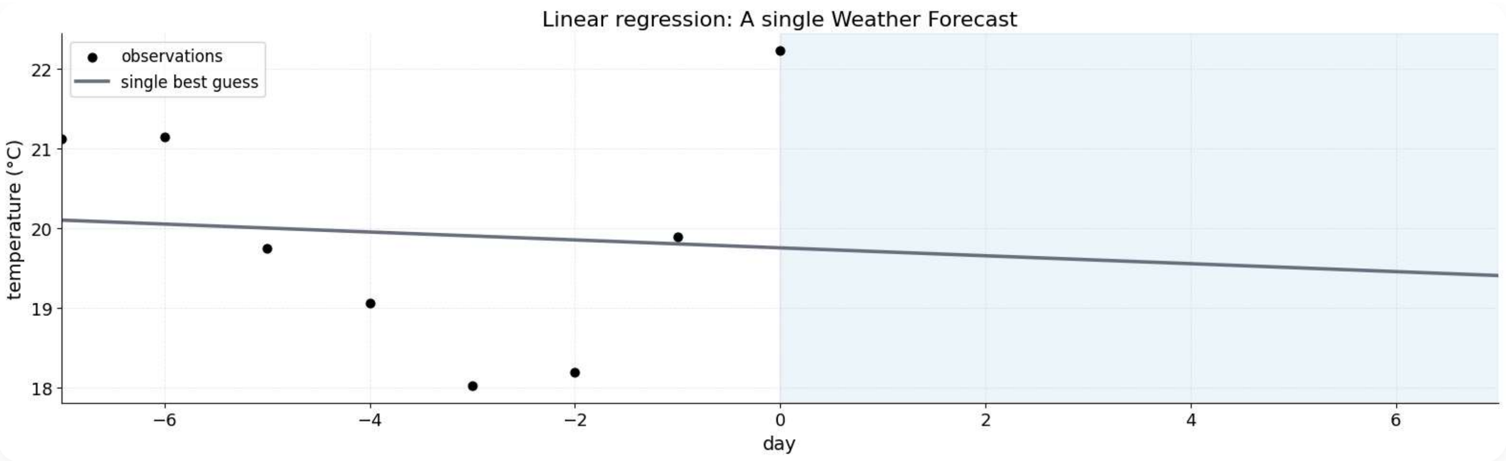
Exploring Uncertainty with Gaussian Processes: An Interactive Tutorial

1. Introduction to Gaussian processes and why they are useful

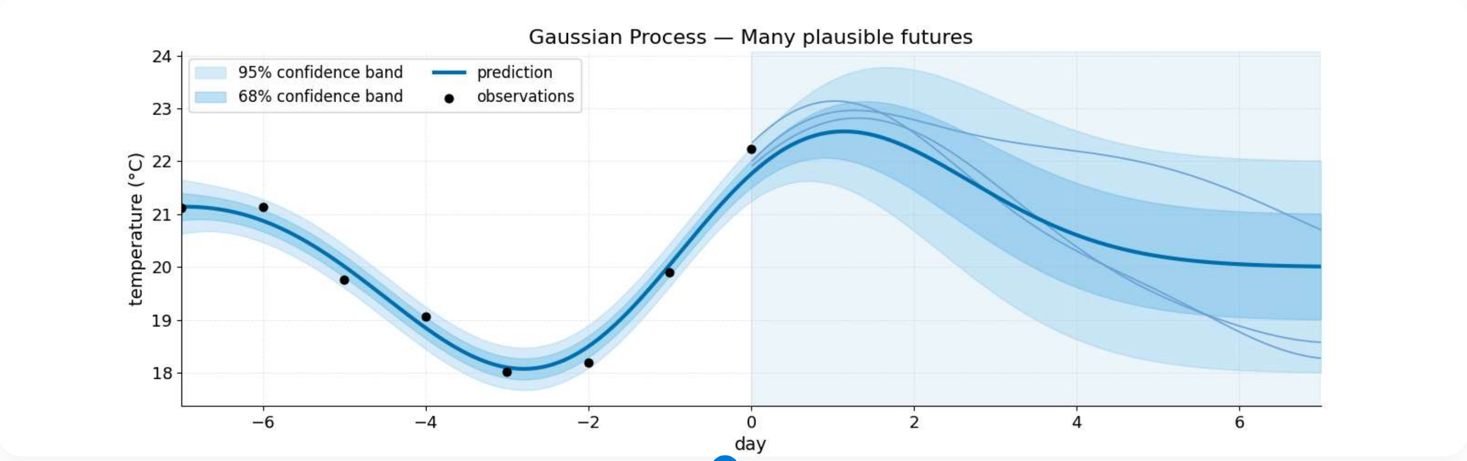
Every day we make predictions:

- *Will it rain tomorrow?*
- *What will the temperature be next week?*
- *How much traffic will I face on the way to work?*

Most models give us a **single best prediction** based on seen data. Linear regression for example fits a single line.



