Response to *Critique of Dream Investigation Results*

Minecraft Speedrunning Team

December 2020

1 Introduction

Before going into the details of the flaws in Dream's response paper, we would like to clarify a few important points.

First of all, the response paper attempts to estimate an entirely different probability from ours, and even then, does so invalidly. That is, its "1 in 10 million" calculation is both invalid and not directly comparable to the "1 in 7.5 trillion" number from the moderator report. Even if the analysis that produced their number were performed correctly, that would not in any way show our analysis to be incorrect. One would have to demonstrate that our statistical techniques are invalid, not just that asking a different question leads to a different answer.

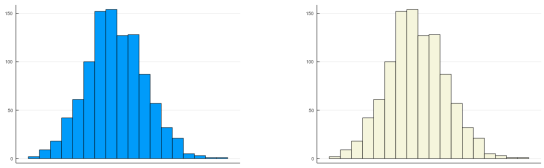
Second, most of the direct criticisms of our analysis in the response paper are blatantly incorrect, disputing the accuracy of extremely standard statistical techniques firmly grounded in probability theory. The only criticism of our analysis which even arguably holds any water is the critique of our choice of 10 as the number of RNG factors to correct for. We strongly disagree that 37 is a suitable number, but even if, despite that, it *were* used, it would not change our conclusion.

2 The Binomial Distribution

Dream's response paper suggested that per-run stopping has to be accounted for, as compared to a binomial distribution with an overall stopping rule. In this section we explain why this is incorrect. We argue that using a binomial distribution with a "worst-case scenario" stopping rule (having a binomial p-value less or equal to Dream's) fully accounts for all stopping rule issues.

The issue can be described as follows. Suppose we have a sequence of Bernoulli trials with probability 0.1, and we stop after the first successful trial. The last trial that we have is necessarily a success, leading to biased results if we assumed a standard fixed-n sampling scheme. The author of Dream's response alleges that Dream's streams are more accurately modeled as the sum of variables with such a negative binomial stopping rule (where each variable corresponds to a run), rather than a single variable with an unknown stopping rule. However, the "stops" that are alleged to be a problem are not true stops. Dream continues speedrunning the next run, and hence the Bernoulli sequence continues. The division of the sequence into "runs" or "streams" is arbitrary and the distribution can be modelled without taking it into account. The only way that having a data-dependent stopping rule *per run* influences the data is by influencing the stopping rule of the *full data*, which was accounted for as admitted in the response paper. For example, the sequence of n negative binomial subsequences that require x successes each is equivalent to a single negative binomial sequence requiring $k = nx$ successes.

Analogously, if you keep flipping a coin until you get heads twice, you are likelier to observe more heads than tails as compared to a fixed number of tosses. However, if you simply take a break after getting two heads and return afterwards, it doesn't affect the numbers whatsoever.



(a) Chunked Negative Binomial (b) Direct Negative Binomial

Figure 1: Distribution Comparison

2.1 Example Simulation

We can illustrate this point with a rather straightforward example. Suppose that we have a sequence of Bernoulli trials succeeding with probability 0.1 each. We stop after 200 successes, which is an *overall* stopping rule at "$k = 200$" — a negative binomial setup. We do this in chunks called "runs" that each have a stopping rule of "stop if $x_{\text{run}} = 2$" where x is the number of successes in that particular run. Effectively, we will stop after successfully completing 100 runs. Here, simulation yields the distribution shown in Figure 1a for the number of trials. However, using the same seed in a simulation of a pure negative binomial setup without per-run stopping yields *the exact same result*, as shown in 1b.

This example illustrates that when the same stopping rule is used overall, the stopping rule of the individual runs do not matter. Again, to reiterate, the "runs" are entirely arbitrary separations. The only way the per-run stopping rule matters is in how it influences the overall stopping rule.

3 Sampling Bias Corrections

The response paper alleges that our bias correction was incorrect. The paper proposes that our correction cannot properly handle "streaks" of successes, and gives some examples to illustrate. However, the numbers given by the paper's author for their own examples are incorrect.

