Probabilistic Modeling: Fitting Distributions to Data

Probability as a Modeling Tool

So far in this course we have clustered data by measuring **distances** between points.

Today we start treating data as the outcome of a **random process** and ask a different question:

Lecture Overview

Four things we *do* with probability:

Fitting a Distribution to Data

A First Dataset: Boston Daily Temperatures

We will use daily temperature records for Boston Logan International Airport from NOAA, station `USW00014739`, going back to 1936.

For each day we have the maximum and minimum temperature. We take the daily mean temperature to be their average.

▶ Download Boston daily temperatures from NOAA (helper)

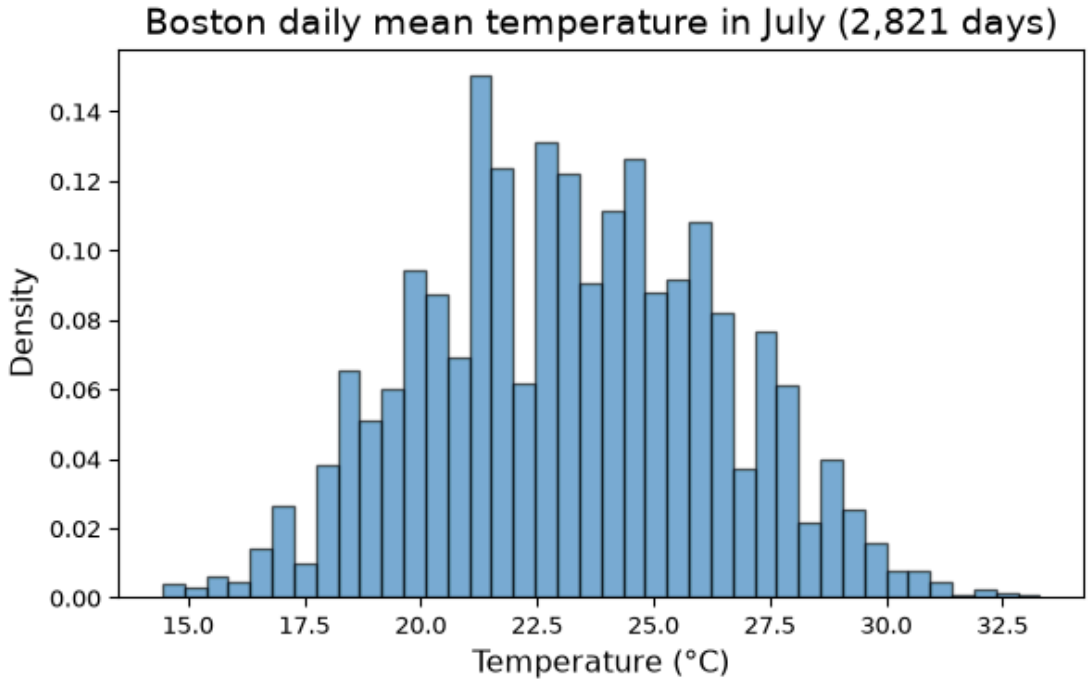
```
33,099 daily records, 1936 - 2026
```

	DATE	TMAX	TMIN	TMEAN
0	1936-01-01	1.7	-6.1	-2.20
1	1936-01-02	1.7	-6.1	-2.20
2	1936-01-03	12.2	1.7	6.95
3	1936-01-04	7.8	1.7	4.75
4	1936-01-05	6.1	0.6	3.35

Look at the Data First

Let’s pull out all the July days – roughly 90 years times 31 days – and look at how the daily mean temperature is distributed.

▶ Code

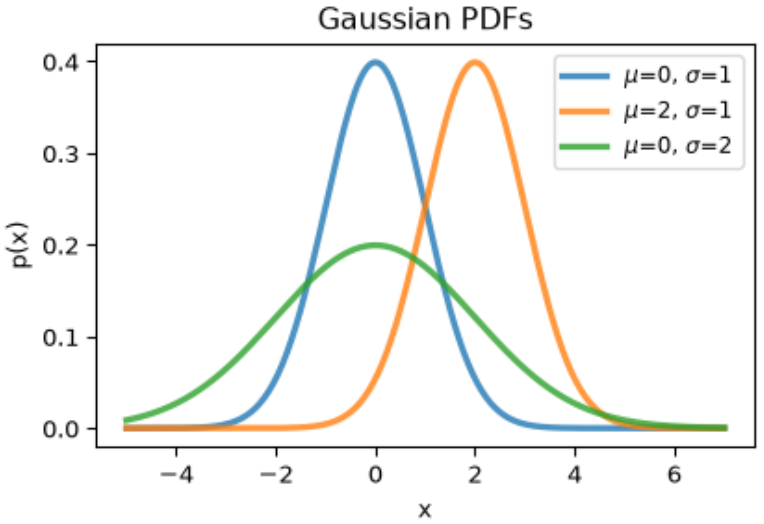


The Model: The Gaussian Distribution

The Gaussian (or Normal) distribution is the workhorse of probabilistic modeling.

A Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ has probability density function

$$p_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$



It has exactly **two parameters**:

- μ, the **mean** – where the bell is centered, and
- σ^2, the **variance** (σ is the **standard deviation**) – how wide the bell is.

Fitting the Model

A Gaussian model for July temperatures is fully specified once we choose μ and σ.

The natural choice: set μ to the **sample mean** and σ to the **sample standard deviation** of the observed July days.

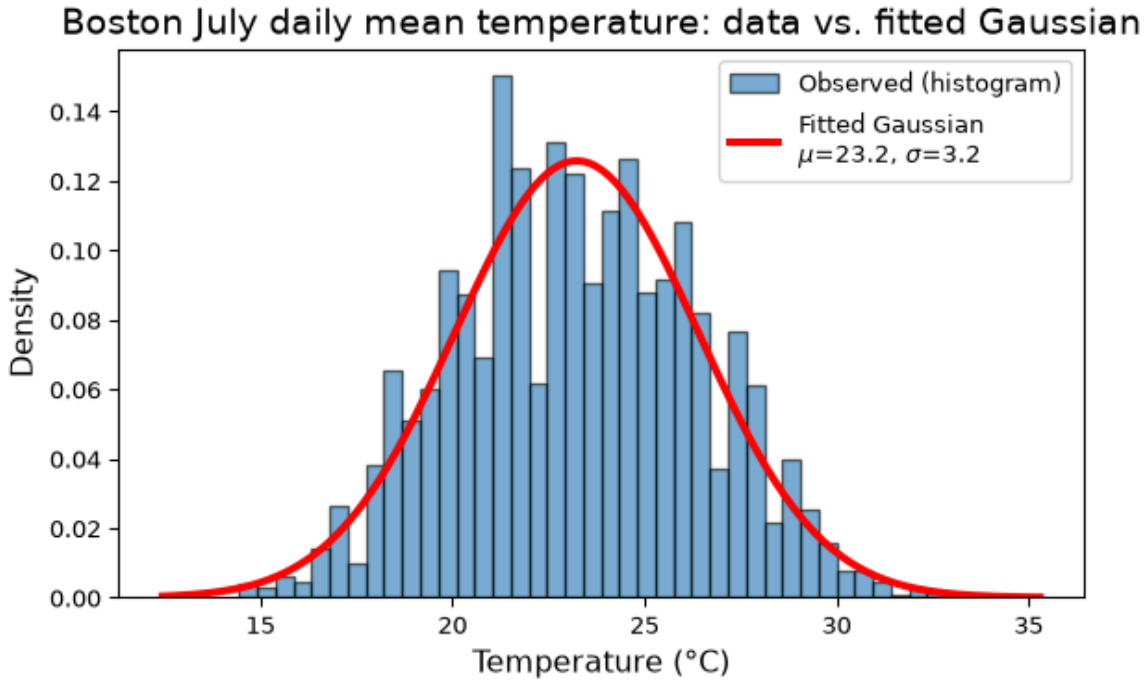
▶ Code

```
Fitted Gaussian for July:  mu = 23.23 °C,  sigma = 3.17 °C
```

Checking the Fit

A model that we don’t check is just an assumption. The simplest check is visual: overlay the fitted density on the histogram.

