

Can You Fix the Random Number Generator?



June 2026

A Bayesian-inference puzzle from The Fiddler.¹

Problem

A calculator's random number key should return a value spread evenly between 0 and 1, with every value equally likely. This one is broken: it returns a value spread evenly between 0 and some fixed a , where a is an unknown number between 0 and 1. Before you switch it on you have no idea what a is, so you treat it as equally likely to be anywhere in that range. You press the key once and read off exactly 0.5. Given that single reading, what is the expected value of a ?

For extra credit: a second broken calculator works the same way, with its own unknown a spread evenly over 0 to 1. This time you do not see the reading itself, only that it landed somewhere between 0 and 0.5. Now what is the expected value of a ?

Solution

Everything turns on the ceiling a , the value the broken calculator never climbs above. Before the key is pressed a could sit anywhere in $(0, 1)$ with equal claim, a flat prior $f(a) = 1$, and the single reading is the only thing that sharpens that belief. The expected value we are after is the average of the sharpened belief, so the task is to see how the reading reshapes it.

Start from what a calculator of a known ceiling does. One with ceiling a scatters its output evenly across $(0, a)$, so it returns a value near x with density $1/a$, as long as x does not exceed a , for a machine can never report a number above its own ceiling. Now read this the other way, as a statement about the ceiling for the value we actually saw, and two things stand out. A reading of 0.5 throws out every ceiling below 0.5 outright, since those machines could not have produced it; and among the ceilings that survive, the lower ones are the better bet, because a calculator with a narrow range packs its output into a short interval and is the likelier source of any particular reading. That second effect is precisely the factor $1/a$.

Those two observations are Bayes' rule for this problem. The sharpened belief in a , its posterior density, is the prior cut down to the survivors and tilted by $1/a$,

$$f(a | x) \propto \frac{1}{a}, \quad x \leq a \leq 1,$$

¹Zach Wissner-Gross, *The Fiddler*, 19 June 2026, thefiddler.substack.com.

and dividing by its total $\int_x^1 \frac{da}{a} = -\ln x$ turns the proportionality into a genuine density. The average ceiling is this density weighted by a and summed across the survivors, where the weight a neatly cancels the $1/a$ and leaves a flat integrand,

$$\mathbb{E}[a | X = x] = \int_x^1 a f(a | x) da = \frac{1}{-\ln x} \int_x^1 da = \frac{1-x}{-\ln x}.$$

For the reading $x = \frac{1}{2}$, where $-\ln \frac{1}{2} = \ln 2$, this comes to

$$\mathbb{E}\left[a | X = \frac{1}{2}\right] = \frac{1/2}{\ln 2} = \boxed{\frac{1}{2 \ln 2} \approx 0.7213}.$$

The reading holds the ceiling above 0.5, and the pull toward smaller ceilings then drags the average down from the midpoint 0.75 of the surviving range, settling at 0.721.

Another way. The tilt $1/a$ carries a tidy reading of its own. Setting $u = \ln a$ turns the weight da/a into plain du , so once the reading is in hand $\ln a$ is spread evenly between $\ln x$ and 0. The ceiling is therefore $a = x^{1-V}$ for V uniform on $[0, 1]$, sliding from the floor x up to 1 as V runs from 0 to 1, and averaging over V returns the same value,

$$\mathbb{E}[a | x] = \int_0^1 x^{1-v} dv = \frac{1-x}{-\ln x},$$

which is the mean of a log-uniform value: the plain gap $1-x$ over the logarithmic gap $-\ln x$.

