Data Science and Ecological Restoration: Using Stats to save Time and Money

This is part 2 of a series on Ecological Restoration. You can find part 1 here[<https://medium.com/@f.vasquez.tavera/data-science-and-ecological-restoration-4-steps-to-action-with-a-real-life-case-study-e5003578b082>].

**Introduction**

Last time, I used basic data science techniques to show how they can be applied towards the results of data gathering. This time, I want to show how the application of statistics can save you time and money from the get go. To clarify: the dataset in part 1 has 2744 data points, which takes a team of two people about 4 days to gather. Is this necessary? Well, technically yes, that particular planting is contractually obligated to replace every dead individual so it’s convenient to gather data while checking for mortality. However, this restoration project has other sites, which aren’t bound by this particular stipulation, and they’re bigger, much bigger. This new dataset that I’m working with has all the same characteristics as the previous dataset, but has roughly 10,000 planted individuals. The same team of 2 people would then require 16 days to gather the same amount of data, which as you can imagine, would really eat into other project goals. Fortunately, it was deemed acceptable to only gather the data of 3986 individuals, so time saved, right?

But can we do better? 3986 is an arbitrary number (just trust me on this one) and is still quite large. Can this sample size be reduced even more, and still capture a reasonable amount of the natural variation in the whole population of plants? Spoilers: the answer is yes. Sample size is a huge topic of attention in scientific research, and the most important parts you can read about in RCSI Sample size handbook[https://www.researchgate.net/publication/324571619\_The\_RCSI\_Sample\_size\_handbook], and to make a long story short, while sources differ, assuming your metrics are normally distributed, an adequate sample size is about 100-300 individuals. This is, however, only applicable for single metrics, and our dataset has 14 metrics that we want to measure, and as we saw before, we might want to see how the distribution changes depending on various factors such as Species, Location, and even how the plants were arranged. You can apply statistical theory to every permutation of these factors, and structure a sample plan based on this, but we already have a large historical data set, so we can run a simulation using approximations of real parameters.

**Method**

What we do here is a variation of a Monte Carlo simulation[https://www.ibm.com/cloud/learn/monte-carlo-simulation]: we know a ‘real’ population mean for any given metric (Ideally, we would know the actual population mean, but with a sample size that large we’re likely to be very close anyway), so now we can simply repeatedly sample an unknown smaller amount of plants that will give us a sufficiently similar statistic. If this sampling gives us a similar mean enough of the time, we can accept that this is a sufficient sample size to capture the variation of the whole population.

Before we run any simulations, we need to define four things: The acceptable range for simulation means, the desired confidence, the number of times we simulate the sampling, and our criteria for sub-dividing the dataset so we properly capture the variation within the population that we want.

The acceptable range is similar to effect size in traditional sampling. If it’s too small, then the only way to satisfy this requirement is to sample every plant, and if it’s too large, then we run significant risk of accepting a sample that doesn’t represent the actual variance within the population. What you consider to be acceptable is a matter of practical significance: depending on your metric you might accept a different margin of error. Since we have 14 different metrics to consider, it’s best to adapt a general heuristic that will consider the natural variation of each metric. Taking a small portion of the standard deviation captures this variance, but this is more easily evaluated by finding the z-score of any given population mean.

If our z-score is lower than a certain value, we can accept that mean. In this case I chose a value of 0.1, which is just under 10% of the natural variation in the population.

Figure: An example of how minimum sample size varies as our acceptable Z-Score range changes. 0.1 is roughly intermediate.

For the number of simulated selections, I chose 10,000 per iteration. This is generally considered an acceptable value for Monte Carlo simulations, but you can go higher if you like.

The desired confidence level is simply how often our sample produces a value that we accept. 95% means we reject 1 in 20 simulated samples, 99% means we reject 1 in 100. If we fail to match this confidence in a simulation for a particular sample size, we simply run the simulation again with a larger sample size. I chose to increment this in intervals of 10. 95% is generally considered sound scientific practice, but I went for 99% in this case to demonstrate the power of this simulation.

