Chapter 5
Describing, Resampling, Validating, and Simplifying the Model

5.1 Describing the Fitted Model

5.1.1 Interpreting Effects

Before addressing issues related to describing and interpreting the model and its coefficients, one can never apply too much caution in attempting to interpret results in a causal manner. Regression models are excellent tools for estimating and inferring *associations* between an X and Y given that the "right" variables are in the model. Any ability of a model to provide *causal* inference rests entirely on the faith of the analyst in the experimental design, completeness of the set of variables that are thought to measure confounding and are used for adjustment when the experiment is not randomized, lack of important measurement error, and lastly the goodness of fit of the model.

The first line of attack in interpreting the results of a multivariable analysis is to interpret the model's parameter estimates. For simple linear, additive models, regression coefficients may be readily interpreted. If there are interactions or nonlinear terms in the model, however, simple interpretations are usually impossible. Many programs ignore this problem, routinely printing such meaningless quantities as the effect of increasing age^2 by one day while holding age constant. A meaningful age change needs to be chosen, and connections between mathematically related variables must be taken into account. These problems can be solved by relying on predicted values and differences between predicted values.

Even when the model contains no nonlinear effects, it is difficult to compare regression coefficients across predictors having varying scales. Some analysts like to gauge the relative contributions of different predictors on a common scale by multiplying regression coefficients by the standard deviations of the predictors that pertain to them. This does not make sense for nonnormally distributed predictors (and regression models should not need

F.E. Harrell, Jr., *Regression Modeling Strategies*, Springer Series in Statistics, DOI 10.1007/978-3-319-19425-7_5

to make assumptions about the distributions of predictors). When a predictor is binary (e.g., sex), the standard deviation makes no sense as a scaling factor as the scale would depend on the prevalence of the predictor.[a]

It is more sensible to estimate the change in Y when X_j is changed by an amount that is subject-matter relevant. For binary predictors this is a change from 0 to 1. For many continuous predictors the interquartile range is a reasonable default choice. If the 0.25 and 0.75 quantiles of X_j are g and h, linearity holds, and the estimated coefficient of X_j is b; $b \times (h - g)$ is the effect of increasing X_j by $h - g$ units, which is a span that contains half of the sample values of X_j.

For the more general case of continuous predictors that are monotonically but not linearly related to Y, a useful point summary is the change in $X\beta$ when the variable changes from its 0.25 quantile to its 0.75 quantile. For models for which $\exp(X\beta)$ is meaningful, antilogging the predicted change in $X\beta$ results in quantities such as interquartile-range odds and hazards ratios. When the variable is involved in interactions, these ratios are estimated separately for various levels of the interacting factors. For categorical predictors, ordinary effects are computed by comparing each level of the predictor with a reference level. See Section 10.10 and Chapter 11 for tabular and graphical examples of this approach.

The model can be described using *partial effect plots* by plotting each X against $X\hat{\beta}$ holding other predictors constant. Modified versions of such plots, by nonlinearly rank-transforming the predictor axis, can show the relative importance of a predictor[336].

For an X that interacts with other factors, separate curves are drawn on the same graph, one for each level of the interacting factor.

Nomograms[40, 254, 339, 427] provide excellent graphical depictions of all the variables in the model, in addition to enabling the user to obtain predicted values manually. Nomograms are especially good at helping the user envision interactions. See Section 10.10 and Chapter 11 for examples.

5.1.2 Indexes of Model Performance

5.1.2.1 Error Measures

Care must be taken in the choice of accuracy scores to be used in validation. Indexes can be broken down into three main areas.

Central tendency of prediction errors: These measures include mean absolute differences, mean squared differences, and logarithmic scores. An absolute measure is mean $|Y - \hat{Y}|$. The mean squared error is a commonly used and sensitive measure if there are no outliers. For the special case

[a] The s.d. of a binary variable is, aside from a multiplier of $\frac{n}{n-1}$, equal to $\sqrt{a(1-a)}$, where a is the proportion of ones.

where Y is binary, such a measure is the Brier score, which is a quadratic proper scoring rule that combines calibration and discrimination[b]. The logarithmic proper scoring rules (related to average log-likelihood) is even more sensitive but can be harder to interpret and can be destroyed by a single predicted probability of 0 or 1 that was incorrect.

Discrimination measures: A measure of pure discrimination is a rank correlation of $\hat{Y}$ and Y, including Spearman's ρ, Kendall's τ, and Somers' D_{xy}. When Y is binary, $D_{xy} = 2 \times (c - \frac{1}{2})$ where c is the concordance probability or area under the receiver operating characteristic curve, a linear translation of the Wilcoxon-Mann-Whitney statistic. R^2 is *mostly* a measure of discrimination, and R^2_{adj} is is a good overfitting-corrected measure, if the model is pre-specified. See Section 10.8 for more information about rank-based measures.

Discrimination measures based on variation in $\hat{Y}$: These include the regression sum of squares and the g–Index (see below).

Calibration measures: These assess absolute prediction accuracy. *Calibration–in–the–large* compares the average $\hat{Y}$ with the average Y. A *high-resolution calibration curve* or *calibration–in–the–small* assesses the absolute forecast accuracy of predictions at individual levels of $\hat{Y}$. When the calibration curve is linear, this can be summarized by the calibration slope and intercept. A more general approach uses the *loess* nonparametric smoother to estimate the calibration curve[37]. For any shape of calibration curve, errors can be summarized by quantities such as the maximum absolute calibration error, mean absolute calibration error, and 0.9 quantile of calibration error.

The g-index is a new measure of a model's predictive discrimination based only on $X\hat{\beta} = \hat{Y}$ that applies quite generally. It is based on Gini's mean difference for a variable Z, which is the mean over all possible $i \neq j$ of $|Z_i - Z_j|$. The g-index is an interpretable, robust, and highly efficient measure of variation. For example, when predicting systolic blood pressure, $g = 11$mmHg represents a typical difference in $\hat{Y}$. g is independent of censoring and other complexities. For models in which the anti-log of a difference in $\hat{Y}$ represents meaningful ratios (e.g., odds ratios, hazard ratios, ratio of medians), g_r can be defined as $\exp(g)$. For models in which $\hat{Y}$ can be turned into a probability estimate (e.g., logistic regression), g_p is defined as Gini's mean difference of $\hat{P}$. These g–indexes represent e.g. "typical" odds ratios, and "typical" risk differences. Partial g indexes can also be defined. More details may be found in the documentation for the `R rms` package's `gIndex` function. 5

[b] There are decompositions of the Brier score into discrimination and calibration components.

5.2 The Bootstrap

When one assumes that a random variable Y has a certain population distribution, one can use simulation or analytic derivations to study how a statistical estimator computed from samples from this distribution behaves. For example, when Y has a log-normal distribution, the variance of the sample median for a sample of size n from that distribution can be derived analytically. Alternatively, one can simulate 500 samples of size n from the log-normal distribution, compute the sample median for each sample, and then compute the sample variance of the 500 sample medians. Either case requires knowledge of the population distribution function.

