A calibration hierarchy for risk models was defined: from utopia to empirical data

Ben VAN CALSTER[a,b], Daan NIEBOER[b], Yvonne VERGOUWE[b], Bavo DE COCK[a], Michael J. PENCINA[c,d], Ewout W. STEYERBERG[b]

[a] KU Leuven, Department of Development and Regeneration, Leuven, Belgium
[b] Department of Public Health, Erasmus MC, Rotterdam, the Netherlands
[c] Duke Clinical Research Institute, Duke University, Durham (NC), USA
[d] Department of Biostatistics and Bioinformatics, Duke University, Durham (NC), USA

Accepted for publication in Journal of Clinical Epidemiology – post-refereeing version, not proofread/edited by the journal

Corresponding author:
Ben Van Calster
KU Leuven
Department of Development and Regeneration
Herestraat 49 box 7003
3000 Leuven
Belgium
T 003216377788
E ben.vancalster@med.kuleuven.be

Abstract

Objective. Calibrated risk models are vital for valid decision support. We define four levels of calibration and describe implications for model development and external validation of predictions.

Study Design and Setting. We present results based on simulated datasets.

Results. A common definition of calibration is "having an event rate of R% among patients with a predicted risk of R%", which we refer to as 'moderate calibration'. Weaker forms of calibration only require the average predicted risk (mean calibration) or the average prediction effects (weak calibration) to be correct. 'Strong calibration' requires that the event rate equals the predicted risk for every covariate pattern. This implies that the model is fully correct for the validation setting. We argue that this is unrealistic: the model type may be incorrect, at model development the linear predictor is only asymptotically unbiased, and all nonlinear and interaction effects should be correctly modeled. In addition, we prove that moderate calibration guarantees non-harmful decision-making. Finally, results indicate that a flexible assessment of calibration in small validation datasets is problematic.

Conclusion. Strong calibration is desirable for individualized decision support, but unrealistic and counter-productive by stimulating the development of overly complex models. Model development and external validation should focus on moderate calibration.

Keywords

Calibration; Decision curve analysis; External validation; Loess; Overfitting; Risk prediction models

What is new?

Key Findings

- We defined a new hierarchy of four increasingly strict levels of calibration, referred to as mean, weak, moderate, and strong calibration.
- Strong calibration of risk prediction models implies that the model was correct given the included predictors. We argue that this is unrealistic.
- Moderate calibration of risk prediction models guarantees that decision-making based on the model does not lead to harm.
- The reliability of calibration assessments, most notably of flexible calibration plots, is highly dependent on the sample size of the validation dataset.

What this adds to what is known

- The evaluation of risk prediction models in terms of calibration is often described as a crucial aspect of model validation. However, a systematic framework for levels of calibration for risk prediction models was lacking, and the characteristics of different levels were unclear.
- We find that strong calibration of risk models occurs only in utopia, while moderate calibration does not and is sufficient from a decision-analytic point of view.

Implications of the findings

- At model development, researchers should not aim to develop the correct model. This is practically impossible and may backfire by developing overly complex models that overfit the available data. Our focus should be on achieving moderate calibration, for example by controlling model complexity and shrinking predictions towards the average.
- At model validation, sufficiently large datasets should be available to reliably assess moderate calibration. We suggest a minimum of 200 events and 200 non-events.

1. Introduction

There is increasing attention for the use of risk prediction models to support medical decision-making. Discriminatory performance is commonly the main focus in the evaluation of performance, while calibration commonly receives less attention [1]. A prediction model is calibrated in a given population if the estimated risks are reliable, i.e. correspond to observed proportions of the event. Commonly, calibration is defined as 'for patients with an estimated risk of R%, on average R out of 100 should indeed suffer from the disease or event of interest'. Calibration is a pivotal aspect of model performance [2-4]: "For informing patients and medical decision making, calibration is the primary requirement" [2], "If the model is not […] well calibrated, it must be regarded as not having been validated. […] To evaluate classification performance […] is inappropriate" [4].

Recently, a stronger definition of calibration has been emphasized in contrast to the definition of calibration given above [4, 5]. Models are considered strongly calibrated if estimated risks are accurate for each and every covariate pattern. In this paper, we aim to define different levels of calibration and describe implications for model development, external validation of predictions, and clinical decision-making. We focus on predicting binary endpoints (event vs no event), and assume that a logistic regression model is developed in a derivation sample with performance assessment in a validation sample. We expand on examples used in recent work by Vach [5].

2. Assessing calibration at external validation

2.1. Methods

We assume that the predicted risks are obtained from a previously developed prediction model for outcome Y (1=event, 0=non-event), e.g. based on logistic regression analysis. The model provides a constant (model intercept) and a set of effects (model coefficients). The linear combination of the coefficients with the covariate values in a validation set defines the linear predictor L: $L = a + b_1 \times x_1 + b_2 \times x_2 + \cdots + b_i \times x_i$, where

aa is the model intercept, b_1 to b_i a set of regression coefficients, and x_1 to x_i the predictor values that define the patient's covariate pattern. Ideally the observed proportions in the validation set equal the predicted risks, resulting in a diagonal line in the plot (e.g. Figure 1A).

Calibration of risk predictions is often visualized in calibration plots. These plots show the observed proportion of events associated with a model's predicted risk [6]. The observed proportions per level of estimated risk cannot be directly observed. We consider their estimation in three ways. First, the observed event rates can be obtained after categorizing the predicted risks, for example using deciles. This is commonly done for the Hosmer-Lemeshow test [7]. Then, for each group the average predicted risk can be plotted versus the observed event rate to obtain a calibration curve, see [8] for an example. Second, the logistic recalibration framework can be used [9, 10], where a logistic model is used for the outcome Y as a function of L. More technically, the logistic recalibration framework fits the following model: $logit(Y) = a + b_L \times L$. Using the results of this model to estimate the observed proportions results in a logistic calibration curve. If $b_L = 1$ and $a = 0$, the logistic calibration curve coincides with the diagonal line. The coefficient b_L is the calibration slope that gives an

indication of the level of overfitting ($b_L < 1$) or underfitting ($b_L > 1$). Overfitting is most common, reflected in a linear predictor that gives too extreme values for the validation data: high risks are overestimated and low risks are underestimated. The intercept a can be interpreted when fixing b_L at 1, i.e. $a|b_L=1$. This calibration intercept is obtained by fitting the model $logit(Y) = a + offset(L)$, where the slope b_L is set to unity by entering L as an offset term to the model. Predicted risks are on average underestimated if $a|b_L=1 > 0$, and overestimated if $a|b_L=1 < 0$.

Third, a flexible, non-linear, calibration curve can be considered using the model $logit(Y) = a + f(L)$. Here, f may be a continuous function of the linear predictor L, such as loess or spline transformations [6, 11]. We used a loess smoother in this paper .

2.2. Illustration: Examples 1-5

For illustration, we consider five simulated examples, as previously presented [5]. We randomly generate four independent predictor variables x_1 to x_4. These predictor variables are ordinal with three categories (-1, 0, and 1) that each have 33% prevalence in order to visualize calibration by covariate pattern. Let outcome Y be generated by an underlying logistic regression model with the true linear predictor $L = 0.21 \times x_1 + 0.37 \times x_2 + 0.64 \times x_3 + 0.77 \times x_4$, hence the intercept equals 0, and x_1 to x_4 are assumed to have linear main effects [5]. In simulations, the true probability of event $P(Y = 1|L)$ is calculated using the linear predictor and the observed outcome Y is generated as a Bernoulli variable from $P(Y = 1|L)$. Then, we develop a model to predict Y based on x_1 to x_4, which means that we are using the correct model formulation. Finally, we validate on a new dataset that is generated with the exact same procedure. We repeat this process four times

to illustrate the influence of random variability: (1) using 100 simulated patients for development and 100 for validation, (2) using 100 for development and 10,000 for validation, (3) using 10,000 for development and 100 for validation, (4) using 10,000 for development and 10,000 for validation, and (5) using 10 million for development and 10 million for validation. Estimation of model coefficients was unstable when the development data contains only 100 patients (Table 1), despite having more than 10 events per variable (4 parameters, 44 non-events, 56 events). Risk estimates were overfitted, as evidenced by the calibration slope and calibration curves at validation (Example 2 in Table 1, and Figure 1B) as well as the calibration curves. Further, the calibration intercept was negative which is suggestive of general overestimation. Enlarging the development dataset alleviated the problems (Example 4 in Table 1, Figure 1D). Similar issues emerge when using a small validation dataset (Figures 1A and 1C. Calibration results for Examples 1 and 3 are unstable, in particular the flexible calibration curve. With 10 million patients the coefficient estimates and calibration results were perfect (Table 1, Figure 1E).

3. A hierarchy of risk calibration

In the following we propose a hierarchy of increasingly strict levels of calibration, starting with the basic level of 'calibration-in-the-large', followed by weak, moderate, and strong calibration (Table 2). Higher levels of calibration require stronger conditions and imply that the conditions of lower levels are satisfied.

3.1. Level 1: mean calibration (Calibration-in-the-large)

The most basal type of calibration simply evaluates whether the observed event rate in the data equals the average estimated risk as a measure of the estimated event rate. There was

some miscalibration in mean risk for Examples 1-2 where the development sample was small, even though the correct underlying model formulation was estimated. For larger development datasets the disagreement between observed and predicted risks disappeared (Examples 3-5). The logistic recalibration framework can be used to investigate calibration-in-the-large by estimating the calibration intercept $a|b_L=1$, and if desired by testing the null hypothesis that $a|b_L=1=0$ using a likelihood ratio test with 1 degree of freedom [9, 10]. Fixing b_L at 1 implies that we keep the relative risks fixed. Calibration-in-the-large is insufficient as the sole criterion. For example, it is satisfied when the estimated risk for each patient would equal the true event rate.

