

# An Introduction to Bootstrap Methods using Arc

Iain Pardoe & Sanford Weisberg

School of Statistics

University of Minnesota, St. Paul, MN 55108

Technical Report Number 631

Work Supported by the National Science Foundation, Grant DMS 0103983

Revised version, October 9, 2001

## Abstract

This report presents (1) the basic ideas of bootstrapping when applied in regression problems, as described in [2, 3], and (2) how to implement these ideas using `Arc`, the computer package that accompanies [1]. This add-on works for linear, generalized linear, nonlinear and generalized nonlinear models.

## 1 Introduction

This report provides documentation for using the bootstrap add-on for most regression problems in `Arc`, [1], and an introduction to bootstrap methods in linear regression analyses [2, 3]. We discuss first in Section 3 how to use the add-on for the bootstrap, illustrating the use in several examples. Following this section, we discuss some of the details of the bootstrap in Section 5. In Section 6 we discuss the use of the bootstrap in comparing models. A few generalizations are presented in Section 7.

## 2 Getting the add-on

The web site <http://www.stat.umn.edu/arc> includes the most recent version of `Arc`. The bootstrap add-on is available from the Add-ons page on this same web site in the file `boot.lsp`.

To use `boot.lsp`, download it from the Internet, and put it in the folder `Extras` in your `Arc` directory. Further directions are given at the top of the add-ons page on the web site.

## 3 Using the bootstrap

### 3.1 A simple example of the bootstrap

Suppose we know that we are sampling from a particular distribution  $F$ , for example a standard normal distribution,  $y \sim N(0, 1)$ . Suppose further that we are taking a sample of size  $n$ . What is the standard error of the sample median?

We can obtain an approximate answer this problem by computer simulation, set up as follows. Let  $F$  be the cumulative distribution function of the  $N(0, 1)$  distribution:

1. Obtain a random sample  $y_1^*, \dots, y_n^*$  from  $F$ .
2. Compute and save the median of the sample in step 1.
3. Repeat steps 1 and 2 a large number, say  $B$ , times. The larger the value of  $B$ , the more precise the ultimate answer.
4. Estimate the standard error of the median by the standard deviation of the medians estimated in step 2.

To use this algorithm, we need to be able to simulate observations from the distribution  $F$ , and this is possible for most choices of  $F$ . For standard distributions like the normal, exponential, and so on, functions already exist for this, and for others there is an extensive literature on simulating values from a distribution (e.g., [4]). We could also get the standard error of any function of the sample data values by replacing the median with whatever other function is of interest. This is an example of a *parametric bootstrap*, in which the computer is used to sample repeatedly from a known parametric distribution.

Suppose next that rather than knowing  $F$  exactly, we only have a random sample  $y_1, \dots, y_n$  from  $F$ , and once again we want to estimate the standard deviation of the median of the sample of size  $n$ . In addition, we may want to make a confidence statement concerning the population median, or perhaps perform a test. While we no longer assume that  $F$  is known, we do have an estimate of  $F$ , namely the *sample cumulative distribution function*  $\hat{F}$  based on the available sample. We can now carry out the above algorithm, but sample from  $\hat{F}$ , not from  $F$ :

1. Obtain a random sample  $y_1^*, \dots, y_n^*$  from  $\hat{F}$  by sampling with replacement from the observed values  $y_1, \dots, y_n$ .
2. Compute and save the median of the sample in step 1.
3. Repeat steps 1 and 2 a large number, say  $B$ , times. The larger the value of  $B$ , the more precise the ultimate answer.
4. Estimate the standard error of the median by the standard deviation of the medians estimated in step 2. We can further use the sample distribution of the  $B$  bootstrap estimates of the median to perform tests or get confidence intervals.

For example, if  $B = 999$ , we can order the bootstraps from largest to smallest, and the interval between the 25th smallest value and the 975th largest value provides an estimated 95% confidence interval for the median. A test at the 5% level concerning the population median can be rejected if the hypothesized value of the median does not fall in the confidence interval. In particular applications of the bootstrap, improvements over these simple percentile-based methods are available, but we do not generally pursue them here.

This algorithm is an example of a *case bootstrap*, in which we do the resampling by taking samples from the sample cumulative distribution of the observed cases. In regression we will have another type of bootstrap based on resampling residuals; see Section 5.1.

The remainder of this section is not required for the rest of this report.

