Lecture 29 — Software Statistics

Eric David Hollbach & Mahesh Tripunitara
edhollbach@uwaterloo.ca & tripunit@uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

November 14, 2015

The Setting

The Java SDK specifies an interface called `Set`.

Its semantics is the mathematical notion of a set:
an unordered collection of unique items.

Several implementations of `Set` are provided by the SDK.

Are they all the same?
If not, which one should we use?

The Setting

They are not identical, so we expect that the answer is:
“it depends on your application.”

We need comparison criteria.

We want these two operations to be fast:

1. Addition; and
2. Membership-checking;

The Setting

These criteria, in turn, provide an objective way of evaluating if one implementation is better than another.

The goal is more modest than the absolute criterion of “fast”.

We are just figuring out if one set-implementation is “better than” (faster than) another in these two operations.

Evaluating Implementations

The question now is: how can we evaluate two given implementations in a rigorous and consistent way?

Statistics!

Alice's Process

1. Randomly choose a bunch of integers (let's say 100).
2. Measure the time it takes to add each of those integers to an instance of a set-implementation I_1, starting at $I_1 = \emptyset$. Compute the average time, a_1.
3. Repeat for an instance of another set-implementation I_2 to get the average a_2 for it.
4. If $a_1 < a_2$, conclude that the first set-implementation is better than the second from the standpoint of additions to the set.

Problems with the Above

- How do we know that the random set of integers Alice chooses is representative?
- Do we have to consider the order in which we add the integers?
- Why is it meaningful to use the average to draw our conclusion?
- How do we know that the averages are *statistically significant*?
- Computers demonstrate considerable variations over time in how long they take to perform an operation.

Addressing the Problems

Ideally, we will have good answers to all of the above questions.

Even if we are unable to conclusively address a concern such as the above, it is very important to recognize the concern.

And at the minimum articulate an assumption explicitly.

Statistics Basics

A (mathematical) *function* is defined as follows.

Given two sets, a domain and a range, a function associates a member of the range with every member of the domain.

We can associate the time it takes to add or membership-check an element against a set as a *random variable*.

Random Variables

A random variable is a function f: $E \longrightarrow \mathbb{R}$
E is a set of *events*, and
$\mathbb{R}$ is the set of real numbers.

The range is the time it takes to add a member or check membership.

In this example, E is the set of all integers.

Random Variables

We adopt two random variables:
a (adding), and
m (membership-checking).

Each maps an integer to a time-value.

We associate a random variable with a *probability distribution*.

Probability Distribution

A probability distribution is itself a function. It maps each event in E to a probability that it occurs.

For example, it is possible that the probability that we add the integer 15 to a set, is different from the probability that we add 32.

Thus, the probability we associate with the event $\langle 15 \rangle$ is different from the probability we associate with the event $\langle 32 \rangle$.

Using our Random Variables

If we can discover this $a(e)$ and $m(e)$ for a "typical" $e \in E$ for two different set-implementations, then we can draw a conclusion.

This has two challenges:
what is a "typical" $e \in E$; and
how do we compute a and m even if we know this e?

Characterizing a and m

The domain of a random variable (E, above) is called a *population*.

We need to characterize the random variable

Determine to what value in the range it maps every member of the population; then compute a *central tendency* (*centre*) of those values.

The Centre

A centre, as its name suggests, represents the “middle” of all possible values.

The *average*, the sum of all values divided by the number of values, is a centre.

Other examples of centres are the *median* and *trimean*.

Choice of a Centre

The choice of a centre is crucial.

Not every centre is *robust* to *outliers*.

An outlier is a value that occurs infrequently.

Robustness is the property that a centre is representative of the “middle” notwithstanding the presence of outliers.

Sampling

It is often impractical or even impossible to compute a centre for the entire population.

If the population is all Canadians, determining the value of a random variable for every Canadian may be deemed too expensive.

Consequently, we collect data *samples*.

The procedure by which we collect samples is important.

Otherwise, the samples may not reflect the properties of the population.

Sampling

For a random variable X whose underlying probability distribution is P, a well-accepted property of samples is that they are:

1. *independent and identically distributed* (iid); and
2. each sample, when viewed as a random variable in its own right, has the same distribution P as X.

Sampling Example

Suppose X is the random variable that is the height of a person.

The manner in which we collect samples must reflect the distribution of heights amongst individuals.

If the probability that there exists a person of height h in the entire population is ρ:

The probability that height h is in our set of samples must be ρ.

How To Sample

There are three issues to resolve in sampling:

1. How do we choose a sample?
2. What is our choice for centre? and
3. How do we compute our choice of centre for the random variables a, m given the samples we choose?

