Sutton said, "that's where the money is". Working for a startup is really hard. The best part is that there are no rules. The worst part, so far, is that we have no revenues, although we're moving nicely to change this. Life is fun and it's a great time to be doing statistical graphics.

I'm looking forward to seeing everyone in NYC this August. If you don't see me, it means that I may have failed to solve our revenue problem. Please consider getting involved with section activities. The health of our section, one of the largest in ASA, depends on volunteers.

Stephen G. Eick, Ph.D.
Co-founder and CTO Visintuit
eick@visintuit.com
630-778-0050

TOPICS IN STATISTICAL COMPUTING

An Introduction to the Bootstrap with Applications in R

A. C. Davison and **Diego Kuonen**

kuonen@statoo.com

Introduction

Bootstrap methods are resampling techniques for assessing uncertainty. They are useful when inference is to be based on a complex procedure for which theoretical results are unavailable or not useful for the sample sizes met in practice, where a standard model is suspect but it is unclear with what to replace it, or where a 'quick and dirty' answer is required. They can also be used to verify the usefulness of standard approximations for parametric models, and to improve them if they seem to give inadequate inferences. This article, a brief introduction on their use, is based closely on parts of Davison and Hinkley (1997), where further details and many examples and practicals can be found. A different point of view is given by Efron and Tibshirani (1993) and a more mathematical survey by Shao and Tu (1995), while Hall (1992) describes the underlying theory.

Basic Ideas

The simplest setting is when the observed data $y_1, \ldots, y_n$ are treated as a realisation of a random sample $Y_1, \ldots, Y_n$ from an unknown underlying distribution F. Interest is focused on a parameter θ, the outcome of applying the statistical functional $t(\cdot)$ to F, so $\theta = t(F)$. The simplest example of such a functional is the average, $t(F) = \int y\, dF(y)$; in general we think of $t(\cdot)$ as an algorithm to be applied to F.

The estimate of θ is $t = t(\hat{F})$, where $\hat{F}$ is an estimate of F based on the data $y_1, \ldots, y_n$. This might be a parametric model such as the normal, with parameters estimated by maximum likelihood or a more robust method, or the empirical distribution function (EDF) $\hat{F}$, which puts mass n^{-1} on each of the y_j. If partial information is available about F, it may be injected into $\hat{F}$. However $\hat{F}$ is obtained, our estimate t is simply the result of applying the algorithm $t(\cdot)$ to $\hat{F}$.

Typical issues now to be addressed are: what are bias and variance estimates for t? What is a reliable confidence interval for θ? Is a certain hypothesis consistent with the data? Hypothesis tests raise the issue of how the null hypothesis should be imposed, and are discussed in detail in Chapter 4 of Davison and Hinkley (1997). Here we focus on confidence intervals, which are reviewed in DiCiccio and Efron (1996), Davison and Hinkley (1997, Chapter 5) and Carpenter and Bithell (2000).

Confidence Intervals

The simplest approach to confidence interval construction uses normal approximation to the distribution of T, the random variable of which t is the observed value. If the true bias and variance of T are

$$\begin{aligned} b(F) &= \mathrm{E}(T \mid F) - \theta = \mathrm{E}(T \mid F) - t(F), \quad (1) \\ v(F) &= \mathrm{var}(T \mid F), \end{aligned}$$

then we might hope that in large samples

$$Z = \frac{T - \theta - b(F)}{v(F)^{1/2}} \dot{\sim} N(0,1);$$

the conditioning in (1) indicates that T is based on a random sample $Y_1, \ldots, Y_n$ from F. In this case an approximate $(1 - 2\alpha)$ confidence interval for θ is

$$t - b(F) - z_{1-\alpha} v(F)^{1/2}, \quad t - b(F) - z_\alpha v(F)^{1/2}, \quad (2)$$

where z_α is the α quantile of the standard normal distribution. The adequacy of (2) depends on F, n, and T and cannot be taken for granted.

