Package 'pmvalsampsize'

November 16, 2023

Version 0.1.0

Date 2023-11-14

Language en-GB

Title Sample Size for External Validation of a Prediction Model

Maintainer Joie Ensor <j.ensor@bham.ac.uk>

Depends R (>= 2.1)

Imports graphics, pROC, utils

Suggests

Description Computes the minimum sample size required for the external validation of an existing multivariable prediction model using the criteria proposed by Archer (2020) <doi:10.1002/sim.8766> and Riley (2021) <doi:10.1002/sim.9025>.

License GPL (>= 3)

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation no

Author Joie Ensor [aut, cre]

Repository CRAN

Date/Publication 2023-11-16 20:00:02 UTC

R topics documented:

pmvalsampsize *pmvalsampsize - Sample Size for External Validation of a Prediction Model*

Description

pmvalsampsize computes the minimum sample size required for the external validation of an existing multivariable prediction model using the criteria proposed by Archer *et al.* and Riley *et al.* pmvalsampsize can currently be used to calculate the minimum sample size for the external validation of models with binary outcomes.

Continuous and survival (time-to-event) outcome model calculations are a work in progress.

Usage

```
pmvalsampsize(
  type,
  cslope = 1,
  csciwidth = 0.2,
  oe = 1,
  oeciwidth = 0.2,
  cstatistic = NA,
  cstatciwidth = 0.1,
  simobs = 1e+06,
  lpnormal = NA,
  lpbeta = NA,
  lpcstat = NA,
  tolerance = 5e-04,
  increment = 0.1,
  oeseincrement = 1e-04,
  graph = FALSE,
  trace = FALSE,
  prevalence = NA,
  seed = 123456,
  sensitivity = NA,
  specificity = NA,
  threshold = NA,
  nbciwidth = 0.2,
  nbseincrement = 1e-04
)
```

Arguments

type	specifies the type of analysis for which sample size is being calculated

- "b" specifies sample size calculation for a prediction model with a binary outcome

`cslope` specifies the anticipated c-slope performance in the validation sample. Default conservatively assumes perfect c-slope=1. The value could alternatively be based on a previous validation study for example. For **binary outcomes** the c-slope calculation requires the user to specify a distribution for the assumed LP in the validation sample (or alternatively the distribution of predicted probabilities in the validation sample). See `lp*()` options below.

`csciwidth` specifies the target CI width (acceptable precision) for the c-slope performance. Default assumes CI width=0.2.

`oe` specifies the anticipated O/E performance in the validation sample. Default conservatively assumes perfect O/E=1.

`oeciwidth` specifies the target CI width (acceptable precision) for the E/O performance. Default assumes CI width=0.2. The choice of CI width is context specific, and depends on the event probability in the population. See Riley et al. for further details.

`cstatistic` specifies the anticipated c-statistic performance in the validation sample. This is a **required input**. May be based on the optimism-adjusted c-statistic reported in the development study for the existing prediction model. Ideally, this should be an optimism-adjusted c-statistic. NB: This input is also used when using the `lpcstat()` option.

`cstatciwidth` specifies the target CI width (acceptable precision) for the c-statistic performance. Default assumes CI width=0.1.

`simobs` specifies the number of observations to use when simulating the LP distribution for c-slope calculation in criteria 2. Default observations=1,000,000. Higher `simobs()` values will reduce random variation further.

`lpnormal` defines parameters to simulate the LP distribution for criteria 2 from a normal distribution. The user must specify the mean and standard deviation (in this order) of the LP distribution.

`lpbeta` defines parameters to simulate the distribution of predicted probabilities for criteria 2 from a beta distribution. The user must specify the alpha and beta parameters (in this order) of the probability distribution. The LP distribution is then generated internally using this probability distribution.

`lpcstat` defines parameters to simulate the LP distribution for criteria 2 assuming that the distribution of events and non-events are normal with a common variance. The user specifies a single input value - the expected mean for the non-events distribution. This could be informed by clinical guidance. However, this input is taken as a starting value and an iterative process is used to identify the most appropriate values for the event and non-event distributions so as to closely match the anticipated prevalence in the validation sample. NB: this approach makes strong assumptions of normality and equal variances in each outcome group, which may be unrealistic in most situations.

`tolerance` for use with `lpcstat` option. Sets the tolerance for agreement between the simulated and expected event proportion during the iterative procedure for calculating the mean for the non-events distribution.

`increment` for use with `lpcstat` option. Sets increment by which to iterate the value of the mean for the non-events distribution. Trial and error may be necessary as it is dependent on how close the initial input for the non-event mean in `lpcstat` is to the required value. If the process takes a particularly long time then the user could try an alternative increment value, or an alternative non-event mean value in `lpcstat`. The `trace` option may be useful in such circumstances.

`oeseincrement` sets the increment by which to iterate when identifying the SE(ln(OE)) to meet the target CI width specified for OE. The default iteration increment=0.0001. In the majority of cases this will be suitably small to ensure a precise SE is identified. The user should check the output table to ensure that the target CI width has been attained and adjust the increment if necessary.

`graph` specifies that a histogram of the simulated LP distribution for criteria 2 is produced. The graph also details summary statistics for the simulated distribution. Useful option for checking the simulated LP distribution against the source of input parameters. Also useful for reporting at publication.

`trace` for use with `lpcstat` option. Specifies that a trace of the values obtained in each iteration when identifying the non-event mean is output. Useful when finding the appropriate values for `lpcstat` & `increment()` is proving difficult!