3.2. Level 2: weak calibration

The next level is to have weak calibration of predictions, defined as a calibration intercept ($a|b_L=1$) of 0 and a calibration slope of 1. As explained above, these values indicate that there is no over- or underfitting and no systematic over- or underestimation of predicted risks. Deviations from the ideal calibration values can readily be evaluated using confidence intervals, or tested by a Cox recalibration test, a likelihood ratio test with 2 degrees of freedom for the null hypothesis that $a|b_L=1=0$ and $b_L=1$ [10]. In Examples 1-2, the prediction model suffers from overfitting and overestimation due to the small development sample (Figure 1A-B). We consider logistic calibration to be only a weak form of calibration for two reasons. First, this approach lacks flexibility because the calibration curve is summarized by only two parameters through a logistic model. Second, this level of calibration is by definition achieved on the dataset on which the prediction model was developed if standard estimation methods are used, such as maximum likelihood for logistic regression models. For example, it generally does not matter whether and how nonlinear effects of continuous predictors were accounted for or whether important interaction terms were

included. Nevertheless, this approach can be useful at external validation because the calibration intercept and slope provide a general and concise summary of potential problems with risk calibration. In addition, when dealing with relatively small validation samples a simple calibration assessment may be preferred over more flexible alternatives (see Example 1) [12].

3.3. Level 3: moderate calibration

Moderate calibration refers to the common definition of calibration: a risk model is moderately calibrated if, among patients with the same predicted risk, the observed event rate equals the predicted risk [4, 13, 14]. For example, among patients with an estimated risk of 25%, 1 in 4 should have the disease. Moderate calibration can be investigated using flexible calibration curves or using categorizations of predicted risk, preferably with the addition of confidence limits [15]. These approaches are more flexible and can reveal miscalibration that is not picked up by the logistic calibration framework. For example, strong interactions or nonlinearities may lead to miscalibration in the development sample although weak calibration is perfect [11].

A recent measure that bears resemblance to Harrell's Emax and to the Brier score [6] is the estimated calibration index (ECI) [16]. This measure builds upon a flexible calibration analysis by computing the average squared difference between predicted risk and the observed proportion, and transforming the result to obtain a value between 0 and 1. If the flexible calibration curve is perfect, ECI equals 0. Because ECI summarizes a flexible calibration curve into a single number, it was mainly suggested as a measure to easily compare calibration between competing models [16].

3.4. Missing non-linearity or interaction: examples 6-7

We simulate patient outcomes with the same procedure as above. For Example 6 we assume that the outcome is generated by a logistic regression formula with the following true linear predictor: $0.21 \times x_1 + 0.37 \times x_2 + 0.64 \times x_3 + 1.2 \times log(x_4)$, where x_1 to x_3 are ordinal variables as defined above and x_4 is a continuous variable with a lognormal distribution (e.g. a biomarker). We develop a model to predict Y based on x_1 to x_4 (without log-transformation) using 10,000 simulated patients, with calibration curves for the development data (Figure 2A). By definition, the logistic calibration curve is perfect. The flexible calibration is not, because the effect of x_4 is not appropriately addressed. Thus, the model is weakly but not moderately calibrated on the development data.

For Example 7 we assume that the outcome is generated by a logistic regression formula with the following true linear predictor:

$0.21 \times x_1 + 0.37 \times x_2 + 0.64 \times x_3 + 0.77 \times x_4 - 1 \times x_2 \times x_4$, so with a strong interaction effect between x_2 and x_4. Variables x_1 to x_4 are ordinal variables as defined above. We develop a model to predict Y based on x_1 to x_4, without the interaction, using 10,000 simulated patients, with calibration curves for the development sample (Figure 2B). Again, only the flexible calibration curve reveals that calibration is problematic.

3.5. Level 4: strong calibration

The most stringent definition of calibration requires predicted risks to correspond to observed event rates for each and every covariate pattern [4, 5]. This definition of strong calibration disentangles different covariate patterns that may be associated with the same predicted risk.

Requiring strong calibration is sensible from a clinical point of view [5]. If a model is moderately but not strongly calibrated, we may provide biased risk estimates depending on an individual patient's covariate values. Note that calibration is always assessed relative to the predictors in the model. Thus, if a model is strongly calibrated it is still possible that patients with the same covariate pattern have different observed event rates after stratification for another variable that is not included as a predictor. This would not invalidate the strong calibration of the model.

3.6. Model misspecification: Examples 8-9

Figure 3 presents three calibration plots to illustrate strong calibration. Each plot contains a logistic calibration curve, a flexible calibration curve, and results per covariate pattern. Figure 3A represents the validation for Example 5 (model developed on 10 million patients and validated on another 10 million patients). This model exhibits perfect strong calibration: both calibration curves and all covariate patterns coincide with the diagonal. Example 8 (Figure 3B) uses the same validation data as Example 5, but now a model with the following linear predictor is validated: $0.40 \times x_1 + 0.31 \times x_2 + 0.68 \times x_3 + 0.63 \times x_4$. Two coefficients are overestimated and two underestimated relative to the true values (Table 1). This model exhibits perfect moderate calibration but lacks strong calibration because results per covariate pattern are scattered around the diagonal. Example 9 (Figure 3C) uses the same validation data to validate a model with the following linear predictor:

$0.40 \times x_1 + 0.04 \times x_2 - 0.06 \times x_3 + 1.62 \times x_4$. The calibration curves show that this model is not weakly calibrated due to overfitting. Results per covariate pattern are scattered around the calibration curves. (This last plot is based on the third example in [5].)

3.7. Can strong calibration be assessed?

Ideally, we would check for strong calibration as in Figure 3 by calculating the observed event rate for every covariate pattern observed in the data, and construct a plot where every covariate pattern is represented by its estimated risk vs observed event rate. In practice this approach is hardly ever feasible because of limited sample size and/or the presence of continuous predictors: in such situations there may be as many patients as there are distinct covariate patterns. The impact of sample size is illustrated in Figure 4 for Examples 1-5. Figure 4A-B present the validation of the same model on a small or large dataset. In the small dataset there were 100 patients for 81 covariate patterns, hence many covariate patterns contained a single patient. These cells had an observed event rate of 0 or 1. For the model developed on 10,000 patients and validated on a different but equally large sample (Example 4, Figure 4D), the covariate patterns still did not lie on the diagonal line. Only when the development and validation datasets were extremely large (Example 5, Figure 4E, 10M patients), results were near perfect.

An approach that may be considered as an attempt to assess calibration beyond moderate calibration involves the categorization of patients based on (combinations of) predictor values rather than on predicted risk. For different subgroups defined by values of one or more predictors, the average predicted risk may be compared with the observed event rate [17]. This is an insightful exercise, but the 'curse of dimensionality' is still looming: increasingly detailed categorizations will inevitably lead to small subgroups and hence unreliable results.

4. Calibration, decision-making, and clinical utility

Strong calibration implies that an accurate risk estimate is obtained for every covariate pattern. Hence a strongly calibrated model allows the communication of accurate risks to every individual patient. In contrast, a moderately calibrated model allows the communication of a reliable average risk for patients with the same estimated risk: among patients with an estimated risk of 70% on average 70 out of 100 have the event, although there may exist relevant subgroups with different covariate patterns and different event rates.

Previous work has shown that miscalibration decreases clinical utility compared to moderate calibration [18]. Clinical utility was evaluated with the Net Benefit measure. This is a simple and increasingly adopted summary measure that appropriately takes the relative clinical consequences of true and false positives into account [19, 20]. Net Benefit accounts for the different consequences of true and false positives through the risk threshold, i.e. the risk at which one is indifferent about classifying a patient as having the event (and providing treatment) or not. The odds of the risk threshold is the harm-to-benefit ratio, i.e. the ratio of the harm of a false positive and the benefit of a true positive classification [21]. For example, a threshold of 0.2 implies a 1:4 odds, suggesting that one true positive is worth four false positives. Net Benefit at threshold *t* equals $(N_{TP} - odds(t) \times N_{FP})/N$, with N_{TP} the number of true positives, N_{FP} the number of false positives, and N the sample size. Plotting Net Benefit for different choices for the risk threshold yields a decision curve. The model's Net Benefit need to be compared with the Net Benefit of two default strategies in which either all patients are classified as having the event (treat all) or as not having the event (treat none). If the model's Net Benefit is lower than that of a default strategy, then using the model to support decision-making can be considered as clinically harmful.

In terms of decision-making the question is whether strong calibration improves clinical utility over a moderately calibrated model? Let us consider Example 9 (Figure 3C). The model presented is not calibrated in a weak sense. We derived decision curves to assess the clinical utility of (1) the original miscalibrated model, (2) a recalibrated model to ensure moderate calibration, and (3) the strongly calibrated true model (Figure 5). Moderate recalibration was obtained by replacing original risk estimates with those from a flexible recalibration analysis based on 10 million patients. The original model has Net Benefit below that of treat all or treat none for a subset of risk thresholds (Figure 5). The moderately recalibrated version does not have that problem anymore: Net Benefit is now at least as high as that of the default strategies. Nevertheless, the decision curve for the true model is the best one. This suggests that moderate calibration prevents the model from becoming harmful [18] but that strong calibration may further enhance the model's clinical utility. More specifically, we prove (see Appendix) that clinical utility in terms of Net Benefit will not be lower than the default strategies that either classify everyone as having the event or as not having the event.

