

## Lecture 04.12 Confidence

Ahem.<sup>4</sup>

One must have it to give a lecture named it. Confidence is used in the common sense, although we do endow it with a mathematical definition to scare business majors, who aren't actually impressed, but indifferent. Approximately: if, under some reasonable assumptions (probabilistic model), we estimate the probability of some event to be P%, we say we have P% confidence in it. I mean, business majors are all, "Supply and demand? Let's call that a 'law,'" so I think we're even.

So we're back to computing probability from distributions—probability density functions (PDFs) and probability mass functions (PMFs). Usually we care most about estimating the mean of our distribution. Recall from the previous lecture that when several samples are taken, each with its own mean, the mean is itself a random variable—with a mean, of course. **Meanception.**

But, more importantly (just kidding—equally so), the mean has a probability distribution of its own. The *central limit theorem* has as one of its implications that, as the sample size N gets large, *regardless of the sample distributions, this distribution of means approaches the Gaussian distribution.*

But sometimes I always worry I'm being lied to, so let's check.

```
clear; close all; % clear kernel
```

### 04.12.1 Generate some data to test the central limit theorem

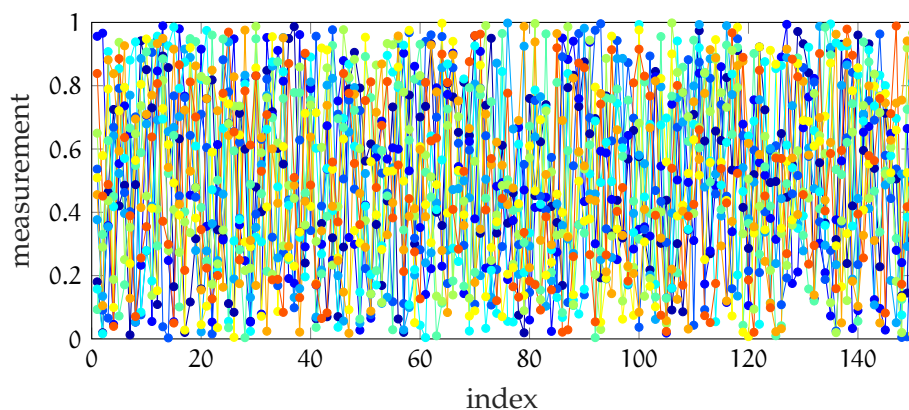
Data can be generated by constructing an array using a (seeded for consistency) random number generator. Let's use a uniformly distributed PDF between 0 and 1.

```
N = 150; % sample size (number of measurements per sample)
M = 120; % number of samples
n = N*M; % total number of measurements
mu_pop = 0.5; % because it's a uniform PDF between 0 and 1

rng(11); % seed the random number generator
```

---

<sup>4</sup>The source for this exercise lecture is in a *Matlab* kernel *Jupyter* notebook. For more information, see [jupyter.org](http://jupyter.org). See [ricopic.one/measurement/code](http://ricopic.one/measurement/code) for the source code in both *Jupyter* notebook and *Matlab* m-file form. Note, however, that simply running the *Matlab* code in the usual m-file environment is easier to get started.



**Figure 04.13:** raw data with colors corresponding to samples.

```
signal_a = rand(N,M); % uniform PDF
size(signal_a) % check the size
```

```
ans =
    150    120
```

Let's take a look at the data by plotting the first ten samples (columns), as shown in [Figure 04.13](#).

This is something like what we might see for continuous measurement data. Now, the histogram.