The Arc add-on for the bootstrap includes two very simple functions for the above two simulations. For example,

```
> (def ans1 (parametric-bootstrap 20))
> (mean ans1)
-0.00591493
> (standard-deviation ans1)
0.274598
```

The first line computed  $B = 999$  bootstrap samples from a standard normal distribution, computing the median of each sample. The mean of the  $B$  samples is  $-0.0059$ , not far from the true value of zero. The standard deviation estimates the standard error of the median, which is  $0.275$ . If you redo this example, you will of course get slightly different answers because this is a simulation, not an exact computation. The command

```
> (def ans2 (parametric-bootstrap 30
             :dist #'(lambda (n) (poisson-rand n 3))
             :statistic #'(lambda(x) (quantile x .25))
             :B 499))
> (mean ans2)
1.86072
> (standard-deviation ans2)
0.419337
```

returns 499 bootstraps, each from the Poisson distribution with mean  $\lambda = 3$ , and on each bootstrap the 25-th percentile will be computed. The average value estimates the expected 25-th percentile, about 1.86. The standard error of an estimated 25-th percentile is 0.419.

The function *case-bootstrap* is similar to the last function, except it does a case bootstrap. For example,

```
> (def data (poisson-rand 30 3))
> (def ans3 (case-bootstrap data
                 :statistic #'(lambda(x) (quantile x .25))
                 :B 499))
> (mean ans3)
2.24449
> (standard-deviation ans3)
```

```

0.388521
> (quantile ans3 '(.025 .975))
(2 3)

```

first computes a fixed sample of data with 30 Poisson random numbers, and then does a case bootstrap based on resampling from  $\hat{F}$  determined by these 30 numbers. The estimated standard deviation of the 25-th percentile is .388, and a 95% confidence interval for the 25-th percentile based on the observed sample is from 2 to 3 (recall the Poisson is a discrete distribution).

### 3.2 The bootstrap for regression

The use of the bootstrap add-on is best illustrated by example. We consider the transactions data described in Section 7.3.3 of [1]. The fitted linear regression is given in part by:

```

Data set = Transactions, Name of Fit = L1
Normal Regression
Kernel mean function = Identity
Response      = Time
Terms         = (T1 T2)
Coefficient Estimates
Label      Estimate      Std. Error    t-value    p-value
Constant  144.369             170.544      0.847     0.3980
T1         5.46206           0.433268    12.607    0.0000
T2         2.03455           0.0943368   21.567    0.0000

R Squared:           0.909053
Sigma hat:           1142.56
Number of cases:     261
Degrees of freedom:  258

```

In this output, the coefficients are the least squares estimates, and the standard errors are computed assuming that the assumptions of the linear model (correct mean function, constant variance) hold. Confidence statements and tests generally rely of the asymptotic normality of the estimated coefficients. However, the assumption of constant variance may not hold in these data, and an alternative method of inference might be useful.

To perform the bootstrap, select “Bootstrap” from the model’s menu. This will give the dialog shown in Figure 1. In this dialog, select the number of bootstraps you would like to perform. The default of  $B = 999$  is appropriate in easy-to-compute problems like linear regression, but may be too large for routine use in problems like nonlinear regression that require iterative computations. Some guidance on selecting  $B$  for other regression models is given in Section 3.3. You can also select the bootstrap method, either the parametric bootstrap, Section 5.3, case bootstrap, Section 5.2, or the residual bootstrap described in Section 5.1. For now, we select the case bootstrap method that was described in the last section, and press the OK button.

The computer program then computes  $B$  bootstrap samples, and saves the resulting coefficient estimates from each of the samples. When the computation is completed,

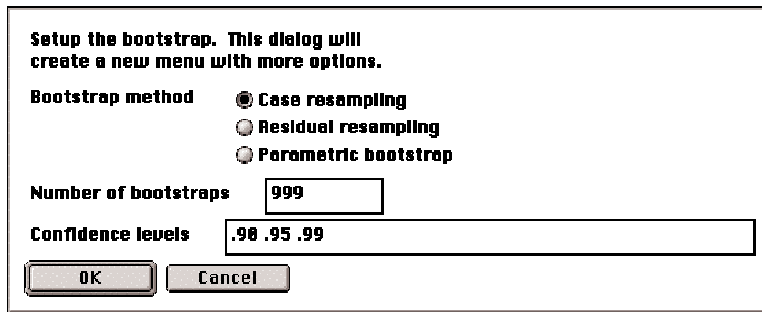


Figure 1: The bootstrap dialog. This dialog is the same for all regression models.

a probability plot is displayed, like the one in Figure 2. This figure displays all the bootstrap samples for one of the coefficients; for linear models with an intercept, the bootstrap sample intercepts are displayed. If normal theory is applied to these data, these would be like a sample from a  $t$  distribution with  $n - 3 = 258$  df, and so this distribution is used in the probability plot. A straight plot suggests that the usual normal theory will be appropriate for inference about this parameter. Probability plots for the other coefficients are obtained by pressing the mouse in the slide-bar that initially says Intercept; all three appear approximately straight.