▶ Code



Checking the Fit Quantitatively

A Gaussian makes concrete predictions we can compare against the data: about 68% of values should fall within $\mu \pm \sigma$ and about 95% within $\mu \pm 2\sigma$.

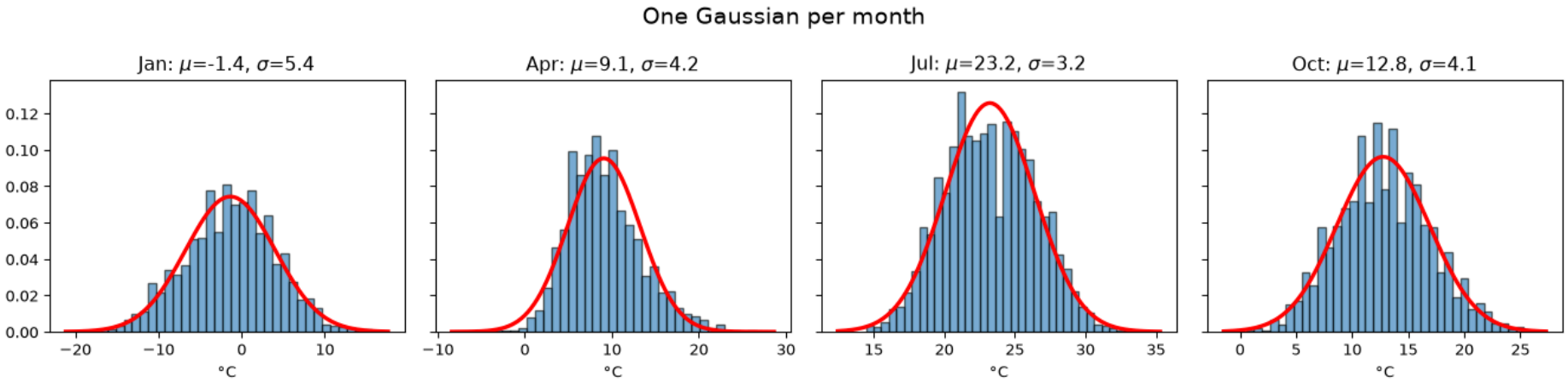
▶ Code

```
within 1 sigma:  observed 0.664   Gaussian predicts 0.683
within 2 sigma:  observed 0.966   Gaussian predicts 0.954
within 3 sigma:  observed 0.999   Gaussian predicts 0.997
```

Does the Same Model Fit Every Month?

Fitting is cheap, so let's fit a separate Gaussian to each month and look at four of them.

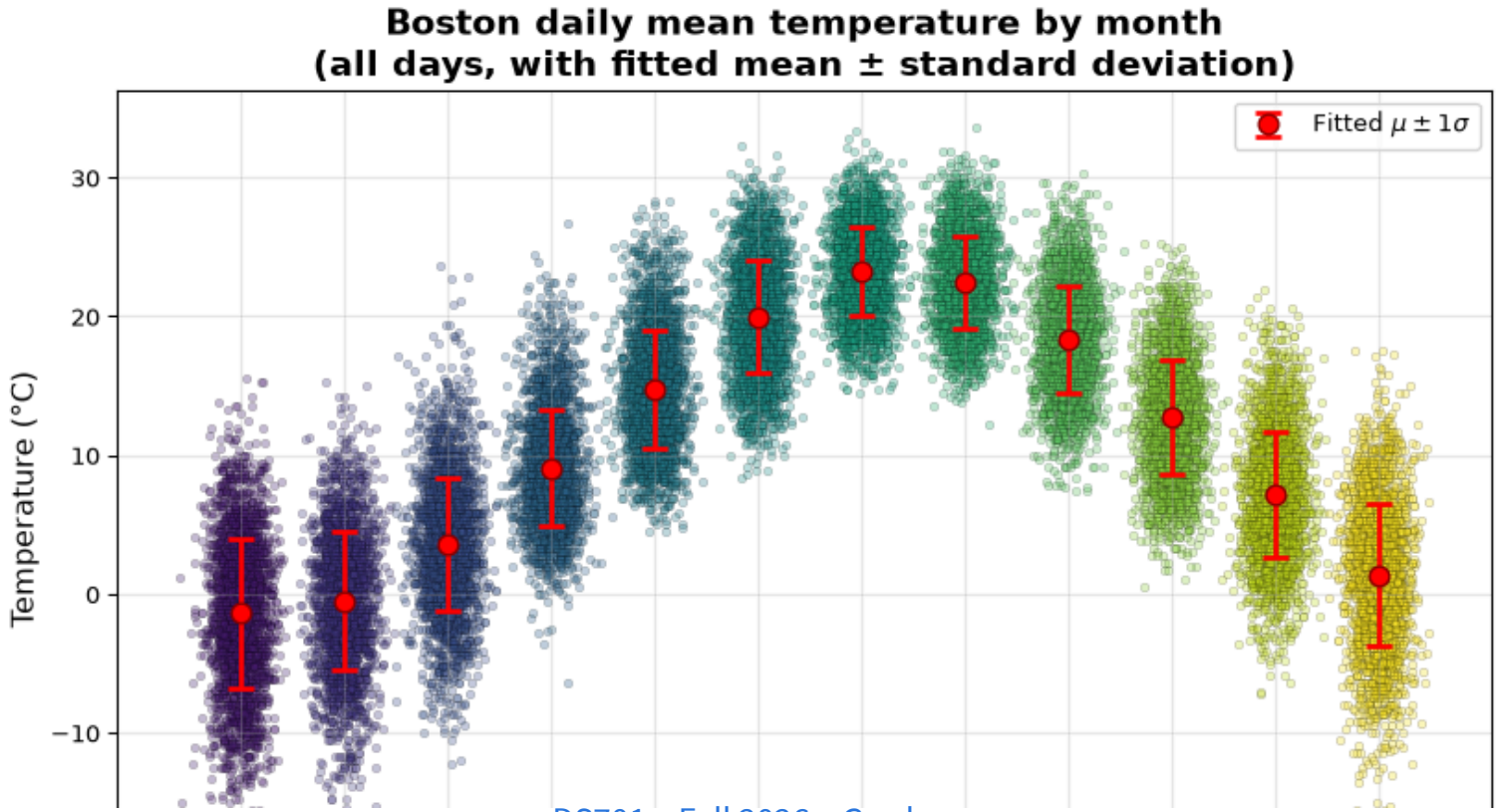
▶ Code



The Whole Year at a Glance

Plotting every day, plus each month's fitted $\mu \pm \sigma$, summarizes the twelve fitted models in one picture.