5. Strong calibration: realistic or utopic?

In line with Vach's work [5], we find that moderate calibration does not imply that the prediction model is 'valid' in a strong sense. In principle, we should aim for strong calibration since this makes predictions accurate at the individual patient's level as well as at the group level leading tobetter decisions on average. However, we consider four problems in empirical analyses. First, strong calibration requires that the model form (e.g. a generalized linear model such as logistic regression) is correct. Such model formulations are often sensible approximations, but in reality, the true model is unknown, may not have a generalized linear form, or may not even exist [22]. Second, using maximum likelihood estimation, as is done

for logistic regression and Cox proportional hazards regression for time to event outcomes, yields only asymptotically unbiased estimates of individual coefficients [23]. However, even when the model coefficients are estimated without bias there is a tendency for overfitting: their combination in the linear predictor leads to too extreme risk prediction, resulting in a calibration slope smaller than one when the model is internally or externally validated. Ensuring sufficient events per variable (EPV) is important to control the amount of overfitting [24]. Also, for predictive purposes, estimators that impose shrinkage to reduce variance at the expense of inducing bias to the individual coefficients have been shown to be superior to standard maximum likelihood [25-27]. Such shrinkage methods include uniform shrinkage, ridge regression, LASSO, elastic net, and the Garotte [2, 27].

Third, strong calibration not only requires correctly estimated regression coefficients for the main effects of model predictors, but also requires fully correct modeling of all nonlinear and non-additive effects. For a limited number of categorized predictors we can imagine fitting a 'full model' including all first- and higher-order interaction terms, but the use of continuous predictors makes finding the correct model unrealistic. We stress that calibration is assessed relative to the predictors included in the model. Failing to include relevant predictors is therefore not an argument for stating that strong calibration is unrealistic, although the fact that many relevant covariates may exist outside of the model should make us modest in claiming that we can define a "true model".

Fourth, measurement error of the predictors is in practice often ignored although it is known that this is a common phenomenon that can bias regression coefficients [28]. Moreover, measurement error may not be transportable across various settings [29]. This means that at

least part of the measurement error is systematic, and further thwarts the concept of the existence of a true model.

In sum, aiming for strong calibration requires the assumption that the model formulation is fully correct, and that unbiased model coefficients as well as unbiased linear predictors are obtained. Such a true model can only be identified in an infinitely large dataset: in utopia. As Vach [30] writes: "the idea to identify the "true" model by statistical means is just a great wish which cannot be fulfilled" (p202).

6. Moderate calibration: a pragmatic guarantee for non-harmful decision-making

Focusing on finding at least moderately calibrated models has several advantages. First, it is a realistic goal in epidemiologic research, where empirical data sets are often of relatively limited size, and the signal to noise ratio is unfavorable [31]. Second, moderate calibration guarantees that decision-making based on the model is not clinically harmful. Conversely, it is an important observation that calibration in a weak sense may still result in harmful decision-making [18]. Third, simpler models can be aimed for, although it is still advisable to have sufficient events per variable (EPV) to appropriately investigate important deviations from linearity for continuous predictors, and to assess whether some (preferably prespecified) interaction terms are indispensable. We emphasize that continuous predictors should not be categorized in a naïve attempt to achieve strong calibration. The disadvantages of categorization are too numerous to summarize here [32]. Examining nonlinear and interaction terms may help to reduce the deviation from strong calibration and obtain better individual risk estimates, but at the risk of overfitting. While still providing sensible risk predictions,

simple models that are for example moderately but not strongly calibrated often have many practical advantages such as transparency or ease of use [33].

7. A link with model updating

In model updating, we adapt a model that has poor performance at external validation [34]. Basic updating approaches include, in order of complexity, intercept adjustment, recalibration, and refitting [34, 35]. There are parallels between updating methods and levels of calibration. Intercept adjustment updates the linear predictor L to $a + L$. This will only address calibration-in-the-large, but does not guarantee weak calibration. A more complex updating method involves logistic recalibration, where the linear predictor L is updated to $a + b_L \times L$. This method addresses lack of weak calibration, but does not guarantee moderate calibration unless all coefficients were biased by the same rate. In model refitting, model coefficients for the predictors are re-estimated. Refitting should lead to moderate calibration, although this may also require reassessment of nonlinear effects.

8. Sample size at external validation

Our simulations have shown the impact of sample size on how reliably calibration can be assessed (Figures 1 and 4). It is clear that observed calibration curves will easily deviate from the diagonal line even when the model is moderately or strongly calibrated. We have extended our simulations by validating the correct (and hence strongly calibrated) model from Examples 1-5 on datasets with sample size between 100 and 1000. Given an overall event rate of 50%, the number of events and non-events varied from 50 to 500, yet the observed number of events per simulated dataset may vary due to random variation. Figure A.1 shows flexible

calibration curves for 50 randomly drawn validation datasets per sample size, Figure A.2 shows boxplots of the calibration slope and ECI for 200 randomly drawn validation datasets. Given that we know that the model is correct, ECI can be used to quantify the variability of the flexible calibration curve around the true line. The results suggest that the flexible calibration curve is more variable and hence requires more data than the calibration slope for a stable assessment. This is in line with related work on this topic [11, 12, 36]. Confidence intervals are useful to properly interpret the obtained results.

9. Statistical testing for calibration

We mainly focused on conceptual issues in assessing calibration of predictions from statistical models. We did not consider statistical testing in detail, and in this area the assessment of statistical power needs further study. In previous simulations, the Hosmer-Lemeshow test showed such poor performance that it may not be recommended for routine use [7, 37]. In practice, indications of uncertainty such as confidence intervals are far more important than a statistical test.

10. Conclusion and recommendations

We conclude that strong calibration, although desirable for individual risk communication, is unrealistic in empirical medical research. Focusing on obtaining prediction models that are calibrated in the moderate sense is a better attainable goal, in line with the most common definition of the notion of 'calibration of predictions'. In support of this view, we proved that moderate calibration guarantees that clinically non-harmful decisions are made based on the model. This guarantee cannot be given for prediction models that are only calibrated in the

weak sense. Based on these findings, we make the following recommendations. When externally validating prediction models, (1) perform a graphical assessment for moderate calibration including pointwise 95% confidence limits, and (2) provide the summary statistics for weak calibration, specifically the calibration slope (b_L) for the overall effect of the predictors and the calibration intercept ($a|b_L=1$). If sample size is limited, flexible calibration curves may become highly unstable and can be omitted [11]. In line with related work, we recommend at least 100 events and 100 non-events to assess the calibration intercept and slope, and at least 200 events and 200 non-events to derive flexible calibration curves [11, 12, 36]. The figures in this paper are based on an adaptation of Harrell's val.prob function in R [6], see Supplementary Material.

At internal validation, e.g. using cross-validation or bootstrapping, we recommend to focus on the calibration slope to provide a shrinkage factor for the estimated risks. The calibration intercept is not relevant because internal validation implies that the model is validated for the same setting, where the mean of predictions matches the mean event rate according to standard statistical estimation methods such as maximum likelihood. When developing or updating prediction models, we recommend to focus on simple models and, to avoid overfitting, to focus more on non-linearity than on interaction terms, but always in balance with the effective sample size. In addition, flexible calibration curves on the development or updating dataset are important to evaluate moderate calibration.

Acknowledgments

We thank Laure Wynants for proofreading the manuscript.

Funding

This study was supported in part by the Research Foundation – Flanders (FWO) (grant G049312N) and by Internal Funds KU Leuven (grant C24/15/037).

Conflict of interest

None.

Table 1. Development and validation results for Examples 1 to 5. Results are shown for a single random draw.

	True model	**Example 1. N_D=100 N_V=100**	**Example 2. N_D=100 N_V=10,000**	**Example 3. N_D=10,000 N_V=100**	**Example 4. N_D=10,000 N_V=10,000**	**Example 5. N_D=10M N_V=10M**
Development results, shown as estimate or estimate (SE)						
C statistic	0.724	0.694		0.718		0.724
Intercept	0	0.24 (0.22)		0.01 (0.02)		0.00 (0.0007)
Coefficient x_1	0.21	-0.12 (0.27)		0.17 (0.03)		0.21 (0.0008)
Coefficient x_2	0.37	0.74 (0.28)		0.38 (0.03)		0.37 (0.0008)
Coefficient x_3	0.64	0.23 (0.26)		0.59 (0.03)		0.64 (0.0009)
Coefficient x_4	0.77	0.59 (0.27)		0.77 (0.03)		0.77 (0.0009)
Validation results, shown as estimate or estimate (SE)						
C statistic	0.724	0.623	0.668	0.673	0.717	0.724
$a \mid b_L = 1$	0	-0.18 (0.21)	-0.28 (0.02)	0.03 (0.21)	-0.04 (0.02)	0.00 (0.0007)
b_L	1	0.71 (0.29)	0.80 (0.03)	0.75 (0.27)	1.00 (0.03)	1.00 (0.0009)
Event rate	0.50	0.49	0.49	0.49	0.49	0.50
Average risk	0.50	0.53	0.55	0.48	0.50	0.50

N_D: development sample size; N_V: validation sample size; 10M: ten million; SE: standard error; $a \mid b_L = 1$: calibration intercept; b_L: calibration slope

Table 2. A hierarchy of calibration levels for risk prediction models.