Finally, how we choose to sub-divide the dataset is what will determine how complex our model needs to be. For example, if we decide that we need a representative sample of every species, that means we need to simulate a sample for each of the 36 species, and the number of planted individuals ranges from 1 to over 800. This weakens the power of the model somewhat, but let’s see by how much. I chose to simulate three criteria: Species, Scenario, and Type of arrangement. The scenario refers to three different locations: The plants were sown in grassy fields, forest fringes, and within secondary forest. The type of arrangement is binary: There’s a type 1 and a type 2 for each scenario, except for the forest which is uniform, and so those are designated type 0.

As an additional benefit, the dataset contains data from 4 separate sampling events, which gives some repetition to the model to show that it’s consistent.

A final consideration, since we are randomly sampling, will we be sampling with replacement, or without replacement? I chose to sample without replacement since it more appropriately matches the sampling methodology we like to employ in the field.

With all of these criteria established, I wrote a python script to perform this programmatically. If you’re curious, please check it out on my GitHub[https://github.com/Slid61/plant-restoration-size].

The gist of the process is that we performed the simulation for 14 metrics across 123 sample size configurations, which were then plotted against the old size for improvement. Check out this gallery[link] for the full set. We identified then a couple of columns for which the simulation didn’t work well: ‘Flooding’ and ‘Plant not found’. It turns out that these had such few data points in some cases, that the standard deviation was essentially 0. This made the model not work, so I removed them, but justifiably so for two reasons: First, if this happens so rarely, it’s likely not a problem, and second, it makes no sense to simulate something with such few data points anyway.

\*\*\* Figure: This metric performed particularly poorly because of not enough data \*\*\*

Then, to reduce these simulations to a single number, I calculated the interquartile range (IQR) of the new distributions, and any population size that fell outside of 1.5 IQR outside of the mean was treated as an outlier. Then to be safe, I just chose the highest value inside of that range as the new minimum sample size for the entire criterion.

Figure: A visual representation of my selection process for population sizes.

What’s curious is that the spread of values across the board seems to normalize as time goes on. The third and fourth data gathering periods have significantly smaller sample sizes needed to capture their variance.

**Results**

After performing an analysis, we finally arrived at stable results, which can be summarized as such: If we classify by species the minimum sample size was 210 individuals, with an average reduction in size of 22%, when classified by arrangement type, the minimum sample size is much larger at 810, but the average reduction in sample size was 64%, finally when classifying by scenario the minimum sample size was 770, and the average reduction is a whopping 76%. Of course, this isn’t the full picture, because we now have to select that many individuals from each sub-grouping, and add them all up for our new total sample size. This will vary depending on the selection, since sometimes criteria overlap, but one instance generated the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Min Sample Size** | **Total Sample** | **% Improvement** |
| Species | 210 | 2433 | 39% |
| Type | 810 | 1842 | 54% |
| Scenario | 770 | 1236 | 69% |
| All Together | - | 3199 | 20% |
| Original Sample Size | - | 3986 | - |

After our simulations, even with the most complicated criteria we could apply, we still reduced the necessary amount of effort by a whole 20%. That’s a day of work saved. At this point it would be up to the project leaders to decide what is most important, because sometimes dropping a criterion or two can reduce the workload enormously, saving real time and money.

But wait, there’s more!

There’s actually a weakness to this simulation: it’s less effective for metrics that aren’t distributed normally. 9 of our metrics our binary, and a 10th is categorical. If we use only the continuous variables gathered (height, diameter, etc.), and the most recent data, the precision of the model increases even further:

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Min Sample Size** | **Total Sample** | **% Improvement** |
| Species | 180 | 2162 | 46% |
| Type | 110 | 314 | 92% |
| Scenario | 110 | 292 | 93% |
| All Together | - | 2339 | 41% |
| Original Sample Size | - | 3986 | - |

Now we’re looking at a worst-case 41% improvement, which is massive. If you have to sample data multiple times, you’re saving multiple weeks of work in a year. If you choose more permissive criteria, you’re reducing over a week of work to something that can be carried out in a single day.

So why did this model work so much better with just our continuous variables? Height, diameter, and others are normally distributed. We could probably tune this model further by scoring each metric against a different type of distribution, such as an F or Chi-squared distribution. This may further reduce sample size, but it’s likely that we capture enough of the variation with a sample size of 200-300, with over 99% confidence, which is already enough for enormous gains in efficiency.