Additional bootstrap options are contained in the Bootstrap pop up menu. Here are the items available in this menu:

**Display summaries** Gives a printed summary of common statistics based on the bootstrap. The output for the transactions data is shown in Table 1. In this table, the column marked “Observed estimate” is the original estimate based on the original data. The “Bootstrap bias” is the difference between the observed estimate and the mean of the  $B$  bootstrap samples. The “Model SE” is the nominal standard error computed from the model that was used to fit the data. The “Bootstrap SE” is the standard error based on the bootstrap. In this data set, the bootstrap standard errors are about 50% larger than the model-based standard errors, suggesting that the model is underestimating variability. The last two columns give  $p$ -values for the test that the coefficient is equal to zero, after adjusting for the others, first based on standard theory, and then based on the bootstrap.

Beneath this table is a summary of confidence intervals for the coefficient estimates. The column marked “Normal theory” is computed from the usual method of “estimate  $\pm$  multiplier  $\times$  standard error.” The percentile bootstrap uses percentiles of the sample distribution of the bootstrap samples to estimate the ends of the confidence intervals. The BCa method is described in [[3]]. We see from the output that normal theory probably has intervals that are too short in this example.

**Show Histograms** Display the bootstrap estimates in histograms.

**Save as a dataset** This will create a new Arc dataset whose variables are the bootstrap replicates. Using this dataset, you can compute other quantities that were not

Table 1: Bootstrap summary for the transactions data.

```

Data set = Transactions, Name of Fit = L1
Normal Regression
Kernel mean function = Identity
Response      = Time
Terms         = (T1 T2)
Bootstrap based on resampling cases
Number of replications = 999
      Observed Bootstrap      Model Bootstrap  Model  Boot
      Estimate Bias          SE           SE    pvalue pvalue
Intercept 144.37 -15.642      170.54   193.10  0.3980 0.4104
T1         5.4621 -5.4692E-2 0.43327  0.64311 0.0000 0.0000
T2         2.0345  1.3128E-2 9.4337E-2 0.14662 0.0000 0.0000
Coefficient confidence intervals, using three methods
Level Normal Theory  Percentile Bootstrap  BCa Bootstrap
Coefficient: Intercept
0.9 (-137.162 425.9) (-153.363 475.496) (-194.484 438.335)
0.95 (-191.466 480.205) (-206.884 544.436) (-236.417 492.953)
0.99 (-298.196 586.935) (-337.296 667.591) (-396.469 594.105)
Coefficient: T1
0.9 (4.74683 6.17729) (4.46815 6.51093) (4.19983 6.33105)
0.95 (4.60886 6.31525) (4.26133 6.79872) (3.89807 6.4477)
0.99 (4.33772 6.5864) (3.83344 7.0646) (3.51959 6.85134)
Coefficient: T2
0.9 (1.87882 2.19028) (1.77011 2.26107) (1.82777 2.32249)
0.95 (1.84878 2.22032) (1.73101 2.31182) (1.78805 2.38479)
0.99 (1.78974 2.27935) (1.66394 2.40912) (1.71666 2.44001)