But real life is uncertain: even if a model predicts the temperature tomorrow to be 20 degrees - it could easily be 18°C or 23°C. Thus, we need not just a single prediction, but also a sense of how confident the model is. This is where **Gaussian Processes (GPs)** are useful. Instead of committing to one best function, a GP represents an infinite number of functions that could explain the data. Some functions are likelier than others, but all are possible.



○ Once ● Loop ○ Reflect

In the plot, we visualise the uncertainty around each prediction with darker shaded regions for more confident predictions.

Gaussian Processes (GPs) are a natural extension of well-known regression models and are defined by just two ingredients:

- a **mean function**: the average trend (often set to zero to “let the data speak”),
- a **covariance function** (kernel): defining how input points (x-values) move together, i.e. how smooth, noisy the world is.

Why Gaussian Processes are useful

- They provide not only predictions, but also a measure of uncertainty.
- Confidence grows where we’ve seen lots of data, and fades where we haven’t.
- This makes them powerful for guiding decisions, such as where to collect more data or when to be cautious about a forecast.

In this tutorial, we’ll build an intuition for GPs: how to express our beliefs with mean and covariance functions, how GPs make predictions at new points, and how those predictions adapt as new data comes in.

2. From Gaussian Distributions to Gaussian Processes

Whenever we want to model uncertainty, we use probability distributions. The building block of Gaussian Processes is the **Gaussian Distribution**.

Univariate Gaussian: one prediction with uncertainty

For a single prediction — say *today's temperature* — we can write: $x \sim \mathcal{N}(\mu, \sigma^2)$

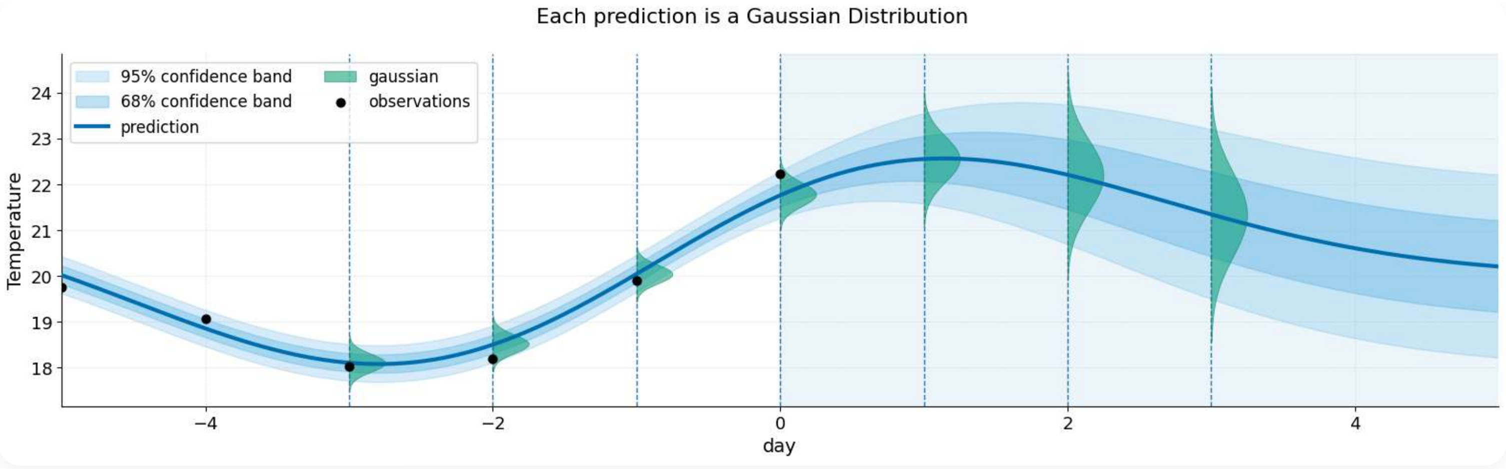
It is fully described by two parameters:

- Mean **(μ)** — the expected value
- Variance **(σ^2)** — how uncertain or spread out the prediction is

Instead of a single number, the Gaussian gives us a **bell curve** around the prediction.

