Chapter 3: Convergence of the p-value algorithm

The value for $\hat{p}$ from Step 3 of the p-value algorithm is an estimate of the true p-value. If this estimate is based on just a few random data sets, it won't be very reliable: the estimate could easily be quite far from the true value. A good estimate $\hat{p}$ is one that is close to p; the distance $|\hat{p} - p|$ is small. Would one hundred random data sets be enough to ensure a good estimate? One thousand? One million? In this chapter you will investigate the behavior of the distance $|\hat{p} - p|$ and how it is related to the number (*NRep*) of random data sets used to estimate p, in order to decide what value of *NRep* to use in your simulations. To focus your work in what follows, keep in mind the question, "How many repetitions do I need if I want $|\hat{p} - p|$ to be less than .01?"

In real situations, you estimate p because you don't know its value. This means that, in real situations, you can never know the value of the distance $|\hat{p} - p|$. To investigate the behavior of that distance, then, you have to work with artificial situations for which you know the true p and can compute the value of $|\hat{p} - p|$. The simplest of these is tossing a fair coin.

Activity 3.1 (Physical Simulation): How many tosses?

Step 1. Toss a coin, record the outcome, and compute (# Heads / # Tosses), which will either be 1 or 0. Call this value $\hat{p}_{(1)}$ to indicate that it is an estimate based on 1 repetition.

Step 2. Toss the coin again. Use this toss, together with the first one, to estimate the chance of heads using (# Heads / # Tosses). Call this value $\hat{p}_{(2)}$.

Step 3. Continue in this way until you have completed 20 tosses. After each toss, update the ratio (# Heads / # Tosses), using all your results to that point. Record your results in a record sheet like the sample in Display 3.1 below.

Step 4. Plot $\hat{p}_{(n)}$ versus n. (By convention, the first variable goes on the y-axis, the second on the x-axis: "y versus x".) In a separate graph, plot $|\hat{p}_{(n)} - p|$ versus n. Join the points with line segments to make the patterns easier to see.

Toss # (n)	1	2	3	4	5	6	7	…	20
Outcome	T	H	H	T	H	H	T	…	H
1 or 0	0	1	1	0	1	1	0	…	0
# heads	0	1	2	2	3	4	4	…	12
$\hat{p}_{(n)}$	0	½	2/3	2/4	3/5	4/6	4/7	…	12/20
Decimal	0	.50	.67	.50	.60	.67	.57	…	.60
$\|\hat{p}_{(n)} - p\|$	.50	.00	.17	.00	.10	.17	.07	…	.60

Display 3.1 Results of 20 coin tosses

Discussion

Your graphs, like all random phenomena, exhibit some features that are predictable, and others features that vary from one occasion to another. The goal of all statistical thinking is to distinguish the repeatable patterns form the "random noise."

1. Describe, as precisely as you can based on your understanding at this point, any patterns in your graph that you would expect to see again if you made another 20 tosses and another pair of graphs.

2. What features of your graph would you expect to be different?

3. At this point, what is your best informed guess for the number n of tosses needed to ensure that $\hat{p}_{(n)}$ will be within .01 of the true p? (Choose a power of 10: 100, 1000, 10000, 100000, 1000000.)

Investigation 3.2 (Computer Simulation): How many tosses?

Step 1. a. Enter and run the following S-plus code:

```
NRep <- 10
p <- .5
x <- rbinom(NRep,1,p)
x
cumsum(x)
1:NRep
cumsum(x)/1:NRep
```

b. Tell what the commands `rbinom` and `cumsum` do.

c. Explain why `cumsum(x)/1:NRep` gives values of $\hat{p}_{(n)}$.

Step 2. a. Modify the code to the following and run it:

```
par(mfrow=c(1,2))                           # 2 graphs per page
NRep <- 20
p <- .5
x <- rbinom(NRep,1,p)
pHat <- cumsum(x)/1:NRep
#
plot(1:NRep,pHat, type="n",ylim=c(0,1))     # plot the axes
points(1:NRep,pHat,type="l")                # plot the points
abline(h=p)                                 # horizontal line at y = p
#
plot(1:NRep,abs(pHat-p),type="n",ylim=c(0,.5))
points(1:NRep,abs(pHat-p),type="l")
abline(h=0)
```

Note: The S-plus command `plot` puts the first variable on the x-axis and the second variable on the y-axis: `plot(x,y)`.

b. Run the code several more times to develop your intuition about the features that are common to all runs. Compare what you see here with your answers to the discussion questions (1) and (2).

