THE DEATH COUNTERS OF MORTORIOUS

KERRY VEENSTRA

Contents

1. Introduction

Mortorious is an art project about death. On average, someone dies in the United States about every 11 seconds. Of the various causes, the most common are heart disease (one death every 48 seconds), cancers and tumors (one death every 52 seconds), and accidents (one death every 3 minutes 3 seconds). To make these and other relative death rates more tangible, the Burning Man art project *Mortorious* uses several digital "death counters" each of which shows the estimated number of deaths that have occurred somewhere in the United States by a specific cause since the start of the 2019 event.

We want each of the death counters to increment at an appropriate rate, but we also want to avoid the distraction of unrealistic uniformity. Since a death by heart disease doesn't occur *exactly* every 48 seconds we want the counters to show some apparent randomness. In the sections that follow, we explain how each of the Mortorious death counters uses a pseudorandom number generator to avoid masking death's unpredictability.

2. Modeling Uncorrelated Random Events

Nationwide deaths are, for the most part, uncorrelated and random. Using the simplifying assumption that death rates are not affected by age (our *spherical cow*[1]), we model

Date: May 19, 2019. Corrected Table 2 on January 3, 2021.

[1] `https://en.wikipedia.org/wiki/Spherical_cow`

deaths as a *Poisson point process*. In a Poisson point process the durations between consecutive events (the "interarrival" times T) are distributed exponentially.[2] The probability density function[3] $f(T)$ of such a distribution is shown in (1) where $\overline{T} = \frac{1}{\lambda}$ is the mean interarrival time.

$$f(T) = \begin{cases} \lambda e^{-\lambda T} & T \geq 0 \\ 0 & T < 0 \end{cases} \tag{1}$$

In our example of heart disease, $\overline{T} = 48$ sec. Then $\lambda = \frac{1}{48}$.

Now that we have an exponential-distribution probability density function for T, we seek a source of random numbers that is so distributed.

3. Exponentially Distributed Pseudorandom Number Generators

We can use nearly any programming language to generate a sequence of seemingly random numbers that represent the interarrival times of independent, random events. If the programming language that we are using includes an exponential-distribution pseudorandom number generator (PRNG), then we can use it directly. Python is such a programming language: use its function `expovariate()`.

However, if our programming language does not directly support an exponential-distribution PRNG, then we need to construct one. In that case, our solution will take two steps. First we will need a uniform-distribution PRNG. Second, we will need to convert that generator's sequence of uniformly distributed numbers into a sequence of exponentially distributed numbers that we then can use as inter-death times.

3.1. Starting With Uniformly Distributed Pseudorandom Numbers.

Most programming languages provide at least one library function that acts as a uniform-distribution PRNG. Table 1 lists the names of PRNG functions that are available in a few programming languages.

Table 1. Pseudorandom Number Generators

Language	Function	Range
C	`rand()`	integral value r where $0 \leq r \leq$ `RAND_MAX`
PHP	`rand()`	integral value r where $0 \leq r \leq$ `getrandmax()`
Java	`Math.random()`	real value r where $0 \leq r < 1$
JavaScript	`Math.random()`	real value r where $0 \leq r < 1$

One can see from the table that the ranges of various PRNG functions vary. Either the PRNG function has the practical floating-point range $[0, 1)$, as with Java and JavaScript, or the function has another range that can be converted into $[0, 1)$. In the case of the C programming language, the range of the `rand()` function is integral values $\in [0,$ `RAND_MAX`$]$.

[2] `https://en.wikipedia.org/wiki/Exponential_distribution`

[3] `https://en.wikipedia.org/wiki/Exponential_distribution\#Probability_density_function`

Converting this range into $[0, 1)$ requires scaling the `rand()` function's result. As long as the value of `RAND_MAX` can be held in a variable of type `double`, we can use `rand()` in the computation of a floating-point value $u \in [0, 1)$:

```
double u = rand() / (RAND_MAX + 1.0);
```

Once we have a uniform-distribution PRNG that generates values $u \in [0, 1)$, then it is straightforward to convert its sequence of numbers into interarrival times that are distributed exponentially.

