Exalted Probability Study for Funsies

Blackwell

March 21, 2016

Introduction

I'll be using this document to compiling my findings as I dig through the various probabilities associated with Exalted.

Die-Rolling Function

I found a neat R package called **discreteRV** that lets you work with custom discrete random variables. This is super handy because unlike most common probability distributions, the results of a die roll are discrete. So, I built a function that defines the sides of the die and the odds of seeing each side, based on inputs of target number, which numbers to double, which numbers get rerolled indefinitely and which ones get rerolled once.[1] These all default to the Exalted standard: target number of 7, double 10, no rerolls.

[1] Future work here to account for reroll-and-keep-successes; this is more complicated because it can't be accounted for simply by changing the probability of each face appearing.

```
library(knitr)
library(discreteRV)
library(foreach)

build_a_die <- function(tn = 7, doubles = c(10),
    reroll_all = NA, reroll_once = NA) {
    face <- c(1:10)
    result <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    odds <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    die <- data.frame(face, result, odds)
    die[die$face >= tn, ]$result <- 1
    die[die$face %in% doubles, ]$result <- 2
    if (!is.na(reroll_all)) {
        die <- die[-reroll_all, ]
    }

    die$odds <- 1/length(die$face)
    if (!is.na(reroll_once)) {
        die[die$face %in% reroll_once, ]$odds <- 0
        die$odds <- die$odds + (sum(die$face %in%
            reroll_once)/length(die$face)^2)
    }
    a_roll <- RV(die$result, die$odds)
```

```
    return(a_roll)
}

a_roll <- build_a_die()


E(a_roll)

## [1] 0.5

V(a_roll)

## [1] 0.45

rr1_die <- build_a_die(reroll_all = 1)
E(rr1_die) * 18

## [1] 10

`?`(var())


d = 27
E(SofIID(a_roll, d))

## [1] 13.5

E(SofIID(rr1_die, d))

## [1] 15

E(SofIID(a_roll, d + 3))

## [1] 15

V(SofIID(a_roll, d))

## [1] 12.15

V(SofIID(rr1_die, d))

## [1] 12.66666

V(SofIID(a_roll, d + 3))

## [1] 13.49999

E
```

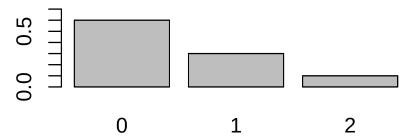

Figure 1: Result Histogram, Single Die, Normal Conditions

```
## function (X)
## {
##     isjoint <- length(grep(",", X)) > 0
##     if (isjoint) {
##         val <- lapply(strsplit(outcomes(X), ","), function(test) {
##             prod(as.numeric(test)) * P(X == (paste(test, collapse = ",")))
##         })
##         return(sum(unlist(as.numeric(val))))
##     }
##     return(sum(as.numeric(X) * probs(X)))
## }
## <environment: namespace:discreteRV>
```

```
a_roll
```

```
## Random variable with 3 outcomes
##
## Outcomes    0    1    2
## Probs     3/5 3/10 1/10
```

```
rr1_die
```

```
## Random variable with 3 outcomes
##
## Outcomes   0   1   2
## Probs    5/9 1/3 1/9
```

```
var(c(0, 0, 0, 0, 0, 0, 1, 1, 1, 2))
```

```
## [1] 0.5
```

```
var(c(0, 0, 0, 0, 0, 1, 1, 1, 2))
```

```
## [1] 0.5277778
```

The Basics

HAVING RIGOROUSLY DEFINED A SINGLE DIE, we can use **discreteRV::SofIID** to calculate the probability of any given number of successes. This works because we defined "result" as the number of successes the face was worth, rather than the number on the face. The RV function took care of reducing our individual faces down to distinct results. A single die roll (our "a_roll" from above) looks like this:

```
## Random variable with 3 outcomes
##
## Outcomes    0    1    2
## Probs     3/5 3/10 1/10
```

A two-die pool is computed like this:

```
SofIID(a_roll, 2)
```

```
## Random variable with 5 outcomes
##
## Outcomes      0      1      2      3      4
## Probs      9/25   9/25 21/100   3/50  1/100
```

Now that we have this random variable, it's trivial to compute the probabilty of succeeding at a given difficulty for a given number of dice. Lets build a table of sample dice pools against difficultes from 1 to 10.