- Scatter of all days with monthly mean ± 1 std



What a Fitted Model Buys You

Once we trust the model, we can ask questions the raw data answers only clumsily.

For example: *what is the probability that a July day in Boston has a mean temperature above 30 °C?*

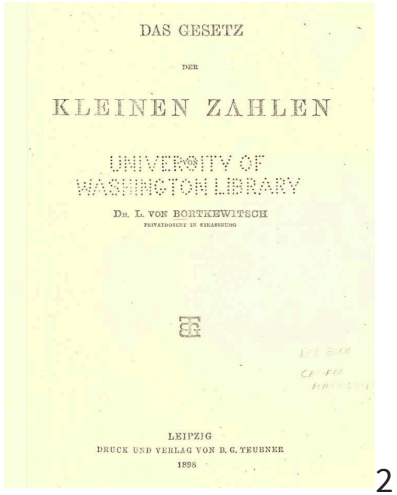
▶ Code

```
Gaussian model:  P(T > 30 °C) = 0.0165
Empirical:       fraction of July days above 30 °C = 0.0117
```

Fitting a Model to Counts

A Second Dataset: Deaths by Horse Kick

1



2

- Ladislaus Bortkiewicz (1868 – 1931)
- Wrote book "Law of Small Numbers" in 1898
- Studied the number of Prussian cavalry soldiers killed by horse kicks, per army corps per year, 1875 – 1894
- Some years a corps had 3 or 4 such deaths – was something going on, or was this what randomness looks like?

1. By no conegut - MacTutor History of Mathematics: http://www-history.mcs.st-andrews.ac.uk/PictDisplay/Bortkiewicz.html, Public Domain, https://commons.wikimedia.org/w/index.php?curid=79219622
2. Law of Small Numbers

The Horse-Kick Data

Twenty years of counts for fourteen corps (the data is in `data/HorseKicks.txt`).

▶ Code

	GC	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C14	C15
Year														
1875	0	0	0	0	0	0	0	1	1	0	0	0	1	0
1876	2	0	0	0	1	0	0	0	0	0	0	0	1	1
1877	2	0	0	0	0	0	1	1	0	0	1	0	2	0
1878	1	2	2	1	1	0	0	0	0	0	1	0	1	0
1879	0	0	0	1	1	2	2	0	1	0	0	2	1	0
1880	0	3	2	1	1	1	0	0	0	2	1	4	3	0
1881	1	0	0	2	1	0	0	1	0	1	0	0	0	0
1882	1	2	0	0	0	0	1	0	1	1	2	1	4	1

The Model: The Poisson Distribution

The Poisson distribution models "how many events occur in a fixed interval" when events happen at a constant average rate but otherwise at random.

Definition. A random variable X taking values $0, 1, 2, \ldots$ has a *Poisson distribution* with parameter $\lambda > 0$ if

$$p(k) = P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{for } k = 0, 1, 2, \ldots$$

Fitting the Poisson

With one parameter there is only one thing to estimate: the rate λ. The natural estimate is the **sample mean** count.

▶ Code

```
200 corps-years,  122 deaths,  lambda_hat = 0.610 deaths per corps per year
```

Checking the Fit: Observed vs. Predicted Counts

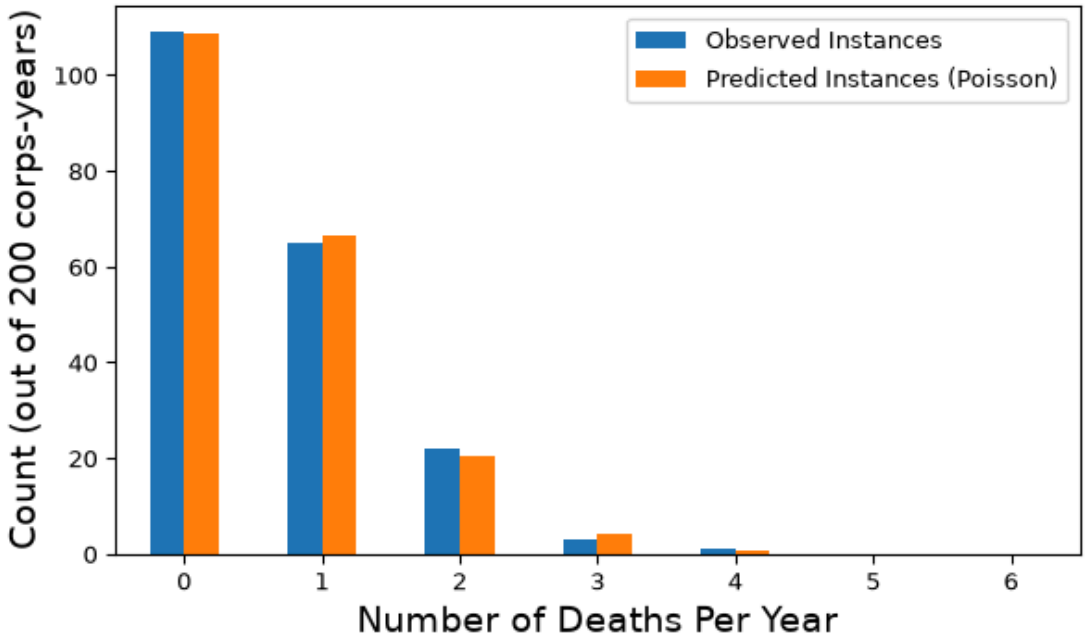
If the Poisson model is right, then out of 200 corps-years we expect $200 \cdot p(k)$ of them to have exactly k deaths.

▶ Code

Number of Deaths Per Year	Observed Instances	Predicted Instances (Poisson)
0	109	108.67
1	65	66.29
2	22	20.22
3	3	4.11
4	1	0.63
5	0	0.08

Checking the Fit: Observed vs. Predicted Counts

▶ Code



Checking the Fit: A Property Check

The Poisson has a built-in consistency check: its mean and variance are both λ. Does the data agree?

▶ Code

```
sample mean     = 0.610
sample variance = 0.608
```

The Pattern So Far

Both examples followed the same four steps:

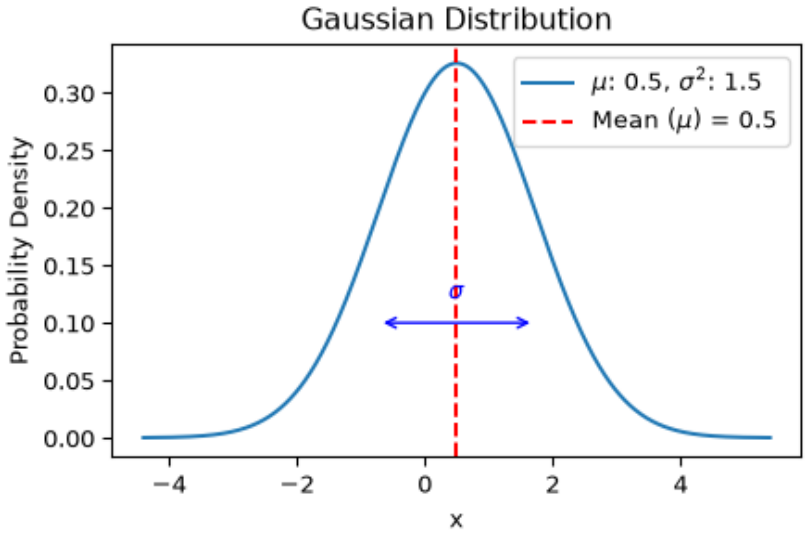
Maximum Likelihood Estimation (MLE)

Motivation

Probability distributions are specified by their parameters.