3.2. **Computing Exponentially Distributed Pseudorandom Numbers.** In this and the following section we present a method of converting a sequence of *uniformly* distributed pseudorandom numbers into a sequence of *exponentially* distributed pseudorandom numbers. See Section 4 for derivations.

Given a sequence of numbers $u \in [0, 1)$ from a uniform distribution, we can compute an exponentially distributed sequence of numbers $T \in [0, \infty)$ with mean $\overline{T} = \frac{1}{\lambda}$ using the equation

$$T = -\frac{1}{\lambda} \ln(1 - u) \tag{2}$$

Observe that when $u \approx 1$ this equation will generate very large values of T relative to $\overline{T}$. If very large values of T will cause practical concerns, read on.

3.3. **Practical Limits: Chopping the Exponential Tail.** Inevitably some values of $T \in [0, \infty)$ generated by (2) will be quite large. If extreme values of T will cause practical concerns, there is a method to chop the tail of the exponential distribution but keep the mean $\overline{T} = \frac{1}{\lambda}$. The approach is to choose a new range $T \in [\frac{a}{\lambda}, \frac{b}{\lambda}]$ using values of a and b from the table below. Then modify the uniform-distribution PRNG to generate a new range of numbers $u \in [u_a, u_b] = [1 - e^{-a}, 1 - e^{-b}]$. For example, if you are comfortable with the tail being five times as long as the mean, then use $[u_a, u_b]$ from the row with $b = 5$. If you want to use a value of b that is not in the table, then continue to Section 4.

TABLE 2. Several (a, b) pairs along with corresponding limits u_a and u_b for a uniform-distribution PRNG.

a	b	$u_a = 1 - e^{-a}$	$u_b = 1 - e^{-b}$
0.406376	2	0.333940	0.864665
0.178561	3	0.163527	0.950213
0.0793096	4	0.0762461	0.981684
0.0348858	5	0.0342843	0.993262
0.0150988	6	0.0149854	0.997521
6.42431×10^{-3}	7	6.40372×10^{-3}	0.999088
4.54206×10^{-4}	10	4.54102×10^{-4}	0.9999546
4.12231×10^{-8}	20	4.12231×10^{-8}	0.99999999794

Another method to compensate for the shift in the distribution's mean when chopping the exponential tail is to reduce the rate value λ. In this case, the target range of $T \in [0, \frac{b}{\lambda}]$ includes zero, but a modified rate $m\lambda$ is used in (3).

$$T = -\frac{1}{m\lambda}\ln(1-u) \tag{3}$$

TABLE 3. Several values of b for chopping the tail of an exponential-distribution PRNG along with values of u_b for a uniform-distribution PRNG and values of m for modifying the rate used in (3).

a	b	m	u_a	$u_b = 1 - e^{-mb}$
0	2.25	0.298517	0	0.489142
0	2.5	0.491973	0	0.707688
0	2.75	0.623616	0	0.820026
0	3	0.716373	0	0.883413
0	4	0.898378	0	0.972498
0	5	0.960201	0	0.991779
0	6	0.983833	0	0.997269
0	7	0.993351	0	0.999045
0	10	0.999544	0	0.9999544
0	20	0.9999999588	0	0.99999999794

4. Derivations

4.1. **Exponential Distribution PRNG.** To derive these methods, we start with an observation about the cumulative distribution function (CDF) of a PRNG. The CDF $F(T)$ for a PRNG returns the fraction of generated numbers that are less than T. For example, consider the uniform-distribution PRNG that returns values in $[0, 1)$ and whose PDF is $f(T) = 1$. We expect that half of the generated numbers will be less than 0.5, a quarter of the generated numbers will be less than 0.25, three quarters of the generated numbers will be less than 0.75, etc. So let's start with the definition of a CDF, compute the uniform-distribution PRNG's CDF, and check.

$$\begin{aligned} F(T) &= \int_0^T f(T)dx \\ &= \int_0^T 1dx \\ &= x|_{x=0}^{T} \\ &= T - 0 \\ F(T) &= T \end{aligned}$$

Then, as expected, $F(0.5) = 0.5$, $F(0.25) = 0.25$, and $F(0.75) = 0.75$, etc.

