

Doing Statistics by Rule-of-Thumb.*

Greg Kochanski
<http://kochanski.org/gpk>

2006/08/30 07:59:32 UTC

Statistics:

If your experiment needs statistics, then you ought to have done a better experiment. [Ernest Rutherford (1871–1937); Nobel prize for Chemistry 1908.]

It's certainly true that if your experiment needs statistics, then you ought to have done a better experiment. So, why do statistics?

- Sometimes, you need something to do until you manage to think of that better experiment (and it may be a long wait).
- Sometimes, you need to do an experiment that needs statistics to help you figure out that better experiment.
- Sometimes, you need statistics to know whether or not your experiment needs statistics. As Donald Rumsfeld said[†],

... because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don't know we don't know.

It's these unknown unknowns that can really bite you.

[†]12 Feb 2002, talking about the invasion of Iraq in a news conference.

1 Introduction

In this handout I won't tell you how to do an experiment that doesn't need statistics, but I will tell you how to a lot of the statistics you might ever need, easily, by hand maybe even in your head, using basic arithmetic. Don't be dissuaded by a bit of complication in the middle: it takes some hard work to make a simple estimate.

2 Ingredients of a Statistical Test

Any statistical test compares a hypothesis to some data. If the data and hypothesis match, the hypothesis passes the test, but if the data is too far from the hypothesis, it fails, and we condemn it to haunt the halls of the history

*This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA. This work is available under <http://kochanski.org/gpk/teaching/06010xford>.

of science along with phlogiston [Thall, 2004, Allchin, 1999] and the ether [Rubin, 2006]. So, we care about how much the hypothesis differs from the data; thus we need to subtract

$$\text{Hypothesis} - \text{Data}, \tag{1}$$

To do this, we need to turn our data into a single number so that we can subtract it; normally this is done by taking the mean.

It would be a bit of a coincidence if the data exactly matched the hypothesis, even if the hypothesis is true. Inevitably, experimental errors will lead to some discrepancies. So, we don't want to throw out the hypothesis too easily; we can't throw it out any time there is a small discrepancy. The trick is knowing how big a discrepancy we can allow.

The allowable discrepancy is proportional to the standard deviation for the data, and the constant of proportionality depends on how certain you wish to be. Writing these ideas out formally, we allow ourselves to keep the hypothesis so long as

$$-z \cdot \sigma_{\text{mean}} < H - \text{mean}(D) < z \cdot \sigma_{\text{mean}}, \tag{2}$$

H is the hypothesis, $\text{mean}(D)$ is the data mean, and z is a constant that is related to the degree of certainty that we wish to have. Note that since we are comparing the *mean* of the data to the hypothesis, we need to get the standard deviation of the mean: σ_{mean} . This should not be confused with the standard deviation of the data itself: they differ by a factor of \sqrt{N} where N is the number of data points (the size of the sample). Specifically,

$$\sigma_{\text{mean}} = \frac{1}{\sqrt{N}} \sigma_{\text{data}}. \tag{3}$$

Now, formal z -tests obey this equation, as do Student's t -tests. All we need to do is figure out z . If one wishes to be precise, z varies with the test; one-sided tests differ from two-sided tests t -tests differ from z -tests, t -test differ amongst themselves depending on the sample size, and it all depends on the confidence level required. It sounds like a mess that requires a statistician. But, have no fear. Adopting a milder version of Ernest Rutherford's attitude, we simply take the largest value of z that you're likely to need. Our attitude will be "If the statistical details matter, you ought to do a better experiment."

That way, we may make it a bit too hard to reject hypotheses, and keep a few silly ideas around, but the gain in simplicity is extreme. To the extent we over-estimate the "right" value of z , we'll just have a stricter confidence limit, perhaps $P < 0.01$ or $P < 0.001$ instead of $P < 0.05$.¹

We could take $z = 2.55$, which would give a 95% or better confidence as long as there are five or more data. That would work well as long as we were willing to compute means and standard deviations, but in fact, we are trying to be even lazier than that. We will compute the median instead of the mean, and the inter-quartile range instead of the standard deviation. Consequently, we cannot just look up the equivalent of z - we shall have to compute it ourselves for the equation analogous to Eq. 2:

$$-z \cdot \sigma_m < H - \text{median}(D) < z \cdot \sigma_m. \tag{4}$$

In Equation 4, σ_m will be a quick-and-dirty estimate² of the standard deviation of the median, specifically

$$\sigma_m = R(D)/\sqrt{N}, \tag{5}$$

where $R(D)$ is a quick-and-dirty approximation to the Inter-Quartile Range, which we use as a quick-and-dirty approximation to the standard deviation of the data.

