

A list with two components:

| | |
|---|---|
| `ci.cl` | value entered as confidence level for the AUC. |
| `message` | table with the warning messages generated by the function. |

---

`check_conf_int-internal` *Checks for parameters to compute the confidence intervals for the AUC*

---

## Description

Check of the consistency of the parameters indicated to compute the confidence intervals for the AUC.

## Usage

```
check_conf_int(conf.int, ci.cl, ci.meth, ci.nboots, parallel, ncpus)
```

## Arguments

| | |
|---|---|
| `conf.int` | parameter indicating whether confidence intervals for the AUC will be computed (**"T"**) or not (**"F"**). The default is "F". |
| `ci.cl` | confidence level at which the confidence intervals for the AUC will be calculated. The default value is 0.95. |
| `ci.meth` | method for computing the confidence intervals according to doi:10.1515/ijb-20190097. There are three options:<br>• **"E"**, for the **E**mpirical variance estimation.<br>• **"V"**, for the theoretical **V**ariance estimation.<br>• **"B"**, for the **B**ootstrap Percentile.<br>The empirical method **E** is taken as default. This parameter is ignored if `conf.int` is set to "F". |
| `ci.nboots` | number of bootstrap samples to be generated when the chosen `ci.meth` is **"B".** The default values is 500. |
| `parallel` | indicates whether parallel computing will be done (**"T"**) or not (**"F"**), when computing the variance of the AUC through the methods **"V"** and **"B"**. |
| `ncpus` | number of CPUS that will be used when parallel computing is chosen. |

## Value

A list with the following components:

| | |
|---|---|
| ci.cl | value entered as confidence level for the AUC. |
| ci.meth | value entered as method for computing the confidence intervals. |
| ci.nboots | value entered as number of bootstrap samples. |
| ci.ncpus | value entered as number of CPUs chosen. |
| message | table with the warning messages generated by the function. |

---

# check_grid-internal *Check grid*

---

## Description

Cheking of the parameter grid.

## Usage

```
check_grid(grid)
```

## Arguments

| | |
|---|---|
| grid | grid size for computing the ROC curve estimate. The default value is 1000. |

## Details

Verifies if the parameter entered as grid is a numerical value greater than 0.

## Value

A list with two components:

| | |
|---|---|
| grid | grid size entered. |
| message | table with the warning messages generated by the function. |

check_marke_timeic-internal *Check of prognosis scenarios under interval censorship*

## Description

Checks the consistency of the parameters entered for prognosis scenarios under interval censorship.

## Usage

```
check_marker_timeic(marker, left, right, time, probs, sd.probs)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. It is a mandatory parameter. |
| left | vector containing the lower edges of the observed intervals. It is mandatory in prognosis scenarios and interval censorship and ignored in other situations. |
| right | vector with the upper edges of the observed intervals. It is mandatory as well in prognosis scenarios and interval censorship and ignored in other situations. |
| time | point of time at which the sMS ROC curve estimator will be computed. It is a mandatory parameter. The default value is 1. |
| probs | vector containing the probabilities corresponding to the predictive model when it has been externally computed. Only values between [0,1] are admissible. |
| sd.probs | vector with the standard deviations of the probabilities entered in probs. It is an optional parameter. |

## Value

The function returns a list with the following components:

| | |
|---|---|
| marker | vector containing the biomarker values. |
| left | vector containing the lower edges of the observed intervals. |
| right | vector containing the upper edges of the observed intervals. |
| probs | vector with the probabilities corresponding to the predictive model. |
| sd.probs | vector containing the standard deviations of the predictive model if they have been manually entered. |
| outcome | vector with the condition of the subjects at the time time, as positive, negative or unknown. |
| controls | number of negative subjects. |
| cases | number of positive subjects. |
| misout | number of subjects whose condition is unknown. |
| message | table containing the warning messages generated during the execution of the function. |

## See Also

check_marker_binout and check_marker_timerc

---

check_marker_binout-internal
*Checks of diagnosis scenarios*

---

## Description

Checks the consistency of the parameters entered for diagnosis scenarios.

## Usage

```
check_marker_binout(marker, status, probs, sd.probs)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. It is a mandatory parameter. |
| status | numeric response vector. Only two values will be taken into account. The highest one is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. It is a mandatory parameter in diagnosis scenarios. |
| probs | vector containing the probabilities corresponding to the predictive model when it has been externally computed. Obviosly, only values between [0,1] are admissible. |
| sd.probs | vector with the standard deviations of the probabilities entered in probs. It is an optional parameter. |

## Value

The ouput is a list with the following components:

| | |
|---|---|
| marker | vector containing the biomarker values. |
| outcome | vector with the condition of the subjects as positive or negative. |
| probs | vector with the probabilities corresponding to the predictive model. |
| sd.probs | vector containing the standard deviations of the predictive model if they have been manually entered. |
| controls | number of negative subjects. |
| cases | number of negative subjects. |
| misout | number of subjects whose outcome value is not known. |
| message | table containing the warning messages generated during the execution of the function. |

## See Also

check_marker_timerc and check_marker_timeic

`check_marker_timerc-internal` *Check of prognosis scenarios under right censorship*

## Description

Checks the consistency of the parameters entered for prognosis scenarios under right censorship.

## Usage

```
check_marker_timerc(marker, status, observed.time, time, probs, sd.probs)
```

## Arguments

| | |
|---|---|
| `marker` | vector with the biomarker values. It is a mandatory parameter. |
| `status` | numeric response vector. Only two values will be taken into account. The highest one is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. It is mandatory in prognosis scenarios and right censorship. |
| `observed.time` | vector with the observed times for each subject. These values can be the event times or the censoring times. It is mandatory when dealing with time-dependent outcomes under right censorship. |
| `time` | point of time at which the sMS ROC curve estimator will be computed. It is a mandatory parameter. The default value is 1. |
| `probs` | vector containing the probabilities corresponding to the predictive model when it has been externally computed. Only values between [0,1] are admissible. |
| `sd.probs` | vector with the standard deviations of the probabilities entered in `probs`. It is an optional parameter. |

## Value

The function returns a list with the following components:

| | |
|---|---|
| `marker` | vector containing the biomarker values. |
| `status` | response vector. |
| `observed.time` | vector containing the observed time. Recall event/censoring time. |
| `probs` | vector with the probabilities corresponding to the predictive model. |
| `sd.probs` | vector containing the standard deviations of the predictive model if they have been manually entered. |
| `outcome` | vector with the condition of the subjects at the time given in `time` as positive, negative or unknown. |
| `controls` | number of negative subjects. |
| `cases` | number of positive subjects. |
| `misout` | number of unknown subjects. |
| `message` | table containing the warning messages generated during the execution of the function. |

## See Also

check_marker_binout and check_marker_timeic

---

check_meth-internal *Check the method for estimating the predictive model*

---

## Description

When the predictive model is entered manually by the user, this function ensures that no other method by default is used to compute it.