Table 1: Odds of success with a standard die (target number 7, double 10s)

	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7	Diff 8	Diff 9	Diff 10
5d	0.922	0.728	0.469	0.242	0.099	0.032	0.008	0.001	0.000	0.000
10d	0.994	0.964	0.886	0.750	0.572	0.388	0.233	0.123	0.057	0.023
15d	1.000	0.996	0.982	0.948	0.879	0.773	0.635	0.483	0.337	0.216
20d	1.000	1.000	0.998	0.991	0.975	0.941	0.881	0.794	0.680	0.551
25d	1.000	1.000	1.000	0.999	0.996	0.988	0.970	0.938	0.887	0.812

This is a matrix of results rerolling 1s:

Table 2: Odds of success with indefinitely-rerolled 1s (target number 7, double 10s, reroll 1s)

	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7	Diff 8	Diff 9	Diff 10
5d	0.947	0.788	0.545	0.304	0.134	0.046	0.012	0.002	0.000	0.000
10d	0.997	0.980	0.929	0.827	0.673	0.492	0.320	0.183	0.092	0.040
15d	1.000	0.999	0.992	0.974	0.933	0.858	0.748	0.610	0.461	0.320
20d	1.000	1.000	0.999	0.997	0.990	0.973	0.940	0.884	0.800	0.692
25d	1.000	1.000	1.000	1.000	0.999	0.996	0.989	0.975	0.948	0.904

With two matrices, it's a simple matter to see the difference:

Table 3: Change in odds of rerolled 1s over baseline

	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7	Diff 8	Diff 9	Diff 10
5d	0.025	0.060	0.076	0.062	0.035	0.014	0.004	0.001	0.000	0.000
10d	0.003	0.017	0.044	0.077	0.101	0.104	0.087	0.060	0.035	0.017
15d	0.000	0.003	0.010	0.027	0.053	0.086	0.113	0.127	0.123	0.104
20d	0.000	0.000	0.002	0.006	0.015	0.033	0.059	0.090	0.120	0.140
25d	0.000	0.000	0.000	0.001	0.003	0.008	0.019	0.036	0.061	0.091

When you're talking about static difficulty (rather than counting excess successes) it's often interesting to see this in context with the **relative** difference; how much do my odds of success improve?

Table 4: % change in odds of success rerolled 1s over baseline

	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7	Diff 8	Diff 9	Diff 10
5d	0.027	0.083	0.163	0.255	0.355	0.454	0.550	0.631	0.694	0.694
10d	0.003	0.017	0.049	0.103	0.176	0.267	0.373	0.490	0.618	0.754
15d	0.000	0.003	0.010	0.028	0.061	0.111	0.178	0.264	0.365	0.481
20d	0.000	0.000	0.002	0.006	0.016	0.035	0.067	0.113	0.176	0.255
25d	0.000	0.000	0.000	0.001	0.003	0.009	0.020	0.039	0.069	0.113

But sometimes even more interesting is flipping the ratios around and looking at the way the odds of failure change:

Table 5: % change in odds of failure rerolled 1s over baseline

	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7	Diff 8	Diff 9	Diff 10
5d	-0.319	-0.222	-0.143	-0.081	-0.039	-0.015	-0.004	-0.001	0.000	0.000
10d	-0.537	-0.460	-0.383	-0.308	-0.235	-0.170	-0.113	-0.069	-0.037	-0.018
15d	-0.685	-0.629	-0.570	-0.508	-0.443	-0.377	-0.310	-0.246	-0.186	-0.133
20d	-0.785	-0.746	-0.704	-0.657	-0.607	-0.552	-0.495	-0.435	-0.374	-0.313
25d	-0.850	-0.827	-0.797	-0.763	-0.726	-0.685	-0.641	-0.593	-0.541	-0.487

Some Specific Questions

We now have all the tools we need to start exploring.

Excellent Strike vs Melee Excellency

After this post by Swooper: `http://forum.theonyxpath.com/forum/main-category/exalted/837333-interesting-numbers?p=841015#post841015`

The Solar Charm **Excellent Strike** works like our example above of indefinite rerolls of 1s. Because it makes the odds of success (or double-success) slightly higher on every die, its relative value scales with the size of the attack dice pool when it is used. For the same cost (3m), **Melee Excellency** adds three dice to the pool. This is extremely impactful when a pool is relatively small, but usually has diminishing returns as the underlying dice pool gets bigger and bigger. Which is better, then, depends on where the break-even point is located, somewhere between "no dice" (where the Excellency is most useful) and "arbitrarily many dice" (where the Excellent Strike is best).

To start, let's imagine that we've got a pool of some specific size, and we're guaging which Charm is better based on our odds of hitting a particular difficulty (this isn't quite how combat works, but we'll get to that in a minute).

Table 6: Absolute difference between odds of success, 3m Melee Excellency minus Excellent Strike

	Diff 2	Diff 4	Diff 6	Diff 8	Diff 10	Diff 12	Diff 14	Diff 16	Diff 18	Diff 20
5d	-0.084	-0.208	-0.094	-0.005	0.002	0.000	0.000	0.000	0.000	0.000
10d	-0.010	-0.082	-0.182	-0.165	-0.072	-0.015	-0.001	0.000	0.000	0.000
15d	-0.001	-0.017	-0.079	-0.169	-0.194	-0.132	-0.056	-0.015	-0.003	0.000
20d	0.000	-0.003	-0.021	-0.076	-0.159	-0.206	-0.177	-0.105	-0.044	-0.013
25d	0.000	0.000	-0.004	-0.023	-0.073	-0.150	-0.209	-0.207	-0.150	-0.081
30d	0.000	0.000	-0.001	-0.005	-0.024	-0.071	-0.142	-0.207	-0.225	-0.188
35d	0.000	0.000	0.000	-0.001	-0.006	-0.025	-0.068	-0.134	-0.202	-0.236
40d	0.000	0.000	0.000	0.000	-0.001	-0.007	-0.025	-0.065	-0.127	-0.196

