CLASSROOM SIMULATION: THE MARGIN OF ERROR IN A PUBLIC OPINION POLL

Bruce E. TRUMBO, Eric A. SUESS, and Shuhei OKUMURA; *California State University, East Bay*.
Department of Statistics; Hayward Hills Campus, California State University, East Bay; Hayward, CA 94542 USA
bruce.trumbo@csueastbay.edu or eric.suess@csueastbay.edu

Abstract: Appropriate simulations can be used effectively in a beginning statistics course to illustrate important principles—either before the underlying theory is accessible or along with a presentation of the theory. Here we use a very few fundamental functions in R to illustrate the margin of error of a public opinion poll. Polls using 25 and 2500 subjects from a population with a known proportion "in favor" are simulated repeatedly. Summaries of results illustrate that point estimates of the proportion in favor are variable and approximately normally distributed, and that errors are smaller for the larger number of subjects. Interval estimates are discussed. A slightly more advanced simulation illustrates that, during the process of a poll, the proportion of steps at which the running fraction in favor is higher than the population probability is much more likely to be near 0 or 1 than to be near half, following Feller's arcsine law (a "bathtub shaped" beta distribution).

Key Words: Public opinion poll, margin of error, simulation, Feller arcsine law, teaching, R.

1. Introduction. In this paper we use R statistical software to do some simple simulations and make graphical and numerical summaries of the results. R can be downloaded online easily and without charge [1], and it performs the operations we need here, including simulations based on a pseudorandom number generator of very high quality. While R is not as easy for beginners to use as point-and-click software, we have found that students have little difficulty learning what is needed for elementary simulations if some restraint is exercised in the complexity of the code and the number of functions used. We review statistical ideas briefly, mainly to establish notation. But we explain the most important R code in a little more detail, and complete documented code is available online [2].

We begin by using simulation to illustrate the margin of error in a pre-election public opinion poll. Consider a population in which the proportion $\pi = 0.53$ of the population favors Candidate A (has trait A) and the remainder has trait B (non-A). In practice, π would be unknown and we might wonder whether π is greater than one half.

Accounts of most public opinion polls include information about the margin of sampling error based on 95% confidence. Assume simple random sampling, a *population* proportion π between 0.4 and 0.6, and a number n of subjects large enough to justify use of the normal approximation. Then an approximate 95% confidence interval for π is $p \pm n^{-1/2}$. Here p denotes the *sample* proportion favoring A, computed as $p = X/n$, where X is the number of subjects favoring A. Thus, the margin of sampling error is $E = n^{-1/2}$. For example, $E = 0.02$ would be reported for a poll based on $n = 2500$ subjects, and $E = 0.04$ for $n = 625$. If $\pi = 0.53$, then 2500 subjects would likely be enough to detect that more than half of the population favors A, but 625 subjects might not be enough.

For values of π nearer 0 or 1, smaller values of E suffice. More generally, an approximate 95% interval is $p \pm 1.96[p(1-p)/n]^{1/2}$. Especially for smaller n, we can improve the accuracy of these approximations if we first increase n by 4 and X by 2 [3, 4]. Of course, none of these computations account for nonsampling errors such as nonresponse bias, misinterpreted questions, or dishonest answers.

2. A very small poll. We begin by simulating a poll with only $n = 25$ subjects. As an example, the R code

```
> sample(0:1, 25, rep=T, prob=c(.47, .53))
```

might return

```
 [1] 1 0 1 1 0 0 1 0 0 1 0 0 0
[13] 0 1 0 1 0 1 1 1 0 1 1 0
```

Here, the first argument `0:1` specifies that we will sample opinions 0 or 1, where 1 represents a respondent favoring A and 0 a respondent not willing to declare for A—maybe opposed, maybe undecided. [If j and k are integers with $j < k$, then `j:k` represents the vector $(j, j+1, j+2, ..., k)$; so `0:1` is the same as `c(0,1)`.] The second argument `25` is the number of "subjects" to be sampled, the third specifies sampling with replacement, and the last specifies that the proportions of 0s and 1s in the population are 0.47 and 0.53, respectively.