Level	Definition	Assessment
Mean	Observed event rate equals average risk estimate; "calibration-in-the-large"	* Compare event rate with average predicted risk; * evaluate $a \mid b_L = 1$ (with 1 df test $a \mid b_L = 1 = 0$)
Weak	No systematic over- or underfitting and/or over- or underestimation of risks; "logistic calibration"	Logistic calibration analysis to evaluate $a \mid b_L = 1$ and b_L (with Cox recalibration test: a 2 df test of the null hypothesis that $a \mid b_L = 1 = 0$ and $b_L = 1$)
Moderate	Predicted risks correspond to observed event rates	Calibration plot (e.g. using loess or splines), or analysis by grouped predictions (including Hosmer-Lemeshow test)
Strong	Predicted risks correspond to observed event rates for each and every covariate pattern	Scatter plot of predicted risk and observed event rate per covariate pattern; impossible when continuous predictors are involved

Figure 1. Calibration curves on the validation data for examples 1 to 5, with pointwise 95% confidence limits for flexible curves: (A) trained on 100, validated on 100; (B) trained on 100, validated on 10,000; (C) trained on 10,000, validated on 100; (D) trained and validated on 10,000; (E) trained and validated on 10 million patients.

Figure 2. Calibration plots on the development data for (A) example 6 in which a true nonlinear effect is ignored and (B) example 7 in which a true interaction effect is ignored.

Figure 3. Calibration plots illustrating (A) strong calibration, (B) moderate but not strong calibration, (C) miscalibration.

Figure 4. Calibration curves for examples 1 to 5 including results for individual covariate patterns, for examples 1 to 5: (A) Trained on 100, validated on 100; (B) Trained on 100, validated on 10,000; (C) Trained on 10,000, validated on 100; (D) Trained and validated on 10,000; (E) Trained and validated on 10 million patients.

Figure 5. Decision curves for Example 8 to assess clinical usefulness.

References

[1] Collins GS, de Groot JA, Dutton S, Omar O, Shanyinde M, Tajar A, et al. External validation of multivariable prediction models: a systematic review of methodological conduct and reporting. BMC Med Res Methodol. 2014;14:40.
[2] Steyerberg EW. Clinical Prediction Models. A Practical Approach to Development, Validation, and Updating. New York: Springer-Verlag; 2009.
[3] Kim KI, Simon R. Probabilistic classifiers with high-dimensional data. Biostatistics. 2011;12:399-412.
[4] Pepe MS, Janes H. Methods for evaluating prediction performance of biomarkers and tests. In: Lee MLT, Gail M, Pfeiffer R, Satten G, Cai T, Gandy A, editors. Risk Assessment and Evaluation of Predictions. New York: Springer-Verlag; 2013. p. 107-42.
[5] Vach W. Calibration of clinical prediction rules does not just assess bias. J Clin Epidemiol. 2013;66:1296-301.
[6] Harrell FE, Jr. Regression Modeling Strategies. With Applications to Linear Models, Logistic Regression, and Survival Analysis. New York: Springer-Verlag; 2001.
[7] Hosmer DW, Hosmer T, Le Cessie S, Lemeshow S. A comparison of goodness-of-fit tests for the logistic regression model. Stat Med. 1997;16:965-80.
[8] Ankerst DP, Boeck A, Freedland SJ, Thompson IM, Cronin AM, Roobol MJ, et al. Evaluating the PCPT risk calculator in ten international biopsy cohorts: results from the Prostate Biopsy Collaborative Group. World J Urol. 2012;30:181-7.
[9] Cox DR. Two Further Applications of a Model for Binary Regression. Biometrika. 1958;45:562-5.
[10] Miller ME, Langefeld CD, Tierney WM, Hui SL, McDonald CJ. Validation of probabilistic predictions. Med Decis Making. 1993;13:49-58.
[11] Austin PC, Steyerberg EW. Graphical assessment of internal and external calibration of logistic regression models by using loess smoothers. Stat Med. 2014;33:517-35.
[12] Collins GS, Ogundimu EO, Altman DG. Sample size considerations for the external validation of a multivariable prognostic model: a resampling study. Stat Med. 2015.
[13] Steyerberg EW, Vickers AJ, Cook NR, Gerds T, Gonen M, Obuchowski N, et al. Assessing the performance of prediction models: a framework for traditional and novel measures. Epidemiology. 2010;21:128-38.
[14] Vickers AJ, Cronin AM. Everything you always wanted to know about evaluating prediction models (but were too afraid to ask). Urology. 2010;76:1298-301.
[15] Austin PC, Steyerberg EW. Bootstrap confidence intervals for loess-based calibration curves. Stat Med. 2014;33:2699-700.
[16] Van Hoorde K, Van Huffel S, Timmerman D, Bourne T, Van Calster B. A spline-based tool to assess and visualize the calibration of multiclass risk predictions. J Biomed Inform. 2015;54:283-93.
[17] Decarli A, Calza S, Masala G, Specchia C, Palli D, Gail MH. Gail model for prediction of absolute risk of invasive breast cancer: independent evaluation in the Florence-European Prospective Investigation Into Cancer and Nutrition cohort. J Natl Cancer Inst. 2006;98:1686-93.
[18] Van Calster B, Vickers AJ. Calibration of risk prediction models: impact on decision-analytic performance. Med Decis Making. 2015;35:162-9.
[19] Van Calster B, Vickers AJ, Pencina MJ, Baker SG, Timmerman D, Steyerberg EW. Evaluation of markers and risk prediction models: overview of relationships between NRI and decision-analytic measures. Med Decis Making. 2013;33:490-501.

[20] Vickers AJ, Elkin EB. Decision curve analysis: a novel method for evaluating prediction models. Med Decis Making. 2006;26:565-74.
[21] Pauker SG, Kassirer JP. Therapeutic decision making: a cost-benefit analysis. N Engl J Med. 1975;293:229-34.
[22] Chatfield C. Model Uncertainty, Data Mining and Statistical Inference. Journal of the Royal Statistical Society Series A (Statistics in Society). 1995;158:419-66.
[23] Nemes S, Jonasson JM, Genell A, Steineck G. Bias in odds ratios by logistic regression modelling and sample size. BMC Med Res Methodol. 2009;9:56.
[24] Austin PC, Steyerberg EW. The number of subjects per variable required in linear regression analyses. J Clin Epidemiol. 2015;68:627-36.
[25] Steyerberg EW, Eijkemans MJC, Habbema JDF. Application of Shrinkage Techniques in Logistic Regression Analysis: A Case Study. Statistica Neerlandica. 2001;55:76-88.
[26] Van Houwelingen JC. Shrinkage and Penalized Likelihood as Methods to Improve Predictive Accuracy. Statistica Neerlandica. 2001;55:17-34.
[27] Ambler G, Seaman S, Omar RZ. An evaluation of penalised survival methods for developing prognostic models with rare events. Stat Med. 2012;31:1150-61.
[28] Fuller WA. Measurement Error Models. New York: John Wiley & Sons; 1987.
[29] Carroll RJ, Delaigle A, Hall P. Nonparametric Prediction in Measurement Error Models. J Am Stat Assoc. 2009;104:993-1014.
[30] Vach W. Regression Models as a Tool in Medical Research. Boca Raton: Chapman and Hall/CRC; 2013.
[31] Ennis M, Hinton G, Naylor D, Revow M, Tibshirani R. A comparison of statistical learning methods on the Gusto database. Stat Med. 1998;17:2501-8.
[32] Royston P, Altman DG, Sauerbrei W. Dichotomizing continuous predictors in multiple regression: a bad idea. Stat Med. 2006;25:127-41.
[33] Marewski JN, Gigerenzer G. Heuristic decision making in medicine. Dialogues Clin Neurosci. 2012;14:77-89.
[34] Steyerberg EW, Borsboom GJ, van Houwelingen HC, Eijkemans MJ, Habbema JD. Validation and updating of predictive logistic regression models: a study on sample size and shrinkage. Stat Med. 2004;23:2567-86.
[35] Van Hoorde K, Vergouwe Y, Timmerman D, Van Huffel S, Steyerberg EW, Van Calster B. Simple dichotomous updating methods improved the validity of polytomous prediction models. J Clin Epidemiol. 2013;66:1158-65.
[36] Vergouwe Y, Steyerberg EW, Eijkemans MJ, Habbema JD. Substantial effective sample sizes were required for external validation studies of predictive logistic regression models. J Clin Epidemiol. 2005;58:475-83.
[37] Hosmer DW, Hjort NL. Goodness-of-fit processes for logistic regression: simulation results. Stat Med. 2002;21:2723-38.

Appendix

Moderate calibration guaranties non-harmful decisions in terms of Net Benefit and decision curve analysis: proof

Notation:

$\pi(\mathbf{x})$: true probability of event given covariate vector $\mathbf{x}$, where $\mathbf{x} = (x_1, \ldots, x_i)$

$\hat{\pi}(\mathbf{x})$: estimated probability of event given covariate vector $\mathbf{x}$ given by a prediction model.

Y: outcome (1=event, 0=non-event)

$f(\mathbf{x})$: probability density of covariate vector $\mathbf{x}$

P_1 and P_0: event rate $P(Y = 1)$ and 1 minus event rate $P(Y = 0)$

For a moderately calibrated model the following property holds, among patients with an estimated risk of R% on average R out of 100 of these patients have the event.