Choice of Samples

We do not know what the underlying distribution of integers in the population is.

That is, we do not know what kinds of integers are added/membership-checked in them.

We resolve this issue by assumption.

Aside on Assumption

We would of course like to minimize the number and nature of assumptions in any study.

Practicality may require that we make some.

It is okay to make assumptions, provided we clearly state them.

...and remember them when the time comes to draw conclusions.

Assumptions in the Example

We have restricted ourselves to integer items only.

We make the assumption that every integer is equally likely to be added to, or queried for, in a set.

This restriction & assumption limits the applicability of our results.

Choice of Centre

Our choice of centre is the average, or mean.

We choose this for convenience — it is easy to compute an average.

The mean is not *robust* against outliers.

The median, on the other hand, is robust.

The median of a continuous random variable is a value, m, such that the probability that a given value lies above or below m is $1/2$.

More Assumptions

We assume that the population is distributed normally.

In such a case, the mean and median are the same.

It is possible to test for normality, rather than simply assume it.

We ask the forgiveness of Math profs.

Computing the Centre

Given our above assumptions, how to compute the centre?

Pick random integers to add/query;
check how long it takes;
compute the average for some large number of such trials.

But, it’s never that simple…

Practical Issues in Computing the Centre

The time for individual additions/queries can be so small that the measurement mechanism we use can be highly unreliable.

We resolve this by performing several adds/queries, and measuring the time for those. We have chosen 250,000 adds/queries.

Practical Issues in Computing the Centre

Java VMs demonstrate strange behaviour when running experiments.

Specifically, there is a “warm up” period for every VM after it starts to run that we need to exclude.

A way that to deal with this is to measure a statistic called the coefficient of variance (CoV) across some number of measurements.

Then, check whether this CoV is below some threshold.

Hypothesis Testing

Now we can compare the three implementations of set across each of the two axes, add-time and query-time.

We do this by comparing two implementations at a time.

An effective way to establish this is via a hypothesis test.

Hypothesis Testing

In a hypothesis test, we articulate a hypothesis, that is, an assertion that may be true or false.

We then ask whether the hypothesis should be accepted or rejected.

In statistics, we typically enunciate a *null hypothesis*.

It is called that because it is usually a hypothesis of ‘no effect,’ or ‘no difference.’

Alternate Hypothesis

The complement of the null hypothesis is called an *alternate hypothesis*.

The alternate hypothesis is that there is a difference in performance.

Theories and Facts

It is important to point out that our null hypothesis was posed before we designed the statistical experiments.

Posing hypotheses after looking at data is dangerous, as it can compromise objectivity.

As Sherlock Holmes says...

Threshold

Once we choose a null hypothesis, we choose a probability *p*.

This is the threshold at which we *reject* the null hypothesis and adopt the alternate hypothesis.

The meaning of a value for *p* is:

> *If the null hypothesis is true, then there is a probability p of observing a difference in the averages as large as we observe.*

Size of p

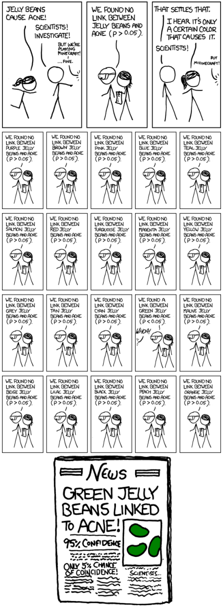
A smaller value for p as threshold is better (suggests more statistical significance).

This is because if the null hypothesis is indeed true, then there is only a small chance of observing the difference.

Therefore:
the fact that we did observe it, and
can presumably reproduce the observations, suggests:

Probabilistically, the null hypothesis is false.

About Statistical Significance


JELLY BEANS CAUSE ACNE!
SCIENTISTS! INVESTIGATE!
BUT WE'RE PLAYING MINECRAFT!
...FINE.
WE FOUND NO LINK BETWEEN JELLY BEANS AND ACNE (P > 0.05).
THAT SETTLES THAT.
I HEAR IT'S ONLY A CERTAIN COLOR THAT CAUSES IT.
SCIENTISTS!
BUT MIIIINECRAFT!
WE FOUND NO LINK BETWEEN PURPLE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BROWN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PINK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLUE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TEAL JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN SALMON JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN RED JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TURQUOISE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN MAGENTA JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN YELLOW JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN GREY JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN CYAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND A LINK BETWEEN GREEN JELLY BEANS AND ACNE (P < 0.05).
WHOA!
WE FOUND NO LINK BETWEEN MAUVE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BEIGE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN LILAC JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLACK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PEACH JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN ORANGE JELLY BEANS AND ACNE (P > 0.05).
NEWS
GREEN JELLY BEANS LINKED TO ACNE!
95% CONFIDENCE
ONLY 5% CHANCE OF COINCIDENCE!
SCIENTISTS...