## Usage

```
check_meth(meth, probs)
```

## Arguments

| | |
|---|---|
| meth | method for approximating the predictive model $P(D\|X = x)$. |
| probs | vector containing the probabilities corresponding to the predictive model when it is entered manually by the user. |

## Details

If the predicitve model has been manually indicated, this function sets the parameter to **"M"** ignoring other options. In this way, none of the function computing the predictive model will be called.

## Value

A list with one component:

| | |
|---|---|
| meth | value set up for the method parameter.It can take eiter the value entered by the user or its default, if the predictive model was not manually indicated, or the value **"M"**, when the predicitive model was entered in the parameter probs. |

## check_nboots-internal *Check number of bootstrap samples*

### Description

This function checks if the value entered as number of bootstrap samples is correct.

### Usage

```
check_nboots(nboots)
```

### Arguments

| | |
|---|---|
| nboots | number of bootstrap samples to be run. The default value is 500. |

### Value

A list with two components:

| | |
|---|---|
| nboots | value entered as number of bootstrap samples. |
| message | table with the warning messages generated by the function. |

## check_ncpus-internal *Check number of CPUs*

### Description

Checks the number of CPUs to be used when parallel computing is performed. The default value is 1 and the maximum is 2.

### Usage

```
check_ncpus(ncpus)
```

### Arguments

| | |
|---|---|
| ncpus | number of CPUs to be used when performing parallel computing. |

### Value

A list with two components:

| | |
|---|---|
| ncpus | value entered as number of CPUs chosen. |
| message | table with the warning messages generated by the function. |

## check_tim-internal *Check tim*

### Description

Cheking of the parameter time.

### Usage

```
check_tim(time)
```

### Arguments

| | |
|---|---|
| time | point of time at which the ROC curve estimate in prognosis scenarios will be computed. It is mandatory in this scenario. |

### Value

A list with two components:

| | |
|---|---|
| time | value entered as time. |
| message | table with the warning messages generated by the function. |

## check_type_outcome-internal *Check the type of scenario (diagnosis/prognosis)*

### Description

Determines the type of scenario handled: diagnosis or prognosis, under right or interval censorship, according to the parameters entered by the user.

### Usage

```
check_type_outcome(status, observed.time, left, right)
```

### Arguments

| | |
|---|---|
| status | numeric response vector. Only two values will be taken into account. The highest one is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. |
| observed.time | vector with the observed times for each subject, when dealing with time-dependent outcomes under right censorship. These values can be the event times or the censoring times. |

| | |
|---|---|
| `left` | vector containing the lower edges of the observed intervals. It is mandatory when dealing with prognosis scenarios and interval censorship, and will be ignored in other situations. |
| `right` | vector with the upper edges of the observed intervals. It is mandatory as well in prognosis scenarios and interval censorship and ignored in other situations. |

## Details

If both the vectors status and observed time are indicated the funtion assumes a prognosis scenario and right censorship. When only the vector status is entered, a diagnosis scenario is set up. If none of these parameters are indicated but the left and right ones, a prognosis scenario and interval censorship is assumed. Any other case, the function is not able to determine the type of scenario.

## Value

A list with a single component:

`type.outcome` string of length 6 with the following values:

- **"binout"**, in the case of diagnosis scenarios.
- **"timerc"**, for prognosis scenarios and right censorship.
- **"timeic"**, for prognosis scenarios and interval censorship.
- **"unknow"**, if it is not possible to determine the type of scenario.

---

# `compute_ROC-internal` *Weighted empirical ROC curve estimator*

---

## Description

Computes the weighted empirical ROC curve estimator associated to the input biomarker.

## Usage

```
compute_ROC(marker, probs, grid)
```

## Arguments

| | |
|---|---|
| `marker` | vector with the biomarker values. |
| `probs` | vector containing the probabilities corresponding to the predictive model. |
| `grid` | grid size. |

## Details

This function computes the weighted empirical estimators for the sensitivity (SE) and specificity (SP) using as weights the probabilities given by the predictive model. Then, the ROC curve is approximated through linear interpolation of 1 - SP and SE and computed at a partition of the $[0, 1]$ interval of size `grid`.

## Value

The returned value is a list with the following components:

| | |
|---|---|
| SE | vector with the weighted empirical estimator of the sensitivity. |
| SP | vector with the weighted empirical estimator of the specificity. |
| u | vector containing the points between 0 and 1 at which the ROC curve estimator will be computed. Its size is determined by the grid parameter. |
| ROC | ROC curve approximated at each point of the vector u. |
| auc | area under the weighted empirical ROC curve estimator. |
| marker | vector with the ordered biomarker values. |
| probs | vector with the probabilities of the predictive model corresponding to each biomarker value. |

## See Also

sMSbinout, sMStimerc and sMStimeic

---

# conf_int_print *AUC and confidence intervals*

---

## Description

Prints the AUC estimate value and its confidence intervals computed by the sMSROC function.

## Usage

```
conf_int_print(sMS)
```

## Arguments

| | |
|---|---|
| sMS | object of class sMS returned by the function sMSROC. |

## Details

This function reads the AUC, lower and upper edges of its confidence intervals and the confidence level at which they were computed and prints this information in a single line.

## Value

Printed string in the console containing the AUC, its confidence intervals and the confidence level at which they were computed.

## See Also

sMSROC

## Examples

```
data(diabet)
roc <- sMSROC(marker=diabet$stab.glu, status=diabet$glyhb, conf.int="T")
conf_int_print(roc)
```

---

`diabet` *Diabetes dataset*

---

## Description

This dataset contains part of the Diabetes Dataset (see References), courtesy of Dr John Schorling from the Department of Medicine, University of Virginia School of Medicine. This version contains 3 variables on 403 subjects interviewed to understand the prevalence of several cardiovascular risks factors in central Virginia for African Americans.