As it stands (2) is useless, because it depends on the unknown F. A key idea, sometimes called the *bootstrap* or *plug-in principle*, is to replace the unknown F with its known estimate $\hat{F}$, giving bias and variance estimates $b(\hat{F})$ and $v(\hat{F})$. For all but the simplest estimators T these cannot be obtained analytically and so

simulation is used. We generate R independent bootstrap samples $Y_1^*, \ldots, Y_n^*$ by sampling independently from $\hat{F}$, compute the corresponding estimator random variables $T_1^*, \ldots, T_R^*$, and then hope that

$$b(F) \doteq b(\hat{F}) = \mathrm{E}(T \mid \hat{F}) - t(\hat{F}) \quad (3)$$

$$\doteq R^{-1} \sum_{r=1}^{R} T_r^* - t = \bar{T}^* - t, \quad (4)$$

$$v(F) \doteq v(\hat{F}) = \mathrm{var}(T \mid \hat{F}) \quad (5)$$

$$\doteq \frac{1}{R-1} \sum_{r=1}^{R} (T_r^* - \bar{T}^*)^2. \quad (6)$$

There are two errors here: statistical error due to replacement of F by $\hat{F}$, and simulation error from replacement of expectation and variance by averages. Evidently we must choose R large enough to make the second of these errors small relative to the first, and if possible use $b(\hat{F})$ and $v(\hat{F})$ in such a way that the statistical error, unavoidable in most situations, is minimized. This means using approximate pivots where possible.

If the normal approximation leading to (2) fails because the distribution of $T - \theta$ is not close to normal, an alternative approach to setting confidence intervals may be based on $T - \theta$. The idea is that if $T^* - t$ and $T - \theta$ have roughly the same distribution, then quantiles of the second may be estimated by simulating those of the first, giving $(1 - 2\alpha)$ *basic bootstrap* confidence limits

$$t - (T^*_{((R+1)(1-\alpha))} - t), \quad t - (T^*_{((R+1)\alpha)} - t),$$

where$T^*_{(1)} < \cdots < T^*_{(R)}$are the sorted T_r^*'s. When an approximate variance V for T is available and can be calculated from $Y_1, \ldots, Y_n$, *studentized bootstrap* confidence intervals may be based on $Z = (T - \theta)/V^{1/2}$, whose quantiles are estimated from simulated values of the corresponding bootstrap quantity $Z^* = (T^* - t)/V^{*1/2}$. This is justified by Edgeworth expansion arguments valid for many but not all statistics (Hall, 1992).

Unlike the intervals mentioned above, *percentile* and *bias-corrected adjusted* (BCa) intervals have the attractive property of invariance to transformations of the parameters. The percentile intervals with level $(1 - 2\alpha)$ is $(T^*_{((R+1)\alpha)}, T^*_{((R+1)(1-\alpha))})$, while the BCa interval has form $(T^*_{((R+1)\alpha')}, T^*_{((R+1)(1-\alpha''))})$, with α' and α'' cleverly chosen to improve the properties of the interval. DiCiccio and Efron (1996) describe the reasoning underlying these intervals and their developments.

The BCa and studentized intervals are second-order accurate. Numerical comparisons suggest that both tend to undercover, so the true probability that a 0.95 interval contains the true parameter is smaller than 0.95, and that BCa intervals are shorter than studentized ones, so they undercover by slightly more.

Bootstrapping in R

R (Ihaka and Gentleman, 1996) is a language and environment for statistical computing and graphics. Additional details can be found at `www.r-project.org`. The two main packages for bootstrapping in `R` are `boot` and `bootstrap`. Both are available on the 'Comprehensive R Archive Network' (CRAN, `cran.r-project.org`) and accompany Davison and Hinkley (1997) and Efron and Tibshirani (1993) respectively. The package `boot`, written by Angelo Canty for use within `S-Plus`, was ported to R by Brian Ripley and is much more comprehensive than any of the current alternatives, including methods that the others do not include. After downloading the package from CRAN and installing the package, one simply has to type
`require(boot)`
at the R prompt. Note that the installation could also performed within R by means of
`install.packages(boot)` A good starting point is to carefully read the documentations of the R functions `boot` and `boot.ci`
`?boot`
`?boot.ci`
and to try out one of the examples given in the 'Examples' section of the corresponding help file. In what follows we illustrate their use.