Efron's *bootstrap*[150,177,178] is a general-purpose technique for obtaining estimates of the properties of statistical estimators without making assumptions about the distribution giving rise to the data. Suppose that a random variable Y comes from a cumulative distribution function $F(y) = \text{Prob}\{Y \leq y\}$ and that we have a sample of size n from this unknown distribution, $Y_1, Y_2, \ldots, Y_n$. The basic idea is to repeatedly simulate a sample of size n from F, computing the statistic of interest, and assessing how the statistic behaves over B repetitions. Not having F at our disposal, we can estimate F by the empirical cumulative distribution function

$$F_n(y) = \frac{1}{n} \sum_{i=1}^{n} [Y_i \leq y]. \tag{5.1}$$

F_n corresponds to a density function that places probability $1/n$ at each observed datapoint (k/n if that point were duplicated k times and its value listed only once).

As an example, consider a random sample of size $n = 30$ from a normal distribution with mean 100 and standard deviation 10. Figure 5.1 shows the population and empirical cumulative distribution functions.

Now pretend that $F_n(y)$ is the original population distribution $F(y)$. Sampling from F_n is equivalent to sampling with replacement from the observed data $Y_1, \ldots, Y_n$. For large n, the expected fraction of original datapoints that are selected for each bootstrap sample is $1 - e^{-1} = 0.632$. Some points are selected twice, some three times, a few four times, and so on. We take B samples of size n with replacement, with B chosen so that the summary measure of the individual statistics is nearly as good as taking $B = \infty$. The bootstrap is based on the fact that the distribution of the *observed* differences between a resampled estimate of a parameter of interest and the original estimate of the parameter from the whole sample tells us about the distribution of *unobservable* differences between the original estimate and the unknown population value of the parameter.

As an example, consider the data $(1, 5, 6, 7, 8, 9)$ and suppose that we would like to obtain a 0.80 confidence interval for the population median, as well as an estimate of the population expected value of the sample median (the latter

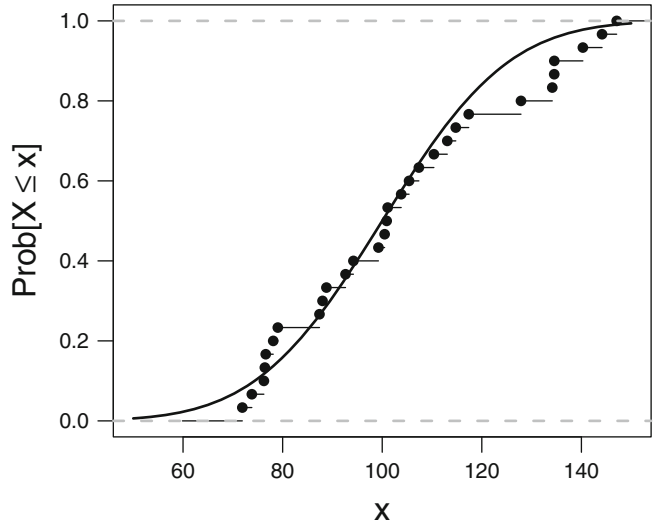

Fig. 5.1 Empirical and population cumulative distribution function

is only used to estimate bias in the sample median). The first 20 bootstrap samples (after sorting data values) and the corresponding sample medians are shown in Table 5.1.

For a given number B of bootstrap samples, our estimates are simply the sample 0.1 and 0.9 quantiles of the sample medians, and the mean of the sample medians. Not knowing how large B should be, we could let B range from, say, 50 to 1000, stopping when we are sure the estimates have converged. In the left plot of Figure 5.2, B varies from 1 to 400 for the mean (10 to 400 for the quantiles). It can be seen that the bootstrap estimate of the population mean of the sample median can be estimated satisfactorily when $B > 50$. For the lower and upper limits of the 0.8 confidence interval for the population median Y, B must be at least 200. For more extreme confidence limits, B must be higher still.

For the final set of 400 sample medians, a histogram (right plot in Figure 5.2) can be used to assess the form of the sampling distribution of the sample median. Here, the distribution is almost normal, although there is a slightly heavy left tail that comes from the data themselves having a heavy left tail. For large samples, sample medians are normally distributed for a wide variety of population distributions. Therefore we could use bootstrapping to estimate the variance of the sample median and then take ± 1.28 standard errors as a 0.80 confidence interval. In other cases (e.g., regression coefficient estimates for certain models), estimates are asymmetrically distributed, and the bootstrap quantiles are better estimates than confidence intervals that are based on a normality assumption. Note that because sample quantiles are more or less restricted to equal one of the values in the sample, the boot-

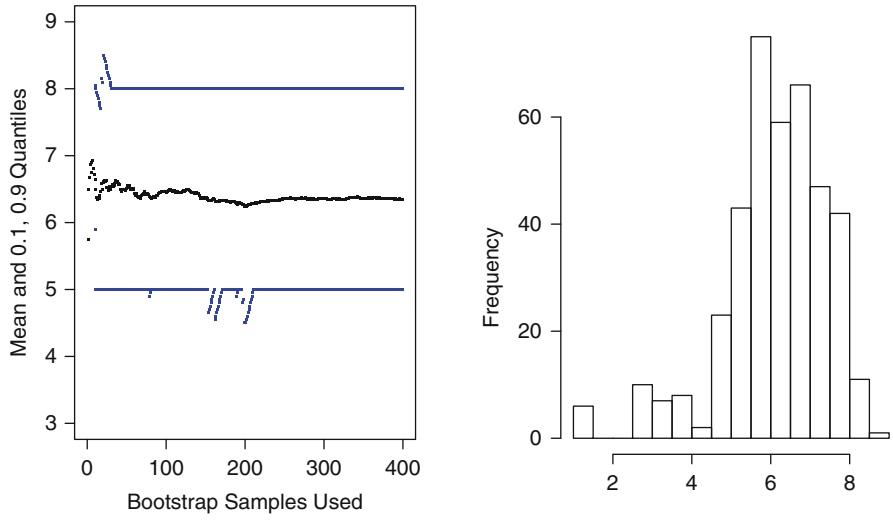

Fig. 5.2 Estimating properties of sample median using the bootstrap

Table 5.1 First 20 bootstrap samples

Bootstrap Sample	Sample Median
1 6 6 7 8 9	6.5
1 5 5 5 6 8	5.0
5 7 8 9 9 9	8.5
7 7 7 8 8 9	7.5
1 5 7 7 9 9	7.0
1 5 6 6 7 8	6.0
7 8 8 8 8 8	8.0
5 5 5 7 9 9	6.0
1 5 5 7 7 9	6.0
1 5 5 7 7 8	6.0
1 1 5 5 7 7	5.0
1 1 5 5 7 8	5.0
1 5 5 7 7 8	6.0
1 5 6 7 8 8	6.5
1 5 6 7 9 9	6.5
6 6 7 7 8 9	7.0
1 5 7 8 8 9	7.5
6 6 8 9 9 9	8.5
1 1 5 5 6 9	5.0
1 6 8 9 9 9	8.5

strap distribution is discrete and can be dependent on a small number of outliers. For this reason, bootstrapping quantiles does not work particularly well for small samples [150, pp. 41–43].