Fortunately, there is a simple way to untangle all these quick-and-dirty approximations and get a precise answer. We will do a Monte-Carlo simulation [Kochanski, 2004] to find out what value of z is sufficient in the worst case, which we'll take to be 6 data points for a two-sided 95% confidence limit. To do this we simply

¹ $P < 0.05$ is a pretty weak confidence limit anyway. It means that *if* you have normally distributed data where each measurement is independent of every other measurement, then 1 out of every 20 of your conclusions will be wrong. If your data isn't independent or is not normal, you could easily be wrong more often. (Personally, I never take a $P < 0.05$ result seriously.) It's best to aim for tighter nominal confidence limit, so that you might still be right even if your data doesn't quite meet the requirements.

² The estimate σ_m captures the correct asymptotic behaviour, that the standard deviation of the median scales as $N^{-1/2}$, but it misses out the alternation between even and odd N . The alternation happens because the median is computed differently for odd N vs. even N .

generate a set of six independent random numbers from a Gaussian distribution, then calculate the inter-quartile range (IQR) and the median. Then, we'll do it 100,000 times and choose a z -value that gives the right answer 95% of the time on a double-sided test. (We'll also do it for sets of seven numbers, just on suspicion, because the median and the IQR are done rather differently for odd vs. even numbers.)

The program can be seen in its entirety in Appendix A. Running it several times, we get $z = 2.63$. Now, by chance, z is pretty close to the square root of 6 ($\sqrt{6} \approx 2.5$), and we'll make use of that coincidence in simplifying Equations 4 and 5. It turns out that you can reject hypothesis H at a 95% confidence level³ or better, if

$$|H - \text{median}(D)| < R(D) \cdot \sqrt{6/N}. \tag{6}$$

(The vertical bars denote the absolute value.) Now, all we have to do is estimate the median and this quasi-Inter Quartile Range simply.

2.1 The Median

The median can be estimated by progressively ticking off the highest and lowest data value until there are only one or two points left. Suppose your data is 1 2 21 4 51 1 22 11 5 2 17 28. Tick off the largest and smallest, and you get 1 2 21 4 51 1 22 11 5 2 17 28, then 1 2 21 4 51 1 22 11 5 2 17 28, then 1 2 21 4 51 1 22 11 5 2 17 28, and so forth, until 1 2 21 4 51 1 22 11 5 2 17 28. The median is then the average of the two remaining items: $(11 + 5)/2 = 16/2 = 8$. It's even easier with an odd number of data: you simply take the single remaining value as the median.

2.2 The Quasi Inter-Quartile Range

This comes from the same procedure. All you have to do is count the data and divide by four (rounding down) to figure out how many points to remove from each end. Again, the data is 1 2 21 4 51 1 22 11 5 2 17 28, so we have $N=12$. $N/4 = 3$, so we remove the three largest and three smallest data. As it turns out, we've already done that in the course of computing the median. It's the third step in the process: 1 2 21 4 51 1 22 11 5 2 17 28. We can then take the difference between the largest and smallest remaining data, which is $21 - 2 = 19$, in this case. That is our $R(D)$.

2.3 That Nasty Square-Root

The trick here is to interpolate between square roots you know. We need to take $\sqrt{6/N} = \sqrt{6/12} = \sqrt{1/2}$. What square roots ought we to know? Well, $\sqrt{1} = 1$ is a good start, and $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, so $\sqrt{1/4} = 1/2$. Now, we know that $1/2$ is in between 1 and $1/4$, so $\sqrt{1/2}$ will be in between 1 and $1/2$. You won't be too far wrong if you take $\sqrt{1/2} = 3/4 = 0.75$ (the true value is 0.707...).

2.4 Example t-Test

Putting it all together, a hypothesis H passes the test against our example data if

$$|H - 8| < 19 \cdot 0.75, \tag{7}$$

and 19 is almost 20, so $3/4$ of 19 is just under 15. Thus, any hypothesis within 15 of 8 (i.e. from -7 to 23) is consistent with the data, and all hypotheses larger than 23 or smaller than -7 can be junked.

3 Comparisons

How does this compare with standard Gaussian statistics? We can do a t-test using the mean and standard deviation, and find that hypotheses from 4.2 to 23.3 are allowed, but any hypotheses smaller than 4.2 or larger than 23.3 are junk.

³ Well, not quite 95% for a two-sided test for $N = 6$, but almost. However, for a one-sided test, you get better than 95% for $N = 6$.