## Usage

```
data("diabet")
```

## Format

A data frame with 403 observations on the following 3 variables.

`stab.glu` a numeric vector indicating the level of stabilized glucose.

`glyhb` a numeric vector indicating the level of glycosolated hemoglobin.

`age` age in years of the participants.

`diab` a numeric vector indicating whether the subject is diagnosed as diabetic (value = 1) or not (value = 0).

## Details

The **diab** variable is not present in the original dataset. Here, values of glycosolated hemoglobin > 7.0 were taken as a positive diagnosis of diabetes (**diab** = 1) and those of glycosolated hemoglobin <= 7.0 as a negative diagnosis (**diab** = 0).

## Source

Full dataset can be downloaded at https://hbiostat.org/data.

## References

Willems JP, Saunders JT, Hunt DE, Schorling JB. Prevalence of coronary heart disease risk factors among rural blacks: a community-based study. South Med J. 1997 Aug;90(8):814-20. PMID: 9258308.

## Examples

```
data(diabet)
summary(diabet)
```

`evol_auc` *Evolution of the AUCs*

## Description

Plots, in prognosis scenarios, the areas under the ROC curves computed by the sMSROC estimator for a sequence of times.

## Usage

```
evol_auc(marker, status, observed.time, left, right,
         time = 1, meth = c("L", "S", "E"), grid = 500)
```

## Arguments

| | |
|---|---|
| `marker` | vector with the biomarker values. |
| `status` | numeric response vector. Only two values will be taken into account. The highest one is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. It is mandatory in prognosis scenarios and right censorship. |
| `observed.time` | vector with the observed times for each subject, for prognosis scenarios under right censorship. Notice that these values may be the event times or the censoring times. |
| `left` | vector containing the lower edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. |
| `right` | vector with the upper edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. The infinity is admissible as value (indicated as **inf**). |
| `time` | vector of times at which the sMS ROC curve estimator will be computed. The default value is 1. |
| `meth` | method for approximating the predictive model $P(D\|X = x)$. There are several options available:<br>• **"E",** allocates to each individual their own status value as having the event of interest or not. Those with missing status values or censored at a fixed point of time **t** are dismissed.<br>• **"L",** for proportional hazards regression models.<br>• **"S",** for smooth models. |
| `grid` | grid size for computing the AUC. Default value 500. |

## Details

This function calls the `sMSROC` function at each of the times indicated in the vector `time`, and the AUC is computed according to the parameters indicated.

## Value

A list with the following components:

| | |
|---|---|
| evol.auc | object of class ggplot. A graphic line plotting the AUCs at the considered times. |
| time | vector with the ordered values of the time entered as parameter. |
| auc | vector with the values of the AUCs computed at the times indicated at the time parameter. |

## See Also

sMSROC

## Examples

```
# Example of the use of the evol.AUC function
data(ktfs)
DT = ktfs
aucs <- evol_auc(marker = DT$score,
                 status = DT$failure,
                 observed.time = DT$time,
                 time = seq(2:3),
                 meth = "E")
aucs$evol.auc
```

---

# explore_plot *Graphical exploratory data analysis*

---

## Description

Plots the kernel density estimations of the biomarker distributions on positive and negative individuals.

## Usage

```
explore_plot(marker, status, observed.time, left, right, time)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| status | numeric response vector. The highest value is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. |
| observed.time | vector with the observed times for each subject, for prognosis scenarios under right censorship. Notice that these values may be the event times or the censoring times. |
| left | vector containing the lower edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. |

| | |
|---|---|
| right | vector with the upper edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. The infinity is admissible as value (indicated as **inf**). |
| time | point of time at which the sMS ROC curve estimator will be computed. The default value is 1. |

## Value

The ouput is a list with three components:

| | |
|---|---|
| plot | object of class ggplot. |
| neg | vector with the biomarker values on negative individuals. |
| pos | vector with the marker values on positive individuals. |

## See Also

explore_table

## Examples

```
data(diabet)
explore_plot(marker=diabet$stab.glu, status=diabet$diab)
```

---

# explore_table *Exploratory data analysis*

---

## Description

This function provides descriptive statistics for the pooled sample and the samples of positive, negative individuals and those whose condition is unknown.

## Usage

```
explore_table(marker, status, observed.time, left, right, time, d, ...)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. It is a mandatory parameter. |
| status | numeric response vector. The highest value is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. |
| observed.time | vector with the observed times for each subject, for prognosis scenarios under right censorship. Notice that these values may be the event times or the censoring times. |
| left | vector containing the lower edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. |

| | |
|---|---|
| right | vector with the upper edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. The infinity is admissible as value (indicated as **inf**). |
| time | point of time at which the sMS ROC curve estimator will be computed. The default value is 1. |
| d | number of decimal figures to which the results will be rounded. |
| ... | additional parameters of the `flextable` function which allow to customize the output table. |

## Details

The function computes the following descriptive statistics for the pooled sample and the samples of the different groups of individuals: minimum, maximun, mean, variance, standard deviation, and first, second and third quartiles.

## Value

The ouput is a list with two components:

| | |
|---|---|
| summary | matrix whose columns are the statistics described above and the rows show the corresponding results for each sample. |
| table | object of class `flextable` that represent the matrix in `summary` in a customizable table. |

## See Also

`explore_plot`

## Examples

```
data(diabet)
explore_table(marker=diabet$stab.glu, status=diabet$diab)
```

---

# fibrosis *Fibrosis dataset*

---

## Description

Synthetic dataset generated to simulate data from a study that aimed to assess the predictive ability of a constructed score to determine the worsening in the fibrosis stage in individuals infected by the hepatitis C (HC) virus. When participants underwent a target revision, their fibrosis stage, certain polymorphisms, and other clinical variables were collected. Highest stages of fibrosis were considered a worsening in the disease. See References for more information about the study.

## Usage

```
data("fibrosis")
```

## Format

A data frame with 722 observations and the following variables:

`Id` Identification label for each participant.

`Score` Score proposed to stratify participants infected by the HC virus according to the risk of a worsening in their fibrosis stage.