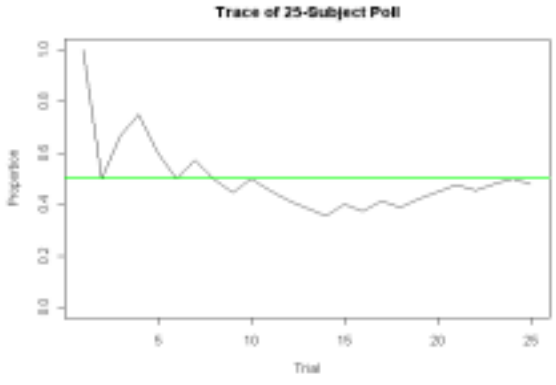

Figure 1

Copyright © 2004 by Bruce E. Trumbo. All rights reserved. DRAFT: Please email btrumbo@csuhayward.edu with suggestions or corrections.

```
n <- 25;  Trial <- 1:n;  p.ba <- c(.47, .53)
x <- sample(0:1, n, rep=T, prob=p.ba)
run.tot <- cumsum(x)
Proportion <- run.tot/Trial
plot(Trial, Proportion, type="l", ylim= c(0,1))
abline(h=.5, col="green", lwd=2)
```

Display 2

```
      Trial x run.tot Proportion
 [1,]     1 1       1      1.000
 [2,]     2 0       1      0.500
 [3,]     3 1       2      0.667
 [4,]     4 1       3      0.750
 [5,]     5 0       3      0.600
 [6,]     6 0       3      0.500
 [7,]     7 1       4      0.571
 [8,]     8 0       4      0.500
 [9,]     9 0       4      0.444
[10,]    10 1       5      0.500
```

Display 3

In the output, the brackets give the index of the first observation printed in each row: the 13th number in the output vector was 0. Of the 25 simulated "respondents" 12 favored A, a misleading result because $p = 12/25 < 1/2$ while $\pi > 1/2$. Because this is a simulation of a random process, the result will be different each time you execute the same command in the R console window.

In Figure 1, we graph the results of this simulation by plotting the *cumulative fraction* of the sample favoring A at each trial against the *trial number*. The first five cumulative fractions are 1/1 = 1, 1/2, 2/3, 3/4, and 3/5; the 25th is 12/25.

The R-script in Display 2 was used to make Figure 1. In it we used the assignment operator `<-` to establish a constant `n`, a vector `Trial` of the integers from 1 through 25, a vector `p.ba` of two probabilities, a vector `x` of 25 simulated poll results, and a vector `cum.tot` of cumulative sums of x. The vector `Proportion` consists of 25 fractions: Each element of `cum.tot` is divided by the corresponding element of `Trial`. (See Display 3.)

Two of the arguments of the plot function are `plot="l"` (with the letter *ell*) to plot lines instead of points (the default), and `ylim=c(0,1)` to force the y-axis to run from 0 to 1. Finally, the `abline` function plots the slightly thick, green horizontal reference line at $p = 1/2$.

Before going on, it is crucial for students to understand the process that leads to Figure 1. Some hand work may be in order. Depending on the level and size of the class, you might "poll" $n = 10$ or so students in sequence with A = Female, and fill in a table with n rows and with columns: Trial Number, F or M, 1 or 0, Running Sum, and Running Proportion. Alternatively, or in addition, you might append the following line of code to Display 1, which makes a similar table in R:

```
round(cbind(Trial, x, run.tot, Proportion), 3)
```

Here `cbind` "binds together" the four vectors used as arguments to make a 25-by-4 matrix, and `round` with second argument 3 rounds the running proportions to thee places. Display 3 shows the first five rows of the resulting output with the same simulation as in Figure 1.

In the classroom, the next step would be to run the simulation of Display 2 several times to see that the shape of the trace (as in Figure 1) is very different from one run to the next: There is no stability in the behavior of this small poll. Also, notice that the endpoints of the traces vary considerably. These endpoints are the point estimates p of the population proportion $\pi = 0.53$. Typically, the values of p will lie between 0.33 and 0.73. To save space here, we do not show multiple versions of Figure 1, but Figure 4 shows superimposed traces for 20 runs obtained from the code in Display 5.