Example

Figure 1shows data from an experiment in which two laser treatments were randomized to eyes on patients. The response is visual acuity, measured by the number of letters correctly identified in a standard eye test. Some patients had only one suitable eye, and they received one treatment allocated at random. There are 20 patients with paired data and 20 patients for whom just one observation is available, so we have a mixture of paired comparison and two-sample data.

```
blue <- c(4,69,87,35,39,79,31,79,65,95,68,
          62,70,80,84,79,66,75,59,77,36,86,
          39,85,74,72,69,85,85,72)
red <-c(62,80,82,83,0,81,28,69,48,90,63,
        77,0,55,83,85,54,72,58,68,88,83,78,
        30,58,45,78,64,87,65)
acui<-data.frame(str=c(rep(0,20),
           rep(1,10)),red,blue)
```

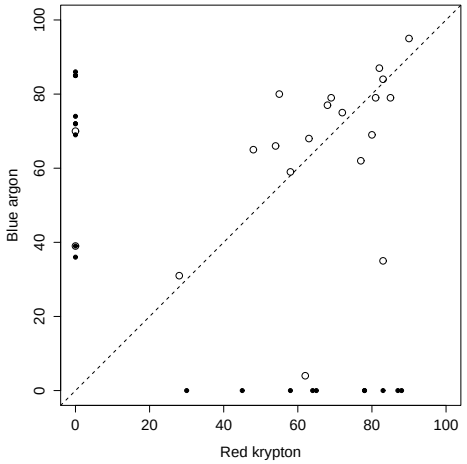

Figure 1:Paired (circles) and unpaired data (small blobs).

We denote the fully observed pairs $y_j = (r_j, b_j)$, the responses for the eyes treated with red and blue treatments, and for these n_d patients we let $d_j = b_j - r_j$. Individuals with just one observation give data $y_j = (?, b_j)$ or $y_j = (r_j, ?)$; there are n_b and n_r of these. The unknown variances of the d's, r's and b's are σ_d^2, σ_r^2 and σ_b^2.

For illustration purposes, we will perform a standard analysis for each. First, we could only consider the paired data and construct the classical Student-t 0.95 confidence interval for the mean of the differences, of form $\bar{d} \pm t_{n-1}(0.025)s_d/n_d^{1/2}$, where $\bar{d} = 3.25$, s_d is the standard deviation of the d's and $t_{n-1}(0.025)$ is the quantile of the appropriate t distribution. This can be done in R by means of

```
> acu.pd <- acui[acui$str==0,]
> dif <- acu.pd$blue-acu.pd$red
> n <- nrow(acu.pd)
>tmp<-qt(0.025,n-1)*sd(dif)/sqrt(n)
> c(mean(dif)+tmp, mean(dif)-tmp)
[1] -9.270335 15.770335
```

But a Q-Q plot of the differences looks more Cauchy than normal, so the usual model might be thought unreliable. The bootstrap can help to check this. To perform a nonparametric bootstrap in this case we first need to define the *bootstrap function*, corresponding to the algorithm $t(\cdot)$:

```
acu.pd.fun <- function(data, i){
 d <- data[i,]
 dif <- d$blue-d$red
 c(mean(dif), var(dif)/nrow(d)) }
```

A set of $R = 999$ bootstrap replicates can then be easily obtained with `acu.pd.b<-boot(acu.pd,acu.pd.fun,R=999)` The resulting nonparametric 0.95 bootstrap confidence intervals can be calculated as shown previously or using directly

```
> boot.ci(acu.pd.b,
 type=c("norm","basic","stud"))
...
Normal          Basic             Studentized
(-8.20,14.95) (-8.10,15.05)   (-8.66,15.77)
```

The normal Q–Q plot of the $R = 999$ replicates in the left panel of Figure 2underlines the fact that the Student-t and the bootstrap intervals are essentially equal.