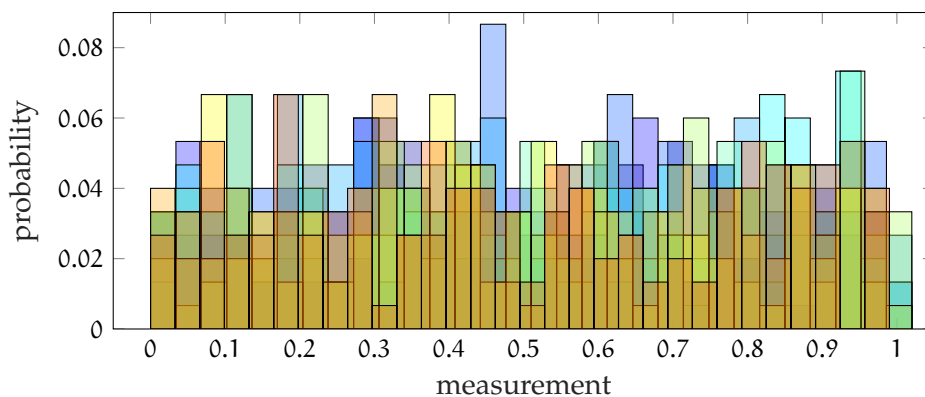
This isn't a great plot, but it shows roughly that each sample is fairly uniformly distributed.

#### 04.12.2 Sample statistics

Now let's check out the sample statistics. We want the sample mean and standard deviation of each column. Let's use the built-in functions `mean` and `std`.

```
mu_a = mean(signal_a,1); % mean of each column
s_a = std(signal_a,1); % std of each column
```

Now we can compute the mean statistics, both the mean of the mean  $\bar{\bar{X}}$  and the standard deviation of the mean  $s_{\bar{X}}$ , which we don't strictly need for



**Figure 04.14:** a histogram showing the approximately uniform distribution of each sample (color).

this part, but we're curious. We choose to use the direct estimate instead of the  $s_X/\sqrt{N}$  formula, but they should be close.

```
mu_mu = mean(mu_a)
s_mu = std(mu_a)
```

```
mu_mu =
    0.4987

s_mu =
    0.0236
```

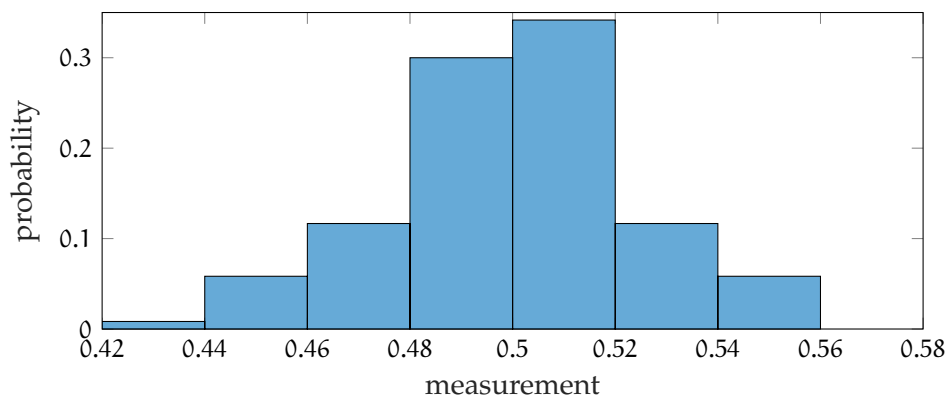
### 04.12.3 The truth about sample means

It's the moment of truth. Let's look at the distribution.

This looks like a Gaussian distribution about the mean of means, so I guess the central limit theorem is legit.

### 04.12.4 Gaussian and probability

We already know how to compute the probability  $P$  a value of a random variable  $X$  lies in a certain interval from a PMF or PDF (the sum or the integral, respectively). This means that, for sufficiently large sample size



**Figure 04.15:** a histogram showing the approximately normal distribution of the means.

$N$  such that we can assume from the central limit theorem that the sample means  $\bar{x}_i$  are normally distributed, *the probability a sample mean value  $\bar{x}_i$  is in a certain interval* is given by integrating the Gaussian PDF. The Gaussian PDF for random variable  $Y$  representing the sample means is

where  $\mu$  is the population mean and  $\sigma$  is the population standard deviation.