Mathematically this can be translated as:

$$E[Y|\hat{\pi}(\mathbf{x}) = r] = \frac{\int_{\mathbf{x}:\hat{\pi}(\mathbf{x})=r} \pi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}}{\int_{\mathbf{x}:\hat{\pi}(\mathbf{x})=r} f(\mathbf{x}) d\mathbf{x}} = r$$

The expected net benefit at threshold *t* is given by:

$$E[\mathrm{NB}_t] = E\left[\frac{N_{TP} - \mathrm{odds}(t) N_{FP}}{N}\right] = P(Y = 1, \hat{\pi}(\mathbf{x}) \geq t) - \mathrm{odds}(t) P(Y = 0, \hat{\pi}(\mathbf{x}) \geq t).$$

To ensure that a moderately calibrated model is not harmful, the expected net benefit should be larger than (a) the net benefit of treating all patients and (b) the net benefit of treating no patients.

$$P(Y = 1, \hat{\pi}(\mathbf{x}) \geq t) = \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} \pi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

$$= \int_t^1 \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})=r} \pi(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} dr$$

$$= \int_t^1 r \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})=r} f(\mathbf{x}) d\mathbf{x} dr$$

$$\geq t \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} f(\mathbf{x}) d\mathbf{x}.$$

$$P(Y = 0, \hat{\pi}(\mathbf{x}) \geq t) = \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} (1 - \pi(\mathbf{x})) f(\mathbf{x}) d\mathbf{x}$$

$$= \int_t^1 \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})=r} (1 - \pi(\mathbf{x})) f(\mathbf{x}) d\mathbf{x} dr$$

$$= \int_t^1 (1 - r) \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})=r} f(\mathbf{x}) d\mathbf{x} dr$$

$$\leq (1 - t) \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} f(\mathbf{x}) d\mathbf{x}.$$

From above we get:

$$\begin{aligned} E[\mathrm{NB}_t] &= P(Y = 1, \hat{\pi}(\mathbf{x}) \geq t) - \mathrm{odds}(t) P(Y = 0, \hat{\pi}(\mathbf{x}) < t) \\ &\geq t \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} f(\mathbf{x}) d\mathbf{x} - \frac{t}{1-t}(1 - t) \int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} f(\mathbf{x}) d\mathbf{x} = 0 \\ &= 0 \end{aligned}$$

So with moderate calibration using the model is better than treating no patients. Remains to show that for t below the event rate (P_1) using the model gives a higher expected net benefit than treating all patients.

$$\begin{aligned} E[NB_t] - E\left[NB_{\text{treat all},t}\right] &= P(Y=1,\hat{\pi}(\mathbf{x}) \geq t) - \frac{t}{1-t}P(Y=0,\hat{\pi}(\mathbf{x}) \geq t) - \\ &\quad \left(P_1 - \frac{t}{1-t}P_0\right) \\ &= (P(Y=1,\hat{\pi}(\mathbf{x}) \geq t) - P_1) - \\ &\quad \frac{t}{1-t}(P(Y=0,\hat{\pi}(\mathbf{x}) \geq t) - P_0) \end{aligned}$$

To achieve a better NB using the model compared to treating all patients we then need

$$(P(Y=1,\hat{\pi}(\mathbf{x}) \geq t) - P_1) \geq \frac{t}{1-t}(P(Y=0,\hat{\pi}(\mathbf{x}) \geq t) - P_0)$$

whenever $t < P_1$.

$$\begin{aligned} \frac{t}{1-t}(P(Y=0,\hat{\pi}(\mathbf{x}) \geq t) - P_0) &\leq \frac{t}{1-t}\left[(1-t)\int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} f(\mathbf{x})d\mathbf{x} - (1-P_1)\right] \\ &= t\int_{\mathbf{x}:\hat{\pi}(\mathbf{x})\geq t} f(\mathbf{x})d\mathbf{x} - \frac{(1-P_1)}{1-t}t \\ &\leq P(Y=1,\hat{\pi}(\mathbf{x}) \geq t) - P_1\frac{(1-P_1)}{1-t} \\ &\leq P(Y=1,\hat{\pi}(\mathbf{x}) \geq t) - P_1 \end{aligned}$$

Figure A1. Flexible calibration curves to simulate external validation of a strongly calibrated model in 50 randomly drawn datasets of size 100 to 1000 (true event rate 50%).

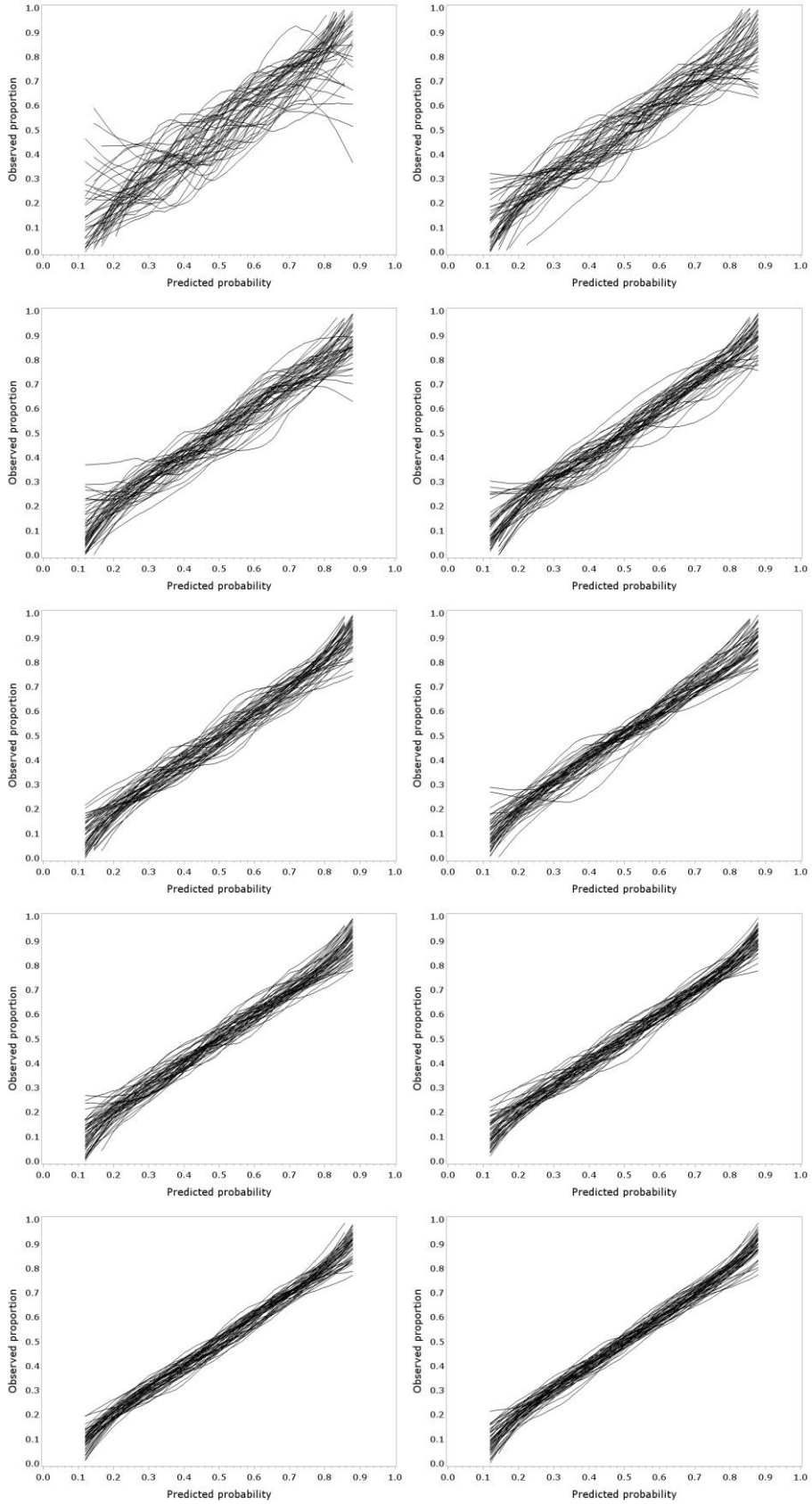
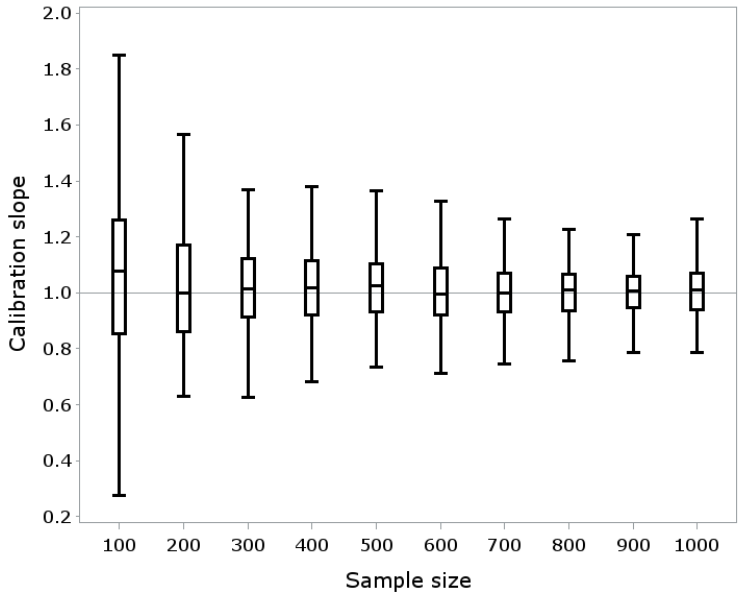
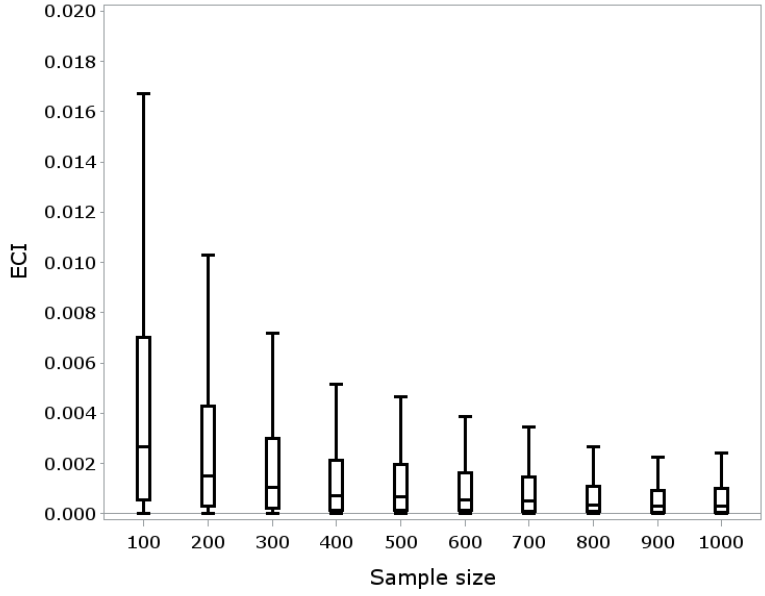


Figure A2. Box plots of the calibration slope (A) and estimated calibration index (ECI) (B) to simulate external validation of a strongly calibrated model in in 200 randomly drawn datasets of size 100 to 1000 (true event rate 50%).

