

# What does it take to be a hero? and other questions from statistical mechanics.

Dan Piloni

September 2024

## 1 We only hear about the survivors

In the classic Star Trek episode *Errand of Mercy*, Spock computes the chance of success:

CAPTAIN JAMES T. KIRK : What would you say the odds are on our getting out of here?

MR. SPOCK : Difficult to be precise, Captain. I should say, approximately 7,824.7 to 1.

And yet they get out of there. Are Spock's probability computations unreliable? Think of it another way. The Galaxy is a large place. There must be tens of thousands of Spocks, and Grocks, and Plocks out there on various missions. But we won't hear (or don't want to hear) about the failures. So they may all be perfectly good at probability theory, but we're only hearing about the lucky ones. This is an example of survivor bias.

## 2 Simulation

We can model this. I've written a small battle simulator for a super-simple made up role-playing game. Our intrepid wannabe hero has to battle

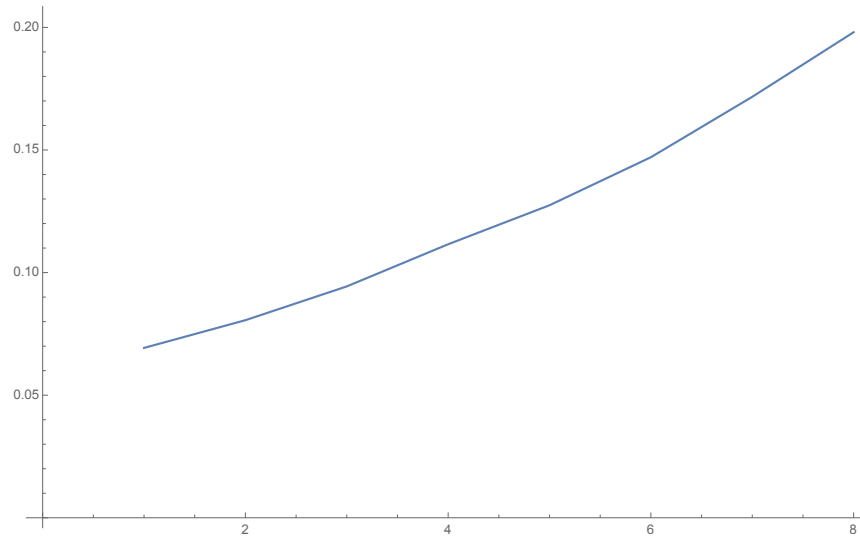


Figure 1: Frequency of damage rolls

a monster with way more hit points. Only very few will survive, but survivor will return home with an unlikely sounding story to tell.

To find out about that story I record some of the die rolls. In my rules, each time the warrior hits they do 1d8 damage. But when we look at the results for the 6 in a million that survive, the distribution of damage rolls is skewed upwards. Although a priori we expect  $P(1) = P(2) = \dots = P(8) = \frac{1}{8}$ , when we condition on surviving we get the table

Roll	1	2	3	4	5	6	7	8
Probability	0.069	0.0806	0.094	0.111	0.127	0.147	0.172	0.198

which I also plot in Figure 1. This corresponds to a mean of 5.27, quite a bit higher than 4.5.

The distribution is significantly skewed upwards. To surviving fighters this is how the world looks - it seems like it's a feature of the world that when you roll d8, you're much more likely to get 8 than 1. (This must be how the world looks to most billionaires too.)

But look closely at the graph and there's something more going on: the graph has a slight convexity. We can explain that too, and here we veer slightly towards statistical physics.

Let's use a simplified model. Suppose we roll 100d8. We expect a mean of 450. Lets say we get 527, which closely matches our experiment above. We can now ask for the conditional distribution of each die roll given that the sum is 527. I'll use Mathematica. Define `num[s,n,m]` to be the number of ways to roll a total of  $s$  using  $n$   $m$ -sided dice:

```
num [s_, n_, m_] := Sum[
  Module[
    {k = s - n - m r},
    (-1)^r
    Binomial[n, r] Binomial[n + k - 1, k]],
  {r, 0, Quotient[s - n, m + 1]]}
```

And now we can use Bayes' theorem to tabulate the probabilities of each roll,  $i$ , by counting how many ways we can score a total of  $527 - i$  using 99 dice:

```
w = Table[num[527 - i, 99, 8], {i, 1, 8}];
N[w/Total[w]]
```

We get

Roll	1	2	3	4	5	6	7	8
Probability	0.069	0.808	0.094	0.110	0.128	0.148	0.172	0.198

These probabilities are very close to our empirical numbers so I think we have a pretty good explanation for our results.