In Display 5, the `set.seed` function ensures that the same 20 simulations occur again on each run of the code; omit it for a different set on each run. The `plot` function sets up the axes and labels, but plots nothing. Here we use parameters rather than variable names (the default) to label the axes. The `lines` function inside the loop plots one trace on these same axes for each of 20 passes through the loop. The code inside the loop is essentially the same as lines 2, 3, and 4 in Display 2.

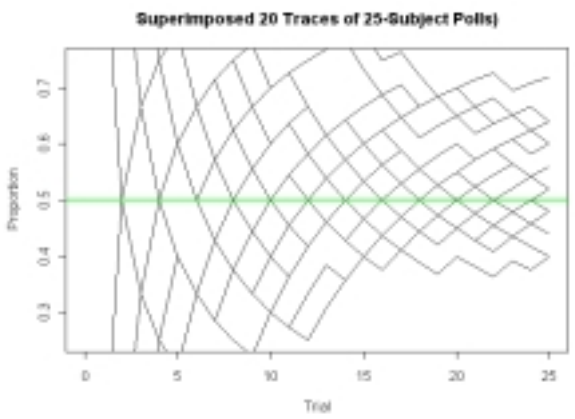

Figure 4

```
set.seed(13)
n <- 25;  m <- 20
plot(0, pch=" ",
   xlim=c(0,n), ylim=c(.25,.75),
     xlab="Trial", ylab="Proportion")
       # Empty plot for axes, labels

for (i in 1:m) {
   x <- cumsum(sample(0:1, n, rep=T,
      prob=c(.47,.53)))/(1:n)
   lines(1:n,x) }

abline(h=.5, col="green", lwd=2)
```

Display 5

Copyright © 2004 by Bruce E. Trumbo. All rights reserved. DRAFT: Please email btrumbo@csuhayward.edu with suggestions or corrections.

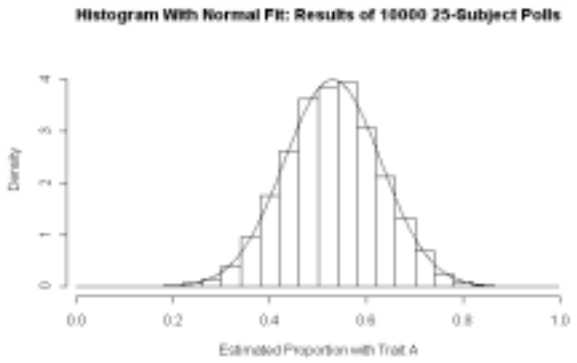

Figure 6

Figure 6 is a histogram of estimates p resulting from m = 10,000 simulated polls, each with n = 25 subjects. It shows that p is approximately normally distributed with E(p) = 0.53 and SD(p) = $[0.53(0.47)/25]^{1/2} \approx 0.10$. We see that $E \approx 0.20$. More than 3800 of the 10,000 estimates p have misleading values below 0.50. While they are so variable as to be useless, the values of p nevertheless follow a specific distribution.

The code for Figure 6 is shown in Display 7. Before the loop we use `numeric(m)` to define `Estimate`, a vector of `m` 0s. On the i^{th} pass through the loop, the i^{th} 0 is replaced by the estimate of π from the i^{th} simulated poll.

A complication in making the histogram is that, with $n = 25$, the m estimates p typically take only about 15 distinct values altogether. The default algorithm in R for choosing intervals works badly for such granular non-integer data. So we define intervals centered on possible values of p with `cutpts`, a sequence of values running from just above 0 to just below 1, and $1/n$ units apart. (For $n > 50$, we let the algorithm choose about 20 cutpoints.) Also, we use `ylim` to make the plotting window a little more than tall enough for the approximating normal density curve, and `prob=T` to put the histogram on a density scale matching that of the normal curve.

```
m <- 10000;  n <- 25;  pa <- .53
pb <- 1 - pa
sd <- sqrt(pa*pb/n)
Estimate <- numeric(m)

for (i in 1:m)  {
  Estimate[i] <- mean(sample(0:1, n,
    rep=T, prob=c(pb, pa))) }

cutpts <- seq(1/(2*n), 1, by=1/n)
if (n > 50) cutpts <- 20
hist(Estimate, breaks=cutpts,
  xlab="Estimated Proportion with Trait A",
    ylim=c(0, 0.45/sd),  prob=T)
zz <- seq(0, 1, length=1000)
zd <- dnorm(zz, pa, sd)
lines(zz, zd)
```