`Start` Lower edge of the observable interval.

`Stop` Upper edge of the observable interval. It can take the value infinity, represented as Inf.

## Source

Synthetic dataset.

## References

Vidal-Castineira JR. al. Genetic contribution of endoplasmic reticulum aminopeptidase 1 polymorphisms to liver fibrosis progression in patients with HCV infection. Journal of Molecular Medicine, 98:1245-1254, 2020. doi:10.1007/s00109020019481

## Examples

```
data(fibrosis)
summary(fibrosis)
```

---

# ktfs *KTFS dataset*

---

## Description

Dataset originally delivered in the `RISCA` package. It contains data from kidney transplant recipients for whom the Kidney Transplant Failure Score (KTFS) was collected. The KTFS is a score proposed by Foucher et al. (2010) (see References) to assess the recipients according to their risk of returning in dialysis.

## Usage

```
data("ktfs")
```

## Format

A data frame with 2169 observations and the following 3 variables:

`time` a numeric vector depicting the follow-up time in years.

`failure` a numeric vector indicating the graft failure at the end of the follow-up (1-Yes, 0-Censoring).

`score` a numeric vector representing the KTFS value.

## Source

This dataset is available at `RISCA` package. More information about the KTFS score can be found at https://www.divat.fr.

## References

Foucher Y. al. A clinical scoring system highly predictive of long-term kidney graft survival. Kidney International, 78:1288-94, 2020. doi:10.1038/ki.2010.232.

## Examples

```
data(ktfs)
summary(ktfs)
```

---

`pred_model_binout-internal`

*Predictive model estimation in diagnosis scenarios*

---

## Description

Estimation of the predictive models in diagnosis scenarios.

## Usage

```
pred_model_binout(marker, status, meth)
```

## Arguments

| | |
|---|---|
| `marker` | vector with the biomarker values. |
| `status` | numeric response vector. The highest value is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. |
| `meth` | method for approximating the predictive model $P(D\|X = x)$. The options are:<br>• **"L",** for **L**inear logistic regression models.<br>• **"S",** for **S**mooth models. |

## Details

- If `meth` = **"L"**, the logit transformation of the predicitive model is approximated by a linear logistic regression model:

$$P(D|X = x) = 1/(1 + \exp -\{\beta_0 + \beta_1 x\}),$$

with $\beta_0, \beta_1 \in \mathcal{R}$.

- If `meth` = **"S"**, the logit transformation of the predicitive model is estimated by the smooth logistic regression,

$$P(D|X = x) = 1/(1 + \exp\{-s(x)\}),$$

being $s(\cdot)$ the smooth function (splines, doi:10.1002/sim.4780080504).

## Value

The returned value is a list with the two components:

| | |
|---|---|
| marker | vector containing the ordered marker values. |
| probs | vector with the probabilities corresponding to each marker value estimated through the predictive model. |

## See Also

sMS_binout and sMSROC

---

# pred_model_emp-internal

*Predictive model (naive estimation)*

---

## Description

Naive estimation of the predictive model.

## Usage

```
pred_model_emp(marker, status)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| status | numeric response vector. The highest value is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. |

## Details

This method for estimating the predictive model is used in both diagnosis and prognosis scenarios. It allocates individuals their own condition as positive or negative. Those with unknown condition are dismissed.

## Value

The returned value is a list with two components:

| | |
|---|---|
| marker | vector containing the ordered marker values. |
| probs | vector with the probabilities corresponding to each marker value estimated through the predictive model. |

## See Also

sMS_binout, sMS_timerc, sMS_timeic and sMSROC

`pred_model_timeic-internal` *Predictive model in prognosis scenarios (I)*

## Description

Estimation of the predictive model in prognosis scenarios under interval censorship.

## Usage

```
pred_model_timeic(marker, left, right, outcome, time, meth)
```

## Arguments

| | |
|---|---|
| `marker` | vector with the biomarker values. |
| `left` | vector containing the lower edges of the observed intervals. It is mandatory in prognosis scenarios and interval censorship and ignored in other situations. |
| `right` | vector with the upper edges of the observed intervals. It is mandatory as well in prognosis scenarios and interval censorship and ignored in other situations. |
| `outcome` | vector with the condition of the subjects as positive, negative or unknown at the considered time `time`. |
| `time` | point of time at which the sMS ROC curve estimator will be computed. |
| `meth` | method for approximating the predictive model $P(D\|X = x)$. The options are:<br>• **"L",** for proportional hazards regression models taking into account the observation intervals (see Details).<br>• **"S",** for proportional hazards regression models **without** taking into account the observation intervals (see Details). |

## Details

- If `meth` = **"L"**, the event times are assumed to come from a Cox proportional hazards regression model and the predictive model is estimated as indicated in doi:10.1080/00949655.2020.1736071.

$$P(T \leq t \mid X = x) = \frac{S(U|x) - S(t|x)}{S(U|x) - S(V|x)},$$

  where $U = \min\{t, L\}$ and $V = \max\{t, R\}$, being L and R the random variables that stand for the edges of the observable interval containing the event time.

- If `meth` = **"S"**, the approximation is done by

$$P(T \leq t \mid X = x) = 1 - S(t|x),$$

  being $S(\cdot)$ the survival function at time **t** given the marker value, estimated through a proportional hazard model for interval censored data according to doi:10.2307/2530698.

## Value

The returned value is a list with three components:

| | |
|---|---|
| marker | vector containing the ordered marker values. |
| probs | vector with the probabilities corresponding to each marker value estimated through the predictive model. |
| outcome | vector with the condition of the subjects as positive, negative or censored at the considered time time. |

## See Also

sMS_timeic and sMSROC

---

# pred_model_timerc-internal

*Predictive model in prognosis scenarios (II)*

---

## Description

Estimation of the predictive model in prognosis scenarios under right censorship.

## Usage

```
pred_model_timerc(marker, status, observed.time, outcome, time, meth)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| status | numeric response vector. The highest value is assumed to stand for the subjects having the event under study. The lowest value, for those who do not. Any other value will not be considered. |
| observed.time | vector with the observed times for each subject. Notice that these values may be the event times or the censoring times. |
| outcome | vector with the status of the subjects as positive, negative or censored (unknown) at the considered time time. |
| time | point of time at which the sMS ROC curve estimator will be computed. |
| meth | method for approximating the predictive model $P(D\|X = x)$. The options are:<br>• **"L",** for proportional hazards regression models (see Details).<br>• **"S",** for smooth models (see Details). |

## Details

- If `meth` = **"L"**, the event times are assumed to come from a Cox proportional hazards regression model:

$$P(T \leq t \mid X = x) = 1 - \exp\{-\Delta_0(t) \cdot \exp\{\beta_0 + \beta_1 \cdot \log(x)\}\},$$

  where $\Delta_0(\cdot)$ is the baseline hazard function and $\beta_0, \beta_1 \in \mathcal{R}$.
- If `meth` = **"S"**, the approximation is done by

$$P(T \leq t \mid X = x) = 1 - \exp\{-\Delta_0(t) \cdot \exp\{s(x)\}\}$$

  being $s(\cdot)$ the smooth function (penalized splines, doi:10.1111/14679868.00125).