The Gaussian distribution is determined by the parameters μ and σ^2, i.e.,

$$f(x|\mu,\sigma^2) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$



Maximum likelihood estimation is a method to estimate the parameters of a probability distribution given a sample of observed data that *best fits the data*.

Likelihood Function

The likelihood function $L(\boldsymbol{\theta}, x)$ represents the probability of observing the given data x as a function of the parameters $\boldsymbol{\theta}$ of the distribution.

The primary purpose of the likelihood function is to estimate the parameters that make the observed data x most probable.

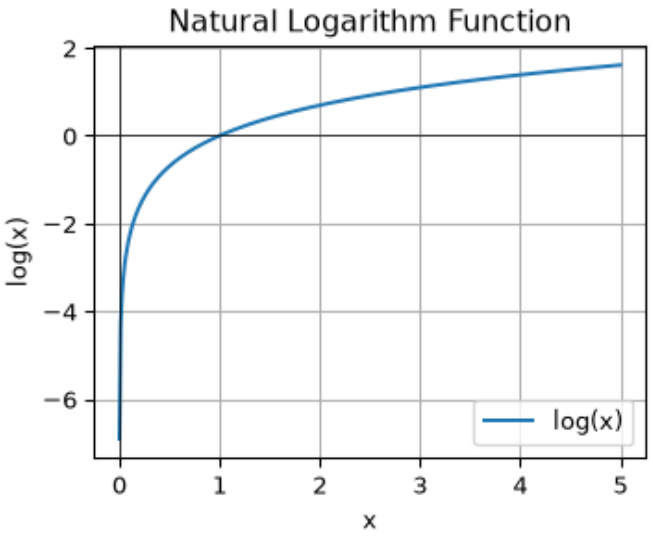
Maximizing the Likelihood

For a particular set of parameters μ, σ^2

- **large** values of $L(\mu, \sigma^2, x_1, \dots, x_n)$ indicate the observed data is very probable (high likelihood) and thus *well modeled by the parameters*
- **small** values of $L(\mu, \sigma^2, x_1, \dots, x_n)$ indicate the observed data is very improbable (low likelihood) and thus *poorly modeled by the parameters*

Log-likelihood

A common manipulation to obtain a more useful form of the likelihood function is to take its natural logarithm.



Advantages of the log-likelihood:

- The log function is monotonically increasing, so the MLE is the same as the log-likelihood estimate
- The product of probabilities becomes a sum of logarithms, which is more numerically stable
- The log-likelihood is easier to work with

Applying the Log

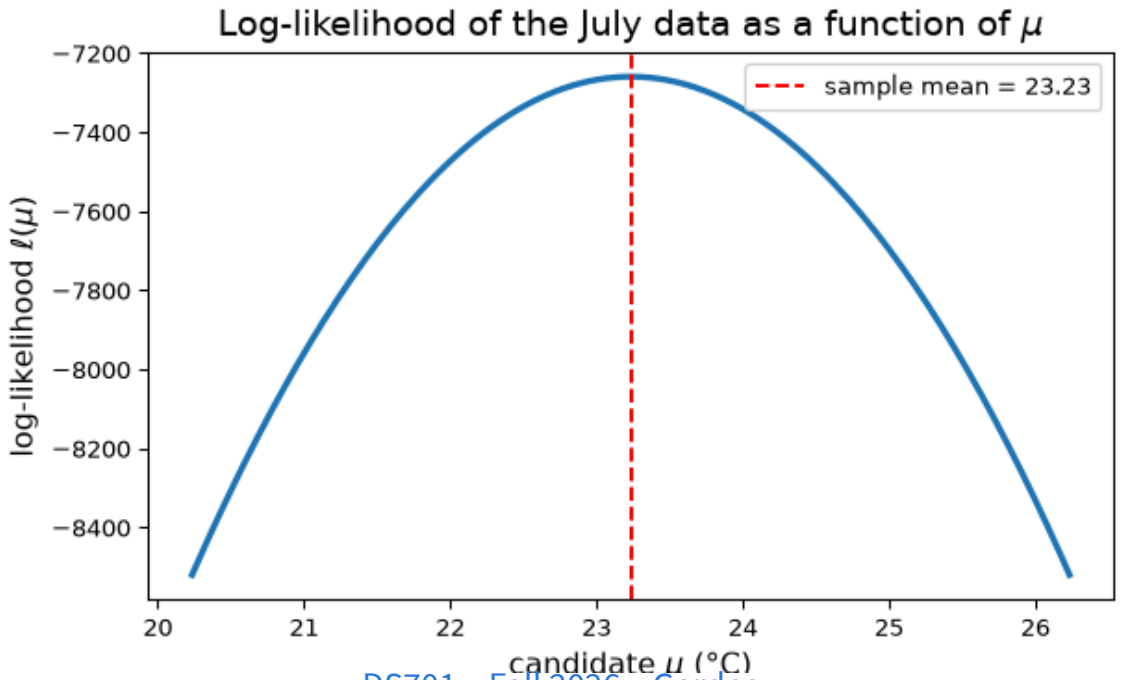
$$
\begin{aligned}
\ell(\mu, \sigma^2, x_1, \dots, x_n) &= \log\left(L(\mu, \sigma^2, x_1, \dots, x_n)\right) \\
&= \log\left(\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}}\right) \\
&= \sum_{n=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}}\right) \\
&= \sum_{n=1}^{N} \left(\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \log\left(e^{-\frac{(x_n-\mu)^2}{2\sigma^2}}\right)\right) \\
&= -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2.
\end{aligned}
$$

Using the log-likelihood we will be able to derive formulas for the maximum likelihood estimates.

Seeing the Log-likelihood on Real Data

Before deriving anything, let's just *compute* the Gaussian log-likelihood of the July temperatures for a range of candidate means μ (holding σ at the sample value).

▶ Code



Maximizing the log-likelihood

Gaussian MLEs

The maximum log-likelihood estimates for a Gaussian distribution are given by

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n,$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N} (x_n - \hat{\mu})^2.$$

The full derivation of these results is provided here.

> Tip: Try deriving these results yourself!

The Same Recipe for the Poisson

Nothing about the recipe was specific to the Gaussian. For counts $k_1, \dots, k_N$ modeled as i.i.d. Poisson(λ):

$$\ell(\lambda) = \sum_{n=1}^{N} \log \frac{\lambda^{k_n} e^{-\lambda}}{k_n!} = \log \lambda \sum_{n=1}^{N} k_n \; - \; N\lambda \; - \; \sum_{n=1}^{N} \log k_n!$$

Summary of MLE

The recipe: write down the probability of the observed data as a function of the parameters (the likelihood), take the log, and find the parameters that maximize it.