The method just presented for obtaining a nonparametric confidence interval for the population median is called the *bootstrap percentile method*. It is the simplest but not necessarily the best performing bootstrap method. 7

In this text we use the bootstrap primarily for computing statistical estimates that are much different from standard errors and confidence intervals, namely, estimates of model performance.

5.3 Model Validation

5.3.1 Introduction

The surest method to have a model fit the data at hand is to discard much of the data. A p-variable fit to $p + 1$ observations will perfectly predict Y as long as no two observations have the same Y. Such a model will, however, yield predictions that appear almost random with respect to responses on a different dataset. Therefore, unbiased estimates of predictive accuracy are essential.

Model validation is done to ascertain whether predicted values from the model are likely to accurately predict responses on future subjects or subjects not used to develop our model. Three major causes of failure of the 8 model to validate are overfitting, changes in measurement methods/changes in definition of categorical variables, and major changes in subject inclusion criteria.

There are two major modes of model validation, *external* and *internal*. The most stringent external validation involves testing a final model developed in one country or setting on subjects in another country or setting at another time. This validation would test whether the data collection instrument was translated into another language properly, whether cultural differences make earlier findings nonapplicable, and whether secular trends have changed associations or base rates. Testing a finished model on new subjects from the 9 same geographic area but from a different institution as subjects used to fit the model is a less stringent form of external validation. The least stringent form of external validation involves using the first m of n observations for model training and using the remaining $n - m$ observations as a test sample. This is very similar to data-splitting (Section 5.3.3). For details about methods for external validation see the `R` `val.prob` and `val.surv` functions in the `rms` package.

Even though external validation is frequently favored by non-statisticians, it is often problematic. Holding back data from the model-fitting phase re-

sults in lower precision and power, and one can increase precision and learn more about geographic or time differences by fitting a unified model to the entire subject series including, for example, country or calendar time as a main effect and/or as an interacting effect. Indeed one could use the following working definition of external validation: validation of a prediction tool using data that were not available when the tool needed to be completed. An alternate definition could be taken as the validation of a prediction tool by an independent research team.

One suggested hierarchy of the quality of various validation methods is as follows, ordered from worst to best.

1. Attempting several validations (internal or external) and reporting only the one that "worked"
2. Reporting apparent performance on the training dataset (no validation)
3. Reporting predictive accuracy on an undersized independent test sample
4. Internal validation using data-splitting where at least one of the training and test samples is not huge and the investigator is not aware of the arbitrariness of variable selection done on a single sample
5. Strong internal validation using 100 repeats of 10-fold cross-validation or several hundred bootstrap resamples, repeating *all* analysis steps involving Y afresh at each re-sample and the arbitrariness of selected "important variables" is reported (if variable selection is used)
6. External validation on a large test sample, done by the original research team
7. Re-analysis by an independent research team using strong internal validation of the original dataset
8. External validation using new test data, done by an independent research team
9. External validation using new test data generated using different instruments/technology, done by an independent research team

Internal validation involves fitting and validating the model by carefully using one series of subjects. One uses the combined dataset in this way to estimate the likely performance of the final model on new subjects, which after all is often of most interest. Most of the remainder of Section 5.3 deals with internal validation.

5.3.2 Which Quantities Should Be Used in Validation?

For ordinary multiple regression models, the R^2 index is a good measure of the model's predictive ability, especially for the purpose of quantifying drop-off in predictive ability when applying the model to other datasets. R^2 is biased, however. For example, if one used nine predictors to predict outcomes of 10 subjects, $R^2 = 1.0$ but the R^2 that will be achieved on future

subjects will be close to zero. In this case, dramatic overfitting has occurred. The *adjusted* R^2 (Equation 4.4) solves this problem, at least when the model has been completely prespecified and no variables or parameters have been "screened" out of the final model fit. That is, R^2_{adj} is only valid when p in its formula is honest— when it includes all parameters ever examined (formally or informally, e.g., using graphs or tables) whether these parameters are in the final model or not.

Quite often we need to validate indexes other than R^2 for which adjustments for p have not been created.[c] We also need to validate models containing "phantom degrees of freedom" that were screened out earlier, formally or informally. For these purposes, we obtain nearly unbiased estimates of R^2 or other indexes using data splitting, cross-validation, or the bootstrap. The bootstrap provides the most precise estimates.

The g–index is another discrimination measure to validate. But g and R^2 measures only one aspect of predictive ability. In general, there are two major aspects of predictive accuracy that need to be assessed. As discussed in Section 4.5, *calibration* or *reliability* is the ability of the model to make unbiased estimates of outcome. *Discrimination* is the model's ability to separate subjects' outcomes. Validation of the model is recommended even when a data reduction technique is used. This is a way to ensure that the model was not overfitted or is otherwise inaccurate.

5.3.3 Data-Splitting

The simplest validation method is one-time *data-splitting*. Here a dataset is split into *training* (model development) and *test* (model validation) samples by a random process with or without balancing distributions of the response and predictor variables in the two samples. In some cases, a chronological split is used so that the validation is prospective. The model's calibration and discrimination are validated in the test set.

In ordinary least squares, calibration may be assessed by, for example, plotting Y against $\hat{Y}$. Discrimination here is assessed by R^2 and it is of interest in comparing R^2 in the training sample with that achieved in the test sample. A drop in R^2 indicates overfitting, and the absolute R^2 in the test sample is an unbiased estimate of predictive discrimination. Note that in extremely overfitted models, R^2 in the test set can be negative, since it is computed on "frozen" intercept and regression coefficients using the formula $1 - SSE/SST$, where SSE is the error sum of squares, SST is the total sum

[c] For example, in the binary logistic model, there is a generalization of R^2 available, but no adjusted version. For logistic models we often validate other indexes such as the ROC area or rank correlation between predicted probabilities and observed outcomes. We also validate the calibration accuracy of $\hat{Y}$ in predicting Y.

of squares, and SSE can be greater than SST (when predictions are worse than the constant predictor $\overline{Y}$).

To be able to validate predictions from the model over an entire test sample (without validating it separately in particular subsets such as in males and females), the test sample must be large enough to precisely fit a model containing one predictor. For a study with a continuous uncensored response variable, the test sample size should ordinarily be ≥ 100 at a bare minimum. For survival time studies, the test sample should at least be large enough to contain a minimum of 100 outcome events. For binary outcomes, the test sample should contain a bare minimum of 100 subjects in the least frequent outcome category. Once the size of the test sample is determined, the remaining portion of the original sample can be used as a training sample. Even with these test sample sizes, validation of extreme predictions is difficult.

Data-splitting has the advantage of allowing hypothesis tests to be confirmed in the test sample. However, it has the following disadvantages.