## Value

The returned value is a list with three components:

| | |
|---|---|
| `marker` | vector containing the ordered marker values. |
| `probs` | vector with the probabilities corresponding to each marker value estimated through the predictive model. |
| `outcome` | vector with the status of the subjects as positive, negative or censored at the considered time `time`. |

## See Also

`sMS_timerc` and `sMSROC`

---

`print.sMSROC` *Print sMSROC*

---

## Description

Prints the estimated AUC and the probabilistic model used to compute the predictive model.

## Usage

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

## Arguments

| | |
|---|---|
| `x` | object of class `sMS` returned by the function `sMSROC`. |
| `...` | Ignored. |

## Details

This function prints the estimated area under the ROC curve computed through the sMSROC estimator and the probabilistic model used to compute the predictive model.

## Value

Printed output in the console containing the information described above.

## See Also

sMSROC

## Examples

```
data(diabet)
roc <- sMSROC(marker=diabet$stab.glu, status=diabet$glyhb, conf.int="T")
print(roc)
```

---

probs_pred *Plot of the predictive model*

---

## Description

This function plots the predicted probabilities for each marker value computed through the predictive model together, with 95% pointwise confidence intervals.

## Usage

```
probs_pred(sMS, var, nboots, parallel, ncpus)
```

## Arguments

| | |
|---|---|
| sMS | object of class sMS returned from function sMSROC. |
| var | parameter indicating whether 95% pointwise confidence intervals for the predictive model will be plotted (value **"T"**) or not (value **"F"**). The default value is "F". |
| nboots | number of bootstrap samples to be generated for computing the pointwise confidence intervals. The default value is 500. |
| parallel | parameter indicating whether parallel computing will be performed (value **"T"**) or not (value **"F"**). The default is "F". |
| ncpus | number of CPUS to be used in the case of carrying out parallel computing. The default value is 1 and the maximum is 2. |

## Details

The function plots the probability function estimation of the predictive model versus the biomaker. It also computes and plots 95% pointwise confidence intervals on the same graphic when the var parameter is set to "T".

The variance of the probability estimates, obtained by the predictive model, is computed via bootstrap with nboots samples.

## Value

A list with these components:

| | |
|---|---|
| `plot` | object of class `ggplot` (graphical output). |
| `thres` | ordered biomarker values (x-axis coordinates). |
| `probs` | predicted probabilities (y-axis coordinates). |
| `sd.probs` | estimates of the standard deviation of the predicted probabilities. |

## See Also

`pred_model_binout`, `pred_model_timerc` and `pred_model_timeic`

## Examples

```
data(ktfs)
DT <- ktfs
roc <- sMSROC(marker = DT$score,
              status = DT$failure,
              observed.time = DT$time,
              time = 5,
              meth = "S")
probs <- probs_pred(sMS = roc)
probs$plot
```

---

# sMS_binout-internal *sMS estimator for diagnostic biomarkers*

---

## Description

Wrap function for computing the sMS estimator in diagnosis scenarios.

## Usage

```
sMS_binout(marker, status, meth, grid, probs, all)
```

## Arguments

| | |
|---|---|
| `marker` | vector with the biomarker values. |
| `status` | numeric response vector. |
| `meth` | method for approximating the predictive model $P(D\|X = x)$.<br>• **"E",** allocates to each individual their own condition as positive or negative. Those whose condition is unknown at time `time` are dismissed.<br>• **"L",** for **L**inear logistic regression models (see details in `sMSROC`).<br>• **"S",** for **S**mooth models (see details `sMSROC`). |
| `grid` | grid size. |

| | |
|---|---|
| probs | vector with the probabilities from the predictive model when it is manually entered. |
| all | parameter indicating whether all probabilities given by the predictive model should be considered (value **"T"**) or just those corresponding to individuals whose condition as positive or negative is unknown (**"F"**). The default value is (**"T"**). |

## Details

The function obtains the probabilities corresponding to the predictive model (first stage of the sMS ROC curve estimator). If they were not manually entered, the functions `pred.mod.emp` or `pred.mod.binout` are called depending on the chosen `meth`. Then, it calls the function `computeROC` to compute the weighted empirical ROC curve estimator (second stage).

## Value

The returned value is a list with the following components:

| | |
|---|---|
| SE | vector with the weighted empirical estimator of the sensitivity. |
| SP | vector with the weighted empirical estimator of the specificity. |
| u | vector containing the points between 0 and 1 at which the ROC curve estimator will be computed. Its size is determined by the `grid` parameter. |
| ROC | ROC curve approximated at each point of the vector `u`. |
| auc | area under the weighted empirical ROC curve estimator. |
| marker | vector with the ordered biomarker values. |
| probs | vector with the probabilities of the predictive model corresponding to each biomarker value. |

## See Also

`pred_mod_emp`, `pred_mod_binout`, `computeROC`, `sMS_timerc` and `sMS_timeic`

---

# sMS_timeic-internal *sMS estimator for prognostic biomarkers and interval censoring*

---

## Description

Wrap function for computing the sMS estimator in prognosis scenarios under interval censorship.