Step 3. You can put several plots on one page. To do this in S-plus, add the five lines shown in boldface below:

```
NRep <- 100
p <- .5
x <- rbinom(NRep,1,p)
pHat <- cumsum(x)/1:NRep
#
par(mfrow=c(4,5))                    # 20 plots in 4 rows by 5 columns
for (i in 1:20){
   x <- rbinom(NRep,1,p)
   pHat <- cumsum(x)/1:NRep
   plot(1:NRep,pHat, type="n",ylim=c(0,1))
   abline(h=p)
   points(1:NRep,pHat,type="l")
}
for (i in 1:20){
   x <- rbinom(NRep,1,p)
   pHat <- cumsum(x)/1:NRep
   plot(1:NRep,abs(pHat-p),type="n",ylim=c(0,.5))
   abline(h=0)
   points(1:NRep,abs(pHat-p),type="l")
}
```

Step 4. a. Instead of making separate plot, you can superimpose several plots on a single set of axes. To change back to one plot per page, change the line shown in bold below. To superimpose plots, move the two lines marked `####` from inside to outside the programming loop, and do the same for the second loop.

```
par(mfrow=c(1,1))
plot(1:NRep,pHat, type="n",ylim=c(0,1)) ####
abline(h=p)                             ####
for (i in 1:20){
   x <- rbinom(NRep,1,p)
   pHat <- cumsum(x)/1:NRep
   points(1:NRep,pHat,type="l")
}
```

b. Rerun the current code using larger values of *NRep* (50? 100? 1000?). Describe the regularities in your graphs. What is your current best guess for the number of repetitions you need to get $\hat{p}_{(n)}$ to be within .01 of the true p?

Step 5. a. A difficulty with the plots in (3) is that as *NRep* increases beyond 50 or so, change is slow, which makes it hard to judge how big *NRep* needs to be. One way around this difficulty would be to compress the horizontal scale. A handy way to do that is to plot the log of *NRep*, instead of *NRep* itself, on the horizontal axis. In S-plus, you can accomplish this by inserting `log=″x″` in the `plot` command:

```
plot(1:NRep,pHat, type="n",ylim=c(0,1),log=″x″)
```

b. Use your new graph to revise your best guess for the value of *NRep* that will give $|\hat{p}_{(n)} - p| < .01$.

Discussion questions: Certainty is too much to ask

4. Suppose you toss a coin 10 times and estimate p using $\hat{p}$ = (# Heads / # Tosses). What is the largest possible value of $|\hat{p}_{(n)} - p|$? Explain how you could get this value. What if you used 100 tosses instead of 10. What is the largest possible value of $|\hat{p}_{(n)} - p|$? Explain how you could get it.

5. Based on your answers in (4), explain why it is not possible to, no matter how many tosses you make, to *guarantee* that the distance $|\hat{p}_{(n)} - p|$ will be at most .01.

Civil engineers who want to protect against floods sometimes use the idea of the "hundred year flood," the one so big it is expected to happen only once a century. The thinking is that there's no such thing as the largest possible flood, and so it isn't practical to try to protect against the largest flood possible. Instead, engineers may decide to protect against all floods smaller than the hundred year flood.

Apply the same logic to the estimation problem: Instead of trying to guarantee that $|\hat{p}_{(n)} - p|$ will *always* be less than .01, we choose n to be large enough to ensure that the distance will *almost always* be less than .01. The most common choice for "almost always" in statistics is 95% of the time, or 19 times out of 20. This leads to a definition:

Def: The **95% margin of error**, or *M*95, is the number for which

$$|\hat{p}_{(n)} - p| \le M95$$

95% of the time.

For statisticians, *M*95 is like the height of a 20-year flood: it is exceeded only 1 time in 20. We can now rephrase the original question a "If you want to estimate p, how many repetitions do you need to make $M95 = .01$?"

6. Let *M*95 be the 95% margin of error. Explain how to find *M*95 by simulation: Suppose you fix *NRep* at 1000 tosses. Each time you do 1000 tosses, you can compute $\hat{p}_{(1000)}$ and $|\hat{p}_{(1000)} - p|$. Tell how to use values of this sort to find *M*95.