The integral of  $f$  over some interval is the probability a value will be in that interval. Unfortunately, that integral is uncool. It gives rise to the definition of the *error function*, which, for the Gaussian random variable  $Y$ , is

This expresses the probability a sample mean being in the interval  $[-y_b, y_b]$  if  $Y$  has mean 0 and variance 1/2.

Matlab has this built-in as `erf`, shown in [Figure 04.16](#).

We could deal directly with the error function, but most people don't and we're weird enough, as it is. Instead, people use the *Gaussian cumulative distribution function*  $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ , which is defined as

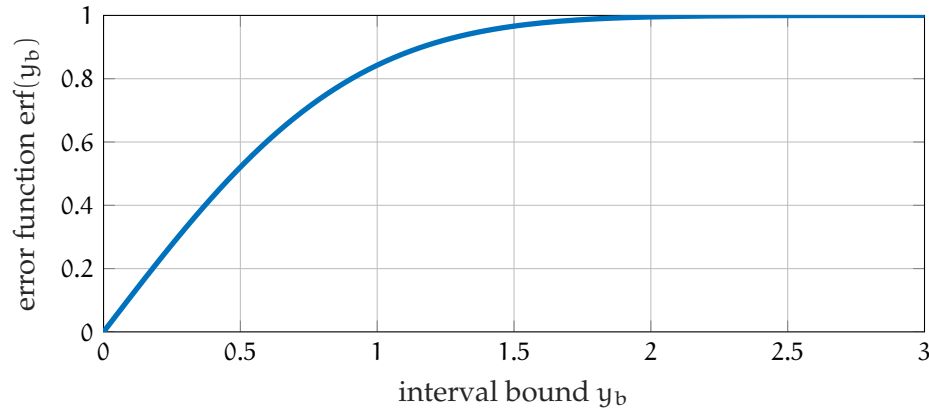


Figure 04.16: the error function.

and which expresses the probability of a Gaussian random variable  $Z$  with mean 0 and standard deviation 1 taking on a value in the interval  $(-\infty, z]$ .

That's great and all, but occasionally we have Gaussian random variables with nonzero means and nonunity standard deviations. It turns out we can shift any Gaussian random variable by its mean and scale it by its standard deviation to make it have zero mean and standard deviation. We can then use  $\Phi$  and interpret the results as being relative to the mean and standard deviation, using phrases like "the probability it is within two standard deviations of its mean." The transformed random variable  $Z$  and its values  $z$  are sometimes called the *z-score*. For a particular value  $x$  of a random variable  $X$ , we can compute its *z-score* (or value  $z$  of random variable  $Z$ ) with the formula

and compute the probability of  $X$  taking on a value within the interval, say,  $x \in [x_{b-}, x_{b+}]$  from the table. (Sample statistics  $\bar{X}$  and  $S_X$  are appropriate when population statistics are unknown.)

For instance, compute the probability a Gaussian random variable  $X$  with  $\mu_X = 5$  and  $\sigma_X = 2.34$  takes on a value within the interval  $x \in [3, 6]$ .