## Usage

```
sMS_timeic(marker, left, right, outcome, time, meth, grid, probs, all)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| left | vector containing the lower edges of the observed intervals. |
| right | vector with the upper edges of the observed intervals. The infinity is admissible as value (indicated as **inf**). |
| outcome | vector containing the condition of the individuals as positive, negative or censored at the time time. |
| time | point of time at which the sMSROC curve estimator will be computed. |
| meth | method for approximating the predictive model $P(D\|X = x)$.<br>• **"E",** allocates to each individual their own condition as positive or negative. Those whose condition is unknown at time time are dismissed.<br>• **"L",** for proportional hazards regression models taking into account the observation intervals (see Details sMSROC).<br>• **"S",** for proportional hazards regression models **without** taking into account the observation intervals (see Details sMSROC). |
| grid | grid size. |
| probs | vector with the probabilities from the predictive model when it is manually entered. |
| all | parameter indicating whether all probabilities given by the predictive model should be considered (value **"T"**) or just those corresponding to individuals whose condition as positive or negative is unknown (**"F"**). The default value is (**"T"**). |

## Details

This function gets the probabilities corresponding to the predictive model (first stage of the sMS ROC curve estimator). If they were not manually entered, the functions pred.mod.emp or pred.mod.timeic are called depending on the chosen meth. Then, it calls the function computeROC to compute the weighted empirical ROC curve estimator (second stage).

## Value

The returned value is a list with the following components:

| | |
|---|---|
| SE | vector with the weighted empirical estimator of the sensitivity. |
| SP | vector with the weighted empirical estimator of the specificity. |
| u | vector containing the points between 0 and 1 at which the ROC curve estimator will be computed. Its size is determined by the grid parameter. |
| ROC | ROC curve approximated at each point of the vector u. |
| auc | area under the weighted empirical ROC curve estimator. |
| marker | vector with the ordered biomarker values. |
| outcome | vector with the condition of the individuals at time time as positive, negative or unknown. |
| probs | vector with the probabilities of the predictive model corresponding to each biomarker value. |

## See Also

pred.mod.emp, pred.mod.binout, computeROC, sMS.binout and sMS.timerc

---

sMS_timerc *sMS estimator for prognostic biomarkers and right censorship*

---

## Description

Wrap function for computing the sMS estimator in prognosis scenarios under right censorship.

## Usage

```
sMS_timerc(marker, status, observed.time, outcome, time,
           meth, grid, probs, all)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| status | numeric response vector. |
| observed.time | vector with the observed times. These values may be the event times or the censoring times. |
| outcome | vector containing the condition of the individuals as positive, negative or censored (unknown) at the time time. |
| time | point of time at which the sMSROC curve estimator will be computed. |
| meth | method for approximating the predictive model $P(D\|X = x)$.<br>• **"E",** allocates to each individual their own condition as positive or negative. Those whose condition is unknown at time time are dismissed.<br>• **"L",** for **L**inear proportional hazards regression models (see details in sMSROC).<br>• **"S",** for **S**mooth models (see details in sMSROC). |
| grid | grid size. |
| probs | vector with the probabilities from the predictive model when it is manually entered. |
| all | parameter indicating whether all probabilities given by the predictive model should be considered (value **"T"**) or just those corresponding to individuals whose condition as positive or negative is unknown (**"F"**). The default value is (**"T"**). |

## Details

This function gets the probabilities corresponding to the predictive model (first stage of the sMS ROC curve estimator). If they were not manually entered, the functions pred.mod.emp or pred.mod.timerc are called depending on the chosen **meth**. Then, it calls the function computeROC to compute the weighted empirical ROC curve estimator (second stage).

## Value

The returned value is a list with the following components:

| | |
|---|---|
| SE | vector with the weighted empirical estimator of the sensitivity. |
| SP | vector with the weighted empirical estimator of the specificity. |
| u | vector containing the points between 0 and 1 at which the ROC curve estimator will be computed. Its size is determined by the grid parameter. |
| ROC | ROC curve approximated at each point of the vector u. |
| auc | area under the weighted empirical ROC curve estimator. |
| marker | vector with the ordered biomarker values. |
| outcome | vector with the condition of the individuals at time time as positive, negative or cenrored (unknown). |
| probs | vector with the probabilities of the predictive model corresponding to each biomarker value. |

## See Also

pred.mod.emp, pred.mod.binout, computeROC, sMS.timerc, sMS.timeic

---

# sMSROC *sMS ROC curve estimator computation*

---

## Description

Core function for computing the sMS ROC estimator which fits the estimation of the ROC curve when the outcome of interest is time-dependent (**prognosis** scenarios) and when it is not (**diagnosis** scenarios).