`prevalence` specifies the overall outcome proportion (for a prognostic model) or overall prevalence (for a diagnostic model) expected within the model validation sample. This is a **required input**. This should be derived based on previous studies in the same population or directly from the validation sample if to hand.

`seed` specifies the initial value of the random-number seed used by the random-number functions when simulating data to approximate the LP distribution for criteria 2.

`sensitivity` specifies the anticipated sensitivity performance in the validation sample at the chosen risk threshold (specified using `threshold`). If sensitivity and specificity are not provided then `pmvalsampsize` uses the simulated LP distribution from criteria 2 and the user-specified risk threshold to estimate the anticipated sensitivity and specificity to be used in calculation of net benefit. NB: net benefit criteria is not calculated if either i) `sensitivity`, `specificity` and `threshold` or ii) `threshold` option are not specified.

`specificity` specifies the anticipated specificity performance in the validation sample at the chosen risk threshold (specified using `threshold`). If sensitivity and specificity are not provided then `pmvalsampsize` uses the simulated LP distribution from criteria 2 and the user-specified risk threshold to estimate the anticipated sensitivity and specificity to be used in calculation of net benefit. NB: net benefit criteria is not calculated if either i) `sensitivity`, `specificity` and `threshold` or ii) `threshold` option are not specified.

`threshold` specifies the risk threshold to be used for calculation of net benefit performance of the model in the validation sample. If sensitivity and specificity are not provided then `threshold` must be given in order for `pmvalsampsize` to assess sample size requirements for net benefit. NB: net benefit criteria is not calculated if either i) `sensitivity`, `specificity` and `threshold` or ii) `threshold` option are not specified.

`nbciwidth` specifies the target CI width (acceptable precision) for the standardised net benefit performance. Default assumes CI width=0.2. The choice of CI width is context specific. See Riley et al. for further details.

`nbseincrement` sets the increment by which to iterate when identifying the SE(standardised net benefit) to meet the target CI width specified for standardised net benefit. The default iteration increment=0.0001. In the majority of cases this will be suitably small to ensure a precise SE is identified. The user should check the output table to ensure that the target CI width has been attained and adjust the increment if necessary.

Details

A series of criteria define the sample size needed to ensure precise estimation of key measures of prediction model performance, allowing conclusions to be drawn about whether the model is potentially accurate and useful in a given population of interest.

For **binary outcomes**, there are three criteria to calculate the sample size (N) needed for:
i) precise estimation of the Observed/Expected (O/E) statistic,
ii) precise estimation of the calibration slope (c-slope), and
iii) precise estimation of the c-statistic.

The sample size calculation requires the user to pre-specify the following;
- the outcome event proportion
- the anticipated O/E performance
- the target precision (CI width) for the O/E
- the anticipated c-slope
- the target precision (CI width) for the c-slope
- the anticipated c-statisitic performance
- the target precision (CI width) for the c-statisitic
- the distribution of estimated probabilities from the model, ideally specified on the log-odds scale
- AKA the Linear Predictor (LP)

With thanks to Richard Riley for helpful feedback

Value

A list including a matrix of calculated sample size requirements for each criteria defined under *'Details'*, and a series of vectors of parameters used in the calculations as well as the the final recommended minimum sample size and number of events required for external validation.

Author(s)

Joie Ensor (University of Birmingham, j.ensor@bham.ac.uk)

References

Archer L, Snell K, Ensor J, Hudda MT, Collins GS, Riley RD. Minimum sample size for external validation of a clinical prediction model with a continuous outcome. *Statistics in Medicine. 2020.*

Riley RD, Debray TPA, Collins G, Archer L, Ensor J, van Smeden M, Snell KIE. Minimum sample size for external validation of a clinical prediction model with a binary outcome. *Statistics in Medicine. 2021.*

Examples

```
# Examples based on those included in the papers referenced below by
# Riley et al. & Archer et al. published in Statistics in Medicine.

# Note that the examples below use a very low simulation sample for criteria
# 2 for brevity. Sizes matching the default or higher are recommended to
# minimise uncertainty in the calculated sample size requirements.

# Binary outcomes (Logistic prediction models)
# Use pmvalsampsize to calculate the minimum sample size required to
# externally validate an existing multivariable prediction model for a
# binary outcome (e.g. mechanical heart valve failure).  Based on previous
# evidence, the outcome prevalence is anticipated to be 0.018 (1.8%) and the
# reported c-statistic was 0.8.  The LP distribution was published and
# appeared normally distributed with mean(SD) of -5(2.5).  We target default
# CI widths for all but O/E CI width=1 (see Riley et al. for details). We can
# use the graph option to check the simulated distribution is appropriate.

pmvalsampsize(type = "b", prevalence = 0.018, cstatistic = 0.8,
lpnorm = c(-5,2.5), oeciwidth = 1, simobs = 1000)


# Alternatively, lets assume that the authors provided a distribution of
# predicted probabilities (e.g. as part of a calibration plot).  We can use
# this to specify parameters for a beta distribution to simulate the LP
# distribution as below:
pmvalsampsize(type = "b", prevalence = 0.018, cstatistic = 0.8,
lpbeta = c(0.5,0.5), oeciwidth = 1, simobs = 1000)

# Finally, we can use the anticipated c-statistic to simulate the event and
# non-event distributions assuming normality and common variances.  We input
# a starting value for the mean of the non-events distribution as below:

pmvalsampsize(type = "b", prevalence = 0.018, cstatistic = 0.8,
lpcstat = -4.7, oeciwidth = 1, seed = 1234, simobs = 10000)
```

Index