For samples $x_1, \dots, x_n$ from a Gaussian, this means maximizing

$$\ell(\mu, \sigma^2, x_1, \dots, x_n) = -\frac{N}{2}\log 2\pi - N\log\sigma - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2,$$

giving

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n, \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu})^2.$$

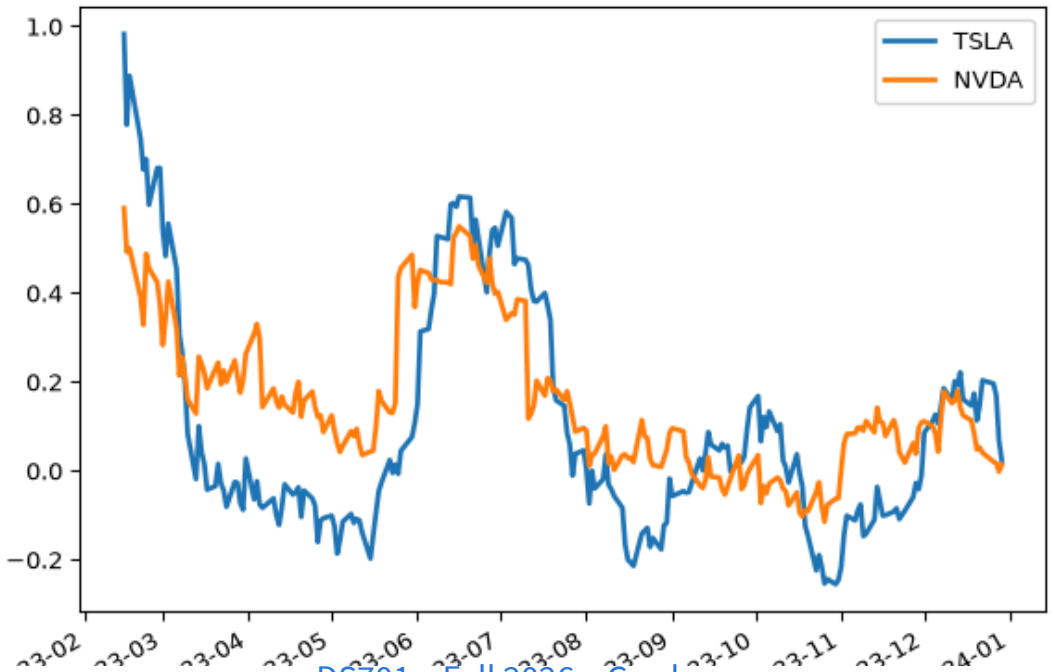
Two Variables at Once

Stocks as Random Variables

So far each model described a **single** quantity. Very often we care about how two quantities vary **together**.

Let's take the daily closing prices of Tesla and NVIDIA in 2023, and look at their 30-day returns:

▶ Code



Covariance

This is captured by the concept of **covariance.**

Definition. For two random variables X and Y, their *covariance* is defined as:

$$\mathrm{Cov}(X, Y) = E\left[(X - \mu_X)(Y - \mu_Y)\right].$$

If covariance is positive, this tells us that X and Y tend to both be above their means together and both below their means together.

We will often denote $\mathrm{Cov}(X, Y)$ as σ_{XY}.

Correlation

If we are interested in asking "how similar" are two random variables, we want to normalize covariance by the amount of variance shown by the random variables.

The tool for this purpose is **correlation**, i.e., normalized covariance:

$$\rho(X,Y) = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y}.$$

$\rho(X,Y)$ takes on values between -1 and 1.

If $\rho(X,Y) = 0$ then X and Y are **uncorrelated**.

> **Note**
>
> Note that this is not the same thing as being independent! It just means there is no linear relationship between the two.
>
> But independence implies uncorrelated. Plug in equations and check.

ρ is sometimes called "Pearson r" after Karl Pearson who popularized it.

Stock Covariance

Let's estimate the covariance of the two closing prices from the data. Just as the sample mean estimates μ, the **sample covariance** estimates $\mathrm{Cov}(X, Y)$:

▶ Code

	TSLA	NVDA
TSLA	1757.018115	387.819338
NVDA	387.819338	115.323299

In the case of Tesla (X) and NVIDIA (Y) above, we find that

$$\mathrm{Cov}(X, Y) \approx 388.$$

The diagonal entries are the variances of each stock's price; the off-diagonal entry is the covariance.

Stock Correlation

How similar are these random variables? Let's compute $\rho(X, Y)$.

▶ Code

	TSLA	NVDA
TSLA	1.000000	0.861555
NVDA	0.861555	1.000000

We observe that

$$\rho(X, Y) \approx 0.86.$$

There appears to be some similarity between the two stocks.

The Multivariate Gaussian

The most common multivariate distribution we will work with is the multivariate Gaussian – it is what a Gaussian looks like in two or more dimensions.

The multivariate normal distribution of a random vector $\mathbf{X} = (X_1, \dots, X_k)^T$ is denoted

$$\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$$

where $\mu = E[\mathbf{X}] = (E[X_1], \dots, E[X_k])^T$

and Σ is the $k \times k$ **covariance matrix** where $\Sigma_{i,j} = \mathrm{Cov}(X_i, X_j)$.

Multivariate Gaussian: Uncorrelated Components

We'll consider two-component random vectors:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}.$$

And our first example will be a simple one

$$\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

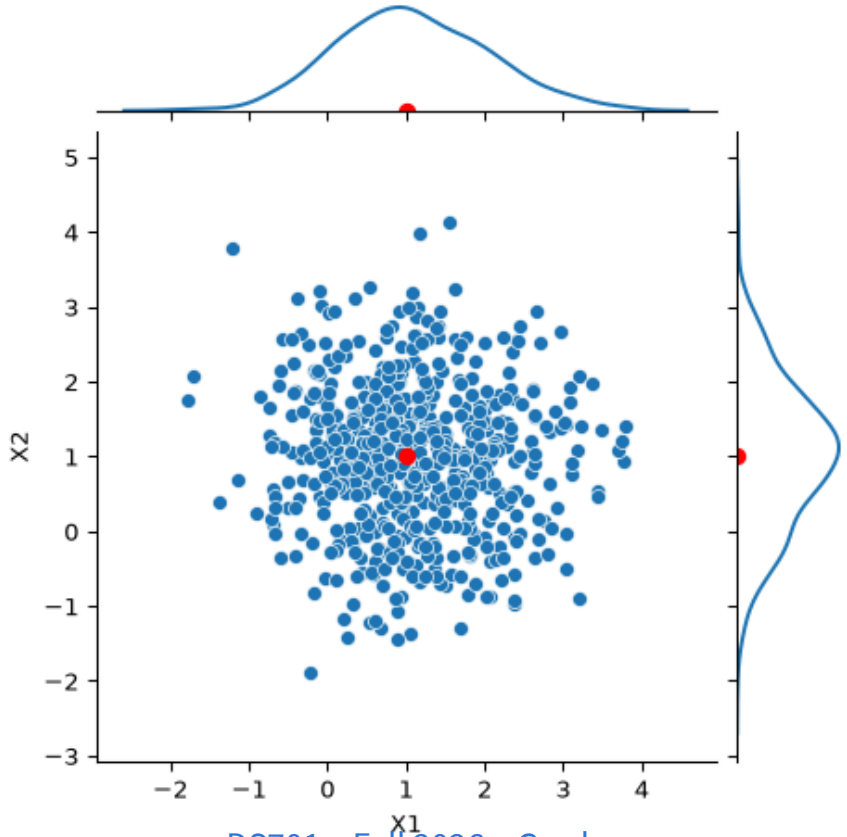
We see that the variance (and standard deviation) of each component is 1.