1. Data-splitting greatly reduces the sample size for both model development and model testing. Because of this, Roecker[528] found this method "appears to be a costly approach, both in terms of predictive accuracy of the fitted model and the precision of our estimate of the accuracy." Breiman [66, Section 1.3] found that bootstrap validation on the original sample was as efficient as having a separate test sample twice as large[36].
2. It requires a larger sample to be held out than cross-validation (see below) to be able to obtain the same precision of the estimate of predictive accuracy.
3. The split may be fortuitous; if the process were repeated with a different split, different assessments of predictive accuracy may be obtained.
4. Data-splitting does not validate the final model, but rather a model developed on only a subset of the data. The training and test sets are recombined for fitting the final model, which is not validated.
5. Data-splitting requires the split before the *first* analysis of the data. With other methods, analyses can proceed in the usual way on the complete dataset. Then, after a "final" model is specified, the modeling process is rerun on multiple resamples from the original data to mimic the process that produced the "final" model.

5.3.4 Improvements on Data-Splitting: Resampling

Bootstrapping, jackknifing, and other resampling plans can be used to obtain nearly unbiased estimates of model performance without sacrificing sample size. These methods work when either the model is completely specified except for the regression coefficients, or all important steps of the modeling process, especially variable selection, are automated. Only then can each

bootstrap replication be a reflection of all sources of variability in modeling. Note that most analyses involve examination of graphs and testing for lack of model fit, with many intermediate decisions by the analyst such as simplification of interactions. These processes are difficult to automate. But variable selection alone is often the greatest source of variability because of multiple comparison problems, so the analyst must go to great lengths to bootstrap or jackknife variable selection.

The ability to study the arbitrariness of how a stepwise variable selection algorithm selects "important" factors is a major benefit of bootstrapping. A useful display is a matrix of blanks and asterisks, where an asterisk is placed in column x of row i if variable x is selected in bootstrap sample i (see p. 263 for an example). If many variables appear to be selected at random, the analyst may want to turn to a data reduction method rather than using stepwise selection (see also [541]).

Cross-validation is a generalization of data-splitting that solves some of the problems of data-splitting. *Leave-out-one cross-validation*,[565,633] the limit of cross-validation, is similar to jackknifing.[675] Here one observation is omitted from the analytical process and the response for that observation is predicted using a model derived from the remaining $n - 1$ observations. The process is repeated n times to obtain an average accuracy. Efron[172] reports that grouped cross-validation is more accurate; here groups of k observations are omitted at a time. Suppose, for example, that 10 groups are used. The original dataset is divided into 10 equal subsets at random. The first 9 subsets are used to develop a model (transformation selection, interaction testing, stepwise variable selection, etc. are all done). The resulting model is assessed for accuracy on the remaining 1/10th of the sample. This process is repeated at least 10 times to get an average of 10 indexes such as R^2. 11

A drawback of cross-validation is the choice of the number of observations to hold out from each fit. Another is that the number of repetitions needed to achieve accurate estimates of accuracy often exceeds 200. For example, one may have to omit $\frac{1}{10}$th of the sample 500 times to accurately estimate the index of interest Thus the sample would need to be split into tenths 50 times. 12 Another possible problem is that cross-validation may not fully represent the variability of variable selection. If 20 subjects are omitted each time from a sample of size 1000, the lists of variables selected from each training sample of size 980 are likely to be much more similar than lists obtained from fitting independent samples of 1000 subjects. Finally, as with data-splitting, cross-validation does not validate the full 1000-subject model.

An interesting way to study overfitting could be called the randomization method. Here we ask the question "How well can the response be predicted when we use our best procedure on random responses when the predictive accuracy should be near zero?" The better the fit on random Y, the worse the overfitting. The method takes a random permutation of the response variable and develops a model with optional variable selection based on the original X and permuted Y. Suppose this yields $R^2 = .2$ for the fitted sample. Apply the

fit to the original data to estimate optimism. If overfitting is not a problem,
13 R^2 would be the same for both fits and it will ordinarily be very near zero.

5.3.5 Validation Using the Bootstrap

Efron,[172,173] Efron and Gong,[175] Gong,[224] Efron and Tibshirani,[177,178] Linnet,[416] and Breiman[66] describe several bootstrapping procedures for obtaining nearly unbiased estimates of future model performance without holding back data when making the final estimates of model parameters. With the "simple bootstrap" [178, p. 247], one repeatedly fits the model in a bootstrap sample and evaluates the performance of the model on the original sample. The estimate of the likely performance of the final model on future data is estimated by the average of all of the indexes computed on the original sample.

Efron showed that an enhanced bootstrap estimates future model performance more accurately than the simple bootstrap. Instead of estimating an accuracy index directly from averaging indexes computed on the original sample, the enhanced bootstrap uses a slightly more indirect approach by estimating the bias due to overfitting or the "optimism" in the final model fit. After the optimism is estimated, it can be subtracted from the index of accuracy derived from the original sample to obtain a bias-corrected or overfitting-corrected estimate of predictive accuracy. The bootstrap method is as follows. From the original X and Y in the sample of size n, draw a sample with replacement also of size n. Derive a model in the bootstrap sample and apply it without change to the original sample. The accuracy index from the bootstrap sample minus the index computed on the original sample is an estimate of optimism. This process is repeated for 100 or so bootstrap replications to obtain an average optimism, which is subtracted from the final
14 model fit's apparent accuracy to obtain the overfitting-corrected estimate.

Note that bootstrapping validates the *process* that was used to fit the original model (as does cross-validation). It provides an estimate of the *expected value* of the optimism, which when subtracted from the original index, pro-
15 vides an estimate of the *expected* bias-corrected index. If stepwise variable selection is part of the bootstrap process (as it must be if the final model is developed that way), and not all resamples (samples with replacement or training samples in cross-validation) resulted in the same model (which is almost always the case), this internal validation process actually provides an unbiased estimate of the future performance of the *process* used to identify markers and scoring systems; it does not validate a single final model. But resampling does tend to provide good estimates of the future performance of the final model that was selected using the same procedure repeated in the resamples.

Note that by drawing samples from X and Y, we are estimating aspects of the *unconditional* distribution of statistical quantities. One could instead draw samples from quantities such as residuals from the model to obtain a distribution that is conditional on X. However, this approach requires that the model be specified correctly, whereas the unconditional bootstrap does not. Also, the unconditional estimators are similar to conditional estimators except for very skewed or very small samples [186, p. 217].

Bootstrapping can be used to estimate the optimism in virtually any index. Besides discrimination indexes such as R^2, slope and intercept calibration factors can be estimated. When one fits the model $C(Y|X) = X\beta$, and then refits the model $C(Y|X) = \gamma_0 + \gamma_1 X\hat{\beta}$ on the same data, where $\hat{\beta}$ is an estimate of β, $\hat{\gamma}_0$ and $\hat{\gamma}_1$ will necessarily be 0 and 1, respectively. However, when $\hat{\beta}$ is used to predict responses on another dataset, $\hat{\gamma}_1$ may be < 1 if there is overfitting, and $\hat{\gamma}_0$ will be different from zero to compensate. Thus a bootstrap estimate of γ_1 will not only quantify overfitting nicely, but can also be used to shrink predicted values to make them more calibrated (similar to [582]). Efron's optimism bootstrap is used to estimate the optimism in $(0, 1)$ and then (γ_0, γ_1) are estimated by subtracting the optimism in the constant estimator $(0, 1)$. Note that in cross-validation one estimates β with $\hat{\beta}$ from the training sample and fits $C(Y|X) = \gamma X\hat{\beta}$ on the test sample directly. Then the γ estimates are averaged over all test samples. This approach does not require the choice of a 16
parameter that determines the amount of shrinkage as does ridge regression or penalized maximum likelihood estimation; instead one estimates how to make the initial fit well calibrated.[123,633] However, this approach is not as reliable as building shrinkage into the original estimation process. The latter allows different parameters to be shrunk by different factors.

