Bootstrap internal validation command for predictive logistic regression models

Fernandez-Felix, B. M.; García-Esquinas, E.; Muriel, A.; Royuela, A.; Zamora, J.

DOI:
10.1177/1536867X211025836

License:
Creative Commons: Attribution-NoDerivs (CC BY-ND)

Document Version
Early version, also known as pre-print

Citation for published version (Harvard):
Fernandez-Felix, BM, García-Esquinas, E, Muriel, A, Royuela, A & Zamora, J 2021, 'Bootstrap internal validation command for predictive logistic regression models', *Stata Journal*, vol. 21, no. 2, pp. 498-509.
https://doi.org/10.1177/1536867X211025836

Bootstrap internal validation command for predictive logistic regression models

Fernandez-Felix, B. M.; Garcia-Esquinas, E.; Muriel, A.; Royuela, A.; Zamora, J

[1]Clinical Biostatistics Unit Hospital Ramón y Cajal (IRYCIS), CIBER Epidemiology and Public Health (CIBERESP), Madrid, Spain, borjam.fernandez@hrc.es

[2]Department of Preventive, Medicine and Public Health, Autonomous University of Madrid and Idipaz, CIBERESP, Madrid, Spain

[3]Clinical Biostatistics Unit, Hospital Ramón y Cajal (IRYCIS), CIBERESP, Madrid, Spain

[4]Biostatistics Unit, Puerta de Hierro Biomedical Research Institute, CIBERESP, Madrid, Spain

[5]Clinical Biostatistics Unit, Hospital Ramón y Cajal (IRYCIS), CIBERESP Madrid, Spain, Institute of Metabolism and Systems Research, University of Birmingham, Birmingham, UK

Abstract. Overfitting is a common problem in the development of predictive models. It leads to an optimistic estimation of apparent model performance. Internal validation using bootstrapping techniques allows one to quantify the optimism of a predictive model and provide a more realistic estimate of its performance measures. Our objective is to build an easy-to-use command, bsvalidation, aimed to perform a bootstrap internal validation of a logistic regression model.

Keywords: st0644, bsvalidation, bootstrap, internal validation, predictive model, performance, logistic, logit

1 Introduction

A multivariable predictive model is a mathematical equation that relates multiple predictors for a particular individual to the probability of future occurrence of an outcome (Royston et al. 2009). Overfitting is a common problem in the development of these models, and it usually yields an overly optimistic model performance (Steyerberg 2009). In this context, internal validation is essential to provide a more realistic estimate of model ability to predict the risk of the outcome in a new subject. Several solutions have been proposed to correct for this optimism (sample splitting, cross-validation, and its variants leave-one-out cross-validation or leave-pair-out cross-validation). Among these strategies, bootstrapping emerges as a popular strategy to correct for optimistic estimates of the apparent performance.

The transparent reporting of a multivariable prediction model for an individual prognosis or diagnosis (TRIPOD) statement is an evidence-based guide of recommendations to standardize reporting of predictive models. The TRIPOD statement recommends bootstrapping techniques to carry out internal model validation and shrinkage methods to adjust overfitted models (Moons et al. 2015; Collins et al. 2015).

Our objective is to develop a new command, bsvalidation, to perform internal model validation using bootstrapping techniques that is executable as a postestimation command after the logistic or logit command. Stata has implemented postestimation commands to assess the apparent performance of the model. First, it has implemented the lroc postestimation command to assess model discrimination. It also has implemented estat gof to assess model calibration with a Hosmer–Lemeshow test. To the best of our knowledge, there is no user-defined internal validation command implemented in Stata to date such as the one we are presenting.

2 Methods

bsvalidation needs to be executed after logistic or logit. The command allows one to estimate different performance measures in terms of overall model fit performance (that is, how close our predictions are to the actual outcome, related to the amount of variability that is explained); discrimination (that is, how well the model distinguishes between those with and without the outcome); and calibration (that is, how well predictions and observations agree). These measures can be observed in table 1.

Table 1. Performance measures

Item	Measure	Characteristics
Overall performance (Steyerberg et al. 2010)	$\text{Brier}_{\text{scaled}}$	Range: [0, 100] High values indicate predictions are closer to the actual outcome.
Discrimination (Riley et al. 2019)	C-statistic	Range: [0.5, 1] High values indicate better discrimination.
Calibration (Riley et al. 2019)	E:O ratio	Ideal value: 1 E:O < 1 indicates the model underestimates for the total number of events. E:O > 1 indicates the model overestimates for the total number of events.
	Calibration-in-the-large (CITL)	Ideal value: 0 CITL < 0 indicates the predictions are systematically too high. CITL > 0 indicates the predictions are systemati cally too low.
	Calibration slope	Ideal value: 1

Slope < 1 indicates the predictions are too ex treme and the model is overfit.
Slope > 1 indicates the predictions are not varied enough and the model is underfit.

