

# Need to aggregate t-test stats

---

*Source:* <http://sci.tech-archive.net/Archive/sci.stat.math/2007-12/msg00529.html>

---

- *From:* [adiamond@xxxxxxxxxxxxxxxx](mailto:adiamond@xxxxxxxxxxxxxxxx)
  - *Date:* Thu, 27 Dec 2007 16:14:24 -0800 (PST)
- 

Here's the setup: My main goal is to "bless" a new database (db) such that "blessing" means that the new database isn't "statistically different" from the old db. One way I thought of doing this is to compare the string length (i.e. number of letters) for a given field with respect (w/r) to each db to see if they were the same. However, I know the database was created by combining data from multiple different sources and each record in the db has the source-origin info. So, at least to me, it makes sense to have a strong test that is based off of testing the consistency of the two dbs string lengths for each different source. That is to say, I am partitioning the two samples into multiple sub-samples where each sample has all the records from a different source.

Rather than continue describe this in painful abstract terms, I will make this concrete:

Imagine that both data bases have a "name" field. The names are collected from different countries (i.e. "source"). It may be that one country messed up their data entry and so I want to test the "name" lengths for each country. So, for instance, both databases could have some names from England, China, and France, etc (hundreds of countries) and lets say that just the french source screwed up their data entry.

I can do a two-sample t-test to compare the length of English names in db1 against db2, then another t-test for the Chinese names, and likewise for the french, etc. Now I have hundreds of t-test p values but I need (like?) to combine the results to give a final answer. The "french t-test" has a low p-value but hey, there are hundreds of tests so that could be a (statistically insignificant) coincidence.

Note, the numbers of records from a given country and from each db can be different (and so each t-test has a different t-distribution given that different sample sizes ==> different d.f.)

If these tests were all homogeneous (measuring the same quantity) I might use Bonferonni or the like but that's not the case.

Furthermore, I simply don't like the idea of lowering the alpha's on each t-test when, if the db's were the same (i.e. my Ho null overall hypothesis), I would expect a distribution of values where most of them would have high (safely Ho) p-values (seems like a great way to get false negatives). In fact, following that logic, my intuition

## Need to aggregate t-test stats

suggests that the "right" thing to do is to use the set of t-statistics obtained from all the different (source) tests and compare that to a t-distribution; if it fits then even if some p-values violate a typical per-test confidence limit I wouldn't care (e.g. if you run a million tests, a single p-value of 0.01 doesn't say anything by itself but if most of them were 0.01...).

Anyway, the above is a non-starter because, as mentioned above, t distributions are a function sample size/degrees of freedom and those are different for each test (country). So, I can't from a single sampled t-distribution.

Another possibility that intuitively feels promising is to somehow work with the final p-values. After all, regardless of the underlying statistic (T vs. <put your stat here>), given a certain number of tests there should only be so many low p-values. Unfortunately, I don't yet see how to put that into practice (assuming it's even sensible!).

Maybe I can put this into a two-way anova framework. That is, each mean should be the same as it's counter part in the other db where the influence of the source has been factored.

.