

# An Introduction to Confidence Intervals

Timothy A. Gonsalves, IIT Mandi

tag@iitmandi.ac.in

10<sup>th</sup> September 2018

## Experiment Design

We may devise an experiment to estimate some performance measure,  $x$ . If there are no errors, the performance measure would be exact. Let this exact value be  $X$ . In practice, we can never guarantee absence of errors, so we can never measure  $X$  exactly. Thus, the measured value  $x$  is an estimate of the true value  $X$ .

If we run the experiment only once, there may be arbitrary errors in the estimate. For instance, if we are estimating the throughput of a link, a burst of noise cause by lightning during the experiment may result in an abnormally low throughput value.

To increase the accuracy of the result, we can repeat the experiment several times and take the average of the readings:  $\bar{x} = \frac{(x_1 + x_2 + \dots + x_n)}{n}$

Assuming that the same errors do not occur systematically in every experiment, as we repeat the experiment more times, we would expect  $\bar{x}$  to become closer to  $X$ .

## Confidence Levels and Intervals

The method of confidence intervals gives a quantitative estimate of the likely difference between  $\bar{x}$  and  $X$ . It enables us to state that the true value  $X$  is within the range  $(x \pm e)$  with probability  $K$ .

That is:  $Probability[|X - \bar{x}| < e] = K$

$K$  is referred to as the confidence level, and range  $(x - e, x + e)$  is the confidence interval [1]. The confidence interval is often given as a percentage of  $\bar{x}$ . In computer applications, a confidence interval of about 10% of the mean value at a confidence level of 90-95% is often considered acceptable.

Given  $n$  independent readings  $x_1 + x_2 + \dots + x_n$  the confidence interval is given by:

$$e = \frac{st(n-1)}{\sqrt{n}}$$

where  $t(h)$  is the value of the Student t-distribution with  $h$  degrees of freedom and  $s$  is the sample standard deviation of  $\{x_i\}$  given by:

$$s = \sqrt{\frac{\sum (\bar{x} - x_i)^2}{n-1}}$$

Values of the Student's t-distribution,  $t(h)$  are given in Table 1 [2]. Note that with  $n$  readings there are  $n-1$  degrees of freedom, so  $h = n-1$ . We see that as the num-

Table 1:  $t(h=n-1)$  for various  $K, n$

<b>K=</b> <b>n</b>	<b>0.90</b>	<b>0.95</b>	<b>0.99</b>
2	6.31	12.71	63.66
3	2.92	4.30	9.93
4	2.35	3.18	5.84
5	2.13	2.78	4.60
10	1.83	2.26	3.25
20	1.73	2.09	2.86
30	1.70	2.05	2.76
$\infty$	1.65	1.96	2.58

ber of repetitions  $n$  increases from 2 to 5, the value of  $t(n-1)$  decreases rapidly. Thereafter, even with large increase in  $n$ ,  $t(n-1)$  decreases only marginally. Hence, it is worthwhile repeating an experiment say 5 times to reduce the confidence interval, but thereafter the benefit of increased experimental effort is subject to the law of diminishing returns.

## Confidence Intervals in Python

Confidence intervals can be easily computed in Python using the `scipy.stats` module.

```
import scipy.stats as st
x = [2.3, 4.5, 1.3, 8.34, 9.0] # array of "readings"
K = 0.95 # typically, K = 0.90, 0.95, rarely K = 0.99
lo, hi = st.t.interval(K, len(x)-1, loc=st.tmean(x), scale=st.sem(x))
```

In this example, the confidence interval at the 95% confidence level is [minus, plus]

## Plotting Confidence Intervals in Python

To plot confidence intervals as error bars in a line graph, use `matplotlib.pyplot` [3].

```
import matplotlib.pyplot as plt
x = [2.3, 4.5, 1.3, 8.34, 9.0] # N readings
e = [.15, .8, .6, .2, 1.5] # N corresponding conf interval/2
plt.errorbar(range(len(y)), y, fmt=".k", ecolor="red")
```

Note that in the 3<sup>rd</sup> parameter to `errorbar()` the confidence intervals are given as  $(hi-lo)/2$ .

## References

1. Raj Jain, *The Art of Computer Systems Performance Analysis*, Wiley, 1991. Section 13.2 "Confidence Interval for the Mean"
2. For the Student's t-distribution, see Table 26.10 in *Handbook of Mathematical Functions*, eds. Abramowitz & Stegun, National Bureau of Standards, USA, 1972. Also available in many books on probability and statistics.
3. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly, 2016. Section "Visualising Errors", pp. 237-240.

## Acknowledgements

Thanks to Shaifu Gupta for the sample Python code.