NOTE: $\text{Brier}_{\text{scaled}} = 1 - \text{Brier}_{\text{score}} / \text{Brier}_{\text{max}}$

After the user has fit a logistic predictive model in the original sample using either the logit or logistic command, the validation command goes over the following algorithm:

1. It determines its apparent performance in the original sample (table 1).
2. It draws a bootstrap sample with replacement from the original sample.
3. It builds a new prediction model (bootstrap model) replicating the same modeling strategy used in the model that is being validated, and it determines its apparent performance in the bootstrap sample (bootstrap performance). If the original model is prespecified (that is, fit without variable selection), bsvalidation uses original model specification without any strategy for variable selection.
4. It applies the bootstrap model to the original sample to determine its performance (test performance).
5. It calculates the model's optimism as the difference between the bootstrap perfor mance and the test performance.
6. It repeats steps 2–5 a userdefined number of times to obtain a stable averaged estimate of the optimism.
7. Finally, it subtracts the averaged optimism estimate obtained in step 6 from the initial apparent performance estimated in step 1 to obtain the optimismcorrected performance estimate.

Also, uniform shrinkage parameters—heuristic (Van Houwelingen and Le Cessie 1990) and bootstrap (Harrell 2015)—are estimated, and the coefficient of the model can be shrunk.

Our bsvalidation command also generates a calibration plot. Calibration is as sessed using a lowess smoother function of predicted and observed risks for the overall sample. It also presents pairs of predicted and observed risks for groups defined by the user according to quantiles of predicted risk.

3 The bsvalidation command

3.1 Syntax

The syntax for bsvalidation is

bsvalidation *varlist* *, options*

If the final model was prespecified, *varlist* will be empty. If the model was built using selection methods (backward, forward, or stepwise), those predictors previously assessed but excluded from the final model during the selection process should be included in *varlist*.

3.2 Options

reps(*#*) specifies the number of bootstrap samples. The default is 50 samples. If you are using Stata/IC, up to 800 bootstrap samples are supported. See help limits.

rseed(*#*) sets the random-number seed. This option can be used to obtain reproducible results. rseed(*#*) is equivalent to typing set seed *#* prior to calling bsvalidation.

adjust(*string*) displays the final model after applying a uniform shrinkage factor to the regression coefficients. *string* is one of the following:

heuristic—uniform heuristic shrinkage parameter from Van Houwelingen and Le Cessie (1990).

bootstrap—uniform bootstrap shrinkage parameter from Steyerberg (2009).

pr(*#*) and pe(*#*) specify the significance level threshold for variables to be removed from or entered into the model, respectively.

pr(*#*) is backward elimination. Variables with *p*-value ≥pr() are eligible to be removed.

pe(*#*) is forward selection. Variables with *p*-value <pe() are eligible to be entered.

pr(*#*) and pe(*#*) indicate backward stepwise.

When a predictor-selection approach is considered, a backward elimination strategy is generally preferred (Harrell 2015).

Furthermore, bsvalidation displays the times each variable is selected in the final model after applying the same selection strategy for each bootstrap sample. Other variable-selection strategies such as lasso (least absolute shrinkage and selection operator) are not included in bsvalidation. See help lasso.

models displays the final model for each bootstrap sample. If the final model is pre-specified, this option does not apply.

eform causes the coefficient table to be displayed in exponentiated form: for each coefficient, exp(b) rather than b is displayed. Standard errors and confidence intervals are also transformed.

graph produces a calibration plot of observed against expected probabilities. Calibration is plotted in groups across the risk spectrum. Confidence intervals for the groupings are displayed as well as a lowess smoother.

This allows one to assess the calibration at the individual level. If adjust() is considered, then the calibration plot will be adjusted.

Other user commands to generate calibration plots can be consulted (Ensor, Snell, and Martin 2018).

group(#) specifies the number of percentiles to divide the predicted risks into. The default is to divide the predicted risks into 10 equally sized groups.

min(#) allows one to fix a lower bound of observed and expected probabilities to be plotted.

If min() is higher than the minimum probability predicted by the model, it is auto matically rounded to the nearest first decimal to minimum.

max(#) allows one to fix an upper bound of observed and expected probabilities to be plotted.

If max() is lower than the maximum probability predicted by the model, it is auto matically rounded to the nearest first decimal to maximum.