Display 7

3. A larger, more useful, poll. Now we look at simulated polls with n = 2500 subjects. Intuitively, with a larger sample we would expect to get better estimates of π — that is, estimates with a smaller margin of error. A program similar to that of Display 2 yields the path shown in Figure 8. (The important changes in the program are that `n <- 2500` and that the vertical plotting window is now set by the parameter `ylim <- c(.25, .75).`)

When the number of trials is still small, the trace seems quite erratic, but it becomes more stable at the right side of the plot, where the number of trials becomes large. If you repeat this simulation several times, you will see that the left side of the traces can be very different in shape from one run to the next. But the right side of each trace is about the same, settling down to very near $\pi = 0.53$ in each case.

Figure 8

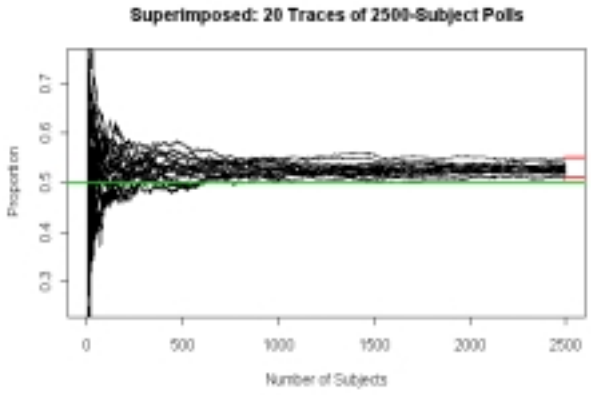

Figure 9

In Figure 9 we illustrate the tendency of such traces to stabilize, by showing 20 of them superimposed. (The code is similar to that of Display 5.) Figure 10 shows a close-up view of the right side of Figure 9. (To adjust the plotting window, we used `xlim=c(n-100, n)` and `ylim=c(.48, .57).`)

Copyright © 2004 by Bruce E. Trumbo. All rights reserved. DRAFT: Please email btrumbo@csuhayward.edu with suggestions or corrections.

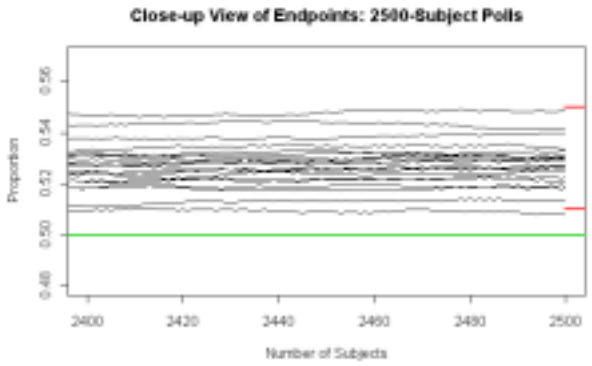

Figure 10

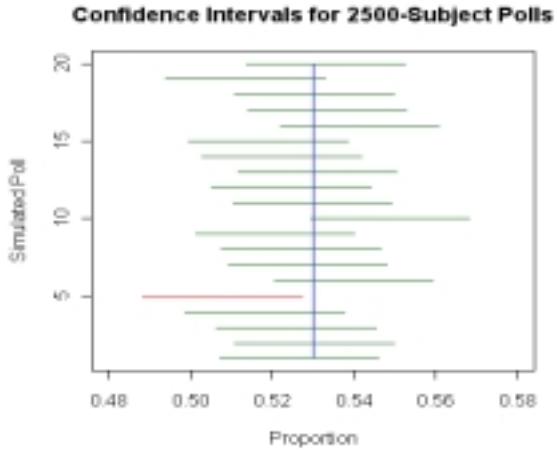

Figure 11

In both Figures 9 and 10 the interval 0.53 ± 0.02 is indicated by heavy (red) lines at the right edge. For the particular 20 traces shown in these figures (based on `set.seed(1)`), it happens that the fraction of traces with endpoints in this interval is 19/20 = 95%. Of course, some random sets of 20 traces are better behaved than these, and some worse.

Looking at the endpoints of the 20 traces of simulated polls in Figures 9 and 10, we see that a poll with $n = 2500$ subjects is large enough that we would usually detect that $\pi = 0.53$ is larger than the proportion 0.50 required to win an election.