An alternative is to consider only the two-sample data and compare the means of the two populations issuing from the patients for whom just one observation is available, namely

```
acu.ts<- acui[acui$str==1,]
```

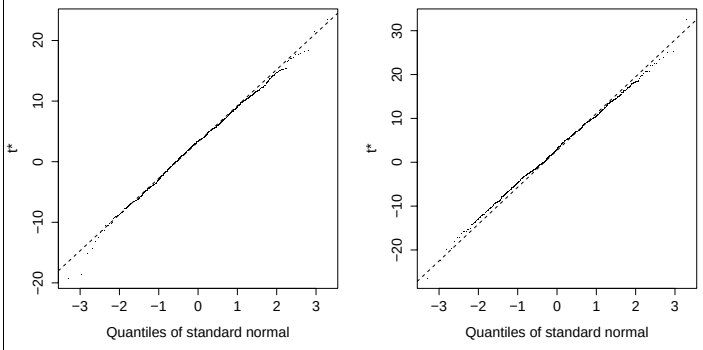

Figure 2:Normal Q–Q plots of bootstrap estimate t^*.
Left: for the paired analysis.
Right: for the two-sample analysis.

The classical normal 0.95 confidence interval for the difference of the means is $(\bar{b} - \bar{r}) \pm z_{0.025}(s_b^2/n_b + s_r^2/n_r)^{1/2}$, where s_b and s_r are the standard deviations of the b's and r's, and $z_{0.025}$ is the 0.025 quantile of the standard normal distribution.

```
>acu.ts <- acui[acui$str==1,]
>dif <- mean(acu.ts$blue)-mean(acu.ts$red)
>tmp <- qnorm(0.025)*
     sqrt(var(acu.ts$blue)/nrow(acu.ts)+
     var(acu.ts$red)/nrow(acu.ts))
> c(dif+tmp, dif-tmp)
[1] -13.76901  19.16901
```

The obvious estimator and its estimated variance are

$$t = \bar{b} - \bar{r}, \quad v = s_b^2/n_b + s_r^2/n_r,$$

whose values for these data are 2.7 and 70.6. To construct bootstrap confidence intervals we generate

$R = 999$ replicates of t and v, with each simulated dataset containing n_b values sampled with replacement from the bs and n_r values sampled with replacement from the rs. In R :

```
y<-c(acui$blue[21:30],acui$red[21:30])
acu<-data.frame(col=rep(c(1,2),c(10,10)),y)
acu.ts.f <- function(data, i){
d <- data[i,]
m <- mean(d$y[1:10])-mean(d$y[11:20])
v <- var(d$y[1:10])/10+var(d$y[11:20])/10
c(m, v) }
acu.ts.boot<-boot(acu,acu.ts.f,R=999,
              strata=acu$col)
```

Here `strata=acu$col` ensures stratified simulation. The Q–Q plot of these 999 values in the right panel of Figure 2 is close to normal, and the bootstrap intervals computed using `boot.ci` differ little from the classical normal interval.

We now combine the analyses, hoping that the resulting confidence interval will be shorter. If the variances σ_d^2, σ_r^2 and σ_b^2 of the ds, rs and bs were known, a minimum variance unbiased estimate of the difference between responses for blue and red treatments would be

$$\frac{n_d\bar{d}/\sigma_d^2 + (\bar{b}-\bar{r})/(\sigma_b^2/n_b + \sigma_r^2/n_r)}{n_d/\sigma_d^2 + 1/(\sigma_b^2/n_b + \sigma_r^2/n_r)}.$$