3.3 Stored results

bsvalidation stores the following in e():

Scalars	
e(N)	number of observations
e(k)	number of parameters in the final model
e(df m)	degrees of freedom
e(k max)	number of parameters in the maximum model
e(boot)	number of bootstrap samples
e(brier)	Brier score for model overall performance
e(opt brier)	optimism of the Brier score
e(cstat)	C-statistic for model discrimination
e(opt cstat)	optimism of the C-statistic
e(eo ratio)	ratio between expected and observed events
for model calibration	
e(citl)	calibration-in-the-large for model calibration
e(slope)	calibration slope for model calibration
e(heur shrink)	uniform heuristic shrinkage
e(boot shrink)	uniform
bootstrap shrinkage Macros	
e(cmd)	bsvalidation
e(depvar)	dependent variable
e(all vars)	independent variables in the maximum model
e(sel vars)	independent variables in the final model
e(model)	regression model
e(properties)	b V
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance
matrix of the estimators Functions	
e(sample)	marks estimation sample

4 Examples

We illustrate the use of bsvalidation with a predictive model developed to estimate the risk of low birthweight using the dataset lbw.dta from Hosmer, Lemeshow, and Sturdivant (2013).

In the first example, the command bsvalidation runs a bootstrap internal valida- tion of a prespecified model.

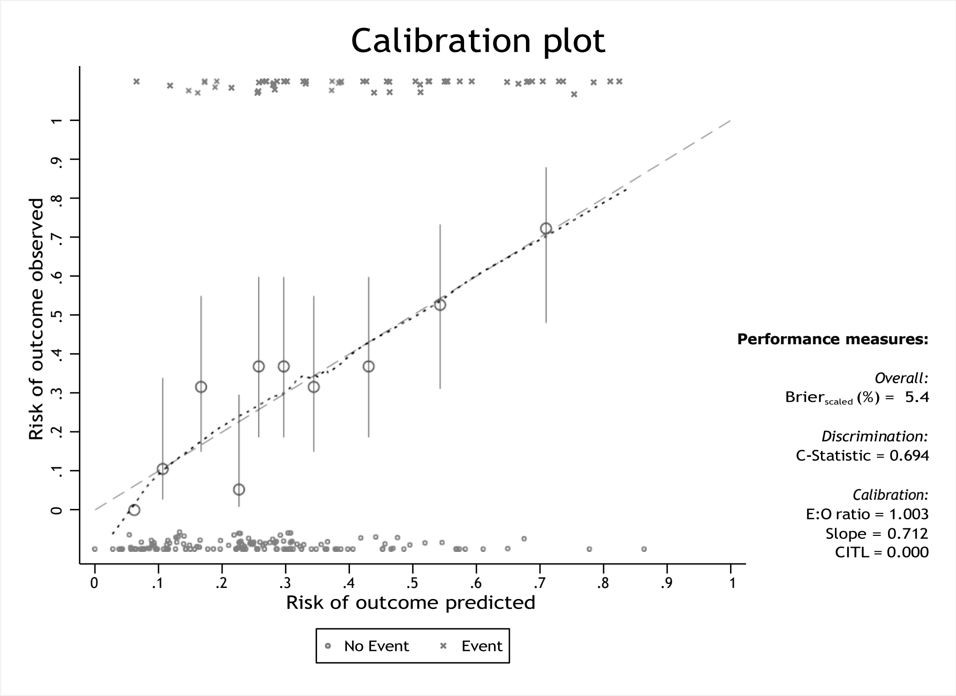

Figure 1. Calibration plot

In this first example, we fit a prespecified logistic model to predict the risk of low birthweight (defined as birthweight lower than 2,500 grams), using the mother's age (age), weight at last menstrual period (lwt), race (race), smoking status during pregnancy (smoke), previous history of premature labor (ptl), hypertension (ht), and uterine irritability (ui) as predictors. The bsvalidation output shows all apparent performance statistics (for example, C-statistic = 0.746). These performance measures are then adjusted for the estimated optimism, which is calculated from 50 (the default number) bootstrap samples (for example, C-statistic = 0.694). Additionally, by using the graph option, we visualize a calibration plot of observed against expected risks of low birthweight in groups defined by deciles of predicted risk, along with a smooth fit ted line. Further, it shows scatterplots with the distribution of events (x symbol) and nonevents (hollow circle symbol) along the *x* axis.

In the second example, bsvalidation performs a bootstrap internal validation of a model that was previously built using a backward-selection strategy with significance level ($p = 0.1$). After the backward-selection strategy, the predictors age and ptl were dropped. The model coefficients are finally adjusted by the bootstrap-estimated uniform shrinkage factor or coefficient.

In the second example, the model is built using a backward-selection strategy in the original data. The predictors selected in the process are lwt, race, smoke, ht, and ui (logistic command). Other candidate predictors (age and ptl) initially assessed, but excluded during the selection process, are added in the *varlist* of the bsvalidation command to replicate the same modeling strategy used during the development of the original model. The output shows both apparent and optimism-adjusted performance measures. Additionally, because the backward-selection strategy is replicated in each bootstrap sample, the output also shows the number of times each predictor is selected in the final model (that is, lwt was included in 75 out of 100 bootstrap models). Finally, the coefficients of the final model are adjusted by bootstrap-based uniform shrinkage to correct overfitting. Thus, coefficients are multiplied by 0.712.