(A)



(B)



SUPPLEMENTARY MATERIAL

R function val.prob.ci.2: description

The R function `val.prob.ci.2`, developed in R version 3.2.1 (R Core Team, 2015), is available as online material for this paper. The function is an adaptation of `val.prob` from Frank Harrell's `rms` package, https://cran.r-project.org/web/packages/rms/rms.pdf. The key feature of `val.prob.ci.2` is the generation of logistic and flexible calibration curves and related statistics. It is used as follows:

```
val.prob.ci.2 <- function(p, y, logit, group,
          weights=rep(1,length(y)), normwt=F, pl=T,
          smooth=c("loess","rcs",F), CL.smooth="fill", CL.BT=F,
          nr.knots=5, logistic.cal=F, xlab="Predicted probability",
          ylab="Observed proportion", xlim=c(-0.02,1), ylim=c(-
          0.15,1), m, g, cuts, emax.lim=c(0,1),
          legendloc=c(0.50,0.27), statloc=c(0,.85), dostats=T,
          roundstats=2, riskdist="predicted", cex=0.75, cex.leg=0.75,
          connect.group=F, connect.smooth=T, g.group=4, evaluate=100,
          nmin=0, d0lab="0", d1lab="1", cex.d01=0.7, dist.label=0.04,
          line.bins=-.05, dist.label2=.03, cutoff, las=1,
          length.seg=1, y.intersp=1, col.ideal=”grey”,
          lwd.ideal=1, ...)
```

New options, or options that are adapted from the original `val.prob` are (default values underlined):

`logistic.cal T` or `TRUE` plots the logistic calibration curve, `F` or `FALSE` suppresses this curve.

smooth "loess" generates a flexible calibration curve based on loess, "rcs" generates a calibration curves based on restricted cubic splines, F or FALSE suppresses the flexible curve. We recommend to use loess unless N is large, for example N>5000.

CL.smooth "fill" shows pointwise 95% confidence limits for the flexible calibration curve with a gray area between the lower and upper limits, T or TRUE shows pointwise 95% confidence limits for the flexible calibration curve with dashed lines, F or FALSE suppresses the confidence limits. To save a plot with filled confidence limits as .eps, we suggest to use cairo_ps: first run cairo_ps(filename="C:/figure.eps"), then run the val.prob.ci.2 function to obtain the figure, and then run dev.off() to avoid that the figure is overwritten by subsequent figures.

CL.BT T or TRUE uses confidence limits based on 2000 bootstrap samples, F or FALSE uses closed form confidence limits.

nr.knots specifies the number of knots (3, 4, or 5) for rcs-based calibration curve. The default as well as the highest allowed value is 5. In case the specified number of knots leads to estimation problems, then the number of knots is automatically reduced to the closest value without estimation problems.

dostats specifies whether and which performance measures are shown in the figure. T or TRUE shows the "abc" of model performance (Steyerberg et al, 2011): calibration intercept, calibration slope, and c statistic. F or FALSE suppresses the presentation of statistics in the figure. A c() list of specific stats shows the specified stats. The key stats which are also mentioned in this paper are "C (ROC)" for the c statistic, "Intercept" for the calibration intercept, "Slope" for the calibration slope, and "ECI" for the

	estimated calibration index (Van Hoorde et al, 2015). The full list of possible statistics is taken from val.prob (Harrell, 2001) and augmented with the estimated calibration index and the scaled Brier score: "Dxy", "C (ROC)", "R2", "D", "D:Chi-sq", "D:p", "U", "U:Chi-sq", "U:p", "Q", "Brier", "Intercept", "Slope", "Emax", "Brier scaled", "Eavg", "ECI". These statistics are always returned by the function.
roundstats	specifies the number of decimals to which statistics are rounded when shown in the figure. Default is 2.
cex, cex.leg	controls the font size of the statistics (cex) or plot legend (cex.leg). Default is 0.75.
d0lab, d1lab	controls the labels for events and non-events (i.e. outcome y) for the histograms. Defaults are d1lab="1" for events and d0lab="0" for non-events.
cex.d01	controls the size of the labels for events and non-events. Default is 0.7.
dist.label	controls the horizontal position of the labels for events and non-events. Default is 0.04.
dist.label2	controls the vertical distance between the labels for events and non-events. Default is 0.03.
line.bins	controls the horizontal (y-axis) position of the histograms. Default is -0.05.
cutoff	puts an arrow at the specified risk cut-off(s). Default is none.
las	controls whether y-axis values are shown horizontally (1) or vertically (0).
length.seq	controls the length of the histogram lines. Default is 1.
y.intersp	character interspacing for vertical line distances of the legend. Default is 1.
col.ideal	controls the color of the ideal line. Default is "grey".
lwd.ideal	controls the line width of the ideal line. Default is 1.

Further options relevant to this paper:

`cuts` provides a list `c()` of actual cutpoints for categorization, in case calibration should be shown for categorizations of predicted risk.

`m` provides average size of categories, in case calibration should be shown for categorizations of predicted risk.

`g` provides number of equally large (quantile) categories, in case calibration should be shown for categorizations of predicted risk. E.g. `g=10` groups predicted risk using deciles as is commonly done.

`group` this provides a stratification variable for which to stratify the calibration analysis. This can be used to get average predicted risks and observed proportions for subgroups of patients based on the stratification variable. `T` or `TRUE` performs this analysis for the whole dataset. A plot with loess calibration curves per subgroup can be obtained with `plot(val.prob.ci.2(p,y,group=z))`.

`g.group` If the stratification variable is continuous, `g.group` defines the number of quantile groups (default is quartiles, i.e. `4`).

`riskdist` indicates whether histograms are based on predicted risk (`"predicted"`) or on calibrated risk (`"calibrated"`). Set to `FALSE` to omit the histograms.

References

R Core Team (2015). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: www.R-project.org/.

Steyerberg EW, Van Calster B, Pencina MJ. Performance measures for prediction models and markers: evaluation of predictions and classifications. *Rev Esp Cardiol* 2011;64:788-94.

Van Hoorde K, Van Huffel S, Timmerman D, Bourne T, Van Calster B. A spline-based tool to assess and visualize the calibration of multiclass risk predictions. *J Biomed Inform*. 2015;54:283-93.

Harrell FE, Jr. *Regression Modeling Strategies. With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York: Springer-Verlag; 2001.