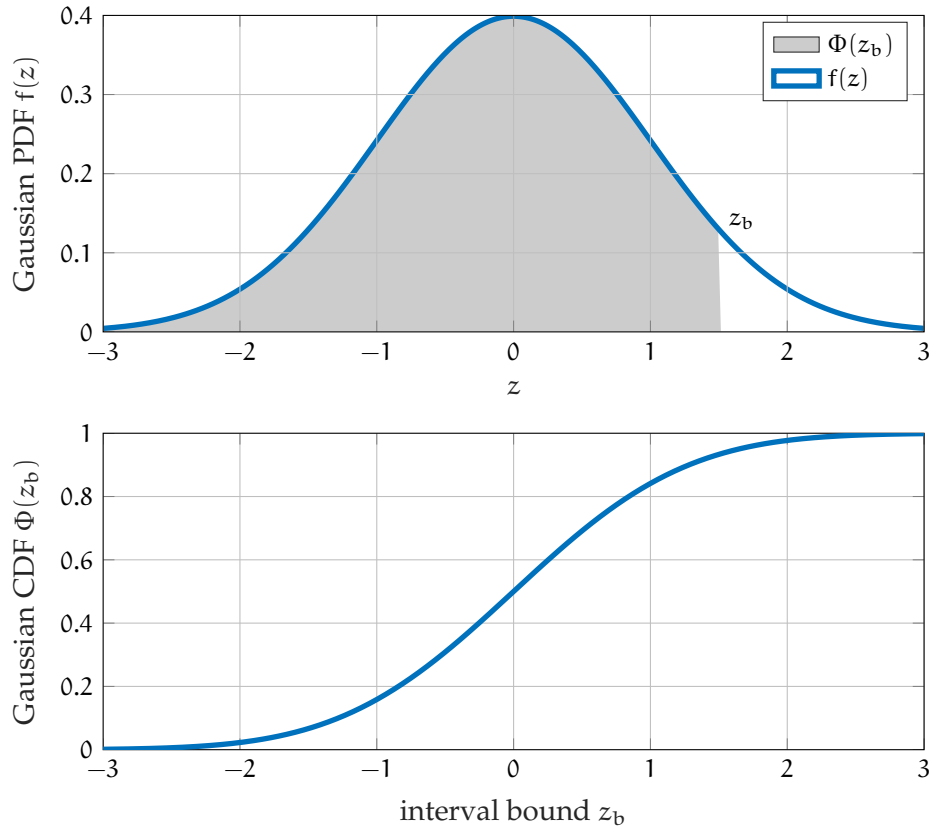


Figure 04.17: the Gaussian PDF and CDF for z-scores.

1. Compute the z-score of each endpoint of the interval:
  
2. Look up the CDF values for  $z_3$  and  $z_6$ , which are  $\Phi(z_3) = 0.1977$  and  $\Phi(z_6) = 0.6664$ .
3. The CDF values correspond to the probabilities  $x < 3$  and  $x < 6$ . Therefore, to find the probability  $x$  lies in that interval, we subtract the lower bound probability:

So there is a 46.89% probability, and therefore we have 46.89% confidence, that  $x \in [3, 6]$ .

Often we want to go the other way, estimating the symmetric interval  $[x_{b-}, x_{b+}]$  for which there is a given probability. In this case, we first look up the  $z$ -score corresponding to a certain probability. For concreteness, given the same population statistics above, let's find the symmetric interval  $[x_{b-}, x_{b+}]$  over which we have 90% confidence. From the table, we want two, symmetric  $z$ -scores that have CDF-value difference 0.9. Or, in maths,

Due to the latter relation and the additional fact that the Gaussian CDF has antisymmetry,

Adding the two  $\Phi$  equations,

and  $\Phi(z_{b-}) = 0.05$ . From the table, these correspond (with a linear interpolation) to  $z_b = z_{b+} = -z_{b-} \approx 1.645$ . All that remains is to solve the  $z$ -score formula for  $x$ :

From this,

and  $X$  has a 90% confidence interval [1.151, 8.849].

#### 04.12.5 Example: Gaussian confidence for a mean

Consider the data set generated above. What is our 95% confidence interval in our estimate of the mean?

Assuming we have a sufficiently large data set, the distribution of means is approximately Gaussian. Following the same logic as above, we need  $z$ -score that gives an upper CDF value of . From the table, we obtain the  $z_b = z_{b+} = -z_{b-}$ , below.

$$z_b = 1.96;$$

Now we can estimate the mean using our sample and mean statistics,

$$\text{mu\_x\_95} = \text{mu\_mu} + [-z_b, z_b] * s\_mu$$

$$\text{mu\_x\_95} =$$

0.4526	0.5449
--------	--------

This is our 95% confidence interval in our estimate of the mean.