However the covariances are zero – the components are uncorrelated.

We will take 600 samples from this distribution.

Multivariate Gaussian: Uncorrelated Components

▶ Code



Multivariate Gaussian: Positively Correlated Components

Next, we look at the case

$$\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

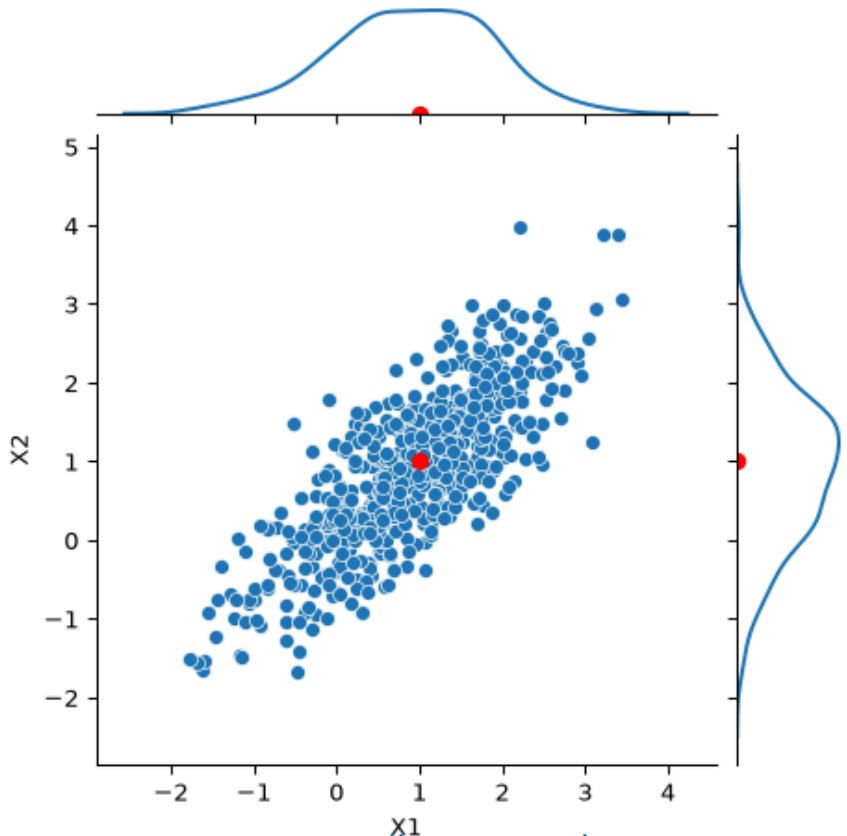
Notice that $\mathrm{Cov}(X_1, X_2) = 0.8$.

We say that the components are **positively correlated.**

Nonetheless, **the marginals are still Gaussian.**

Multivariate Gaussian: Positively Correlated Components

▶ Code



Multivariate Gaussian: Negatively Correlated Components

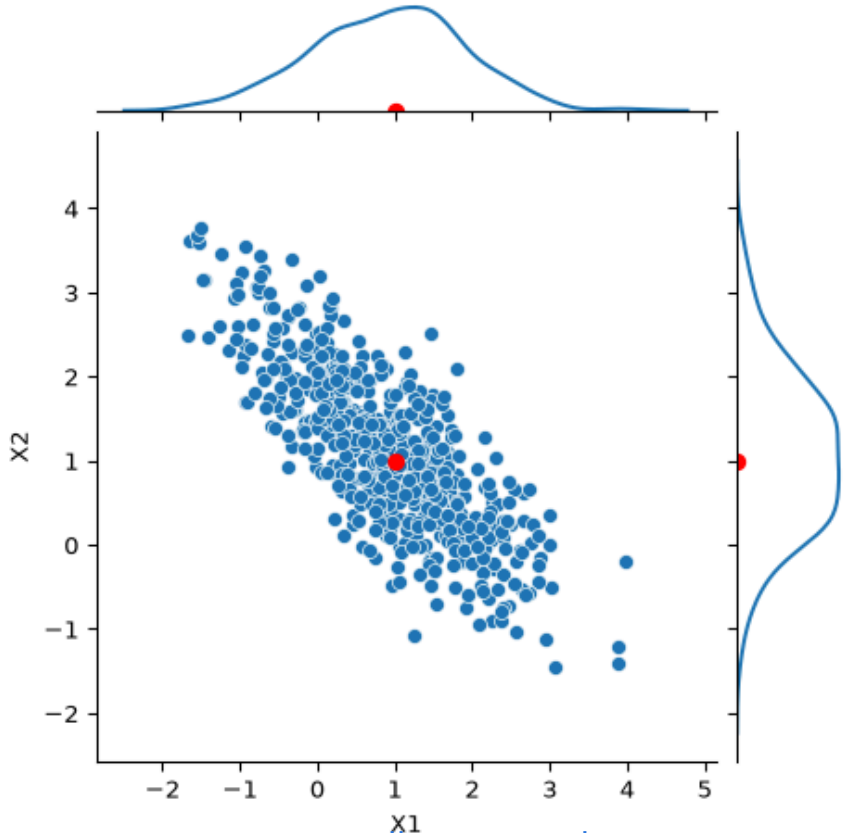
Next, we look at the case

$$\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}.$$

Notice that $\mathrm{Cov}(X_1, X_2) = -0.8$. We say that the components are **negatively correlated** or **anticorrelated.**

Multivariate Gaussian: Negatively Correlated Components

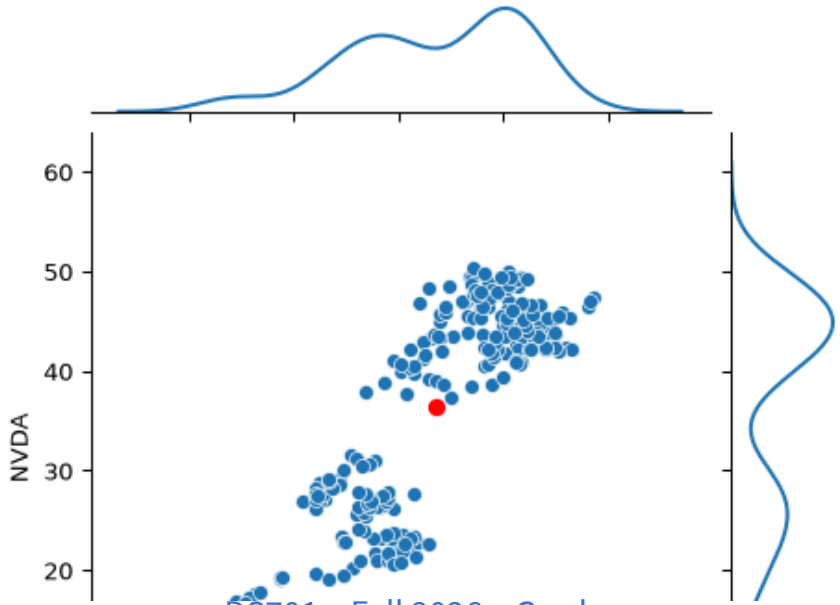
▶ Code



Fitting a Multivariate Gaussian to the Stock Data

The MLE story carries over: the maximum likelihood estimates of μ and Σ are the **sample mean vector** and the **sample covariance matrix** – exactly the `df.mean()` and `df.cov()` we already computed.