Ordinary bootstrapping can sometimes yield overly optimistic estimates of optimism, that is, may underestimate the amount of overfitting. This is especially true when the ratio of the number of observations to the number of parameters estimated is not large.[205] A variation on the bootstrap that improves precision of the assessment is the ".632" method, which Efron found to be optimal in several examples.[172] This method provides a bias-corrected estimate of predictive accuracy by substituting $0.632\times$ [apparent accuracy $-\hat{\epsilon}_0$] for the estimate of optimism, where $\hat{\epsilon}_0$ is a weighted average of accuracies evaluated on observations *omitted* from bootstrap samples [178, Eq. 17.25, p. 253]. 17

For ordinary least squares, where the genuine per-observation .632 estimator can be used, several simulations revealed close agreement with the modified .632 estimator, even in small, highly overfitted samples. In these overfitted cases, the ordinary bootstrap bias-corrected accuracy estimates were significantly higher than the .632 estimates. Simulations[259,591] have shown, however, that for most types of indexes of accuracy of binary logistic regression models, Efron's original bootstrap has lower mean squared error than the .632 bootstrap when $n = 200, p = 30$. Bootstrap overfitting-corrected es- 18
timates of model performance can be biased in favor of the model. Although

Table 5.2 Example validation with and without variable selection

Method	Apparent Rank Correlation of Predicted vs. Observed	Over-Optimism	Bias-Corrected Correlation
Full Model	0.50	0.06	0.44
Stepwise Model	0.47	0.05	0.42

cross-validation is less biased than the bootstrap, Efron[172] showed that it has much higher variance in estimating overfitting-corrected predictive accuracy than bootstrapping. In other words, cross-validation, like data-splitting, can yield significantly different estimates when the entire validation process is repeated.

It is frequently very informative to estimate a measure of predictive accuracy forcing all candidate factors into the fit and then to separately estimate accuracy allowing stepwise variable selection, possibly with different stopping rules. Consistent with Spiegelhalter's proposal to use all factors and then to shrink the coefficients to adjust for overfitting,[582] the full model fit will outperform the stepwise model more often than not. Even though stepwise modeling has slightly less optimism in predictive discrimination, this improvement is not enough to offset the loss of information from deleting even marginally important variables. Table 5.2 shows a typical scenario. In this example, stepwise modeling lost a possible $0.50 - 0.47 = 0.03$ predictive discrimination. The full model fit will especially be an improvement when

1. the stepwise selection deletes several variables that are almost significant;
2. these marginal variables have *some* real predictive value, even if it's slight; and
3. there is no small set of extremely dominant variables that would be easily found by stepwise selection.

19

Faraway[186] has a fascinating study showing how resampling methods can be used to estimate the distributions of predicted values and of effects of a predictor, adjusting for an automated multistep modeling process. Bootstrapping can be used, for example, to penalize the variance in predicted values for choosing a transformation for Y and for outlier and influential observation deletion, in addition to variable selection. Estimation of the transformation of Y greatly increased the variance in Faraway's examples. Brownstone [77, p. 74] states that "In spite of considerable efforts, theoretical statisticians have been unable to analyze the sampling properties of [usual multistep modeling strategies] under realistic conditions" and concludes that the modeling strategy must be completely specified and then bootstrapped to get consistent

20

estimates of variances and other sampling properties.

5.4 Bootstrapping Ranks of Predictors

When the order of importance of predictors is not pre-specified but the researcher attempts to determine that order by assessing multiple associations with Y, the process of selecting "winners" and "losers" is unreliable. The bootstrap can be used to demonstrate the difficulty of this task, by estimating confidence intervals for the ranks of all the predictors. Even though the bootstrap intervals are wide, they actually underestimate the true widths[250].

The following exampling uses simulated data with known ranks of importance of 12 predictors, using an ordinary linear model. The importance metric is the partial χ^2 minus its degrees of freedom, while the true metric is the partial β, as all covariates have $U(0,1)$ distributions.

```
# Use the plot method for anova, with pl=FALSE to suppress
# actual plotting of chi-square - d.f. for each bootstrap
# repetition.  Rank the negative of the adjusted chi-squares
# so that a rank of 1 is assigned to the highest.  It is
# important to tell plot.anova.rms not to sort the results,
# or every bootstrap replication would have ranks of 1,2,3,
# ... for the partial test statistics.
require(rms)
n ← 300
set.seed(1)
d ← data.frame(x1=runif(n), x2=runif(n),  x3=runif(n),
   x4=runif(n), x5=runif(n), x6=runif(n),  x7=runif(n),
   x8=runif(n), x9=runif(n), x10=runif(n), x11=runif(n),
   x12=runif(n))
d$y ← with(d, 1*x1 + 2*x2 + 3*x3 +  4*x4  + 5*x5 + 6*x6 +
               7*x7 + 8*x8 + 9*x9 + 10*x10 + 11*x11 +
              12*x12 + 9*rnorm(n))

f ← ols(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12, data=d)
B ← 1000
ranks ← matrix(NA, nrow=B, ncol=12)
rankvars ← function(fit)
  rank(plot(anova(fit), sort='none', pl=FALSE))
Rank ← rankvars(f)
for(i in 1:B) {
  j ← sample(1:n, n, TRUE)
  bootfit ← update(f, data=d, subset=j)
  ranks[i,] ← rankvars(bootfit)
  }
lim ← t(apply(ranks, 2, quantile, probs=c(.025,.975)))
predictor ← factor(names(Rank), names(Rank))
w ← data.frame(predictor, Rank, lower=lim[,1], upper=lim[,2])
require(ggplot2)
ggplot(w, aes(x=predictor, y=Rank)) + geom_point() +
  coord_flip() + scale_y_continuous(breaks=1:12) +
  geom_errorbar(aes(ymin=lim[,1], ymax=lim[,2]), width=0)
```

With a sample size of $n = 300$ the observed ranks of predictor importance do not coincide with population βs, even when there are no collinearities among

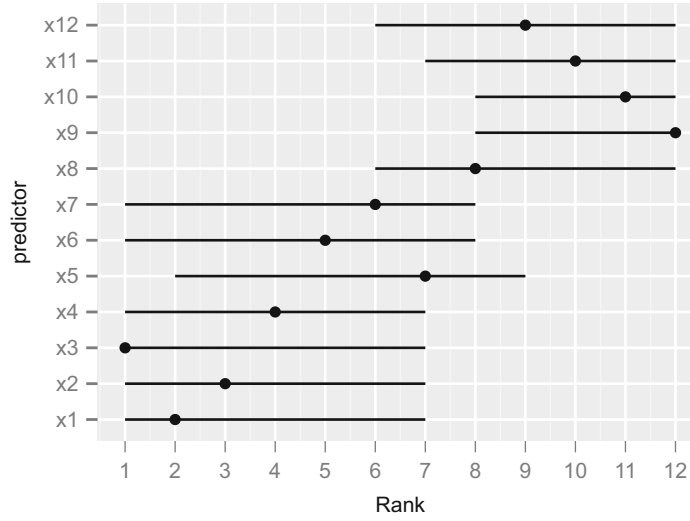

Fig. 5.3 Bootstrap percentile 0.95 confidence limits for ranks of predictors in an OLS model. Ranking is on the basis of partial χ^2 minus d.f. Point estimates are original ranks

the predictors. Confidence intervals are wide; for example the 0.95 confidence interval for the rank of `x7` (which has a true rank of 7) is $[1, 8]$, so we are only confident that `x7` is not one of the 4 most influential predictors. The confidence intervals do include the true ranks in each case (Figure 5.3).