```

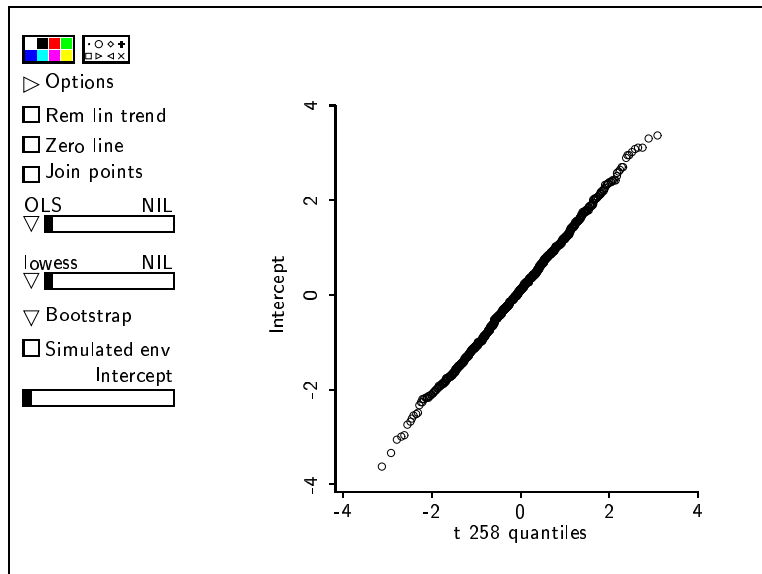


Figure 2: The bootstrap control plot. This is a probability plot of the bootstrap samples for the intercept for the transactions data. Probability plots for the other coefficients are obtained using the slider at the bottom left of the plot. Other bootstrap summaries are obtained using the Bootstrap pop-up menu.

included in the basic output. For example, if interested in a confidence interval for the ratio of the coefficient for  $T_1$  to the coefficient for  $T_2$ , one could: (1) create a new data set; (2) use the “Add a variate” item to compute the ratio; (3) use the “Table data” item to get the statistics of interest, such as the 0.25 and .975 quantiles that give the ends of a confidence interval.

### 3.3 Bootstrap in other regression models

The bootstrap add-on works the same way for linear models, generalized linear models, nonlinear models and generalized nonlinear models. Here are some special considerations for each of these types.

#### 3.3.1 Linear models

The implementation of the bootstrap used here should work well for all linear models. In addition, see Section 6.2 for additional methods for bootstrapping  $F$ -tests to compare models. The residual bootstrap and possibly the case bootstrap are likely to be the most useful here.

In some instances, when using the case bootstrap, some of the bootstrap samples may have fewer linearly independent terms than the original model. This may cause the bootstrap procedure to fail. Should this happen, try using the residual bootstrap.

### 3.3.2 Generalized linear models

The residual bootstrap method is only available for normal generalized linear models. Computations here are iterative, so setting  $B$  to be large can result in very lengthy computations. The case bootstrap is the preferred method, but it, too, may fail if the number of linearly independent terms in the bootstrap data is different from the original data.

### 3.3.3 Nonlinear models and generalized nonlinear models

Nonlinear models are also iterative, and furthermore starting values strongly influence whether or not estimates are actually found. The bootstrap method uses starting values from the original data, and counts the number of bootstrap samples for which the algorithm fails to converge. If the algorithm fails 25 times in a row on 25 different bootstrap samples, the program gives up. Case bootstraps are more likely to fail than residual bootstraps. The residual bootstrap is not available with non-normal generalized nonlinear models.

## 4 Linear regression

In this section, we review linear regression as a prelude to presenting some of the details for the bootstrap. Following the notation and development in [1], regression concerns a *response*  $y$  and  $p$  *predictors*,  $\mathbf{x} = (x_1, \dots, x_p)^T$ . The general goal in regression is to study how the conditional distribution of  $y|\mathbf{x}$  changes as the value of  $\mathbf{x}$  changes, often concentrating on the mean function,  $E(y|\mathbf{x})$ . In many regression problems, the response  $y|\mathbf{x}$  is written

$$y|\mathbf{x} = E(y|\mathbf{x}) + \varepsilon/\sqrt{w}$$

where  $\varepsilon$  is called the statistical *error* and the weights  $w > 0$  are *known, positive* numbers. Another feature of the conditional distribution of  $y|\mathbf{x}$  that is often studied in regression is the variance function  $\text{var}(y|\mathbf{x}) = \text{var}(\varepsilon)/w$ .

Let  $\mathbf{u}$  be a  $k \times 1$  vector of *terms* derived from  $\mathbf{x}$ . Typically,  $\mathbf{u}$  will consist of a constant 1 for an intercept, and  $(k - 1)$  additional functions of  $\mathbf{x}$ , like polynomials or other transformations. The *linear regression model* has mean function

$$E(y|\mathbf{x}) = E(y|\mathbf{u}) = \eta_0 + \eta_1 u_1 + \dots + \eta_{k-1} u_{k-1} = \boldsymbol{\eta}^T \mathbf{u} \quad (1)$$

where  $\boldsymbol{\eta}^T = (\eta_0, \dots, \eta_{k-1})$  is a  $k \times 1$  vector of mean function coefficients, and variance function

$$\text{var}(y|\mathbf{x}) = \sigma^2/w \quad (2)$$

These assumed forms of the mean and variance functions imply that  $E(\varepsilon) = 0$  and  $\text{var}(\varepsilon) = \sigma^2$ . This reflects an alternative way of specifying the general form of the linear regression model—the linear mean function (1) together with the assumption that the distribution of the errors is independent of  $\mathbf{x}$ .