Choice of Test

There are a number of tests from which one could choose.

The main thing to note here is the prerequisites each tests requires for it to be sound.

For example, a number of tests require that the underlying population is distributed normally.

Student’s t-test

The requirements for the Student’s t-test are:

1. The data is continuous,
2. The data follows a normal probability distribution
3. The samples are independent, and
4. They are simple random samples from the respective populations.

Student’s t-test

(1) is true if we assume that any value for time may occur.

We ensure (3) and (4) from the manner in which we choose the samples.

We satisfied (2) by assumption.

Results

Adding 250,000 integers starting from the empty set.

Implementation	**Average**	**Std. Dev.**
`HashSet`	0.183095	0.00124
`LinkedHashSet`	0.20011	0.000609
`TreeSet`	0.3975	0.00222

Results

Querying each of the 250,000 integers that have been added.

Our queries are elements we know to be in the set only.

Implementation	Average	Std. Dev.
HashSet	0.029934	0.000685
LinkedHashSet	0.083366	0.001148
TreeSet	0.097701	0.002612

Hypothesis Tests

We now articulate null hypothesis, and a choice for the threshold *p* value, and conduct our tests.

We have chosen a two-tailed, unpaired Student's t-test.

Again, take a stats course to understand the test in more depth.

Hypothesis Tests

As p value, we choose, say, 0.01, i.e., 1%.

This is arbitrary, and really depends on what your peers will accept.

In conducting a hypothesis test in the context of the discovery of Higgs Boson, for example, the p value was chosen as "five-sigma".

Five sigma is 5 times the standard deviation, which corresponds to a p threshold value of 3×10^{-7}.

We choose a much more conservative threshold of 0.01. We reject the null hypothesis if the calculated probability is less than 0.01.

Hypothesis Tests

The Student's t-test run with our above mean, standard deviation and *N* (number of samples, which in our case is 30) produces:

> *The two-tailed P value is less than* 0.0001.

That is, the computed probability is less than our threshold probability of 0.01.

Therefore, we reject the null hypothesis!

Our Alternate Hypothesis

Our alternate hypothesis is that there is a difference in performance.

But not that a particular one of those is better than the other.

Informally, our experiments suggest that `HashSet` performs better than `LinkedHashSet`, given all our assumptions and choices.

The Other Comparisons

We can perform the other two comparisons, `HashSet` vs. `TreeSet`, and `LinkedHashSet` vs. `TreeSet`, and get similar results.

A Student's t-test for the null hypothesis, "there is no difference in querying 250,000 members of `LinkedHashSet` and `TreeSet`" using our data
from the table above for querying produces the same output as above:

The two-tailed P value is less than 0.0001.

Multiple Comparisons

Are we done? Unfortunately, not quite.

We have performed a pairwise comparison of three groups.

This is an example of a situation that makes us susceptible to a problem known as *multiple comparisons*.

The problem is we posed multiple hypotheses using the same data.

Therefore, the chance that we see statistical significance in some pairwise comparison is increased purely by chance.

Multiple Comparisons

We have a probability of 0.01 of erroneously rejecting each null hypothesis if indeed the null hypothesis is true.

We have, however, posed three such hypotheses.

Therefore, $\Pr\{\geq 1 \text{ erroneous rejection}\} =$
$1 - \Pr\{\text{no erroneous rejection}\} = 1 - (0.99)^3 \approx 0.03 > 0.01$.

Addressing the Multiple Comparisons Problem

Our strategy is the Bonferroni Correction.

To test k hypotheses with a desired significance level of s, we test each hypothesis at a significance level of s/k.

In our case, $k = 3$, and so we need to test each of our pairwise hypotheses at a threshold of 0.0033

Then the family of tests has a threshold of 0.01.

Bonferroni Correction

In our case, the computed probability based on the data is less than 0.0033 for each test, so our conclusions are remain valid.

The Bonferroni correction is conservative, i.e., overkill, and therefore safe.

A more exact correction can be applied instead.

Details omitted from this lecture.

Closing

This is just an introduction to the use of statistics in evaluating software performance.

The techniques are sound but this is a complicated subject and the brief overview in this lecture is by no means a substitute for a full course on statistics.

It should serve to alert you to the issues.