R function val.prob.ci.2: code

```
#-------------------------#
#                         #
#    val.prob.ci.2        #  Adjusted version of Harrell's val.prob
#                         #
#-------------------------#

January 2016

# WHEN USING THIS FUNCTION, PLEASE CITE:
# Van Calster B, Nieboer D, Vergouwe Y, De Cock B, Pencina MJ, Steyerberg
# EW. A calibration hierarchy for risk models was defined: from utopia to
# empirical data. Journal of Clinical Epidemiology, in press (2016).

# Some years ago, Yvonne Vergouwe and Ewout Steyerberg adapted val.prob
# into val.prob.ci:
  # - Scaled Brier score by relating to max for average calibrated Null
  #   model
  # - Risk distribution according to outcome
  # - 0 and 1 to indicate outcome label; set with d1lab="..", d0lab=".."
  # - Labels: y axis: "Observed Frequency"; Triangle: "Grouped
  #   observations"
  # - Confidence intervals around triangles
  # - A cut-off can be plotted; set x coordinate

# In December 2015, Bavo De Cock, Daan Nieboer, and Ben Van Calster adapted
# this to val.prob.ci.2:
  # - Flexible calibration curves can be obtained using loess (default) or
  #   restricted cubic splines, with pointwise 95% confidence intervals
  # - Loess: CI can be obtained in closed form or using bootstrapping
  #   (CL.BT=T will do bootstrapping with 2000 bootstrap samples, however
  #   this will take a while)
  # - RCS: 3 to 5 knots can be used
  #     -> the knot locations will be estimated using default quantiles of
```

```
  #        x (by rcspline.eval, see help rcspline.plot and rcspline.eval)
  #     -> if estimation problems occur at the specified number of knots
  #        (nr.knots, default is 5), the analysis is repeated with
  #         nr.knots-1 until the problem has disappeared
  # - You can now adjust the plot through use of normal plot commands
  #   (cex.axis etc), and the size of the legend now has to be specified in
  #   cex.leg
  # - Label y-axis: "Observed proportion"
  # - Stats: added the Estimated Calibration Index (ECI), a statistical
  #   measure to quantify lack of calibration (Van Hoorde et al., 2015)
  # - Stats to be shown in the plot: by default we shown the calibration
  #   intercept (calibration-in-the-large), calibration slope and c-
  #   statistic. Alternatively, the user can select the statistics of
  #   choice (e.g. dostats=c("C (ROC)","R2") or dostats=c(2,3).
  # - Vectors p, y and logit no longer have to be sorted

  # Example:
  # # simulated data
  # x1 <- as.matrix(rnorm(500))
  # x2 <- as.matrix(rnorm(500))
  # x3 <- as.matrix(rnorm(500))
  # lp0=0.5*x1+1.2*x2+0.75*x3
  # p0true=exp(lp0)/(1+exp(lp0))
  # y <-rbinom(500,1,p0true)
  # data.0 <- data.frame(y,x1,x2,x3)
  #
  # # fit logistic model
  # fit.lrm <- lrm(y~x1+x2+x3,data=data.0)
  # pred.lrm <- predict(fit.lrm,type="fitted")
  #
  # # calibration plot for development data
  # val.prob.ci.2(pred.lrm,y)

val.prob.ci.2 <- function(p, y, logit, group, weights = rep(1, length(y)), normwt = F, pl = T,
                          smooth = c("loess","rcs",F), CL.smooth="fill",CL.BT=F,
                          nr.knots=5,logistic.cal = F, xlab = "Predicted probability", ylab =
                            "Observed proportion", xlim = c(-0.02, 1),ylim = c(-0.15,1), m, g, cuts, emax.lim = c(0, 1),
                          legendloc =  c(0.50 , 0.27), statloc = c(0,.85),dostats=T,roundstats=2,
                          riskdist = "predicted", cex=0.75,cex.leg = 0.75, connect.group =
                            F, connect.smooth = T, g.group = 4, evaluate = 100, nmin = 0, d0lab="0", d1lab="1", cex.d01=0.7,
```

```
                          dist.label=0.04, line.bins=-.05, dist.label2=.03, cutoff, las=1, length.seg=1,
                          y.intersp=1,col.ideal="grey",lwd.ideal=1,...)
{

  if(smooth[1]==F){smooth <- "F"}
  smooth <- match.arg(smooth)
  if(missing(p))
    p <- 1/(1 + exp( - logit))
  else logit <- log(p/(1 - p))
  if(length(p) != length(y))
    stop("lengths of p or logit and y do not agree")
  names(p) <- names(y) <- names(logit) <- NULL
  if(!missing(group)) {
    if(length(group) == 1 && is.logical(group) && group)
      group <- rep("", length(y))
    if(!is.factor(group))
      group <- if(is.logical(group) || is.character(group))
        as.factor(group) else cut2(group, g =
                                   g.group)
    names(group) <- NULL
    nma <- !(is.na(p + y + weights) | is.na(group))
    ng <- length(levels(group))
  }
  else {
    nma <- !is.na(p + y + weights)
    ng <- 0
  }
  logit <- logit[nma]
  y <- y[nma]
  p <- p[nma]
  if(ng > 0) {
    group <- group[nma]
    weights <- weights[nma]
    return(val.probg(p, y, group, evaluate, weights, normwt, nmin)
    )
  }
  require(rms)
  # Sort vector with probabilities
  y     <- y[order(p)]
  logit <- logit[order(p)]
```

```
p     <- p[order(p)]


if(length(p)>5000 & CL.smooth==T){warning("Number of observations > 5000, RCS is recommended.",immediate. = T)}
if(length(p)>1000 & CL.BT==T){warning("Number of observations is > 1000, this could take a while...",immediate. = T)}


if(length(unique(p)) == 1) {
  #22Sep94
  P <- mean(y)
  Intc <- log(P/(1 - P))
  n <- length(y)
  D <- -1/n
  L01 <- -2 * sum(y * logit - log(1 + exp(logit)), na.rm = T)
  L.cal <- -2 * sum(y * Intc - log(1 + exp(Intc)), na.rm = T)
  U.chisq <- L01 - L.cal
  U.p <- 1 - pchisq(U.chisq, 1)
  U <- (U.chisq - 1)/n
  Q <- D - U

  stats <- c(0, 0.5, 0, D, 0, 1, U, U.chisq, U.p, Q, mean((y - p[
    1])^2), Intc, 0, rep(abs(p[1] - P), 2))
  names(stats) <- c("Dxy", "C (ROC)", "R2", "D", "D:Chi-sq",
                    "D:p", "U", "U:Chi-sq", "U:p", "Q", "Brier",
                    "Intercept", "Slope", "Emax", "Eavg", "ECI")
  return(stats)
}
i <- !is.infinite(logit)
nm <- sum(!i)
if(nm > 0)
  warning(paste(nm, "observations deleted from logistic calibration due to probs. of 0 or 1"))
i.2 <- i
f.or <- lrm(y[i]~logit[i])
f <- lrm.fit(logit[i], y[i])
f2<-   lrm.fit(offset=logit[i], y=y[i])
stats <- f$stats
n <- stats["Obs"]
predprob <- seq(emax.lim[1], emax.lim[2], by = 0.0005)
lt <- f$coef[1] + f$coef[2] * log(predprob/(1 - predprob))
calp <- 1/(1 + exp( - lt))
```

```
emax <- max(abs(predprob - calp))
if (pl) {
  plot(0.5, 0.5, xlim = xlim, ylim = ylim, type = "n", xlab = xlab,
       ylab = ylab, las=las,...)
  clip(0,1,0,1)
  abline(0, 1, lty = 1,col=col.ideal,lwd=lwd.ideal)
  do.call("clip", as.list(par()$usr))


  lt <- 1
  lw.d <- lwd.ideal
  leg <- "Ideal"
  marks <- -1
  if (logistic.cal) {
    lt <- c(lt, 1)
    lw.d <- c(lw.d,1)
    leg <- c(leg, "Logistic calibration")
    marks <- c(marks, -1)
  }
  if (smooth=="loess") {
    #Sm <- lowess(p,y,iter=0)
    Sm <- loess(y~p,degree=2)
    Sm <- data.frame(Sm$x,Sm$fitted); Sm.01 <- Sm

    if (connect.smooth==T & CL.smooth!="fill") {
      clip(0,1,0,1)
      lines(Sm, lty = 1,lwd=2)
      do.call("clip", as.list(par()$usr))
      lt <- c(lt, 1)
      lw.d <- c(lw.d,2)
      marks <- c(marks, -1)
    }else if(connect.smooth==F & CL.smooth!="fill"){
      clip(0,1,0,1)
      points(Sm)
      do.call("clip", as.list(par()$usr))
      lt <- c(lt, 0)
      lw.d <- c(lw.d,1)
      marks <- c(marks, 1)
    }
    if(CL.smooth==T | CL.smooth=="fill"){
```

```
to.pred <- seq(min(p),max(p),length=200)
if(CL.BT==T){
  BT.samples <- function(y,p,to.pred){
    data.1 <- cbind.data.frame(y,p)

    # REPEAT TO PREVENT BT SAMPLES WITH NA'S
    repeat{
      BT.sample.rows <- sample(1:nrow(data.1),replace=T)
      BT.sample <- data.1[BT.sample.rows,]
      loess(y~p,BT.sample) ->loess.BT
      predict(loess.BT,to.pred,type="fitted") ->pred.loess
      if(!any(is.na(pred.loess))){break}
    }
    return(pred.loess)
  }
  cat("Bootstrap samples are being generated.\n\n\n")

  replicate(2000,BT.samples(y,p,to.pred)) -> res.BT
  apply(res.BT,1,quantile,c(0.025,0.975)) -> CL.BT
  colnames(CL.BT) <- to.pred

  if(CL.smooth=="fill"){
    clip(0,1,0,1)
    polygon(x = c(to.pred, rev(to.pred)), y = c(CL.BT[2,],
                                                rev(CL.BT[1,])),
            col = rgb(177, 177, 177, 177, maxColorValue = 255), border = NA)
    if (connect.smooth==T) {
      lines(Sm, lty = 1,lwd=2)
      lt <- c(lt, 1)
      lw.d <- c(lw.d,2)
      marks <- c(marks, -1)
    }else if(connect.smooth==F){
      points(Sm)
      lt <- c(lt, 0)
      lw.d <- c(lw.d,1)
      marks <- c(marks, 1)
    }
    do.call("clip", as.list(par()$usr))
    leg <- c(leg, "Flexible calibration (Loess)")
  }else{
```

```
    clip(0,1,0,1)
    lines(to.pred,CL.BT[1,],lty=2,lwd=1);clip(0,1,0,1);lines(to.pred,CL.BT[2,],lty=2,lwd=1)