> At first this seems extremely unlikely as the probability of getting 20 heads in a row is $\frac{1}{2^{20}}$, just less than 1 in a million. Applying the Bonferroni correction and saying that there are 80 choices for the starting position of the 20 successful coin tosses in the string of 100 cases gives $\frac{80}{2^{20}} = 7.629 \times 10^{-5}$ or 1 in 13000... The actual odds come out to be about 1 in 6300, clearly better than the supposed "upper limit" calculated using the methodology in the MST Report. This is due to the facts mentioned above: 1) subsets with different p-values are harder to combine and 2) "lucky streaks" are not average randomly chosen samples, but samples that are specifically investigated because they are lucky.

Applying a Sidak correction, like we used, yields a probability of 7.63×10^{-5}, or one in about 13,000, as they noted. However, reading over the page that they linked[1], we can get the exact result of 3.91×10^{-5}, notably smaller than our Sidak correction value. Proceeding with a simple Monte Carlo simulation, just as the response paper does, we run a simulation for 500 million samples and yield a value of 3.86×10^{-5}, or about one in 25,900, again smaller than the value from our correction. It is unclear how the author of Dream's response paper got their values.

The author proceeds to give another example, but it is unclear what they did. They state that they are finding the probability of three consecutive events with probability 0.01, but do not state out of how many trials these events come from. Equation 2 from the response paper was referenced, but this equation does not appear to be relevant here[2]. However, comparing a simple Monte Carlo simulation with 500 million samples again, considering the case of $n = 100$, we find an exact value of 9.70×10^{-5}, and a Monte Carlo value of 9.71×10^{-5}. In contrast, using the same correction as the original paper, we get the larger value of 9.8×10^{-5}. The author seemed to suggest that our correction is inaccurate due to the p-values for various streams or runners being different. However, it is only Dream's combined p-value that is relevant to the correction, and as has been illustrated above, the correction was not shown to be wrong.

[1] `https://mathworld.wolfram.com/Run.html`

[2] Equation 2 from the response paper is a formula for the probability density function for the product of n iid uniform variables

4 Including all 11 streams

Dream's response paper notes that:

> However, as is discussed throughout this document, choosing to put a break point between the streams after seeing the probabilities would require including a correction for the bias of knowing this result.

This implies that we did not correct for this bias, but we did, as per section 8.2 in our initial paper. Dream's response paper concludes that when including all 11 streams in the analysis, there is "no statistically significant evidence that Dream was modifying the probabilities". This result is expected and meaningless, as Dream is only accused of using a modified game for the last 6 streams; including all streams dilutes the data, yielding inconsistent results.

5 Correction Across Runners

The rebuttal paper states:

> In Section 8.3, they claim that their calculation of p is for a runner within their entire speedrunning career. This is presumably based on the argument from Section 8.2 that they have already corrected for every possible subset of streams... Further, that correction was based on choosing 6 of 11 livestream events from Dream, suggesting that their definition of "career" is 11 multi-hour livestream events comprising about 50 runs.

This is incorrect. The p-value this process generates is the probability that results as extreme as Dream's are obtained if one chooses the most extreme sequence of streams from a runner's entire streaming career. The choice of 11 is only due to the fact that this happens to be the amount of times Dream has streamed speedrun attempts — to calculate that value for a different runner, you would use the number of times they had streamed instead of using 11.

The response paper suggests correcting across livestreams instead of individuals. This is redundant, as the p-value outputted, after correcting for the number of streams, is the p-value for Dream's entire livestream history. Were it applied to someone else, it would also be applied to their entire livestream history. Moreover, their estimation of 300 livestreamed runs per day over the past year is highly implausible. Many runs are not livestreamed, and the estimation is based on current numbers, even though Minecraft speedrunning has grown massively in the recent months.

At the time of Dream's run, there were 487 runners who had times in 1.16 – far under 1000 – and the vast majority of these were unpopular or did not stream. Selection bias could only be induced from *observed runners*, so speedrunners who had no significant viewership watching their attempts should not be included. Frankly, there were probably fewer than 50 runners in any version who might've been examined like this, but we used 1000 as an upper bound.

