Statistics calculations

Pétur Ó. Aðalgeirsson

July 2019

1 Introduction

In the simulations we collect some data on various measures, and want to calculate statistics on those measures, such as the average, standard deviation, skewness and kurtosis. For a given sequence $(x_i)_{i\in[n]}$, we use the following formulas:

Average:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Standard deviation:

$$\delta = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2}$$

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$$

Skewness:

$$g = \frac{1}{s^3}\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^3$$

Kurtosis:

$$u = \frac{1}{s^4}\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^4$$

Now, those are simple enough to calculate when we have all the data. So, if we just collect all the data, we can calculate these statistics afterwards.

But for a simulation comparing 10 different electoral systems on a vote table like that of the Finish election of 2015, with 13 constituencies and 12 parties, then we only need to run something on the order of 100.000 rounds of the simulation before the amount of data collected starts taking up Gigabytes of memory.

So, what we want to do instead, is to calculate these statistics on the fly as we run through the simulation, in such a way that we don't need to store the entire sequence of inputs, but only the latest value, using a one-pass algorithm.

For a histogram, and some statistics like perhaps the (exact) median, we would need the whole sequence, of course. But for those mentioned above, it is entirely possible to derive the exact values from a set of intermediate aggregates, that can be updated after each iteration of the simulation.

2 Derivation

For a sequence $(x_i)_{i\in[n]}$, where $[n] := \{1, \dots, n\}$ and $n \in \mathbb{N}$, let us begin by defining four core aggregates:

$$s_k := \sum_{i=1}^{k} x_i \qquad t_k := \sum_{i=1}^{k} x_i^2 \qquad q_k := \sum_{i=1}^{k} x_i^3 \qquad r_k := \sum_{i=1}^{k} x_i^4$$

Note, that all of those can be updated on the fly:

$$\begin{aligned} s_0 &:= 0 & t_0 &:= 0 & q_0 &:= 0 & r_0 &:= 0 \\ s_{k+1} &:= s_k + x_i & t_{k+1} &:= t_k + x_i^2 & q_{k+1} &:= q_k + x_i^3 & r_{k+1} &:= r_k + x_i^4 \end{aligned}$$

Now, the average is simply

$$\mu = \frac{1}{n} s_n$$

And the other statistics can also be calculated directly from these core aggregates. To see that, let us expand the intermediate sums. Define

$$\begin{aligned} d_k &:= \sum_{i=1}^{k} (x_i - \mu)^2 & h_k &:= \sum_{i=1}^{k} (x_i - \mu)^3 & c_k &:= \sum_{i=1}^{k} (x_i - \mu)^4 \\ d &:= d_n & h &:= h_n & c &:= c_n \end{aligned}$$

Then

$$\begin{aligned} d &:= \sum_{i=1}^{n} (x_i - \mu)^2 = \sum_{i=1}^{n} (x_i^2 - 2\mu x_i + \mu^2) = \sum_{i=1}^{n} x_i^2 - 2\mu \sum_{i=1}^{n} x_i + \mu^2 \sum_{i=1}^{n} 1 \\ &= t_n - 2\mu s_n + \mu^2 n = t_n - 2\mu s_n + \mu n \mu = t_n - 2\mu s_n + \mu s_n \\ &= t_n - \mu s_n \\ h &:= \sum_{i=1}^{n} (x_i - \mu)^3 = \sum_{i=1}^{n} (x_i^3 - 3\mu x_i^2 + 3\mu^2 x_i - \mu^3) = \sum_{i=1}^{n} x_i^3 - 3\mu \sum_{i=1}^{n} x_i^2 + 3\mu^2 \sum_{i=1}^{n} x_i - \mu^3 \sum_{i=1}^{n} 1 \\ &= q_n - 3\mu t_n + 3\mu^2 s_n - \mu^3 n = q_n - 3\mu t_n + 3\mu^2 s_n - \mu^2 s_n \\ &= q_n - 3\mu t_n + 2\mu s_n \\ c &:= \sum_{i=1}^{n} (x_i - \mu)^4 = \sum_{i=1}^{n} (x_i^4 - 4\mu x_i^3 + 6\mu^2 x_i^2 - 4\mu^3 x_i + \mu^4) \\ &= \sum_{i=1}^{n} x_i^4 - 4\mu \sum_{i=1}^{n} x_i^3 + 6\mu^2 \sum_{i=1}^{n} x_i^2 - 4\mu^3 \sum_{i=1}^{n} x_i + \mu^4 \sum_{i=1}^{n} 1 = r_n - 4\mu q_n + 6\mu^2 t_n - 4\mu^3 s_n + \mu^4 n \\ &= r_n - 4\mu q_n + 6\mu^2 t_n - 3\mu^3 s_n \end{aligned}$$

And finally

$$\delta = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2} = \sqrt{\frac{d}{n-1}}$$

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2} = \sqrt{\frac{d}{n}}$$

$$g = \frac{1}{s^3}\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^3 = \frac{h}{ns^3} = \frac{h}{n\sqrt{\frac{d}{n}}^3} = h\sqrt{\frac{n}{d^3}}$$

$$u = \frac{1}{s^4}\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^4 = \frac{c}{ns^4} = \frac{c}{n\frac{d^2}{n^2}} = n\frac{c}{d^2}$$

And of course, we can handle partial sums to calculate these statistics on the fly before finishing the simulation, by simply taking n to be the number of iterations run so far. Equivalently, we could define δ_k, g_k and u_k, but then we'd need to rename s, so as not to clash with the already defined s_k's.