    do.call("clip", as.list(par()$usr))
    leg <- c(leg,"Flexible calibration (Loess)","CL flexible")
    lt <- c(lt,2)
    lw.d <- c(lw.d,1)
    marks <- c(marks,-1)
  }

}else{
  Sm.0 <- loess(y~p,degree=2)
  predict(Sm.0,type="fitted",se=T) -> cl.loess
  clip(0,1,0,1)
  if(CL.smooth=="fill"){
    polygon(x = c(Sm.0$x, rev(Sm.0$x)), y = c(cl.loess$fit+cl.loess$se.fit*1.96,
                                              rev(cl.loess$fit-cl.loess$se.fit*1.96)),
            col = rgb(177, 177, 177, 177, maxColorValue = 255), border = NA)
    if (connect.smooth==T) {
      lines(Sm, lty = 1,lwd=2)
      lt <- c(lt, 1)
      lw.d <- c(lw.d,2)
      marks <- c(marks, -1)
    }else if(connect.smooth==F){
      points(Sm)
      lt <- c(lt, 0)
      lw.d <- c(lw.d,1)
      marks <- c(marks, 1)
    }
    do.call("clip", as.list(par()$usr))
    leg <- c(leg, "Flexible calibration (Loess)")
  }else{
    lines(Sm.0$x,cl.loess$fit+cl.loess$se.fit*1.96,lty=2,lwd=1)
    lines(Sm.0$x,cl.loess$fit-cl.loess$se.fit*1.96,lty=2,lwd=1)
    do.call("clip", as.list(par()$usr))
    leg <- c(leg,"Flexible calibration (Loess)","CL flexible")
    lt <- c(lt,2)
    lw.d <- c(lw.d,1)
    marks <- c(marks,-1)
  }
```

```
        }

      }else{
        leg <- c(leg, "Flexible calibration (Loess)")}
      cal.smooth <- approx(Sm.01, xout = p)$y
      eavg <- mean(abs(p - cal.smooth))
      ECI <- mean((p-cal.smooth)^2)*100
    }
    if(smooth=="rcs"){
      par(lwd=2,bty="n")
      if(!is.numeric(nr.knots)){stop("Nr.knots must be numeric.")}
      if(nr.knots==5){
        tryCatch(rcspline.plot(p,y,model="logistic",nk=5,show="prob", statloc = "none"
                               ,add=T,showknots=F,xrange=c(min(na.omit(p)),max(na.omit(p)))),error=function(e){
                                 warning("The number of knots led to estimation problems, nk will be set to 4.",immediate. =
T)
                                 tryCatch(rcspline.plot(p,y,model="logistic",nk=4,show="prob", statloc = "none"
                                                        ,add=T,showknots=F,xrange=c(min(na.omit(p)),max(na.omit(p)))),error=f
unction(e){
                                                          warning("Nk 4 also led to estimation problems, nk will be set to
3.",immediate.=T)
                                                          rcspline.plot(p,y,model="logistic",nk=3,show="prob", statloc =
"none"
                                                                        ,add=T,showknots=F,xrange=c(min(na.omit(p)),max(na.om
it(p))))
                                                        })
                               })
      }else if(nr.knots==4){
        tryCatch(rcspline.plot(p,y,model="logistic",nk=4,show="prob", statloc = "none"
                               ,add=T,showknots=F,xrange=c(min(na.omit(p)),max(na.omit(p)))),error=function(e){
                                 warning("The number of knots led to estimation problems, nk will be set to 3.",immediate.=T)
                                 rcspline.plot(p,y,model="logistic",nk=3,show="prob", statloc = "none"
                                               ,add=T,showknots=F,xrange=c(min(na.omit(p)),max(na.omit(p))))
                               })
      }else if(nr.knots==3){
        tryCatch(rcspline.plot(p,y,model="logistic",nk=3,show="prob", statloc = "none"
                               ,add=T,showknots=F,xrange=c(min(na.omit(p)),max(na.omit(p)))),
                 error=function(e){
                   stop("Nk = 3 led to estimation problems.")
```

```
            })
    }else{stop(paste("Number of knots = ",nr.knots,sep="", ", only 5 >= nk >=3 is allowed."))}

    par(lwd=1,bty="o")
    leg <- c(leg,"Flexible calibration (RCS)","CL flexible")
    lt <- c(lt,1,2)
    lw.d <- c(lw.d,2,2)
    marks <- c(marks,-1,-1)
  }
  if(!missing(m) | !missing(g) | !missing(cuts)) {
    if(!missing(m))
      q <- cut2(p, m = m, levels.mean = T, digits = 7)
    else if(!missing(g))
      q <- cut2(p, g = g, levels.mean = T, digits = 7)
    else if(!missing(cuts))
      q <- cut2(p, cuts = cuts, levels.mean = T, digits = 7)
    means <- as.single(levels(q))
    prop <- tapply(y, q, function(x)mean(x, na.rm = T))
    points(means, prop, pch = 2, cex=1)
    #18.11.02: CI triangles
    ng  <-tapply(y, q, length)
    og  <-tapply(y, q, sum)
    ob  <-og/ng
    se.ob     <-sqrt(ob*(1-ob)/ng)
    g         <- length(as.single(levels(q)))

    for (i in 1:g) lines(c(means[i], means[i]), c(prop[i],min(1,prop[i]+1.96*se.ob[i])), type="l")
    for (i in 1:g) lines(c(means[i], means[i]), c(prop[i],max(0,prop[i]-1.96*se.ob[i])), type="l")

    if(connect.group) {
      lines(means, prop)
      lt <- c(lt, 1)
      lw.d <- c(lw.d,1)
    }
    else lt <- c(lt, 0)
    lw.d <- c(lw.d,0)
    leg <- c(leg, "Grouped observations")
    marks <- c(marks, 2)
  }
}
```

```
lr <- stats["Model L.R."]
p.lr <- stats["P"]
D <- (lr - 1)/n
L01 <- -2 * sum(y * logit - logb(1 + exp(logit)), na.rm = TRUE)
U.chisq <- L01 - f$deviance[2]
p.U <- 1 - pchisq(U.chisq, 2)
U <- (U.chisq - 2)/n
Q <- D - U
Dxy <- stats["Dxy"]
C <- stats["C"]
R2 <- stats["R2"]
B <- sum((p - y)^2)/n
# ES 15dec08 add Brier scaled
Bmax  <- mean(y) * (1-mean(y))^2 + (1-mean(y)) * mean(y)^2
Bscaled <- 1 - B/Bmax
stats <- c(Dxy, C, R2, D, lr, p.lr, U, U.chisq, p.U, Q, B,
           f2$coef[1], f$coef[2], emax, Bscaled)
names(stats) <- c("Dxy", "C (ROC)", "R2", "D", "D:Chi-sq",
                  "D:p", "U", "U:Chi-sq", "U:p", "Q", "Brier", "Intercept",
                  "Slope", "Emax", "Brier scaled")
if(smooth=="loess")
  stats <- c(stats, c(Eavg = eavg),c(ECI = ECI))

# Cut off definition
if(!missing(cutoff)) {
  arrows(x0=cutoff,y0=.1,x1=cutoff,y1=-0.025,length=.15)
}
if(pl) {
  if(min(p)>plogis(-7) | max(p)<plogis(7)){

    lrm(y[i.2]~qlogis(p[i.2]))-> lrm.fit.1
    if(logistic.cal)  lines(p[i.2],plogis(lrm.fit.1$linear.predictors),lwd=1,lty=1)

  }else{logit <- seq(-7, 7, length = 200)
  prob <- 1/(1 + exp( - logit))
  pred.prob <- f$coef[1] + f$coef[2] * logit
  pred.prob <- 1/(1 + exp( - pred.prob))
  if(logistic.cal) lines(prob, pred.prob, lty = 1,lwd=1)
  }
  #     pc <- rep(" ", length(lt))
```

```
#    pc[lt==0] <- "."
lp <- legendloc
if (!is.logical(lp)) {
  if (!is.list(lp))
    lp <- list(x = lp[1], y = lp[2])
  legend(lp, leg, lty = lt, pch = marks, cex = cex.leg, bty = "n",lwd=lw.d,
         col=c(col.ideal,rep("black",length(lt)-1)),y.intersp = y.intersp)
}
if(!is.logical(statloc)) {
  if(dostats[1]==T){
    stats.2 <- paste('Calibration\n',
                     '...in the large: ', sprintf(paste("%.",roundstats,"f",sep=""), stats["Intercept"]), '\n',
                     '...slope : ', sprintf(paste("%.",roundstats,"f",sep=""), stats["Slope"]), '\n',
                     'Discrimination\n',
                     '...c-statistic : ', sprintf(paste("%.",roundstats,"f",sep=""), stats["C (ROC)"]), sep = '')
    text(statloc[1], statloc[2],stats.2,pos=4,cex=cex)

  }else{
    dostats <- dostats
    leg <- format(names(stats)[dostats]) #constant length
    leg <- paste(leg, ":", format(stats[dostats], digits=roundstats), sep =
                   "")
    if(!is.list(statloc))
      statloc <- list(x = statloc[1], y = statloc[2])
    text(statloc, paste(format(names(stats[dostats])),
                        collapse = "\n"), adj = 0, cex = cex)
    text(statloc$x + (xlim[2]-xlim[1])/3 , statloc$y, paste(
      format(round(stats[dostats], digits=roundstats)), collapse =
        "\n"), adj = 1, cex = cex)
  }
}
if(is.character(riskdist)) {
  if(riskdist == "calibrated") {
    x <- f$coef[1] + f$coef[2] * log(p/(1 - p))
    x <- 1/(1 + exp( - x))
    x[p == 0] <- 0
    x[p == 1] <- 1
  }
  else x <- p
  bins <- seq(0, min(1,max(xlim)), length = 101)
```

```
      x <- x[x >= 0 & x <= 1]
      #08.04.01,yvon: distribution of predicted prob according to outcome
      f0  <-table(cut(x[y==0],bins))
      f1  <-table(cut(x[y==1],bins))
      j0  <-f0 > 0
      j1  <-f1 > 0
      bins0 <-(bins[-101])[j0]
      bins1 <-(bins[-101])[j1]
      f0  <-f0[j0]
      f1  <-f1[j1]
      maxf <-max(f0,f1)
      f0  <-(0.1*f0)/maxf
      f1  <-(0.1*f1)/maxf

      segments(bins1,line.bins,bins1,length.seg*f1+line.bins)
      segments(bins0,line.bins,bins0,length.seg*-f0+line.bins)
      lines(c(min(bins0,bins1)-0.01,max(bins0,bins1)+0.01),c(line.bins,line.bins))
      text(max(bins0,bins1)+dist.label,line.bins+dist.label2,d1lab,cex=cex.d01)
      text(max(bins0,bins1)+dist.label,line.bins-dist.label2,d0lab,cex=cex.d01)

    }
  }
  stats
}
```