Extra credit

This time the reading itself is hidden, and all we are told is that it landed below 0.5. The same machinery carries over, except that the evidence is now a whole event rather than one value, so each ceiling is judged by how often it lands there. A calculator with ceiling $a \leq \frac{1}{2}$ never climbs past $\frac{1}{2}$, so it falls below for certain; one with $a > \frac{1}{2}$ manages it only part of the time, with probability $\frac{1}{2}/a$, the share of its range that lies under $\frac{1}{2}$. Writing $c = \frac{1}{2}$ and weighting the flat prior by this chance gives the posterior

$$f(a) \propto \begin{cases} 1, & 0 < a \leq c, \\ \frac{c}{a}, & c < a < 1. \end{cases}$$

Its total is the overall chance of a low reading,

$$Z = \int_0^c 1 da + \int_c^1 \frac{c}{a} da = c + c(-\ln c) = c(1 - \ln c),$$

and the same integral carrying an extra factor of a gives the unnormalised mean,

$$M = \int_0^c a da + \int_c^1 a \cdot \frac{c}{a} da = \frac{c^2}{2} + c(1 - c) = c - \frac{c^2}{2}.$$

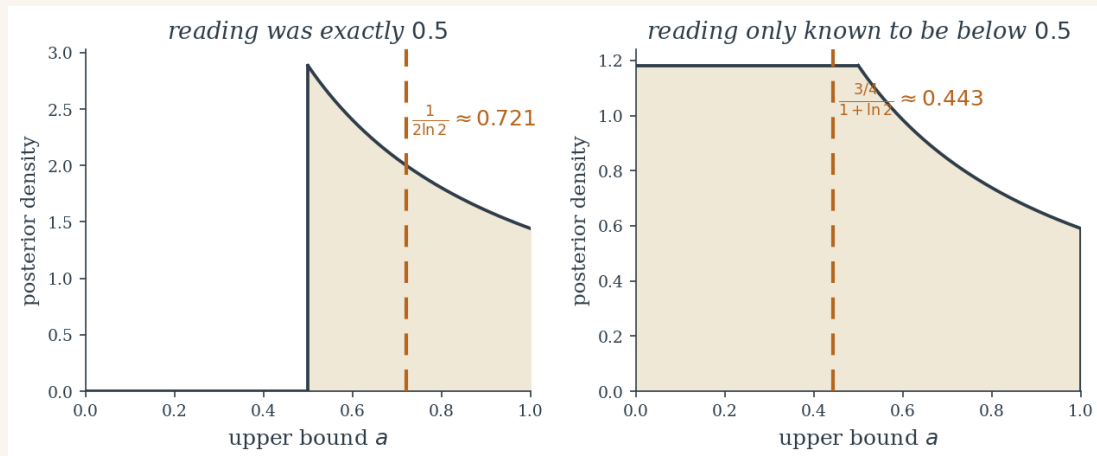
Their ratio is the answer,

$$\mathbb{E}[a | X < c] = \frac{M}{Z} = \frac{c - c^2/2}{c(1 - \ln c)} = \frac{1 - c/2}{1 - \ln c}$$

and at $c = \frac{1}{2}$ it works out to

$$\mathbb{E}\left[a \mid X < \frac{1}{2}\right] = \frac{3/4}{1 + \ln 2} = \frac{3}{4(1 + \ln 2)} \approx 0.4430.$$

The two halves of the puzzle send the same number 0.5 in opposite directions. Seeing it exactly proves the ceiling reaches that high and lifts the estimate to 0.721, whereas learning only that the reading stayed below it points to a low ceiling, since those are the machines bound to land there, and the estimate falls to 0.443.



Another way. The extra credit also drops out of the main answer, by averaging it over the reading we never saw. A reading near x turns up with density $f(x) = \int_x^1 \frac{da}{a} = -\ln x$ across $(0, 1)$, so the law of total expectation gives

$$\mathbb{E}[a \mid X < c] = \frac{\int_0^c \mathbb{E}[a \mid X = x] f(x) dx}{\int_0^c f(x) dx}.$$

The frequency $-\ln x$ is the very denominator buried inside $\mathbb{E}[a \mid X = x] = (1 - x)/(-\ln x)$, so the two cancel and leave

$$\mathbb{E}[a \mid X < c] = \frac{\int_0^c (1 - x) dx}{\int_0^c (-\ln x) dx} = \frac{c - c^2/2}{c(1 - \ln c)},$$

the same value once more.