Note that treating whether or not someone is "observed" as a binary value is a simplification: the less likely extreme luck would be noticed for someone, the less they contribute to sampling bias. We included people who have only a handful of viewers in the calculation even though the amount of sampling bias they introduce is likely negligible.

Additionally, note that this is one of the most important factors shifting the number upwards in the response paper. Severely overestimating the number of livestreamed attempts artificially inflates the final number to a massive degree.

6 The number of RNG types

Dream's response paper corrects across 37 different random factors. It is worth noting that, even using this increased number of factors, the final p-value only changes by a factor of 15. If we accepted this list, it would not change our conclusion, but we still hold that the list is seriously flawed.

Dream suggests that eye breaking odds, various mob spawn rates, dragon perch time, triangulation ability, and various seed-based factors should be counted. However, these are more difficult to cheat

than blaze rods and piglin bartering rates, and in some cases are entirely implausible for us to examine. The dominant theory is that Dream cheated by modifying the internal configuration files in his launcher jar file directly. Other methods are possible as well, but this is likely the most straightforward. Using this method, only entity drops and piglin barters can be modified.

Dream offers frequency of triangulation into stronghold as one factor. However, this isn't random at all, and is instead a skill-based factor[3]. Additionally, many of the factors proposed are seed-based. An extensive amount of time would be required to seedfind for enough randomly generatable world seeds for a livestream, making it not a very plausible method for long-term cheating. Further, it is in principle possible to detect set-seeds based on non-seed random factors. As a simplified example, if we clearly know the LCG state at a fixed length from seed generation, we can backstep to seed generation to find what seed *should've* been generated. Frankly, this would be rather difficult to do, but it would be attempted first instead of statistical analysis.

Some suggested factors rely on strategies that were either defunct or nonexistent at the time of Dream's runs. Monuments, and string from barters, are only important for so-called "hypermodern" strategies, which often skip villages and explore the ocean. These strategies did not exist at the time of Dream's run. Similarly, ender pearl trades are practically never used in 1.16 runs due to it being more difficult and slower to get pearls via trades than via barters. As a result, no top runs in 1.16 utilize villager trading.

Finally, some factors occur too rarely to obtain a large enough sample for analysis. For instance, one only gets to the end portal on nearly completed runs, so there would be very few events to check.

Clearly, the 37 number is entirely unrealistic. It relies on the use of strategies that Dream could not have used, and the investigation of factors that we could not investigate. Again though, even if we accept the full 37 number, it only changes our result by a factor of 15 – not enough to change our conclusion.

[3] How well a player can triangulate based on eye throws.

7 Paradigm Inconsistency

In section 4.2 of Dream's response paper, the author explains they use the Bayesian statistics paradigm instead of the hypothesis testing paradigm used in our report. That is, Dream's response paper attempts to calculate the probability that Dream cheated given the bartering and blaze data; in contrast, our paper calculates the probability of obtaining bartering and blaze results at least as extreme as Dream's under the assumption the game is unmodified. These are entirely different probabilities, but Dream's response paper confuses the two paradigms throughout, producing an uninterpretable result.

7.1 Unclear Corrections

Dream's response paper mimics many of the bias corrections in our original paper, but because the starting value is the posterior probability of an unmodified game and not a p-value, some of these corrections are unjustified. Indeed, it is not trivially obvious that frequentist p-value corrections can be applied to such a probability.

Dream's response paper attempts to correct for the stopping rule. This is perfectly fine under a frequentist paradigm like we used. However, it is inconsistent with the Bayesian paradigm used in the response paper. Bayesians follow the likelihood principle, such that changes to the likelihood by a factor that does not depend on the parameter of interest do not change the results. A well-known feature of the likelihood principle is that stopping rules are irrelevant to analyses that use methods following it. Hence, the author should not have accounted for stopping rules at all, including the dropping of the last data point. Indeed, the response paper itself stated that one of the reasons why a Bayesian approach was used is to avoid having to model the stopping rule of each run. However, despite this statement, the author goes on to drop the last data point in attempt to address the stopping rule.