### 3 Maximum entropy

But there's another way to approximate this using one of my favourite papers: Maximum Entropy and Conditional Probability by Campenhout and Cover. The idea there is that the maximum (relative) entropy distribution is a conditional distribution, and vice versa. That paper really helped me understand max entropy. Using the idea of that paper we want to find the maximum entropy distribution on  $\{1, 2, \dots, 8\}$  consistent with having a mean matching ours of 0.527. I'll leave it as an exercise in calculus and Lagrange multipliers to show that this means the probability of

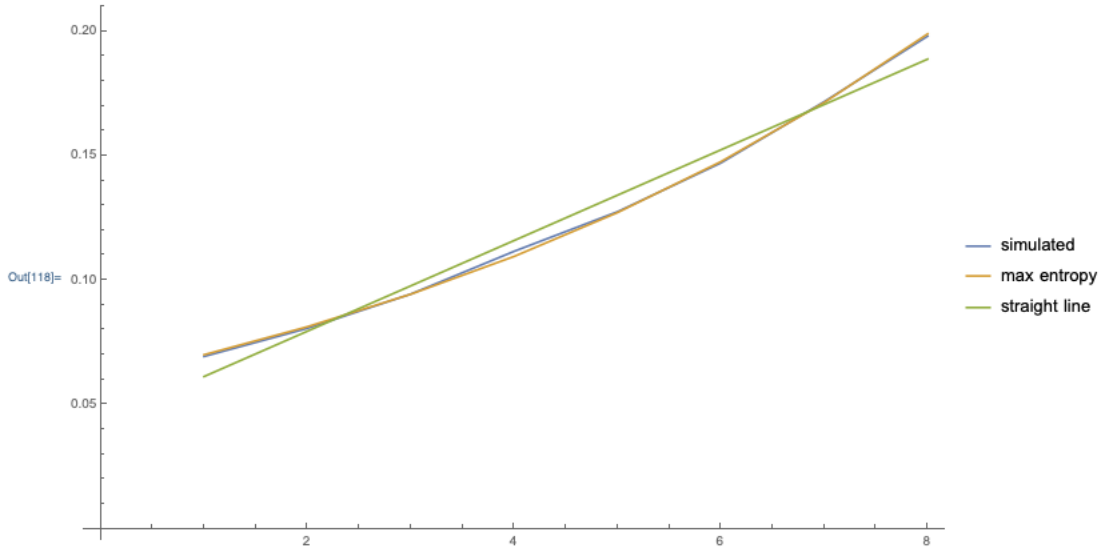


Figure 2: Three distributions compared

rolling  $i$  is given by  $a \exp(-bi)$  for some choice of  $a$  and  $b$ . Again we can use Mathematica:

```
s = Solve[{Sum[a Exp[-b i], {i, 1, 8}] == 1,
  Sum[i a Exp[-b i], {i, 1, 8}] == 5.2654, a \[Element] Reals,
  b \[Element] Reals}, {a, b}];
u = Table[a Exp[-b i] /. s[[1]], {i, 1, 8}]
```

and we get

Roll	1	2	3	4	5	6	7	8
Probability	0.070	0.081	0.094	0.109	0.127	0.148	0.171	0.198

In Figure 2 I plot the simulation results and the max entropy distributions together so you can see how similar they are. I also plot a straight line distribution with the same mean so you can see it is not a good model.

The simulation also tracks the d20 rolls to hit. In the game, these rolls don't combine additively. Instead they act as a kind of "gate" controlling whether the d8's are applied. As a result, the distribution is somewhat different. It's fun to tweak the simulation and see the results. For exam-

ple, try adding a critical hit rule when you roll a 20.

## 4 The thermodynamics connection

The mathematics that derived the skewed distribution is pretty much the same that is used to derive the Boltzmann distribution in statistical physics. In that case, we start with the assumption that particles can have any energy, and then condition on the hypothesis that the total amount of energy in some domain is known. The result is probabilities proportional to  $\exp(-E/kT)$  where  $E$  is the average particle energy,  $k$  is a universal constant and  $T$  is a property of the system.

## 5 Final words

Role-playing game designers realised early that there was a need to give players a little help if their characters were supposed to be the ones we all get to hear about. For example, back in 1980 the RPG "Top Secret" introduced Fame and Fortune Points to allow restricted fudging of rolls. (See A History of Hero Points: Fame, Fortune and Fate.)

To get the results above I had to run 2 billion simulations. Appropriately enough I watched Conan the Barbarian (1982) while my slow Mac did the work. A couple of techniques from the rendering world should speed this up:

*Importance sampling* (Rejection sampling, which is what I already use, is a special case.) Skew the dice rolls upwards a bit so our fighter is more likely to win, but weight the results appropriately so everything balances out.

*"Recursive ray tracing"* Pick some subgoals in the simulation, each less likely than the previous (surviving 3 rounds, surviving 6 rounds, etc.). As well as running simulations from scratch can spawn new simulations that are copies of the state each time a goal is achieved. That way you spend more time running simulations in which many subgoals have been achieved.

Both of the above methods need a bit of fine tuning if you want to get the variance of their results low.

Another point worth noting: if a distribution has a low relative entropy compared to another, it's hard to distinguish from the other distribution. So skewing uniform distributions with an exponential factor is a great way to cheat - for example in a video game where you want players to think they're playing fairly but you want to slightly but consistently help them out.