Following a similar argument, let's look at the CDF of our exponential-distribution PRNG that has a PDF $f(T) = \lambda e^{-\lambda x}$:

$$\begin{aligned} F(T) &= \int_0^T f(T)dx \\ &= \int_0^T \lambda e^{-\lambda x} dx \\ &= -e^{-\lambda x}\Big|_{x=0}^{T} \\ &= -e^{-\lambda T} - (-e^0) \\ &= -e^{-\lambda T} + 1 \\ F(T) &= 1 - e^{-\lambda T} \end{aligned} \tag{4}$$

Since this is the CDF of the PRNG, we expect that half of the generated numbers will be less than $T_{0.5}$ where $F(T_{0.5}) = 0.5$, a quarter of the generated numbers will be less than $T_{0.25}$ where $F(T_{0.25}) = 0.25$, three quarters of the generated numbers will be less than $T_{0.75}$ where $F(T_{0.75}) = 0.75$, etc. We'd like to find an equation that gives us T_u for any $F(T_u) = u$, and we can do this with (4) by setting $F(T) = u$ and solving for T.

$$\begin{aligned} 1 - e^{-\lambda T_u} &= u \\ -e^{-\lambda T_u} &= u - 1 \\ e^{-\lambda T_u} &= 1 - u \\ -\lambda T_u &= \ln(1 - u) \\ T_u &= -\frac{1}{\lambda}\ln(1 - u) \end{aligned} \tag{5}$$

(6)

The result (6) gives us the value $T_u \in [0, \infty)$ for any $u \in [0, 1)$ where u is the fraction of generated values are less than T_u.

If we use (6) to convert a sequence of numbers $u \in [0, 1)$ that have come from a uniform-distribution PRNG into a sequence of numbers $T_u \in [0, \infty)$, the resulting sequence of numbers will be distributed such that u is the fraction of the T sequence's values that are less than T_u. This is exactly the distribution that we want for an exponential-distribution PRNG, and so using (6) lets us generate exponentially distributed pseudorandom numbers.

4.2. **Chopping the Tail.** Since we plan to use the exponential-distribution PRNG to generate a sequence of "interdeath" delays, there is a practical issue. It is quite unlikely but still possible that such a PRNG occasionally will generate a very large delay, up to

several days. To avoid the appearance of a counter stalling for days and appearing broken, we plan on limiting the tail of the distribution.[4]

4.2.1. *Lower and Upper Limits.* Limiting the tail will reduce the mean, making $\overline{T} < \frac{1}{\lambda}$. In addition to being mathematically wrong, such a change will cause the counters to run more quickly than planned. To compensate, we also must limit the distribution at the other end of the mean. We define the distribution's new limits as $T_a = \frac{a}{\lambda}$ and $T_b = \frac{b}{\lambda}$, where $\frac{a}{\lambda} < \frac{1}{\lambda} < \frac{b}{\lambda}$, and then compute the relationship between them using the definition of the mean $\overline{T}$ over the interval $[T_a, T_b]$ of a continuous distribution.

$$\overline{T} = \frac{\int_{T_a}^{T_b} T f(T) dT}{\int_{T_a}^{T_b} f(T) dT}$$

Substituting in the PDF for the exponential distribution:

$$\frac{1}{\lambda} = \frac{\int_{\frac{a}{\lambda}}^{\frac{b}{\lambda}} T\lambda e^{-\lambda T} dT}{\int_{\frac{a}{\lambda}}^{\frac{b}{\lambda}} \lambda e^{-\lambda T} dT}$$

Integrating:

$$\frac{1}{\lambda} = \frac{-\frac{e^{-\lambda T}(\lambda T+1)}{\lambda}\Big|_{T=\frac{a}{\lambda}}^{\frac{b}{\lambda}}}{-e^{-\lambda T}\Big|_{T=\frac{a}{\lambda}}^{\frac{b}{\lambda}}}$$

$$\frac{1}{\lambda} = \frac{\frac{e^{-a}(a+1)}{\lambda} - \frac{e^{-b}(b+1)}{\lambda}}{-e^{-b} + e^{-a}}$$