7. Examine the following S-plus code, and explain what it does:

```
NRep <- 1000
p <- .5
```

```
x <- rbinom(10000,NRep,p)
pHat <- x/NRep
error <- abs(pHat-p)
error <- sort(error)
M95 <- error[.95*10000]
M95
```

Investigation 3.3 (Computer simulation)

Step 1. Enter the S-plus code from Discussion Question (7) and run it.

Step 2. Rerun the same code, changing the value of *NRep*, to investigate the relationship between *M*95 and *NRep*.

Step 3. Suppose we want to run the code for a large set of values for *NRep*, and then plot *M*95 versus *NRep*. To do this efficiently in S-plus, we first define a function that computes *M*95 for any choice of *NRep* and *p*, then we apply the function to a set of values for *NRep*.

a. Enter the code shown in boldface around the lines of code in Step 2, and run it.

```
M95 <- function(NRep,p){
   x <- rbinom(10000,NRep,p)
   pHat <- x/NRep
   error <- abs(pHat-p)
   error <- sort(error)
   M95 <- error[.95*10000]
   return(M95)
}
M95(1000,.5)
```

b. Now replace the line `M95(1000,.5)` with the following four lines, and run the result:

```
NReps <- c(10,20,50,100,200,500,1000,2000,5000,10000,20000,50000)
M95s <- sapply(as.list(NReps),M95,p=.5)
plot(NReps,M95s)
plot(NReps,M95s,log=”x”)
```

Looking for a functional relationship. As you can see from your graphs, there appears to be a fairly simple relationship between the 95% margin of error and the number *n* of repetitions you use. It is reasonable to think that there might even be a simple formula for *M*95 in terms of *n*. Unfortunately, it isn't obvious from the graphs what that formula might be. Fortunately, though, there are strategies for finding it. The first step is to turn the open-ended question, "What is the formula?" into a multiple choice question, "Which one of the following short list of functional forms seems to work best?"

Linear relationship: $M95 = a + bn$, where a and b are unknown constants. You already know the relationship is not linear, because your plot of *M*95 versus *n* in (3b) above doesn't look like a line. All the same, whenever you are working with function, it can't hurt to start by thinking about linear functions, in part because they occur so commonly in applications, but also because linear functions are the simplest of all families of functions.

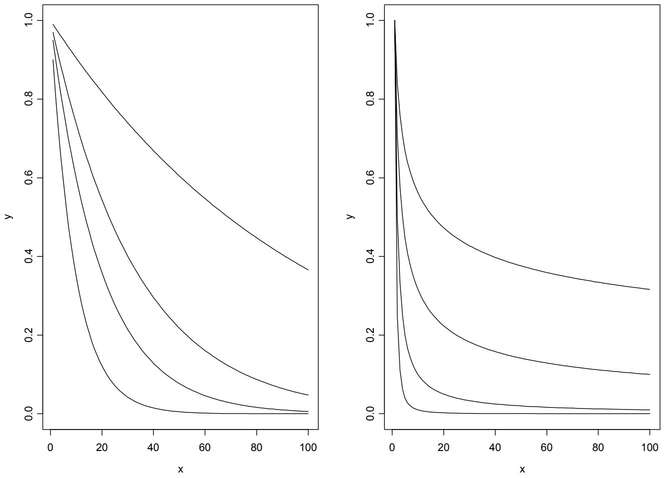
Logarithmic relationship: $M95 = a + b\log(n)$, where a and b are unknown constants. Here, too, you already have the evidence to rule out this relationship. Notice if the relationship is logarithmic, then a plot of $M95$ versus $\log(n)$ would look like a line. Then check that your second plot in (3b) above is a plot of $M95$ versus n, with n in the log scale, and it doesn't look at all like a line.

Here's another way to see that the relationship cannot be logarithmic. As n increases, $M95$ decreases. This means that if $M95 = a + b\log(n)$, then b has to be negative. But if b is negative, then for large enough values of n, the values of $a + b\log(n)$ will be negative, whereas $M95$ cannot be negative.

Geometric (exponential) relationship: $M95 = ab^n$, for some choice of a and b. If the relationship has this form, then we know that a has to be positive, and b has to be both positive and less than 1. (Why?) Relationships of this form are very common, as you may have learned in calculus. Such relationships are used to describe population growth, chemical reactions, radioactive decay, compound interest, and the cooling of hot liquids, to name just a few.