## Usage

```
sMSROC(marker, status, observed.time, left, right, time,
       meth, grid, probs, sd.probs,
       conf.int, ci.cl, ci.meth, ci.nboots, parallel, ncpus, all)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| status | numeric response vector. The highest value is assumed to stand for the subjects having the event under study. The lowest one, for those who do not. Any other value will not be considered. It is a mandatory parameter in diagnosis scenarios. |
| observed.time | vector with the observed times for each subject, for prognosis scenarios under right censorship. Notice that these values may be the event times or the censoring times. |

`left` vector containing the lower edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations.

`right` vector with the upper edges of the observed intervals. It is mandatory in prognosis scenarios under interval censorship and ignored in other situations. The infinity is admissible as value (indicated as **inf**).

`time` point of time at which the sMS ROC curve estimator will be computed. The default value is 1.

`meth` method for approximating the predictive model $P(D|X = x)$. There are several options available:

- **"E",** allocates to each individual their own condition as positive or negative. Those whose condition is unknown at time `time` are dismissed.
- **"L",** for **L**inear logistic regression and proportional hazards regression models (see Details).
- **"S",** for **S**mooth models (see Details).

`probs` vector containing the probabilities corresponding to the predictive model when it has been externally computed. Only values within [0,1] are admissible.

`sd.probs` vector with the standard deviations of the probabilities entered in `probs`. It is an optional parameter.

`grid` grid size for computing the AUC. Default value 1000.

`conf.int` indicates whethet a conficence interval for the AUC will be computed (**"T"**) or not (**"F"**). The default value is (**"F"**).

`ci.cl` confidence level at which the confidence interval for the AUC will be provided. The default value is 95%. This parameter is ignored when `conf.int` is set to **"F"**.

`ci.meth` method for computing the confidence interval for the AUC. There are three options:

- **"E"**, for the **E**mpirical variance estimation.
- **"V"**, for the theoretical **V**ariance estimation.
- **"B"**, for the **B**ootstrap percentile approximation.
  The empirical method **E** is taken as default value and the parameter is ignored too when `ci.cl` value is **"F"**.

`ci.nboots` number of boostrap samples to be run when **B**oostrap is set as `ci.meth` parameter. The default value is 500 and it is not taken into account when no confidence interval is computed.

`parallel` indicates whether parallel computing will be done (**"T"**) or not (**"F"**) when computing the variance of the AUC through the methods **"V"** and **"B"**.

`ncpus` number of CPUS that will be used when parallel computing is chosen. The default value is 1 and the maximum is 2.

`all` parameter indicating whether all probabilities given by the predictive model should be considered (value **"T"**) or just those corresponding to individuals whose condition as positive or negative is unknown (**"F"**). The default value is (**"T"**).

## Details

The Two-stages mixed-subjects (sMSROC) ROC curve estimator links diagnosis and prognosis scenarios through a general predictive model (first stage) and the weighted empirical estimator of the cumulative distribution function of the biomarker (second stage).

The predictive model $P(D|X = x)$ depicts the relationship between the biomarker and the binary response variable. It is approximated through the most suitable probabilistic model.

For **diagnosis** scenarios:

- If `meth` = **"L"**, the logit transformation of the predicitive model is approximated by a linear logistic regression model:

$$P(D|X = x) = 1/(1 + \exp -\{\beta_0 + \beta_1 x\}),$$

  with $\beta_0, \beta_1 \in \mathcal{R}$.

- If `meth` = **"S"**, the logit transformation of the predicitive model is estimated by the smooth logistic regression,

$$P(D|X = x) = 1/(1 + \exp\{-s(x)\}),$$

  being $s(\cdot)$ the smooth function (splines, doi:10.1002/sim.4780080504).

Notice that the predictive model allows to compute the probability of being positive/negative even when the actual belonging group is unknown.

For **prognosis** scenarios and right censorship:

- If `meth` = **"L"**, the event times are assumed to come from a Cox proportional hazards regression model:

$$P(T \leq t \mid X = x) = 1 - \exp\{-\Delta_0(t) \cdot \exp\{\beta_0 + \beta_1 \cdot \log(x)\}\},$$

  where $\Delta_0(\cdot)$ is the baseline hazard function and $\beta_0, \beta_1 \in \mathcal{R}$.

- If `meth` = **"S"**, the approximation is done by

$$P(T \leq t \mid X = x) = 1 - \exp\{-\Delta_0(t) \cdot \exp\{s(x)\}\}$$

  being $s(\cdot)$ the smooth function (penalized splines, doi:10.1111/14679868.00125).

Finally, for **prognosis** scenarios and interval censorship:

- If `meth` = **"L"**, the event times are assumed to come from a Cox proportional hazards regression model and the predictive model is estimated as indicated in doi:10.1080/00949655.2020.1736071.

$$P(T \leq t \mid X = x) = \frac{S(U|x) - S(t|x)}{S(U|x) - S(V|x)},$$

  where $U = \min\{t, L\}$ and $V = \max\{t, R\}$, being L and R the random variables that stand for the edges of the observable interval containing the event time.

- If `meth` = **"S"**, the approximation is done by

$$P(T \leq t \mid X = x) = 1 - S(t|x),$$

  being $S(\cdot)$ the survival function at time **t** given the marker value, estimated through a proportional hazard model for interval censored data according to doi:10.2307/2530698.

The confidence intervals for the AUC can be computed in three different ways according to parameter `ci.meth`. When it is set to **"E"** the variance of the AUC is estimated by the empirical procedure and when the chosen option is **"V"**, the theoretical approximation is used (see doi:10.1515/ijb2019-0097). The third option in by using the Bootstrap percentile.

## Value

The ouput is an objetc of class `sMSROC` with the following components:

| | |
|---|---|
| `thres` | vector containing the biomarker values for which sensitivity and specificity were computed. |
| `SE` | vector with the estimates of the sensitivity. |
| `SP` | vector with the estimates of the specificity. |
| `probs` | vector with the probabilities corresponding to the predictive model. |
| `u` | vector containing the points between 0 and 1 at which the ROC curve estimator will be computed. Its size is determined by the `grid` parameter. |
| `ROC` | ROC curve approximated at each point of the vector `u`. |
| `auc` | area under sMSROC curve estimator. |
| `auc.ci.l` | lower edge of the confidence interval for the AUC. |
| `auc.ci.u` | upper edge of the confidence interval for the AUC. |
| `ci.cl` | confidence level at which the confidence interval for the AUC were computed. |
| `ci.meth` | method chosen for computing the confidence interval for the AUC. |
| `time` | point of time at which the sMS ROC curve estimator was computed in prognosis scenarios. |
| `data` | list contaning several parameters used in the internal functions, when applicable:<br>• data_type - type of scenario handled (diagnosis/prognosis, under right or interval censorship).<br>• grid - grid size.<br>• marker - vector with the biomarker values.<br>• outcome - vector with the condition of the individuals at time `time` as positive, negative or unknown.<br>• ncpus - CPUs used if parallel computing was performed.<br>• ci.nboots - number of bootstrap samples generated for computing the confidence intervals for the AUC.<br>• parallel - was parallel computing performed?<br>• meth - method used to compute the predictive model.<br>• status - response vector.<br>• observed.time - vector with the observed times for each subject.<br>• left - vector with the lower edges of the observed intervals.<br>• right - vector with the upper edges of the observed intervals. |
| `message` | table containing the warning messages generated during the execution of the function. |

## References

S. Díaz-Coto, P. Martínez-Camblor, and N. O. Corral-Blanco. Cumulative/dynamic ROC curve estimation under interval censorship. Journal of Statistical Computation and Simulation, 90(9):1570–1590, 2020. doi:10.1080/00949655.2020.1736071.

S. Díaz-Coto, N. O. Corral-Blanco, and P. Martínez-Camblor. Two-stage receiver operating-characteristic curve estimator for cohort studies. The International Journal of Biostatistics, 17:117–137, 2021. doi:10.1515/ijb20190097.

Finkelstein, Dianne M. A Proportional Hazards Model for Interval-Censored Failure Time Data. Biometrics 42, no. 4 (1986): 845–54. doi:10.2307/2530698.

Durrleman S, Simon R. Flexible regression models with cubic splines. Statistics in Medicine 1989; 8(5): 551-561.doi:10.1002/sim.4780080504

Hurvich C, Simonoff J, Tsai CL. Smoothing parameter selection in nonparametric regression using an improved Akaike 1998. J.R. Statist. Soc. 60 271-293. doi:10.1111/14679868.00125

B. Efron and R. J. Tibshirani. An Introduction to the Bootstrap. CRC press, 1994.

## Examples