5.5 Simplifying the Final Model by Approximating It

5.5.1 Difficulties Using Full Models

A model that contains all prespecified terms will usually be the one that predicts the most accurately on new data. It is also a model for which confidence limits and statistical tests have the claimed properties. Often, however, this model will not be very parsimonious. The full model may require more predictors than the researchers care to collect in future samples. It also requires predicted values to be conditional on all of the predictors, which can increase the variance of the predictions.

As an example suppose that least squares has been used to fit a model containing several variables including race (with four categories). Race may be an insignificant predictor and may explain a tiny fraction of the observed variation in Y. Yet when predictions are requested, a value for race must be inserted. If the subject is of the majority race, and this race has a majority of,

say 0.75, the variance of the predicted value will not be significantly greater than the variance for a predicted value from a model that excluded race for its list of predictors. If, however, the subject is of a minority race (say "other" with a prevalence of 0.01), the predicted value will have much higher variance. One approach to this problem, that does not require development of a second model, is to ignore the subject's race and to get a weighted average prediction. That is, we obtain predictions for each of the four races and weight these predictions by the relative frequencies of the four races.[d] This weighted average estimates the expected value of Y unconditional on race. It has the advantage of having exactly correct confidence limits when model assumptions are satisfied, because the correct "error term" is being used (one that deducts 3 d.f. for having ever estimated the race effect). In regression models having nonlinear link functions, this process does not yield such a simple interpretation.

When predictors are collinear, their competition results in larger P-values when predictors are (often inappropriately) tested individually. Likewise, confidence intervals for individual effects will be wide and uninterpretable (can other variables really be held constant when one is changed?).

5.5.2 Approximating the Full Model

When the full model contains several predictors that do not appreciably affect the predictions, the above process of "unconditioning" is unwieldy. In the search for a simple solution, the most commonly used procedure for making the model parsimonious is to remove variables on the basis of P-values, but this results in a variety of problems as we have seen. Our approach instead is to consider the full model fit as the "gold standard" model, especially the model from which formal inferences are made. We then proceed to approximate this full model to any desired degree of accuracy. For any approximate model we calculate the accuracy with which it approximates the best model. One goal this process accomplishes is that it provides different degrees of parsimony to different audiences, based on their needs. One investigator may be able to collect only three variables, another one seven. Each investigator will know how much she is giving up by using a subset of the predictors. In approximating the gold standard model it is very important to note that there is nothing gained in removing certain nonlinear terms; gains in parsimony come only from removing entire predictors. Another accomplishment of model approximation is that when the full model has been fitted using

[d] Using the `rms` package described in Chapter 6, such estimates and their confidence limits can easily be obtained, using for example `contrast(fit, list(age=50, disease='hypertension', race=levels(race)), type='average', weights=table(race))`.

shrinkage (penalized estimation, Section 9.10), the approximate models will inherit the shrinkage (see Section 14.10 for an example).

Approximating complex models with simpler ones has been used to decode "black boxes" such as artificial neural networks. Recursive partitioning trees (Section 2.5) may sometimes be used in this context. One develops a regression tree to predict the predicted value $X\hat{\beta}$ on the basis of the unique variables in X, using R^2, the average absolute prediction error, or the maximum absolute prediction error as a stopping rule, for example[184]. The user desiring simplicity may use the tree to obtain predicted values, using the first k nodes, with k just large enough to yield a low enough absolute error in predicting the more comprehensive prediction. Overfitting is not a problem as it is when the tree procedure is used to predict the outcome, because (1) given the predictor values the predictions are deterministic and (2) the variable being predicted is a continuous, completely observed variable. Hence the best cross-validating tree approximation will be one with one subject per node. One advantage of the tree-approximation procedure is that data collection on an individual subject whose outcome is being predicted may be abbreviated by measuring only those Xs that are used in the top nodes, until the prediction is resolved to within a tolerable error.

When principal component regression is being used, trees can also be used to approximate the components or to make them more interpretable.

Full models may also be approximated using least squares as long as the linear predictor $X\hat{\beta}$ is the target, and not some nonlinear transformation of it such as a logistic model probability. When the original model was fitted using unpenalized least squares, submodels fitted against $\hat{Y}$ will have the same coefficients as if least squares had been used to fit the subset of predictors directly against Y. To see this, note that if X denotes the entire design matrix and T denotes a subset of the columns of X, the coefficient estimates for the full model are $(X'X)^{-1}X'Y$, $\hat{Y} = X(X'X)^{-1}X'Y$, estimates for a reduced model fitted against Y are $(T'T)^{-1}T'Y$, and coefficients fitted against $\hat{Y}$ are $(T'T)T'X(X'X)^{-1}X'Y$ which can be shown to equal $(T'T)^{-1}T'Y$.

When least squares is used for both the full and reduced models, the variance–covariance matrix of the coefficient estimates of the reduced model is $(T'T)^{-1}\sigma^2$, where the residual variance σ^2 is estimated using the *full* model. When σ^2 is estimated by the unbiased estimator using the d.f. from the full model, which provides the only unbiased estimate of σ^2, the estimated variance–covariance matrix of the reduced model will be appropriate (unlike that from stepwise variable selection) although the bootstrap may be needed to fully take into account the source of variation due to how the approximate model was selected.

So if in the least squares case the approximate model coefficients are identical to coefficients obtained upon fitting the reduced model against Y, how is model approximation any different from stepwise variable selection? There are several differences, in addition to how σ^2 is estimated.

1. When the full model is approximated by a backward step-down procedure against $\hat{Y}$, the stopping rule is less arbitrary. One stops deleting variables when deleting any further variable would make the approximation inadequate (e.g., the R^2 for predictions from the reduced model against the original $\hat{Y}$ drops below 0.95).
2. Because the stopping rule is different (i.e., is not based on P-values), the approximate model will have a different number of predictors than an ordinary stepwise model.
3. If the original model used penalization, approximate models will inherit the amount of shrinkage used in the full fit.