▶ Code



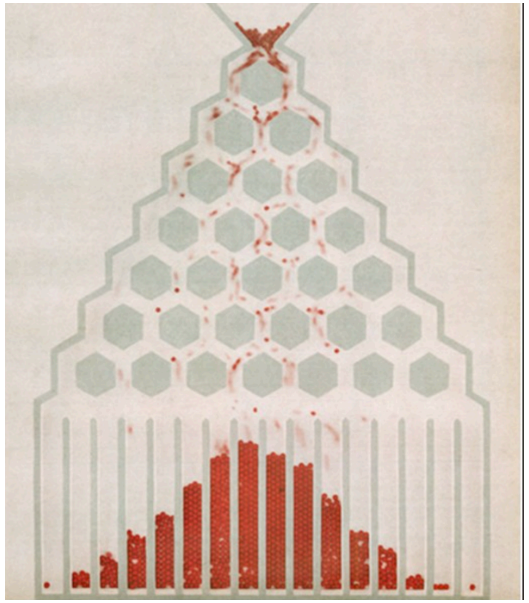
Uncertainty: How Good Is a Sample Mean?

The Central Limit Theorem

We have been estimating means from data all lecture. How much should we trust such an estimate?

The key tool is the celebrated **Central Limit Theorem**. Informally,

> *The sum (or average) of a large number of independent observations from any distribution with finite variance tends to have a Gaussian distribution.*



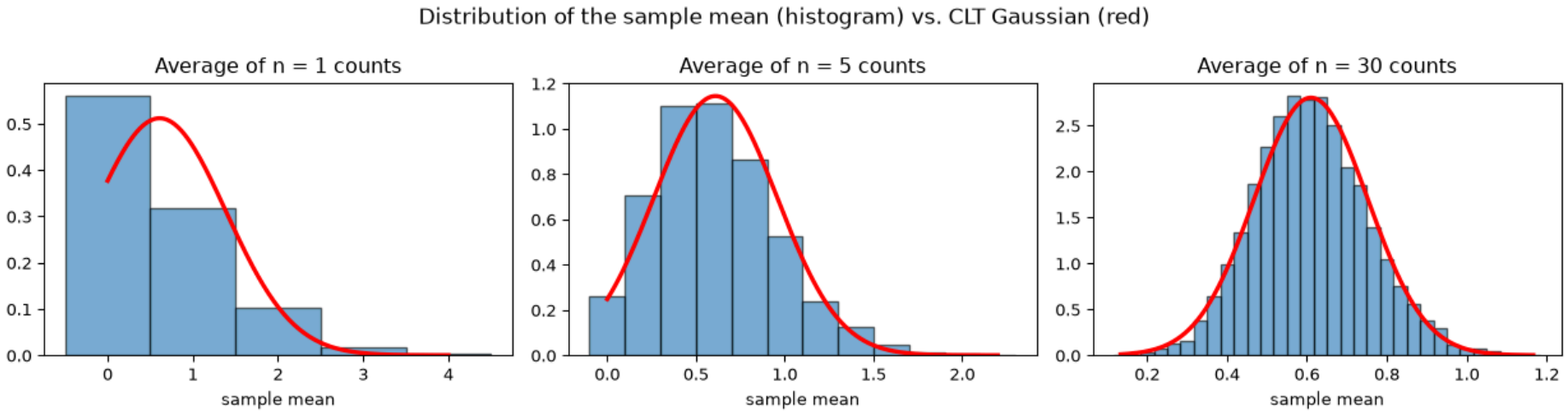
This is also why the Gaussian is such a good default model: measurement errors, daily temperatures, and many other quantities are themselves the accumulation of many small independent effects.

Note the “any distribution” – the individual observations do **not** need to be Gaussian.

The CLT in Action

Let's check it on the horse-kick counts, which are certainly not Gaussian (they are 0, 1, 2, …). We repeatedly draw n corps-years at random and record the average count.

▶ Code



Confidence Intervals

Say you are concerned with some data that we take as coming from a random process.

You want to characterize it as accurately as possible. You measure it, yielding a single value.

How much does that value tell you? Can you rely on it as a description of the random process?

Let’s say you have a dataset and you compute its average value.

How certain are you that the average would be the same if you took another dataset from the same source (i.e., the same random process)?

Confidence Intervals

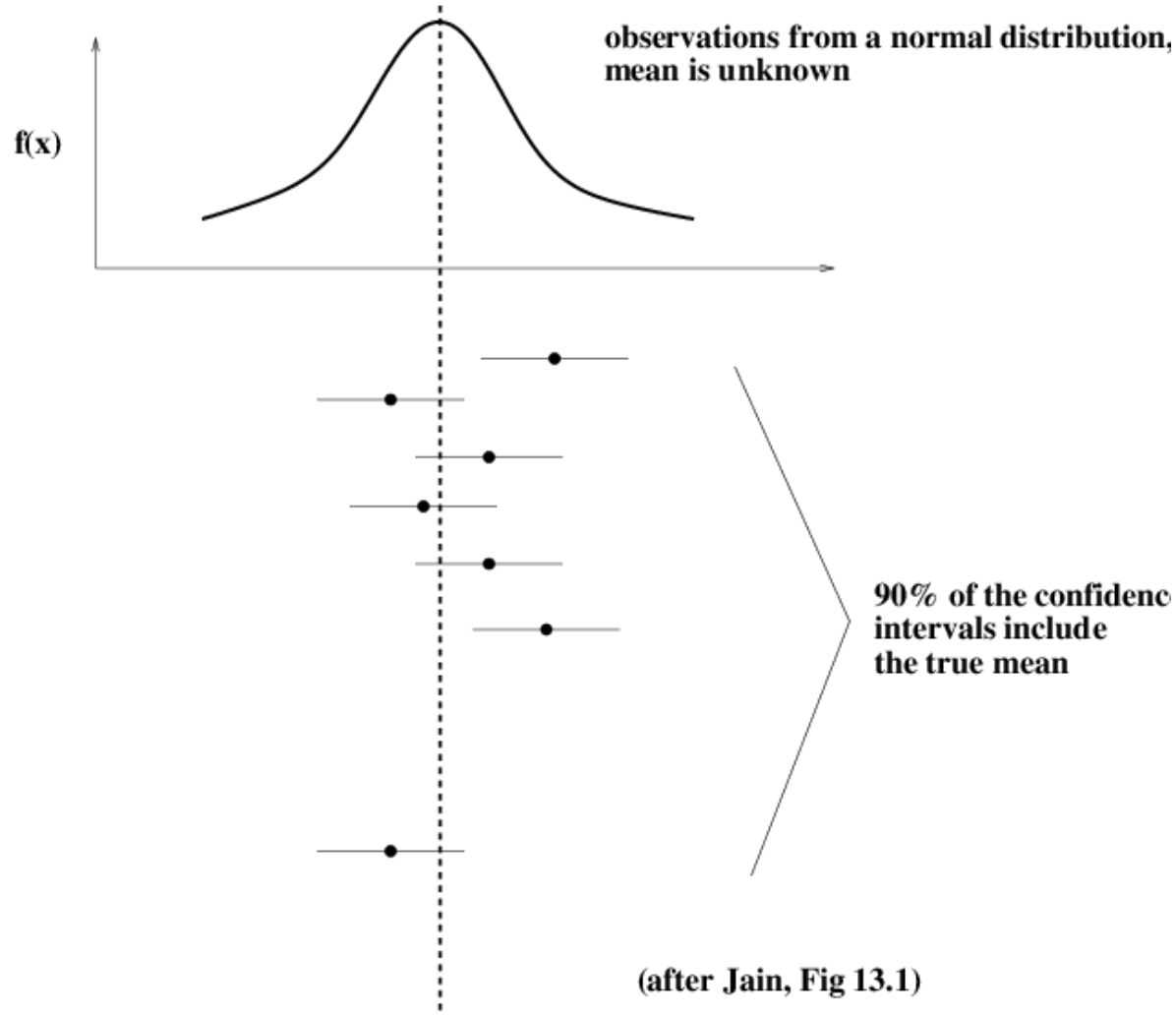
We think of the hypothetical data source as a random variable with a true mean μ.