```
data(ktfs)
DT <- ktfs
sROC <- sMSROC(marker = DT$score, status = DT$failure,
               observed.time = DT$time, time = 5, meth = "L", conf.int = "T",
               ci.cl =0.90, ci.meth = "E")
```

---

`sMSROC_plot` *Plot of the sMS ROC curve estimate*

---

## Description

Provides informative plots of the sMS ROC curve estimates.

## Usage

```
sMSROC_plot(sMS, m.value)
```

## Arguments

| | |
|---|---|
| `sMS` | object of class `sMS` returned from function `sMSROC`. |
| `m.value` | marker value. It is an optional parameter that, when indicated, adds over the graphic of the ROC curve, the point which corresponds to that marker value. |

## Details

The function provides two types of graphics:

- A basic plot approximating the ROC curve by the pairs given by the sequences **1 - SP** and **SE**, from the `sMSROC` object. The layers `geom_roc()` and `roc_style()` from the `plotROC` package were added to this plot, which make possible to take advantage of the functionality of this package.
- A customized graphic of the ROC curve whose class is `ggplot`, obtained approximating the sequences **1 - SP** and **SE**. When te parameter `m.value` is indicated, the final plot displays over the ROC curve estimate the point that corresponds to the entered value.

## Value

A list with the following elements:

| | |
|---|---|
| `basic.plot` | object that can be used and customized by the tools from the `plotROC` package. |
| `roc.plot` | object of class `ggplot`. Although it is already customized (title, colors, axis labels, ..., etc.) the end-users can make their own changes by adding the corresponding layers, with the available tools from the `ggplot2` package. |

## See Also

`sMSROC`

## Examples

```
#  Example of the use of the plot.sMSROC function
data(ktfs)
DT = ktfs
ROC <- sMSROC(marker = DT$score,
              status = DT$failure,
              observed.time = DT$time,
              time = 5,
              meth = "S")
plot <- sMSROC_plot(sMS = ROC, m.value = 4.2)
plot$basicplot; plot$rocplot
```

---

# variance_probs-internal

*Variance of the predictive model*

---

## Description

Estimation of the variance of the predictive model by bootstrap.

## Usage

```
variance_probs(marker, outcome, status, observed.time, left, right, time,
               meth, data_type, grid, probs, ci.nboots, parallel, ncpus, all)
```

## Arguments

| | |
|---|---|
| marker | vector with the biomarker values. |
| outcome | vector with the condition of the subjects as positive, negative or unknown at the considered time time. |
| status | response vector with the outcome values. The highest one is assumed to stand for the subjects having the event under study. |
| observed.time | vector with the observed times for each subject. |
| left | vector with the lower edges of the observed intervals. |
| right | vector with the upper edges of the observed intervals. |
| time | point of time at which the sMS ROC curve estimator will be computed. |
| meth | method for approximating the predictive model $P(D\|X = x)$. |
| data_type | scenario handled. |
| grid | grid size. |
| probs | vector containing the probabilities estimated through to the predictive model. |
| ci.nboots | number of bootstrap samples. |
| parallel | indicates whether parallel computing will be done or not. |
| ncpus | number of CPUs to use if parallel computing is performed. |
| all | indicates whether the probabilities from the predictive model should be considered or not. |

## Value

List with a single component:

| | |
|---|---|
| sd.probs | vector containing the standard deviation of the probabilities of the predictive model. |

# Index


auc_ci_boot *(auc_ci_boot-internal)*, 2
auc_ci_boot-internal, 2
auc_ci_empr *(auc_ci_empr-internal)*, 3
auc_ci_empr-internal, 3
auc_ci_nvar *(auc_ci_nvar-internal)*, 4
auc_ci_nvar-internal, 4

check_ci_cl *(check_ci_cl-internal)*, 5
check_ci_cl-internal, 5
check_conf_int *(check_conf_int-internal)*, 6
check_conf_int-internal, 6
check_grid *(check_grid-internal)*, 7
check_grid-internal, 7
check_marke_timeic-internal, 8
check_marker_binout *(check_marker_binout-internal)*, 9
check_marker_binout-internal, 9
check_marker_timeic *(check_marke_timeic-internal)*, 8
check_marker_timerc *(check_marker_timerc-internal)*, 10
check_marker_timerc-internal, 10
check_meth *(check_meth-internal)*, 11
check_meth-internal, 11
check_nboots *(check_nboots-internal)*, 12
check_nboots-internal, 12
check_ncpus *(check_ncpus-internal)*, 12
check_ncpus-internal, 12
check_tim *(check_tim-internal)*, 13
check_tim-internal, 13
check_type_outcome *(check_type_outcome-internal)*, 13
check_type_outcome-internal, 13
compute_ROC *(compute_ROC-internal)*, 14
compute_ROC-internal, 14

conf_int_print, 15

diabet, 16

evol_auc, 17
explore_plot, 18
explore_table, 19

fibrosis, 20

ktfs, 21

pred_model_binout *(pred_model_binout-internal)*, 22
pred_model_binout-internal, 22
pred_model_emp *(pred_model_emp-internal)*, 23
pred_model_emp-internal, 23
pred_model_timeic *(pred_model_timeic-internal)*, 24
pred_model_timeic-internal, 24
pred_model_timerc *(pred_model_timerc-internal)*, 25
pred_model_timerc-internal, 25
print.sMSROC, 26
probs_pred, 27

sMS_binout *(sMS_binout-internal)*, 28
sMS_binout-internal, 28
sMS_timeic *(sMS_timeic-internal)*, 29
sMS_timeic-internal, 29
sMS_timerc, 31
sMSROC, *4, 5*, *15*, *26*, *28*, *30, 31*, 32
sMSROC_plot, 36

variance_probs *(variance_probs-internal)*, 37
variance_probs-internal, 37