Similarly, the response paper attempts to correct for selection bias *across runners*. This is rather odd, as the goal of these corrections is to control error rates,

a goal that is not shared with Bayesian methods[4]. The likelihoods across individuals are independent of one another, and therefore comparisons across other individuals are irrelevant to a Bayesian analysis.

7.2 Invalid Comparison

The final conclusion of Dream's response paper conflates the posterior probability with the p-value once more.

> In any case, the conclusion of the MST Report that there is, at best, a 1 in 7.5 trillion chance that Dream did not cheat is too extreme for multiple reasons that have been discussed in this document.

Again, the 1 in 7.5 trillion chance does not represent the probability that Dream did not cheat; it represents the probability of any Minecraft speedrunner to get results at least as extreme as Dream's using an unmodified game while streaming. Widening the scope to any streaming speedrunner already artificially enlarges the p-value in Dream's favor and was only done to prevent accusations of p-hacking and the like.

Even if Dream's response calculation were done correctly, the 1 in 10 million posterior probability would not be directly comparable to the 1 in 7.5 trillion figure and would still imply a 99.99999% chance of Dream cheating.

8 Conclusion

The author of Dream's response paper appears to mix frequentist and Bayesian methods, resulting in an uninterpretable final result. Further, these methods are applied incorrectly, preventing valid conclusions being made. Despite these problems being in Dream's favor, the author presents a probability that still suggests that Dream was using a modified game. Hence, our conclusion remains unchanged.

[4] With the exception of matching priors, although such can hardly be considered Bayesian.

Relevant Links:

By Moderators or Dream

1. *Dream Investigation Results*, original moderator paper.
2. *Critique of Dream Investigation Results*, Dream response paper by Photoexcitation.
3. *Did Dream Fake His Speedruns - Official Moderator Analysis*, Moderator YouTube investigation report.
4. *Did Dream Fake His Speedrun - RESPONSE*, Dream response video.

By Others

5. Reddit r/statistics comment by mfb, a particle physicist with a PhD in physics.
6. *The chances of "lucky streaks"*, a Reddit post by particle physicist mfb.
7. *Dream's cheating scandal - explaining ALL the math simply*, YouTube video by Mathemaniac.
8. Blog post by Professor Andrew Gelman.

A Julia Simulation Code

A.1 Stopping Rule Simulations

```
using Random
using Distributions
using Plots

Random.seed!(1234)
nbsplit = []
for i ∈ 1:1000
    n = 0
    nseq = 0
    while nseq != 100
        x = 0
        while x != 2
            x += rand(Bernoulli(0.1))
            n += 1
        end
        nseq += 1
    end
    push!(nbsplit, n)
end

Random.seed!(1234)
nb = []
for i ∈ 1:1000
    x = 0
    n = 0
    while x != 200
        x += rand(Bernoulli(0.1))
        n += 1
    end
    push!(nb, n)
end


#nb: Direct negative binomial result
#nbsplit: Chunked negative binomial result

println(nb == nbsplit)
```

A.2 Coin Flip Simulation

```
using Random
using Distributed


numruns = @distributed (+) for i ∈
     1:500000000
    x = rand(Bool, 100)

    res = false
    count = 0
    for j ∈ 1:length(x)
        if x[j]
            count += 1
        else
            count = 0
        end

        if count == 20
            res = true
            break
        end
    end
    res
end

# probability is numruns / 500000000
```

A.3 1% Event Simulation

```
using Random
using Distributed
using Distributions

numruns = @distributed (+) for i ∈
     1:500000000
    x = rand(Bernoulli(0.01), 100)

    res = false
    count = 0
    for j ∈ 1:length(x)
        if x[j]
            count += 1
        else
            count = 0
        end

        if count == 3
            res = true
            break
        end
    end
    res
end


# probability is numruns / 500000000
```