We would like to find a range within which we are 90% sure that the true mean μ lies.

In other words, we want the probability that the true mean lies in the interval to be 0.9.

This interval is then called the 90% confidence interval.

Confidence Intervals



Confidence Intervals for the Mean

Imagine we have a set of n samples of a random variable, $x_1, x_2, \ldots, x_n$ Let's assume that the random variable has mean μ and variance σ^2.

An estimate of μ is the empirical average of the samples, $\bar{x}$.

Now, the Central Limit Theorem tells us that the sum of a large number n of random variables, each with mean μ and variance σ^2, yields a Gaussian random variable with mean $n\mu$ and variance $n\sigma^2$.

So the distribution of the average of n samples is normal with mean μ and variance σ^2/n. That is,

$$\bar{x} \sim \mathcal{N}(\mu, \sigma^2/n).$$

We usually assume that the number of samples should be 30 or more for the CLT to hold.

While the specific value 30 is a bit arbitrary, we will usually be using very large samples (datasets) in this course for which this assumption is valid.

The Standard Error

The standard deviation of the sample mean, $\sigma/\sqrt{n}$, is called the **standard error**.

Notice that the standard error decreases as we increase the sample size, according to $1/\sqrt{n}$.

So it will turn out that using $\bar{x}$, we can get increasingly "tight" estimates of μ as we increase the number of samples n – exactly what we saw in the CLT simulation.

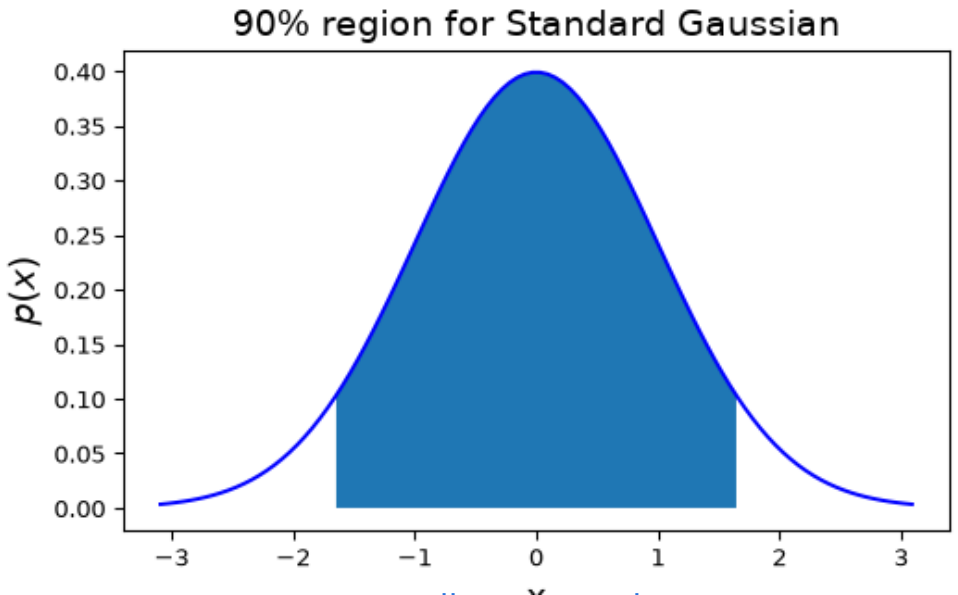
Choosing k

We write $z_{1-\alpha/2}$ to be the $1-\alpha/2$ quantile of the unit normal. That is,

$$P(-z_{1-\alpha/2} < S < z_{1-\alpha/2}) = 1-\alpha.$$

So to form a 90% probability interval for S (centered on zero) we choose $k = z_{0.95}$.

▶ Code



From a Probability Interval to a Confidence Interval

Turning back to $\bar{x}$, the 90% probability interval on $\bar{x}$ would be:

$$\mu - z_{0.95}\sigma/\sqrt{n} < \bar{x} < \mu + z_{0.95}\sigma/\sqrt{n}.$$

The last step: by a simple argument, we can show that the sample mean is in some fixed-size interval centered on the true mean, if and only if the true mean is also in a fixed-size interval (of the same size) centered on the sample mean.

This means that:

$$\begin{aligned} 1 - \alpha &= P(\mu - z_{1-\alpha/2}\sigma/\sqrt{n} < \bar{x} < \mu + z_{1-\alpha/2}\sigma/\sqrt{n}) \\ &= P(\bar{x} - z_{1-\alpha/2}\sigma/\sqrt{n} < \mu < \bar{x} + z_{1-\alpha/2}\sigma/\sqrt{n}). \end{aligned}$$

This latter expression defines the $1 - \alpha$ **confidence interval for the mean.**

The Confidence Interval Formula

We are done, except for estimating σ. We do this directly from the data: $\hat{\sigma} = s$, where s is the sample standard deviation, that is,

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

To summarize: by the argument presented here, a 100(1-α)% confidence interval for the population mean is given by

$$\bar{x} \pm z_{1-\alpha/2} \frac{s}{\sqrt{n}}.$$

A Confidence Interval on Real Data

What is the mean daily temperature in Boston in July? Let's report it the way you should report a mean in a project: **with a confidence interval**.

▶ Code

```
n = 2821,  sample mean = 23.23 °C,  sample std = 3.17 °C
standard error = 0.060 °C
95% CI for the mean July temperature: [23.12, 23.35] °C
```

Reading a Confidence Interval Correctly

Summary

Summary

Probability as a **modeling tool** – four things we did:

Next: Gaussian Mixture Models