Table 7: % difference between odds of failure, 3m Melee Excellency minus Excellent Strike

	Diff 2	Diff 4	Diff 6	Diff 8	Diff 10	Diff 12	Diff 14	Diff 16	Diff 18	Diff 20
5d	1.000	0.496	0.119	0.006	-0.002	0.000	0.000	0.000	0.000	0.000
10d	1.000	0.807	0.512	0.246	0.081	0.016	0.001	0.000	0.000	0.000
15d	1.000	0.918	0.754	0.543	0.332	0.163	0.060	0.016	0.003	0.000
20d	1.000	0.962	0.875	0.742	0.577	0.400	0.239	0.118	0.045	0.013
25d	0.998	0.982	0.934	0.856	0.745	0.609	0.457	0.307	0.179	0.087

	Diff 2	Diff 4	Diff 6	Diff 8	Diff 10	Diff 12	Diff 14	Diff 16	Diff 18	Diff 20
30d	0.989	0.990	0.965	0.919	0.850	0.756	0.639	0.506	0.368	0.239
35d	0.764	0.991	0.981	0.954	0.911	0.850	0.769	0.667	0.549	0.422
40d	0.380	0.982	0.989	0.974	0.948	0.909	0.855	0.783	0.694	0.588

Neat! This result was actually quite counterintuitive to me. It appears that, when we're looking at odds-of-hitting-difficulty, **Excellent Strike** doesn't overtake 3m of **Melee Excellency** until you are throwing *lots* of dice at high difficulties, well outside the "core" pools and difficulties of Exalted 3e. Even then, if you're not trying to hit a high difficulty, the chance of failure is already so low that even if the relative difference is large, the absolute difference is negligible. The exact break-even point depends on the pool size or difficulty, but we can at least tell that Melee Excellency is always better up to 20 dice, and Excellent Strike is only meaningfully better above 30 dice and when difficulty is 14+.

But wait! That's not quite the whole story. Attack rolls aren't just about hitting. Successes beyond the difficulty have value. How do these two Charms differ in that respect?

Table 8: Absolute difference between expected extra successes, 3m Melee Excellency minus Excellent Strike

	Def 1	Def 2	Def 3	Def 4	Def 5	Def 6	Def 7	Def 8	Def 9	Def 10	Def 11	Def 12
2d	0.39	0.47	0.43	0.28	0.13	0.04	0.01	0.00	0.00	0.00	0.00	0.00
4d	0.28	0.31	0.34	0.32	0.24	0.14	0.06	0.02	0.01	0.00	0.00	0.00
6d	0.17	0.18	0.20	0.23	0.23	0.19	0.13	0.07	0.03	0.01	0.00	0.00
8d	0.06	0.06	0.07	0.10	0.13	0.14	0.13	0.10	0.07	0.04	0.02	0.01
10d	-0.06	-0.05	-0.05	-0.03	0.00	0.04	0.07	0.08	0.07	0.05	0.03	0.02
12d	-0.17	-0.17	-0.16	-0.15	-0.13	-0.09	-0.05	-0.01	0.02	0.03	0.03	0.03
14d	-0.28	-0.28	-0.28	-0.27	-0.26	-0.23	-0.19	-0.14	-0.08	-0.04	-0.01	0.01
16d	-0.39	-0.39	-0.39	-0.39	-0.38	-0.36	-0.33	-0.28	-0.22	-0.16	-0.10	-0.05
18d	-0.50	-0.50	-0.50	-0.50	-0.49	-0.48	-0.46	-0.42	-0.37	-0.30	-0.23	-0.16
20d	-0.61	-0.61	-0.61	-0.61	-0.61	-0.60	-0.59	-0.56	-0.52	-0.46	-0.38	-0.30
22d	-0.72	-0.72	-0.72	-0.72	-0.72	-0.72	-0.71	-0.69	-0.66	-0.61	-0.55	-0.46
24d	-0.83	-0.83	-0.83	-0.83	-0.83	-0.83	-0.83	-0.82	-0.80	-0.76	-0.71	-0.63
26d	-0.94	-0.94	-0.94	-0.94	-0.94	-0.94	-0.94	-0.94	-0.92	-0.90	-0.86	-0.80
28d	-1.06	-1.06	-1.06	-1.06	-1.06	-1.06	-1.05	-1.05	-1.04	-1.03	-1.00	-0.96
30d	-1.17	-1.17	-1.17	-1.17	-1.17	-1.17	-1.17	-1.16	-1.16	-1.15	-1.13	-1.10

As before, positive numbers indicate that 3 motes put into Melee Excellency have, overall, higher expected number of successes (post-defense). As before, Excellent Strike doesn't surpss extra dice until high pools.

It appears that until you have LOTS of dice Excellent Strike isn't superior to just adding some dice. It does have some fringe benefits that may make it worthwhile, though. It eliminates the (low, but still) chance of a botch. Also, it doesn't run into dice-adder limits.