For a full parametric analysis, the distribution of  $y|\mathbf{x}$ , or alternatively of  $\varepsilon$ , must be specified. For normally distributed errors, the least squares theory of regression



estimation and inference provides straightforward, exact methods for analysis. But for non-normal errors, these methods have the potential to be inaccurate or misleading. Resampling methods such as the bootstrap provide an alternative methodology, with the potential to both

- reinforce conclusions arrived at using normal theory, and
- to provide estimation and inference techniques in situations where normal theory does not seem to be justified.

For other regression problems, such as generalized linear models, nonlinear models and generalized nonlinear models, the development is similar to that for linear models. In those problems bootstrap methods can be even more useful because usual parametric inferences can be questionable as they may depend more heavily on either asymptotics or assumptions.

## 5 Types of bootstraps

The bootstrap is a data-based simulation method for statistical inference. The basic idea is as follows. To make an inference about a population quantity, say  $\theta$ , we have a data-based estimate,  $\hat{\theta}$ . We would like to get some idea of the distribution of the estimate, without having to make assumptions about the data, such as assuming that the data come from a multivariate normal distribution. One way to do this is to resample with replacement from the data to get a bootstrap sample of the same size as the original sample, and made up of cases from the original sample, some appearing once, some twice, and so on, and some not appearing at all. Repeat this process to get a large number,  $B$ , bootstrap samples, and calculate  $\hat{\theta}$  for each sample. For notation, denote bootstrap estimates with a star, and hence  $\hat{\theta}$  for a bootstrap sample is denoted  $\hat{\theta}^*$ . These  $B$   $\hat{\theta}^*$ 's contain information that can be used to make inferences from the data; essentially,  $\hat{\theta}^*$  is to  $\hat{\theta}$  as  $\theta$  is to  $\theta$ . Some of the types of inference possible in the linear regression context will be explored in more detail in Section 6. In the next three subsections, we present different ways of doing the resampling.

### 5.1 Resampling residuals

The “resampling residuals” paradigm takes the point of view that responses,  $y|\mathbf{x}$ , are sampled from a univariate distribution  $F(y|\mathbf{x})$  whose mean and variance are given by the mean function (1) and variance function (2). Trying to estimate  $F(y|\mathbf{x})$  using bootstrap methods doesn't work directly, because of the conditioning on  $\mathbf{x}$ . But, since  $\mathbf{x}$  is assumed to be fixed under this paradigm, estimate the distribution of  $\varepsilon$  by resampling residuals, defined to be the weighted differences between the observed values of the response and the fitted values under the linear regression model

$$\hat{\varepsilon} = \sqrt{w}(y - \hat{E}(y|\mathbf{x})) = \sqrt{w}(y - \hat{y}) \quad (3)$$

The linear regression model is implicit under this paradigm—using bootstrap methods in regression by resampling residuals essentially assumes that the linear regression model holds.

In the bootstrap add-on, we use a refinement of the residuals that has good properties. Since the distribution of  $\varepsilon$  is assumed to be independent of  $\mathbf{x}$ , we really want to be resampling something whose distribution does not depend on  $\mathbf{x}$ . Assuming  $n$  cases, the  $i$ -th linear model regression residual as defined in (3) has variance  $\text{var}(\hat{\varepsilon}_i|\mathbf{x}_i) = \sigma^2(1 - h_i)$ , where  $h_i$  is the  $i$ -th leverage,  $i = 1, \dots, n$ . The leverage depends on  $\mathbf{x}_i$ , and so if any leverages are particularly high, resampling the residuals will not do a good job of estimating the distribution of  $\varepsilon$ . Since the leverages are known, consider the *modified residuals*

$$\hat{r}_i = \frac{\sqrt{w_i}(y_i - \hat{y}_i)}{\sqrt{(1 - h_i)}} \quad (4)$$

which have constant variance. The modified residuals differ from the *Studentized residuals* described in Section 15.3.1 of [1] only by division by an estimate of  $\sigma^2$ , and thus the two are equivalent.