Power law relationship: $M95 = an^b$, for some choice of a and b.
Like exponential relationship, power law relationships are extremely common. For example, the area of a circle is $A = \pi r^2$, the volume of a cube is $V = s^3$, the distance traveled in t seconds by a falling object is $d = 16t^2$, and the volume of a gas at pressure P is given by $V = cP^{-1}$.

Note that for power law relationships, the base n is the independent variable, and the exponent b is constant. For geometric relationships, the base b is constant, and the exponent n is the independent variable. The difference, small in terms of notation, can be huge in terms of how the relationship behaves. However, it can be hard to tell which family a relationship belongs to just from looking at the graph. Display 3.2 shows plots of four exponential relationships and four power law relationships:



Display 3.2 Four exponential and four power law relationships

Fortunately, there is a simple way to tell the two families apart, brought to you by the magic of logarithms, which convert the apparently hard problem into the simple one of deciding whether a plot looks line a line.

Exponential. If $y = ab^x$, then plotting pairs $(x, \log y)$ gives a straight line with slope $\log a$ and intercept log(b).

Power law. If $y = ax^b$, then plotting pairs $(\log x, \log y)$ gives a straight line with slope b and intercept log(a).

Exercises.

8. Consider the areas of four squares with sides 1”, 2”, 5”, and 10”. Is the relationship between area and length of side exponential or power law? Which plot will give a line? What should its intercept and slope be? Answer the questions first, then make the plot.

9. Consider the perimeters for the same set of four squares. Tell the kind of relationship, what to plot to get a line, and what the intercept and slope should be. Then make the plot.

10. Invent (base, height) pairs for a set of four rectangles, all having the same area of 24 square inches. Which family does the relationship belong to? Which plot gives a line? What should the slope and intercept be? Make the plot.

11. When you put caffeine into your bloodstream, it gets removed at a rate that cuts the concentration in half every four hours. Suppose you start at time $t = 0$ with 80 milligrams of caffeine in your blood. (a) Compute caffeine levels (mg) for $t = 4$, 8,

12, 16 and 20 hours. (b) Which kind of relationship fits the (time, caffeine) pairs? (c) Make the plot that gives a line.

12. Prove that the boxed statements above are correct.

3.4 Investigation: Formula for the 95% margin of error

Step 1. Use the S-plus code from Step 3a of Investigation 3.3 to plot log(*M*95) versus *n* and log(*M*95) versus log(*n*). Is the relationship geometric or power law?

Step 2. Print a copy of whichever plot from Step 1 is linear, and use a ruler to draw a line that fits the points of your plot. Find the slope and intercept of this line.

Step 3. Use your results from Step 2 to write a formula for *M*95 in terms of *n*.

Step 4. Use your formula to answer the original question: How big should *n* be if you want *M*95 = .01?

Extensions: Different values of p, different error rates[1]

So far, everything in this chapter has focused on a single fixed value of *p*, namely .5, and a single fixed error rate, 95%.

13. Investigation and discussion: Different values of *p*
Choose a value of *p* other than .5. (The investigation will be broader in scope if you and the others in your class use a variety of values, chosen from .1, .2, .3, …, .8, .9.) Make suitable plots of *M*95 versus *n* for your value of *p* in order to answer the following questions: (a) Is the functional form of the relationship between M95 and n the same for all values of p? (b) If so, is the value of *a* the same for all values of *p*? If not, graph *a* versus *p*, and try to find a formula. (c) Same question as (b), this time about the other constant, *b*.

14. Investigation and discussion: Different confidence levels
The 95% in M95 is often called the confidence level. Choose a confidence level othe than 95% (The investigation will be broader in scope if you and the others in your class use a variety of values, chosen from 80%, 90%, 97.5%, 99%, 99.5%, 99.9%.) Make suitable plots of your margin of error versus *n* for your confidence level, using p = .5, in order to answer the following questions: (a) Is the functional form of the relationship between the margin of error and *n* the same for all confidence levels? (b) If so, is the value of *a* the same for all confidence levels? If not, graph *a* versus confidence level. (c) Same question as (b), this time about the other constant, *b*.

---

[1] Hint: At some point in working on these extensions, it will be helpful to recall the fact that a circle of radius *r* centered at (x_0,y_0) has equation $(x-x_0)^2 + (y-y_0)^2 = r^2$.