Figure 11 shows confidence intervals for each of the 20 simulated polls in Figures 9 and 10, based on the formula $p \pm 1.96[p(1-p)/n]^{1/2}$. Only one of them fails to cover $\pi = 0.53$, and only a few others extend down near 0.50. (The loop structure of the code is similar to that in Display 5, but with the axes reversed; see [2].)

The histogram in Figure 12 shows results from 10,000 simulated polls. (The code in Display 7 works by changing only `n <- 2500`.) As in Figure 6, the distribution of estimates p is very nearly normal. The important difference between Figures 6 and 11 is seen in the horizontal scale. For polls with 2500 subjects, almost all values of p are in the interval 0.53 ± 0.02 and almost all are above 0.50. In our run of 10,000 polls, only 8 estimates p fell below 0.5 (found using `length(Estimate[Estimate < .5])`). Figure 13 compares the density curves of the approximating normal distributions in Figures 7 and 12.

4. Margin of error for a candidate's lead. We have seen some evidence from simulations that a 95% confidence interval for π in a poll based on n subjects has margin of error $E \approx n^{-1/2}$ in some circumstances of practical importance. However, a frequent mistake is to apply the same margin of error to derived quantities such as the lead one candidate has over another.

Suppose there are only *two* candidates A and B and all subjects express a preference. Then relevant estimates for the population proportions are $p_A = X/n$ for π_A and $p_B = (n - X)/n$ for π_B, where X is the number of subjects preferring A and $\pi_A + \pi_B = 1$. An unbiased estimate of A's lead $\delta_{AB} = \pi_A - \pi_b = 2\pi_A - 1$ is $D_{AB} = p_A - p_B = 2p_A - 1$, which has twice the standard deviation of p_A. Thus the margin of error for the lead is approximately $2E$. Each subject preferring A is also one less subject for B.

If the population is partitioned into *three* traits A, B, and C, then the estimate of A's lead over B is $D_{AB} = p_A - p_B$, which has standard deviation (see [5, 6])

$$\mathrm{SD}(D_{AB}) = \{[\pi_A(1-\pi_A) + \pi_B(1-\pi_B) + 2\pi_A\pi_B] / n\}^{1/2},$$

where the third term in the numerator is $-2\mathrm{Cov}(p_A, p_B)$.

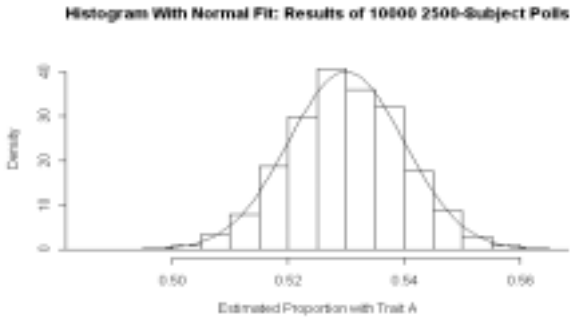

Figure 12

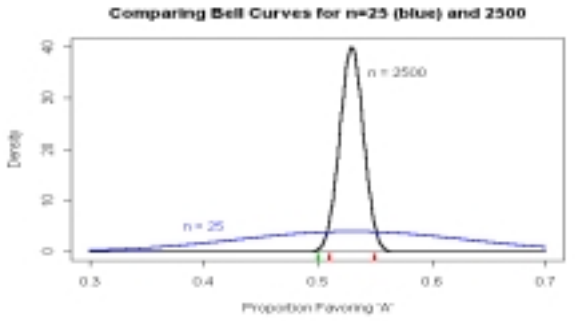

Figure 13

Copyright © 2004 by Bruce E. Trumbo. All rights reserved. DRAFT: Please email btrumbo@csuhayward.edu with suggestions or corrections.

Thus, if $\pi_A = 0.48$, $\pi_B = 0.42$ and $\pi_C = 0.10$, then the margin of error for A's lead over B in a poll with $n = 2500$ would be $1.96\,\mathrm{SD}(D_{AB}) = 1.96(0.0189) = .0371$. In practice, this margin of error would be *estimated* by plugging estimates of the πs into the expression for $\mathrm{SD}(D_{AB})$: p_A for π_A, and so on.