One final adjustment is needed since  $\varepsilon$  is assumed to have mean 0. Thus, we sample from the mean-corrected modified residuals to get bootstrap errors

$$\varepsilon_i^* \text{ randomly sampled with replacement from } \hat{r}_i - \bar{r}$$

where  $\bar{r}$  is the sample average of the modified residuals (4).

Assume that the linear regression model holds, and fix the bootstrap predictor values at the original sample  $\mathbf{x}_i$  values. Also fix bootstrap weights at the original sample  $w_i$  values. The usual weighted least squares estimate of  $\boldsymbol{\eta}$  from the original sample, say  $\hat{\boldsymbol{\eta}}$ , are used to derive bootstrap responses

$$y_i^* = \hat{\boldsymbol{\eta}}^T \mathbf{u}_i + \varepsilon_i^* / \sqrt{w_i}$$

Next, compute the usual weighted least squares estimate of  $\boldsymbol{\eta}$  from this bootstrap sample, say  $\hat{\boldsymbol{\eta}}^*$ , and repeat  $B$  times.

We use this same sampling plan in normal nonlinear models as well, with approximate leverages defined as appropriate for the nonlinear model.

## 5.2 Resampling cases

The “resampling cases” paradigm takes the point of view that cases,  $(y, \mathbf{x})$ , are sampled from a multivariate distribution  $F(y, \mathbf{x})$ . Estimate the distribution of  $(y, \mathbf{x})$  by resampling cases, defined to be the multivariate vectors  $(y, \mathbf{x})$ . The linear regression model lies outside this paradigm—using bootstrap methods in regression by resampling cases essentially makes no assumption about whether or not the linear regression model holds. The linear regression model only arises in this setting by going on to specify the conditional mean and variance of  $y|\mathbf{x}$  as the mean function (1) and variance function (2). Then, it is also necessary to include each case’s weight when resampling, and the cases to be resampled can now be written  $(y, \mathbf{x}, w)$ .

To estimate the distribution of  $(y, \mathbf{x})$  by resampling cases sample from the integers running from 1 to  $n$  to get bootstrap indices

$$j_i^* \text{ randomly sampled with replacement from } (1, 2, \dots, n)$$

Assuming that  $(y, \mathbf{x})$  are sampled from a multivariate distribution  $F(y, \mathbf{x})$ , derive bootstrap cases

$$(y_i^*, \mathbf{x}_i^*, w_i^*) = (y_{j_i^*}, \mathbf{x}_{j_i^*}, w_{j_i^*})$$

Compute the usual weighted least squares estimate of  $\boldsymbol{\eta}$  from this bootstrap sample, and repeat  $B$  times.

With the additional step from a joint distribution to conditional distributions, this resampling method can be used in all the regression models described here. For nonlinear and generalized nonlinear models, however, resampling cases can cause problems because the estimates of parameters can be very different when the design changes. This may lead to convergence problems when trying to find the estimates for the bootstrap samples.

### 5.3 Parametric bootstrap

In a linear regression problem, suppose that  $\hat{E}(y|\mathbf{x})$  are the fitted values and that  $\hat{\sigma}^2$  is the estimated residual variance. Bootstrap samples are obtained by computing  $y^* = \hat{E}(y|\mathbf{x}) + e^*$ , where  $e^*$  is a random sample from the normal  $N(0, \hat{\sigma}^2)$  distribution. This is called a parametric bootstrap because it makes use of the parametric form of the distribution of the residuals.

An advantage of the parametric bootstrap is that it can be applied in virtually any regression problem. For example, suppose we have a logistic regression problem. Suppose that  $\hat{\pi}$  gives the estimated probabilities of success for each of the  $n$  cases in the data, and that the vector  $m$  gives the number of trials for each of the  $m$  cases. We can generate bootstrap samples by obtaining  $y^*$  as binomial random numbers with trials given by  $m$  and probabilities of success given by  $\hat{\pi}$ .

A prime disadvantage of this method is that it relies on a parametric assumption that need not hold, and therefore it is likely to be less accurate than other bootstrap methods in some problems.

### 5.4 Numerical problems

It is possible for case resampling to run into computational problems. In particular, given the numerical tolerances computers must work within, it is possible (albeit unlikely) for a bootstrap sample to produce a model with lower rank than the model for the original sample, particularly when doing case resampling. The Arc bootstrap code does not currently check for this, and if it does happen, an unhelpful error message is returned and the bootstrap routine terminates. Future work will hopefully address this issue so that any lower rank bootstrap samples are ignored, allowing the routine to be completed.