As σ_d^2, σ_r^2 and σ_b^2 are unknown, we replace them by estimates, giving estimated treatment difference and its variance

$$\begin{aligned} t &= \frac{n_d\bar{d}/\hat{\sigma}_d^2 + (\bar{b}-\bar{r})/(\hat{\sigma}_b^2/n_b + \hat{\sigma}_r^2/n_r)}{n_d/\hat{\sigma}_d^2 + 1/(\hat{\sigma}_b^2/n_b + \hat{\sigma}_r^2/n_r)}, \\ v &= \left\{n_d/\hat{\sigma}_d^2 + 1/(\hat{\sigma}_b^2/n_b + \hat{\sigma}_r^2/n_r)\right\}^{-1}. \end{aligned}$$

Here $t = 3.07$ and $v = 4.873^2$, so a naive 0.95 confidence interval for the treatment difference is $(-6.48, 12.62)$.

One way to apply the bootstrap here is to generate a bootstrap dataset by taking n_d pairs randomly with replacement from $\hat{F}_y$, n_b values with replacement from $\hat{F}_b$ and n_r values with replacement from $\hat{F}_r$, each resample being taken with equal probability:

```
acu.f <- function(data, i){
d <- data[i,]
m <- sum(data$str)
if(length(unique((i)==(1:nrow(data))))!=1){
d$blue[d$str==1]<-sample(d$blue,size=m,T)
d$red[d$str==1]<-sample(d$red,size=m,T)}
dif<- d$blue[d$str==0]-d$red[d$str==0]
d2 <- d$blue[d$str==1]
d3 <- d$red[d$str==1]
v1 <- var(dif)/length(dif)
v2 <-var(d2)/length(d2)+var(d3)/length(d3)
v <- 1/(1/v1+1/v2)
c((mean(dif)/v1+(mean(d2)-mean(d3))/v2)*v,v)}
acu.b<-boot(acui,acu.f,R=999,strata=acui$str)
boot.ci(acu.b,type=c("norm","basic","stud",
    "perc","bca"))
```

giving all five sets of confidence limits. The interested reader can continue the analysis.

Regression

A linear regression model has form $y_j = x_j^{\mathrm{T}}\beta + \varepsilon_j$, where the (y_j, x_j) are the response and the $p \times 1$ vector of covariates for the jth response y_j. We are usually interested in confidence intervals for the parameters, the choice of covariates, or prediction of the future response y_+ at a new covariate x_+. The two basic resampling schemes for regression models are

- *resampling cases* $(y_1, x_1), \ldots, (y_n, x_n)$, under which the bootstrap data are

  $$(y_1, x_1)^*, \ldots, (y_n, x_n)^*,$$

  taken independently with equal probabilities n^{-1} from the (y_j, x_j), and
- *resampling residuals*. Having obtained fitted values $x_j^T\hat{\beta}$, we take ε_j^* randomly from centred standardized residuals $e_1, \ldots, e_n$ and set

  $$y_j^* = x_j^T\hat{\beta} + \varepsilon_j^*, \quad j = 1, \ldots, n.$$

Under case resampling the resampled design matrix does not equal the original one. For moderately large data sets this doesn't matter, but it can be worth bearing in mind if n is small or if a few observations have a strong influence on some aspect of the design. If the wrong model is fitted and this scheme is used we get an appropriate measure of uncertainty, so case resampling is in this sense robust. The second scheme is more efficient than resampling pairs if the model is correct, but is not robust to getting the wrong model, so careful model-checking is needed before it can be used. Either scheme can be stratified if the data are inhomogeneous. In the most extreme form of stratification the strata consist of just one residual; this is the *wild bootstrap*, used in non-parametric regressions.

Variants of residual resampling needed for generalized linear models, survival data and so forth are all constructed essentially by looking for the exchangeable aspects of the model, estimating them, and then resampling them. Similar ideas also apply to time series models such as ARMA processes. Additional examples and further details can be found in Davison and Hinkley

(1997, Chapters 6–8). We now illustrate case and residual resampling.