To simulate the situation with traits A, B and C, substitute `-1:1` for `0:1` in the R code, associating A with 1, B with –1, and C with 0. Then the sums will be leads of A over B. Also use `prob=c(.42, .10, .48)` in the sample function. For the histogram in Figure 14, we also modified the code in Display 7 with

```
pa <- .48;  pb <- .42;  pc <- .1
sd <- sqrt((pa*(1-pa)+pb*(1-pb)+2*pa*pb)/n)
zd <- dnorm(zz, pa-pb, sd)
```

This histogram is consistent with the known lead of 0.48 – 0.42 = 0.06 and $\mathrm{SD}(D_{AB}) = 0.0189$. For our run, the code `sqrt(var(Estimate))` returned 0.01878.

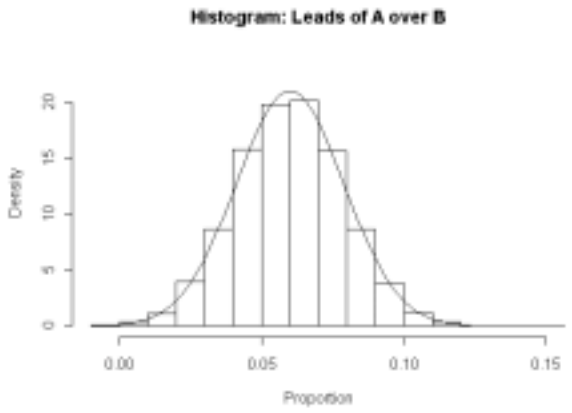

Figure 14

5. Percentage of trials with A in the lead. Now, for simplicity, assume there are two equally matched candidates so that taking a poll is like tossing a fair coin. Because trials are independent there is no "correction" for an initial chance preponderance of subjects favoring one candidate or another. The nearly correct estimate of p in a large poll is often achieved by a sequence that approaches the correct value through values that rather consistently favor either A or B. Feller [7] showed that the percentage of trials with A in the lead is most likely either near 0 or 1. There are typically only a few trials on which the two candidates are tied. (As a small adjustment to ensure symmetry, Feller takes A to be "in the lead" at trial k if A is tied at trial k, but truly in the lead at trial $k + 1$.)

We illustrate this idea with a simple simulation. A histogram of results from $m = 10{,}000$ simulated surveys with $n = 2500$ subjects is shown in Figure 15 along with the "bathtub shaped" distribution, BETA(1/2, 1/2), which has an arcsine function as its cumulative distribution function. The code for the histogram (but not the curve) is shown in Display 16. (See [2] for notes on the code.)

Figure 15

```
set.seed(2);  m <- 10000;  n <- 2500
Leading <- numeric(m)
for (i in 1:m) {
  x <- sample(c(-1,1),n, repl=T)
  cum <- cumsum(x)
  pos <- (cum[1:(n-1)] > 0) | (cum[2:n] > 0)
  Leading[i] <- mean(pos) }
cutpt <- seq(0, 1, by=.1)
hist(Leading, prob=T, breaks=cutpt)
```

Display 16

6. In the classroom. In a precalculus service course, a demonstration based on simulations in Sections 1-4 can be used to introduce confidence intervals for proportions. In beginning statistics courses for majors in science and engineering, we have found that it is feasible for students to understand and run the programs, and to do exercises requiring minor modifications of the code [2]; also perhaps to try other elementary simulations based on the `sample` function [2, 8]. More advanced courses can also benefit from simulations and computations in R [9, 10].

7. References.

[1] `www.r-project.org` (Instructions for downloading and installing R for Windows, Macintosh, Unix.)

[2] `www.csueastbay.edu/~btrumbo/JSM/pollsim` (R code, additional materials for this paper.)

[3] Brown, L. D.; Cai, T. T.; and DasGupta, A.: "Interval estimation for a binomial proportion," *Statistical Science*, 16:2, pages 101–133 (2001)

[4] Moore, David S.: *Basic Practice of Statistics*, 3rd ed., W. H. Freeman (2004) page 479.

[5] Mood, A. M.; Graybill, F. A.; Boes D. C.: *Intro. to the Theory of Statistics*, 3rd ed., McGraw-Hill (1974) page 196.