## 6 Hypothesis tests for regression coefficients

Consider a hypothesis test with null hypothesis, NH, and test statistic,  $T$ . The  $p$ -value for this hypothesis test can be defined as the probability that a random variable with the

same distribution as that of  $T$  when NH is true is as large or larger (in absolute value) than the observed value of  $T$ .

This suggests the following method for estimating a  $p$ -value using bootstrap ideas: figure out a null distribution for the data under NH and generate  $B$  bootstrap values of  $T$  under this distribution. The  $p$ -value estimate is then the proportion of times that the bootstrap  $T^*$ 's are as large or larger (in absolute value) than the (original sample) observed value of  $T$ . To apply this in practice however, we need to choose both a suitable  $T$  and a null distribution for the data under NH.

An alternative strategy uses a *pivot statistic* for  $T$ . Pivot statistics have distributions that do not depend on coefficient values, so there is no need to figure out a null distribution for the data under NH—we can just carry out the simulation under the alternative hypothesis, AH. Again, the  $p$ -value estimate is the proportion of times that the bootstrap  $T^*$ 's are as large or larger (in absolute value) than the (original sample) observed value of  $T$ .

## 6.1 Testing a single regression coefficient

The usual normal theory method for testing a single regression coefficient is a  $t$ -test of the Studentized coefficient estimate.

As an alternative, one way to test a single regression coefficient using the pivot method outlined above is to use the pivot

$$T = \frac{\hat{\eta}_k - \eta_k}{\text{se}(\hat{\eta}_k)}$$

to test the  $k$ -th coefficient ( $k = 0, \dots, p - 1$ ). Using the “ $\hat{\eta}^*$  is to  $\hat{\eta}$  as  $\hat{\eta}$  is to  $\eta$ ” idea mentioned in Section 5,

$$T^* = \frac{\hat{\eta}_k^* - \hat{\eta}_k}{\text{se}(\hat{\eta}_k^*)}$$

The (two-tailed)  $p$ -value estimate for  $\eta_k = 0$  is then the proportion of times that the bootstrap  $T^*$ 's are as large or larger (in absolute value) than the (original sample) observed value of  $T$ ,

$$T_0 = \frac{\hat{\eta}_k}{\text{se}(\hat{\eta}_k)}$$

These  $p$ -values are presented in the bootstrap output. Similar methodology is used for generalized linear, nonlinear and generalized nonlinear models.

## 6.2 Testing a subset of mean function coefficients

This uses special code that is available only for linear models. Consider the following two models stated as hypotheses:

$$\begin{aligned} \text{NH: } E(y|\mathbf{x}) &= \boldsymbol{\eta}_1^T \mathbf{u}_1 \text{ with } \text{var}(y|\mathbf{x}) = \sigma^2/w \\ \text{AH: } E(y|\mathbf{x}) &= \boldsymbol{\eta}_1^T \mathbf{u}_1 + \boldsymbol{\eta}_2^T \mathbf{u}_2 \text{ with } \text{var}(y|\mathbf{x}) = \sigma^2/w \end{aligned}$$

where the vectors of predictors and mean function coefficients have been similarly partitioned into  $\mathbf{u}^T = (\mathbf{u}_1, \mathbf{u}_2)^T$  and  $\boldsymbol{\eta}^T = (\boldsymbol{\eta}_1, \boldsymbol{\eta}_2)^T$ . Let  $\mathbf{U}_1$  be the matrix consisting of the  $n$  sample row vectors  $\mathbf{u}_1^T$  stacked on top of each other. Let  $\mathbf{U}_2$  be defined similarly.

Testing these hypotheses is equivalent to testing  $\boldsymbol{\eta}_2 = \mathbf{0}$  and the usual normal theory method for this is an  $F$ -test of the ratio

$$F = \frac{(\text{RSS}_{\text{NH}} - \text{RSS}_{\text{AH}})/(\text{df}_{\text{NH}} - \text{df}_{\text{AH}})}{\hat{\sigma}^2}$$

where RSS indicates residual sum of squares from the model fit and  $\hat{\sigma}^2$  is the estimate of  $\sigma^2$  from the fit of the AH model.