Typically, though, if one performed ordinary backward step-down against Y using a large cutoff for α (e.g., 0.5), the approximate model would be very similar to the step-down model. The main difference would be the use of a larger estimate of σ^2 and smaller error d.f. than are used for the ordinary step-down approach (an estimate that pretended the final reduced model was prespecified).

When the full model was not fitted using least squares, least squares can still easily be used to approximate the full model. If the coefficient estimates from the full model are $\hat{\beta}$, estimates from the approximate model are matrix contrasts of $\hat{\beta}$, namely, $W\hat{\beta}$, where $W = (T'T)^{-1}T'X$. So the variance–covariance matrix of the reduced coefficient estimates is given by

$$WVW', \tag{5.2}$$

where V is the variance matrix for $\hat{\beta}$. See Section 19.5 for an example. Ambler et al.[21] studied model simplification using simulation studies based on several clinical datasets, and compared it with ordinary backward stepdown variable selection and with shrinkage methods such as the *lasso* (see Section 4.3). They found that ordinary backwards variable selection can be competitive when there is a large fraction of truly irrelevant predictors (something that can be difficult to know in advance). Paul et al.[485] found advantages to modeling the response with a complex but reliable approach, and then developing a parsimoneous model using the *lasso* or stepwise variable selection against $\hat{Y}$. See Section 11.7 for a case study in model approximation.

5.6 Further Reading

1 Gelman[213] argues that continuous variables should be scaled by two standard deviations to make them comparable to binary predictors. However his approach assumes linearity in the predictor effect and assumes the prevalence of the binary predictor is near 0.5. John Fox [202, p. 95] points out that if two predictors are on the same scale and have the same impact (e.g., years of employment and years of education), standardizing the coefficients will make them appear to have different impacts.

2 Levine et al.[401] have a compelling argument for graphing effect ratios on a logarithmic scale.

3 Hankins[254] is a definitive reference on nomograms and has multi-axis examples of historical significance. According to Hankins, Maurice d'Ocagne could be called the inventor of the nomogram, starting with alignment diagrams in 1884 and declaring a new science of "nomography" in 1899. d'Ocagne was at École des Ponts et Chaussées, a French civil engineering school. Julien and Hanley[328] have a nice example of adding axes to a nomogram to estimate the absolute effect of a treatment estimated using a Cox proportional hazards model. Kattan and Marasco[339] have several clinical examples and explain advantages to the user of nomograms over "black box" computerized prediction.

4 Graham and Clavel[231] discuss graphical and tabular ways of obtaining risk estimates. van Gorp et al.[630] have a nice example of a score chart for manually obtaining estimates.

5 Larsen and Merlo[375] developed a similar measure—the median odds ratio. Gönen and Heller[223] developed a c-index that like g is a function of the covariate distribution.

6 Booth and Sarkar[61] have a nice analysis of the number of bootstrap resamples needed to guarantee with 0.95 confidence that a variance estimate has a sufficiently small relative error. They concentrate on the Monte Carlo simulation error, showing that small errors in variance estimates can lead to important differences in P-values. Canty et al.[91] provide a number of diagnostics to check the reliability of bootstrap calculations.

7 There are many variations on the basic bootstrap for computing confidence limits.[150,178] See Booth and Sarkar[61] for useful information about choosing the number of resamples. They report the number of resamples necessary to not appreciably change P-values, for example. Booth and Sarkar propose a more conservative number of resamples than others use (e.g., 800 resamples) for estimating variances. Carpenter and Bithell[92] have an excellent overview of bootstrap confidence intervals, with practical guidance. They also have a good discussion of the unconditional nonparametric bootstrap versus the conditional semiparametric bootstrap.

8 Altman and Royston[18] have a good general discussion of what it means to validate a predictive model, including issues related to study design and consideration of uses to which the model will be put.

9 An excellent paper on external validation and generalizability is Justice et al.[329]. Bleeker et al.[58] provide an example where internal validation is misleading when compared with a true external validation done using subjects from different centers in a different time period. Vergouwe et al.[638] give good guidance about the number of events needed in sample used for external validation of binary logistic models.

10 See Picard and Berk[505] for more about data-splitting.

11 In the context of variable selection where one attempts to select the set of variables with nonzero true regression coefficients in an ordinary regression model, Shao[565] demonstrated that leave-out-one cross-validation selects models that are "too large." Shao also showed that the number of observations held back for validation should often be larger than the number used to train the model. This is because in this case one is not interested in an accurate model (you fit the whole sample to do that), but an accurate estimate of prediction error is mandatory so as to know which variables to allow into the final model. Shao suggests using a cross-validation strategy in which approximately $n^{3/4}$ observations are used in each training sample and the remaining observations are used in the test sample. A repeated balanced or Monte Carlo splitting approach is used, and accuracy estimates are averaged over $2n$ (for the Monte Carlo method) repeated splits.

12 Picard and Cook's Monte Carlo cross-validation procedure[506] is an improvement over ordinary cross-validation.

13 The randomization method is related to Kipnis' "chaotization relevancy principle"[348] in which one chooses between two models by measuring how far each is from a nonsense model. Tibshirani and Knight also use a randomization method for estimating the optimism in a model fit.[611]

14 This method used here is a slight change over that presented in [172], where Efron wrote predictive accuracy as a sum of per-observation components (such as 1 if the observation is classified correctly, 0 otherwise). Here we are writing $m \times$ the unitless summary index of predictive accuracy in the place of Efron's sum of m per-observation accuracies [416, p. 613].

15 See [633] and [66, Section 4] for insight on the meaning of expected optimism.

16 See Copas,[123] van Houwelingen and le Cessie [633, p. 1318], Verweij and van Houwelingen,[640] and others[631] for other methods of estimating shrinkage coefficients.

17 Efron[172] developed the ".632" estimator only for the case where the index being bootstrapped is estimated on a per-observation basis. A natural generalization of this method can be derived by assuming that the accuracy evaluated on observation i that is omitted from a bootstrap sample has the same expectation as the accuracy of any other observation that would be omitted from the sample. The modified estimate of ϵ_0 is then given by

$$\hat{\epsilon}_0 = \sum_{i=1}^{B} w_i T_i, \tag{5.3}$$

where T_i is the accuracy estimate derived from fitting a model on the ith bootstrap sample and evaluating it on the observations omitted from that bootstrap sample, and w_i are weights derived for the B bootstrap samples:

$$w_i = \frac{1}{n} \sum_{j=1}^{n} \frac{[\text{bootstrap sample } i \text{ omits observation } j]}{\#\text{bootstrap samples omitting observation } j}. \tag{5.4}$$

Note that $\hat{\epsilon}_0$ is undefined if any observation is included in every bootstrap sample. Increasing B will avoid this problem. This modified ".632" estimator is easy to compute if one assembles the bootstrap sample assignments and computes the w_i before computing the accuracy indexes T_i. For large n, the w_i approach $1/B$ and so $\hat{\epsilon}_0$ becomes equivalent to the accuracy computed on the observations not contained in the bootstrap sample and then averaged over the B repetitions.

18 Efron and Tibshirani[179] have reduced the bias of the ".632" estimator further with only a modest increase in its variance. Simulation has, however, shown no advantage of this ".632+" method over the basic optimism bootstrap for most accuracy indexes used in logistic models.