Python code

The check simulates the broken calculator rather than the formula: draw a ceiling a uniformly on $(0, 1)$, draw a reading uniformly on $(0, a)$, and keep the runs that match what was observed; the average ceiling over those runs is the answer. An exact reading of 0.5 has probability zero, so we keep readings inside a thin window and watch the estimate settle as the window shrinks; the event “below 0.5” has positive probability, so plain accept-and-average is exact.

```

import random
from math import log
random.seed(0)

# Main: the reading came out exactly 0.5. Condition on a thin window  $|x-0.5|<eps$ ;
# as the window shrinks the estimate approaches  $E[a | X = 0.5]$ .
def estimate_exact(eps, N=20_000_000):
    total = count = 0
    for _ in range(N):
        a = random.random() # prior:  $a \sim \text{Uniform}(0, 1)$ 
        x = random.random() * a # broken:  $x \sim \text{Uniform}(0, a)$ 
        if abs(x - 0.5) < eps:
            total += a; count += 1
    return total / count

for eps in (0.02, 0.01, 0.004):
    print(f"exact reading, window {eps}<5}:  $E[a] \sim \{\text{estimate\_exact}(eps):.4f\}$ ")
    print(f"closed form  $1/(2 \ln 2) = \{1 / (2 * \log(2)):.4f\}$ ")

# Extra credit: the reading is only known to be below 0.5. A finite-probability
# event, so a plain accept/reject Monte Carlo is exact.
def estimate_below(N=40_000_000):
    total = count = 0
    for _ in range(N):
        a = random.random()
        x = random.random() * a
        if x < 0.5:
            total += a; count += 1
    return total / count

print(f"reading below 0.5:  $E[a] \sim \{\text{estimate\_below}():.4f\}$ ")
print(f"closed form  $(3/4)/(1+\ln 2) = \{0.75 / (1 + \log(2)):.4f\}$ ")
# exact reading, window 0.02 :  $E[a] \sim 0.7210$ 
# exact reading, window 0.01 :  $E[a] \sim 0.7211$ 
# exact reading, window 0.004:  $E[a] \sim 0.7209$ 
# closed form  $1/(2 \ln 2) = 0.7213$ 
# reading below 0.5:  $E[a] \sim 0.4429$ 
# closed form  $(3/4)/(1+\ln 2) = 0.4430$ 

```

The two alternate derivations are identities rather than experiments, so a symbolic check suits them: the log-uniform mean for the main puzzle, and the total-expectation quotient for the extra credit, with the marginal $f(x) = -\ln x$ computed rather than assumed.

```

import sympy as sp
x, v, a, c = sp.symbols("x v a c", positive=True)

# Main, log-uniform route:  $a = x^{1-V}$ ,  $V$  uniform on  $[0, 1]$ .
# The  $x < 1$  branch of its mean is  $(x-1)/\ln(x)$ :
main = sp.integrate(x ** (1 - v), (v, 0, 1))
print("main  $E[a] = (x-1)/\ln x$  ; at  $x=1/2 \rightarrow$ ",
      float(((x - 1) / sp.log(x)).subs(x, sp.Rational(1, 2))))

# Extra credit, law of total expectation over the hidden reading.
fX = sp.integrate(1 / a, (a, x, 1)) # marginal of the reading =  $-\ln x$ 
Emean = (x - 1) / sp.log(x) # =  $E[a | X = x]$ 
num = sp.integrate(Emean * fX, (x, 0, c))
den = sp.integrate(fX, (x, 0, c))
tower = sp.simplify(num / den)

```

```
print("EC E[a|X<c] =", tower, "; at c=1/2 ->",  
      float(tower.subs(c, sp.Rational(1, 2))))  
# main E[a] = (x-1)/ln x ; at x=1/2 -> 0.7213475204444817  
# EC E[a|X<c] = (c - 2)/(2*(log(c) - 1)) ; at c=1/2 -> 0.44296208186223096
```