As an alternative, one way to test  $\boldsymbol{\eta}_2 = \mathbf{0}$  using the pivot method outlined above is to choose the pivot

$$T = \frac{(\hat{\boldsymbol{\eta}}_2 - \boldsymbol{\eta}_2)^T (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1}) (\hat{\boldsymbol{\eta}}_2 - \boldsymbol{\eta}_2)}{\hat{\sigma}^2}$$

where  $\mathbf{U}_{2.1} = (\mathbf{I} - \mathbf{U}_1(\mathbf{U}_1^T \mathbf{U}_1)^{-1} \mathbf{U}_1^T) \mathbf{U}_2$ , the part of  $\mathbf{U}_2$  with the linear effect of  $\mathbf{U}_1$  subtracted out. Note that the numerator of  $T$  is just another way of writing  $\text{RSS}_{\text{NH}} - \text{RSS}_{\text{AH}}$ .

Using the “ $\hat{\boldsymbol{\eta}}^*$  is to  $\hat{\boldsymbol{\eta}}$  as  $\hat{\boldsymbol{\eta}}$  is to  $\boldsymbol{\eta}$ ” idea mentioned in Section 5,

$$T^* = \frac{(\hat{\boldsymbol{\eta}}_2^* - \hat{\boldsymbol{\eta}}_2)^T (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1}) (\hat{\boldsymbol{\eta}}_2^* - \hat{\boldsymbol{\eta}}_2)}{\hat{\sigma}^{*2}}$$

The  $p$ -value estimate is then the proportion of times that the bootstrap  $T^*$ 's are as large or larger than the (original sample) observed value of  $T$ ,

$$T_0 = \frac{\hat{\boldsymbol{\eta}}_2^T (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1}) \hat{\boldsymbol{\eta}}_2}{\hat{\sigma}^2}$$

To actually compute  $T^*$  and  $T_0$ , note that

$$\begin{aligned} \hat{\boldsymbol{\eta}}_2 &= (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1})^{-1} \mathbf{U}_{2.1}^T \mathbf{y} \\ \hat{\boldsymbol{\eta}}_2^* &= (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1})^{-1} \mathbf{U}_{2.1}^T \mathbf{y}^* \\ \hat{\boldsymbol{\eta}}_2^* - \hat{\boldsymbol{\eta}}_2 &= (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1})^{-1} \mathbf{U}_{2.1}^T (\mathbf{y}^* - \mathbf{y}) \end{aligned}$$

where  $\mathbf{y}$  is the vector of responses. Thus

$$\begin{aligned} T^* &= \frac{(\mathbf{y}^* - \mathbf{y})^T \mathbf{U}_{2.1} (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1})^{-1} \mathbf{U}_{2.1}^T (\mathbf{y}^* - \mathbf{y})}{\hat{\sigma}^{*2}} \\ T_0 &= \frac{\mathbf{y}^T \mathbf{U}_{2.1} (\mathbf{U}_{2.1}^T \mathbf{U}_{2.1})^{-1} \mathbf{U}_{2.1}^T \mathbf{y}}{\hat{\sigma}^2} \end{aligned}$$

$T^*$  and  $T_0$  can therefore be computed with full model (AH) simulation only, and without the need to actually calculate the bootstrap coefficient estimates.

This procedure can be performed in Arc only using residual resampling, and then only for linear models. This is independent of the “Bootstrap” menu item. Here is an outline of the method:

- Fit the “Full model”. Suppose it is called L1.
- Fit the “subset model.” Suppose it is called L2.

- Type the command

```
> (send L1 :sigtest-boot-pvalue L2 :nboots 999)
```

The program will return the bootstrap significance level corresponding to the  $F$ -test for comparing the two models.

## 7 Generalizations

Some other areas in linear regression for which bootstrap methods have potential include prediction and variable selection. Also, the methods discussed in this report can be extended to work with generalized linear models and nonlinear models. The computer routines in `boot.lisp` can provide a starting point for these other methods.

## References

- [1] Cook, R.D., and Weisberg, S. (1999), *Applied Regression Including Computing and Graphics*, New York: Wiley.
- [2] Davison, A.C., and Hinkley, D.V. (1997), *Bootstrap Methods and their Application*, New York: Cambridge University Press.
- [3] Efron, B., and Tibshirani, R.J. (1993), *An Introduction to the Bootstrap*, New York: Chapman and Hall.
- [4] Thisted, R. (1988), *Elements of Statistical Computing: Numerical Computation*. London: Chapman & Hall.
- [5] Tierney, L. (1990), *Lisp-Stat: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*, New York: Wiley.