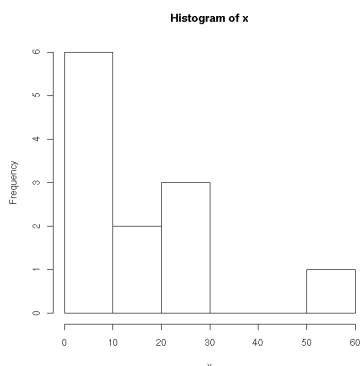


Figure 1: Histogram of sample data. Note the long tail towards the right.

The agreement is not too bad: our confidence interval has a length of 30, and the t-test confidence interval is 19 long. That's not a huge discrepancy, and it's in the right direction. We claimed that this test would be over-cautious sometimes, allowing in some extra hypotheses that more sophisticated tests might exclude, in the interests of simplicity. So, it is good that our confidence interval, from -7 to 23 covers over (almost) all of the t-test confidence interval. We won't reject many hypothesis that the t-test says are consistent with the data.

But, don't jump to the conclusion that the t-test gives the right answer here. T-tests assume that the data is sampled from a Gaussian distribution, but the histogram of the data is strongly non-Gaussian (see Figure 1). We can run the Shapiro-Wilk test in R to see if the distribution is plausibly Gaussian:

```
> x = c(1,2,21,4,51,1,22,11,5,2,17,28)
> shapiro.test(x)
```

Shapiro-Wilk normality test

```
data: x
W = 0.8253, p-value = 0.01845
```

```
>
```

This says it isn't Gaussian, that there is only a 1.8% chance that a Gaussian distribution would give data like x.

We can also compare our confidence interval to some serious statistical tests, like the Wilcoxon Signed Rank Test. This test is nonparametric, which means that it does not assume the data comes from a normal Gaussian distribution. Its results can therefore be trusted for non-Gaussian data. Running that in R, we get:

```
> wilcox.test(x, conf.int=TRUE)
```

Wilcoxon signed rank test with continuity correction

```
data: x
V = 78, p-value = 0.002507
alternative hypothesis: true mu is not equal to 0
95 percent confidence interval:
 2.999945 24.499977
sample estimates:
(pseudo)median
 11.50002
```

Warning messages:

```
1: cannot compute exact p-value with ties in: wilcox.test.default(x, conf.int = TRUE)
```

```
2: cannot compute exact confidence interval with ties in: wilcox.test.default(x, conf.int = TRUE)
>
```

Again, the confidence interval is similar to ours but a little smaller (though bigger than the interval produced by the t -test). Again, it is almost entirely inside the confidence interval we produced with a minimum of effort.

4 Conclusion

We have a trivially simple statistical test that can be done by pencil and paper, or even in the head. It compares well with normal statistical tests, and can be a useful approximation.

References

- Douglas Allchin. Phlogiston after oxygen, 1999. URL <http://www1.umn.edu/ships/updates/after-o2.htm>.
Checked Feb 2006; SHiPs is the “Resource Center for science teachers using Sociology, History and Philosophy of Science”.
- Greg Kochanski, 2004. URL <http://kochanski.org/gpk/teaching/04010xford/MonteCarlo.pdf>.
- Julian Rubin. Michelson-morley: Detecting the ether wind experiment, January 2006. URL <http://www.julianrubin.com/bigten/michelsonmorley.html>. Checked Feb 2006.
- Edwin Thall. Demise of phlogiston, 2004. URL <http://mooni.fccj.org/~ethall/phlogist/phlogist.htm>.
Checked Feb 2006.

A Program for Computing z .

```
#!/usr/bin/env python
```

```
"""This computes a Monte-Carlo estimate of a t-like statistic:
(hypothesis-median)/quasi_Inter_Quartile_Range,
and reports the symmetric 95% confidence interval.
"""

import random
import math

Trials = 100000
N = 7
P = 0.95

def median(x):
    """We compute the median in the obvious way:
    sorting the data, then taking the middle point
    or (if there an even number of points) the average
    of the two midpoints.
    """
    x.sort()
    n = len(x)
    if n%2 == 1:
        return x[n/2]
    return 0.5*(x[(n-1)/2] + x[n/2])

def qIQR(x):
    """This is not quite the inter-quartile range. It's simpler.
    Just count the points, divide by 4 (rounding down), and
    knock that many points off the top and bottom.
    Then, subtract the smallest remaining point from the largest
    remaining point.
    """
    x.sort()
    n = len(x)
    trim = n // 4
    return x[n-1-trim] - x[trim]

if __name__ == '__main__':
    samples = []
    for i in range(Trials):
        x = [random.normalvariate(0.0, 1.0) for k in range(N)]
        med = median(x)
        qiqr = qIQR(x)
        t = med/(qiqr/math.sqrt(N))
        samples.append(abs(t))
    samples.sort()
    m = int(round(P*N))
    print samples[m]
```