19 van Houwelingen and le Cessie[633] have several interesting developments in model validation. See Breiman[66] for a discussion of the choice of X for which to validate predictions. Steyerberg et al.[587] present simulations showing the number of bootstrap samples needed to obtain stable estimates of optimism of various accuracy measures. They demonstrate that bootstrap estimates of optimism are nearly unbiased when compared with simulated external estimates. They also discuss problems with precision of estimates of accuracy, especially when using external validation on small samples.

20 Blettner and Sauerbrei also demonstrate the variability caused by data-driven analytic decisions.[59] Chatfield[100] has more results on the effects of using the data to select the model.

5.7 Problem

Perform a simulation study to understand the performance of various internal validation methods for binary logistic models. Modify the `R` code below in at least two meaningful ways with regard to covariate distribution or number, sample size, true regression coefficients, number of resamples, or number of times certain strategies are averaged. Interpret your findings and give recommendations for best practice for the type of configuration you studied. The `R` code from this assignment may be downloaded from the RMS course wiki page.

For each of 200 simulations, the code below generates a training sample of 200 observations with p predictors ($p = 15$ or 30) and a binary response. The predictors are independently $U(-0.5, 0.5)$. The response is sampled so as to follow a logistic model where the intercept is zero and all regression coefficients equal 0.5. The "gold standard" is the predictive ability of the fitted model on a test sample containing 50,000 observations generated from the same population model. For each of the 200 simulations, several validation methods are employed to estimate how the training sample model predicts responses in the 50,000 observations. These validation methods involve fitting 40 or 200 models in resamples.

g-fold cross-validation is done using the command `validate(f, method='cross', B=g)` using the `rms` package. This was repeated and averaged using an extra loop, shown below.

For bootstrap methods, `validate(f, method='boot' or '.632', B=40 or B=200)` was used. `method='.632'` does Efron's ".632" method[179], labeled `632a` in the output. An ad-hoc modification of the .632 method, `632b` was also done. Here a "bias-corrected" index of accuracy is simply the index evaluated in the observation omitted from the bootstrap resample. The "gold standard" external validations were obtained from the `val.prob` function in the `rms` package. The following indexes of predictive accuracy are used:

D_{xy}: Somers' rank correlation between predicted probability that $Y = 1$ vs. the binary Y values. This equals $2(C - 0.5)$ where C is the "ROC Area" or concordance probability.
D: Discrimination index — likelihood ratio χ^2 divided by the sample size
U: Unreliability index — unitless index of how far the logit calibration curve intercept and slope are from $(0, 1)$
Q: Logarithmic accuracy score — a scaled version of the log-likelihood achieved by the predictive model
Intercept: Calibration intercept on logit scale
Slope: Calibration slope (slope of predicted log odds vs. true log odds)

Accuracy of the various resampling procedures may be estimated by computing the mean absolute errors and the root mean squared errors of estimates (e.g., of D_{xy} from the bootstrap on the 200 observations) against the "gold standard" (e.g., D_{xy} for the fitted 200-observation model achieved in the 50,000 observations).

```
require(rms)
set.seed(1)  # so can reproduce results

n    ← 200             # Size of training sample
reps ← 200             # Simulations
npop ← 50000       # Size of validation gold standard sample
methods ← c('Boot 40','Boot 200','632a 40','632a 200',
            '632b 40','632b 200','10-fold x 4','4-fold x 10',
            '10-fold x 20','4-fold x 50')
R ← expand.grid(sim    = 1:reps,
                p      = c(15,30),
                method = methods)
R$Dxy ← R$Intercept ← R$Slope ← R$D ← R$U ← R$Q ←
  R$repmeth ← R$B ← NA
R$n ← n

## Function to do r overall reps of B resamples, averaging to
## get estimates similar to as if r*B resamples were done

val ← function(fit, method, B, r) {
  contains ← function(m) length(grep(m, method)) > 0
  meth ← if(contains('Boot')) 'boot' else
         if(contains('fold')) 'crossvalidation' else
         if(contains('632')) '.632'
  z ← 0
  for(i in 1:r) z ← z + validate(fit, method=meth, B=B)[
         c("Dxy","Intercept","Slope","D","U","Q"),
         'index.corrected']
  z/r
}
```

```
for(p in c(15, 30)) {

  ## For each p create the true betas, the design matrix,
  ## and realizations of binary y in the gold standard
  ## large sample
  Beta ← rep(.5, p)  # True betas
  X    ← matrix(runif(npop*p), nrow=npop) - 0.5
  LX   ← matxv(X, Beta)
  Y    ← ifelse(runif(npop) ≤ plogis(LX), 1, 0)

  ## For each simulation create the data matrix and
  ## realizations of y
  for(j in 1:reps) {

    ## Make training sample
    x ← matrix(runif(n*p), nrow=n) - 0.5
    L ← matxv(x, Beta)
    y ← ifelse(runif(n) ≤ plogis(L), 1, 0)
    f ← lrm(y ∼ x, x=TRUE, y=TRUE)
    beta ← f$coef
    forecast ← matxv(X, beta)
    ## Validate in population
```

```
    v ← val.prob(logit=forecast, y=Y, pl=FALSE)[
                    c("Dxy","Intercept","Slope","D","U","Q")]

    for(method in methods) {
      repmeth ← 1
      if(method %in% c('Boot 40','632a 40','632b 40'))
        B ← 40
      if(method %in% c('Boot 200','632a 200','632b 200'))
        B ← 200
      if(method == '10-fold x 4') {
        B ← 10
        repmeth ← 4
      }
      if(method == '4-fold x 10') {
        B ← 4
        repmeth ← 10
      }
      if(method == '10-fold x 20') {
        B ← 10
        repmeth ← 20
      }
      if(method == '4-fold x 50') {
        B ← 4
        repmeth ← 50
      }

      z ←  val(f, method, B, repmeth)
      k ← which(R$sim == j & R$p == p & R$method == method)
      if(length(k) != 1) stop('program logic error')
      R[k, names(z)] ← z - v
      R[k, c('B','repmeth')] ← c(B=B, repmeth=repmeth)
    } # end over methods
  } # end over reps
} # end over p
```

Results are best summarized in a multi-way dot chart. Bootstrap nonparametric percentile 0.95 confidence limits are included.

```
statnames ← names(R)[6:11]
w ← reshape(R, direction='long', varying=list(statnames),
            v.names='x', timevar='stat', times=statnames)
w$p ← paste('p', w$p, sep='=')
require(lattice)
s ← with(w, summarize(abs(x), llist(p, method, stat),
                      smean.cl.boot,stat.name='mae'))
Dotplot(method ~ Cbind(mae, Lower, Upper) | stat*p, data=s,
        xlab='Mean |error|')
s ← with(w, summarize(x^2, llist(p, method, stat),
                      smean.cl.boot, stat.name='mse'))
Dotplot(method ~ Cbind(sqrt(mse), sqrt(Lower), sqrt(Upper)) |
        stat*p, data=s,
        xlab=expression(sqrt(MSE)))
```