The relationship between a and b is independent of λ:

$$1 = \frac{e^{-a}(a+1) - e^{-b}(b+1)}{-e^{-b} + e^{-a}}$$

And finally solve for the relationship between a and b:

$$-e^{-b} + e^{-a} = e^{-a}(a+1) - e^{-b}(b+1)$$
$$-e^{-b} + e^{-a} = ae^{-a} + e^{-a} - be^{-b} - e^{-b}$$
$$0 = ae^{-a} - be^{-b}$$

[4]The case described is quite unlikely. For example, if the a C-program uniform-distribution PRNG generates `RAND_MAX`, then the value of T computed with (2) will be about 21 times larger than the mean $\overline{T}$. If the affected counter normally sums 21 deaths per day, then with this value from the PRNG it will sum only one. The decision to limit the tail will depend on each counter's expected daily total. A counter that will sum 6 deaths per day ought to chop the tail since generating the maximum T would stall the counter for three days. On the other hand, the effect on counters that sum thousands of deaths per day will not be noticeable.

$$ae^{-a} = be^{-b} \tag{7}$$

$$\frac{a}{e^a} = \frac{b}{e^b}$$

$$ae^{b} = be^{a} \tag{8}$$

Theses results verify our statements in Section 3.3.

Table 2 contains some useful limits, based on selected values of b. To compute column a from column b in the table, we can use numerical methods, or we can use the Lambert W function,[5] which is defined as the function that returns z when given ze^z:

$$z = W(ze^z) \tag{9}$$

For any value of b, we want to solve for a. Start with (7).

$$ae^{-a} = be^{-b}$$

$$-ae^{-a} = -be^{-b}$$

$$W(-ae^{-a}) = W(-be^{-b})$$

Apply (9) with $z = -a$:

$$-a = W(-be^{-b})$$

$$a = -W(-be^{-b}) \tag{10}$$

Several computer algebra systems will evaluate an equation like (10) that contains the Lambert W function. We used WolframAlpha.[6] As an example, to obtain the value of a for $b = 2$ we entered this expression. (Be aware that WolframAlpha uses square brackets for function arguments.)

```
-LambertW[-2 Exp[-2]]
```

Values for the rest of column a in Table 2 were obtained using this WolframAlpha expression:

```
Table[{N[-LambertW[-b Exp[-b]]], b}, {b, {2, 3, 4, 5, 6, 7, 10, 20}}]
```

4.2.2. *Upper Limit Only.* The calculations for Table 3 derive the mean $\overline{T}$ over the interval $[0, T_b]$ of a continuous distribution.

$$\overline{T} = \frac{\int_0^{T_b} T f(T) dT}{\int_0^{T_b} f(T) dT}$$

[5] https://en.wikipedia.org/wiki/Lambert_W_function

[6] https://www.wolframalpha.com

Substituting in the PDF for the exponential distribution:

$$\frac{1}{\lambda} = \frac{\int_0^{\frac{b}{\lambda}} Tm\lambda e^{-m\lambda T} dT}{\int_0^{\frac{b}{\lambda}} m\lambda e^{-m\lambda T} dT}$$

Integrating:

$$\frac{1}{\lambda} = \frac{-\frac{e^{-m\lambda T}(m\lambda T+1)}{m\lambda}\Big|_{T=0}^{\frac{b}{\lambda}}}{-e^{-m\lambda T}|_{T=0}^{\frac{b}{\lambda}}}$$

$$\frac{1}{\lambda} = \frac{-\frac{e^{-mb}(mb+1)}{m\lambda} + \frac{1}{m\lambda}}{-e^{-mb}+1}$$

The relationship between a and b is independent of λ:

$$1 = \frac{-\frac{e^{-mb}(mb+1)}{m} + \frac{1}{m}}{-e^{-mb}+1}$$

Simplify:

$$\begin{aligned} -e^{-mb}+1 &= -\frac{e^{-mb}(mb+1)}{m} + \frac{1}{m} \\ m(-e^{-mb}+1) &= -e^{-mb}(mb+1)+1 \\ -me^{-mb}+m &= -mbe^{-mb} - e^{-mb} + 1 \\ -m + me^{mb} &= -mb - 1 + e^{mb} \end{aligned}$$

$$e^{mb}(m-1) + mb - m + 1 = 0 \tag{11}$$

For a given b one must solve (11) for m numerically.