The `survival` data (Efron, 1988) are survival percentages for rats at a succession of doses of radiation, with two or three replicates at each dose; see Figure 3. The data come with the package `boot` and can be loaded using

```
> data(survival)
```

To have a look at the data, simply type `survival` at the R prompt. The theoretical relationship between survival rate (`surv`) and dose (`dose`) is exponential, so linear regression applies to

$$x = \texttt{dose}, \quad y = \log(\texttt{surv}).$$

There is a clear outlier, case 13, at $x = 1410$. The least squares estimate of slope is -59×10^{-4} using all the data, changing to -78×10^{-4} with standard error 5.4×10^{-4} when case 13 is omitted.

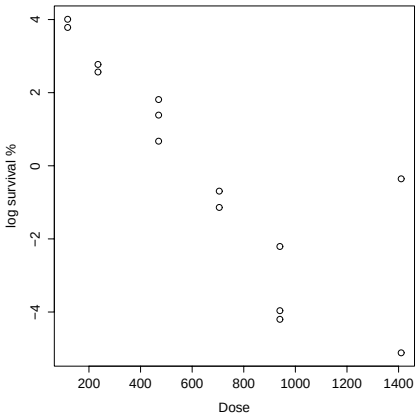

Figure 3:Scatter plot of `survival` data.

To illustrate the potential effect of an outlier in regression we resample cases, using

```
 surv.fun <- function(data, i){
  d <- data[i,]
  d.reg <- lm(log(d$surv)~d$dose)
  c(coef(d.reg))    }
surv.boot<-boot(survival,surv.fun,R=999)
```

The effect of the outlier on the resampled estimates is shown in Figure 4, a histogram of the $R = 999$ bootstrap least squares slopes $\hat{\beta}_1^*$. The two groups of bootstrapped slopes correspond to resamples in which case 13 does not occur and to samples where it occurs once or more. The resampling standard error of $\hat{\beta}_1^*$ is 15.6×10^{-4}, but only 7.8×10^{-4} for samples without case 13.

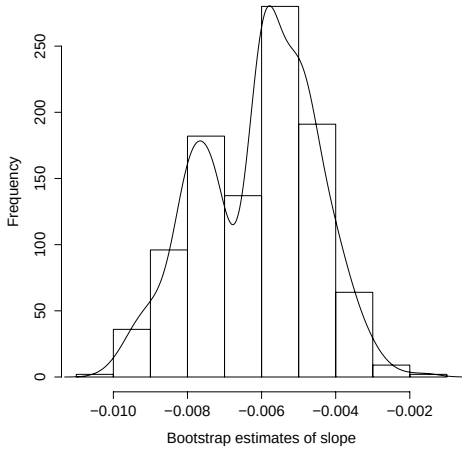

Figure 4: Histogram of bootstrap estimates of slope $\hat{\beta}_1^*$ with superposed kernel density estimate.

A jackknife-after-bootstrap plot (Efron, 1992; Davison and Hinkley, 1997, Section 3.10.1) shows the effect on $T^* - t$ of resampling from datasets from which each of the observations has been removed. Here we expect deletion of case 13 to have a strong effect, and Figure 5obtained through

```
> jack.after.boot(surv.boot, index=2)
```

shows clearly that this case has an appreciable effect on the resampling distribution, and that its omission would give much tighter confidence limits on the slope.

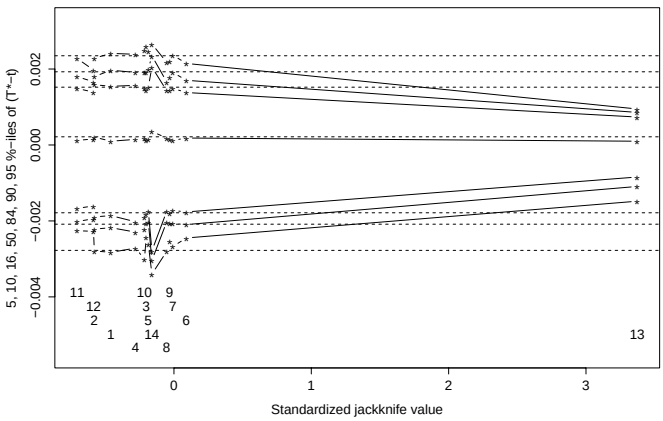

Figure 5: Jackknife-after-bootstrap plot for the slope. The vertical axis shows quantiles of $T^* - t$ for the full sample (horizontal dotted lines) and without each observation in turn, plotted against the influence value for that observation.