[6] `www.sci.csueastbay.edu/statistics/Resources/Quiz/poll.htm`

[7] Feller, William: An Intro. to Probability Theory and Its Applications, Vol. 1, 3rd ed., Wiley (1968)

[8] Trumbo, B. E.; Suess, E. A.; Schupp, C. W.: "Using R to compute probabilities of matching birthdays," *Proc. JSM 2004*.

[9] Trumbo, B. E., Suess, E. A; Fraser, C. M.: "Contemporary statistical simulation methodology for undergraduates," *Proc. JSM 2000.*

[10] Horton, N. J., Brown, E. R., Qian, L.: "Use of R as a toolbox for mathematical statistics exploration," *The American Statistician*, 58:4, 343-357 (2004)

Copyright © 2004 by Bruce E. Trumbo. All rights reserved. DRAFT: Please email `btrumbo@csuhayward.edu` with suggestions or corrections.

Addendum to Draft — Additional Annotated R Code for Figures 10, 11, and 13

(Including labels, colored reference lines and tick marks, etc. Nice for publication. Optional for students.)

Code for Figure 10

```
set.seed(1)  # remove this line for simulation different from that shown
n <- 2500;  m <- 20;  p <- .53
plot(c(-.1*n,1.1*n),c(.5, .5), type="l", col="green", lwd=2,
   xlim = c(n-100,n), ylim=c(p-.05,p+.04),
   ylab="Proportion", xlab="Number of Subjects",
   main="Close-up View of Endpoints: 2500-Subject Polls")
lines(c(n,1.1*n),c(p-.02,p-.02), col="red", lwd=2)
lines(c(n,1.1*n),c(p+.02,p+.02), col="red", lwd=2)
# set up plot window with green reference line and red tick marks
# xlim and ylim parameters provide close-up of endpoints of paths
for (i in 1:m)
       {
       x <- sample(0:1, 2500, rep=T, prob=c(1-p,p))  # poll for ith trace
       r <- cumsum(x)/1:n
       lines(1:n, r)  # plot ith trace
       #
       }
```

Code for Figure 11

```
set.seed(1)
n <- 2500;  m <- 20;  p <- .53
plot(c(p,p), c(1,m), type="l", col="blue",
   xlim = c(.48,.58), ylim=c(1,m),
   ylab="Simulated Poll", xlab="Proportion",
   main="Confidence Intervals for 2500-Subject Polls")
# plot labels and vertical blue reference line
for (i in 1:m)
       {
       x <- sample(0:1, n, rep=T, prob=c(1-p,p))  # sample n subjects
       est <- sum(x)/n  # point estimate of p
       lcl = est - 1.96*sqrt(est*(1-est)/n)
       ucl = est + 1.96*sqrt(est*(1-est)/n)
       # lower and upper confidence limits
       farb="darkgreen"
       if (ucl < p | lcl > p) farb="red"     # green if CI covers p, red if not
       lines(c(lcl, ucl), c(i,i), col=farb)  # draw lines for conf intervals
       }
```

Code for Figure 13

```
zz <- seq(.3, .7, by=.001)  # plotting points, horizontal axis
dz.25 <- dnorm(zz, .53, sqrt(.53*.47/25))
dz.2500 <- dnorm(zz, .53, sqrt(.53*.47/2500))
# y-coordinates of normal curves
plot(zz, dz.2500, type="l", lwd=2, ylim=c(0,40), ylab="Density",
   xlab="Proportion Favoring 'A'",
   main="Comparing Bell Curves for n=25 (blue) and 2500")  # plot tall normal curve
lines(c(.3,.7),c(0,0))  #  draw line for x-axis
lines(zz, dz.25, col=4, lwd=2)  # plot flat normal curve
lines(c(.5,.5),c(-.5,-2), col="green", lwd=3)
lines(c(.51,.51),c(-.5,-2), col="red", lwd=3)
lines(c(.55,.55),c(-.5,-2), col="red", lwd=3)
# tick marks
text(.57,35,"n = 2500")
text(.40,5,"n = 25", col="blue")
# text labels of normal curves
```

Copyright © 2004 by Bruce E. Trumbo. All rights reserved. DRAFT: Please email btrumbo@csuhayward.edu with suggestions or corrections.