5 Conclusion

bsvalidation is a useful command to run bootstrap internal validation of predictive logistic regression models. It makes this internal validation method more accessible to researchers promoting a more complete and better report of predictive models according to TRIPOD guidelines.

6 Limitations

Although bsvalidation helps standardize the internal validation process, a disadvantage of bootstrap validation is that it allows validation only of models built following fixed or automated modeling strategies (that is, without dynamic modeling strategies or stepwise modeling strategies). Other important steps during the modeling process, such as collapsing factor variables, assessing nonlinearities, or testing for interaction terms, cannot be handled by bsvalidation. The command does not handle other shrinkage methods, such as the least absolute shrinkage and selection operator (Tibshirani 1996), and cannot handle missing values.

7 Future works

In the future, we will work to solve some of the previously mentioned limitations, and we will evolve the command to validate other regression models commonly used in biomedical research, such as Cox regression.

8 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 21-2
. net install st0644                    (to install program files, if available)
. net get st0644                        (to install ancillary files, if available)
```

9 References

Collins, G. S., J. B. Reitsma, D. G. Altman, and K. G. M. Moons. 2015. Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): The TRIPOD statement. *British Medical Journal* 350: g7594. https://doi.org/10.1136/bmj.g7594.

Ensor, J., K. I. E. Snell, and E. C. Martin. 2018. pmcalplot: Stata module to produce calibration plot of prediction model performance. Statistical Software Components S458486, Department of Economics, Boston College. https://EconPapers.repec.org/RePEc:boc:bocode:s458486.

Harrell, F. E., Jr. 2015. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. 2nd ed. Cham, Switzerland: Springer.

Hosmer, D. W., Jr., S. Lemeshow, and R. X. Sturdivant. 2013. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley.

Moons, K. G. M., D. G. Altman, J. B. Reitsma, J. P. A. Ioannidis, P. Macaskill, E. W. Steyerberg, A. J. Vickers, D. F. Ransohoff, and G. S. Collins. 2015. Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): Explanation and elaboration. *Annals of Internal Medicine* 162: W1–W73. https://doi.org/10.7326/M14-0698.

Riley, R. D. A., D. van der Windt, P. Croft, and K. G. M. Moons. 2019. *Prognosis Research in Health Care: Concepts, Methods, and Impact*. New York: Oxford University Press. https://doi.org/10.1093/med/9780198796619.001.0001.

Royston, P., K. G. M. Moons, D. G. Altman, and Y. Vergouwe. 2009. Prognosis and prognostic research: Developing a prognostic model. *British Medical Journal* 338: b604. https://doi.org/10.1136/bmj.b604.

Steyerberg, E. W. 2009. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. Cham, Switzerland: Springer.

Steyerberg, E. W., A. J. Vickers, N. R. Cook, T. Gerds, M. Gonen, N. Obuchowski, M. J. Pencina, and M. W. Kattan. 2010. Assessing the performance of prediction models: A framework for traditional and novel measures. *Epidemiology* 21: 128–138. https://doi.org/10.1097/EDE.0b013e3181c30fb2.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* 58: 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Van Houwelingen, J. C., and S. Le Cessie. 1990. Predictive value of statistical models. *Statistics in Medicine* 9: 1303–1325. https://doi.org/10.1002/sim.4780091109.

About the authors

Borja M. Fernandez-Felix is a PhD student in the Department of Epidemiology and

Public Health at the Autonomous University of Madrid. He works as a biostatistician at the Clinical Biostatistics Unit, Ram´on y Cajal University Hospital, Instituto Ram´on y Cajal de Investi gaci´on Sanitaria (IRYCIS), in Madrid, Spain.

Esther Garc´ıa Esquinas works at the Department of Preventive Medicine and Public Health of the Autonomous University of Madrid.

Alfonso Muriel works as a biostatistician at the Clinical Biostatistics Unit, Instituto Ram´on y Cajal de Investigaci´on Sanitaria (IRYCIS) in Madrid, Spain.

Ana Royuela is the head of the Biostatistics Unit at Puerta de Hierro Hospital in Majadahonda, Madrid, Spain.

Javier Zamora is the head of the Clinical Biostatistics Unit at Ram´on y Cajal University Hospital, and he works as a professor of biostatistics in maternal and perinatal health at the University of Birmingham, UK.

All coauthors are members of the CIBER of Epidemiology and Public Health (CIBERESP).