The effect of this outlier on the intercept and slope when resampling residuals can be assessed using

`sim=parametric` in the `boot` call. The required R code is:

```
fit <-lm(log(survival$surv)~survival$dose)
res <- resid(fit)
f <- fitted(fit)
surv.r.mle<- data.frame(f,res)
surv.r.fun<-function(data)
       coef(lm(log(data$surv)~data$dose))
surv.r.sim <- function(data, mle){
data$surv<-exp(mle$f+sample(mle$res,T))
            data                    }
surv.r.boot<- boot(survival,surv.r.fun,
              R=999,sim="parametric",
     ran.gen=surv.r.sim,mle=surv.r.mle)
```

Having understood what this code does, the interested reader may use it to continue the analysis.

Discussion

Bootstrap resampling allows empirical assessment of standard approximations, and may indicate ways to fix them when they fail. The computer time involved is typically negligible — the resampling for this article took far less than the time needed to examine the data, devise plots and summary statistics, and to code (and check) the simulations.

Bootstrap methods offer considerable potential for modelling in complex problems, not least because they enable the choice of estimator to be separated from the assumptions under which its properties are to be assessed. In principle the estimator chosen should be appropriate to the model used, or there is a loss of efficiency. In practice, however, there is often some doubt about the exact error structure, and a well-chosen resampling scheme can give inferences robust to precise assumptions about the data.

Although the bootstrap is sometimes touted as a replacement for 'traditional statistics', we believe this to be misguided. It is unwise to use a powerful tool without understanding why it works, and the bootstrap rests on 'traditional' ideas, even if their implementation via simulation is not 'traditional'. Populations, parameters, samples, sampling variation, pivots and confidence limits are fundamental statistical notions, and it does no-one a service to brush them under the carpet. Indeed, it is harmful to pretend that mere computation can replace thought about central issues such as the structure of a problem, the type of answer required, the sampling design and data quality. Moreover, as with any simulation experiment, it is essential to monitor the output to ensure that no unanticipated complications have arisen and to check that the results make sense, and this entails understanding how the output will be used. Never forget: *the aim of computing is insight, not numbers*; *garbage in, garbage out*.

References

Carpenter, J. and Bithell, J. (2000). Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Statistics in Medicine*, **19**, 1141–1164.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and their Application*. Cambridge University Press.
(`statwww.epfl.ch/davison/BMA/`)

DiCiccio, T. J. and Efron, B. (1996). Bootstrap confidence intervals (with Discussion). *Statistical Science*, **11**, 189–228.

Efron, B. (1988). Computer-intensive methods in statistical regression. *SIAM Review*, **30**, 421–449.

Efron, B. (1992). Jackknife-after-bootstrap standard errors and influence functions (with Discussion). *Journal of the Royal Statistical Society series B*, **54**, 83–127.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. New York: Chapman & Hall.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*. New York: Springer-Verlag.

Ihaka, R. and Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5**, 299–314.

Shao, J. and Tu, D. (1995). *The Jackknife and Bootstrap*. New York: Springer-Verlag.

SOFTWARE PACKAGES

GGobi

Deborah F. Swayne, AT&T Labs – Research

`dfs@research.att.com`

GGobi is a new interactive and dynamic software system for data visualization, the result of a significant redesign of the older XGobi system (Swayne, Cook and Buja, 1992; Swayne, Cook and Buja, 1998), whose development spanned roughly the past decade. GGobi differs from XGobi in many ways, and it is those differences that explain best why we have undertaken this redesign.