▶ Show full Gaussian formula

The next figure illustrates this: each day has its own Gaussian curve, centered at the mean predicted temperature, with the width showing how uncertain the model is about that prediction.



Multivariate Gaussian: linking variables

Real data often involves multiple variables that are not independent.

For example:

- x_1: yesterday's temperature
- x_2: today's temperature

If yesterday was hot, today is also likely to be hot. This dependency is captured by the **multivariate** Gaussian: $x \sim \mathcal{N}(\mu, \Sigma)$

with

$$\mu = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & c \\ c & \sigma_2^2 \end{bmatrix}$$

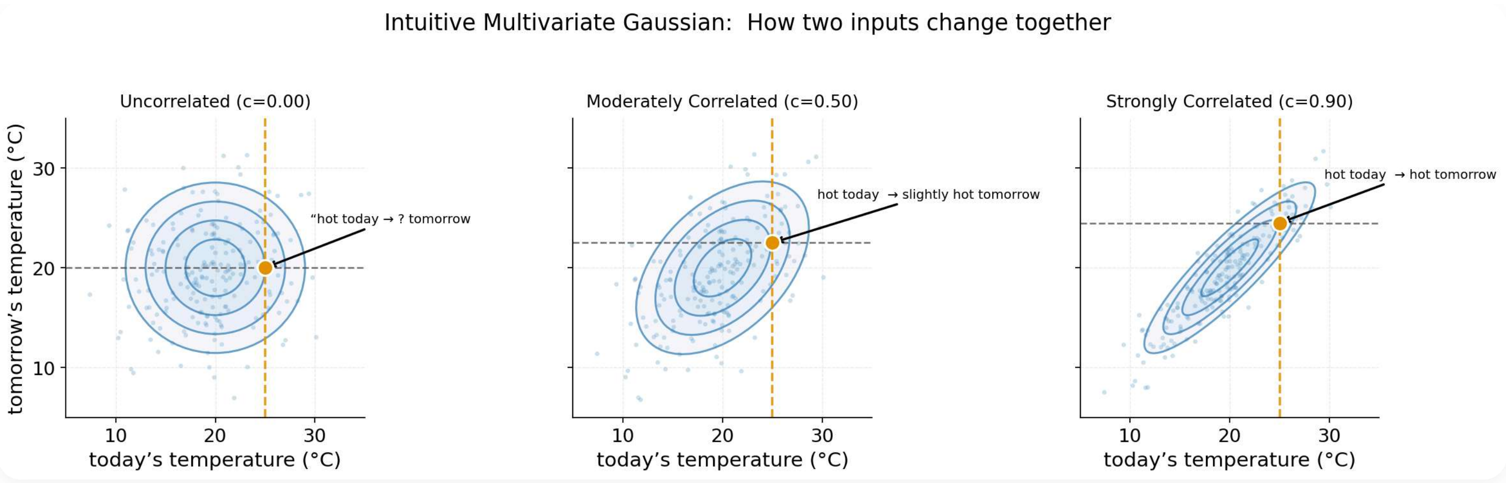
- **Mean vector** μ — expected values for each variable
- **Covariance matrix** Σ where
  - variances σ_1^2, σ_2^2 describe the spread of each variable
  - correlations c showing how strongly they move together

▶ Show full multivariate Gaussian formula

Intuition:

- **High correlation** (large c): if yesterday was hot, today is likely hot as well
- **Low correlation** (small c): yesterday's weather tells us little about today

The contour plot illustrates this visually: circular contours show weak correlations whereas tilted ellipses show strong correlations.



Intuitive Multivariate Gaussian: How two inputs change together

Uncorrelated (c=0.00)

Moderately Correlated (c=0.50)

Strongly Correlated (c=0.90)

“hot today → ? tomorrow

hot today → slightly hot tomorrow

hot today → hot tomorrow

Once we understand Gaussians, the step to **Gaussian Processes (GPs)** is natural. A GP is simply the idea of extending Gaussians to functions over many inputs. In the next section, we will see how this works by defining priors through mean and covariance functions.

3. The Ingredients of a GP: Mean and Covariance Function

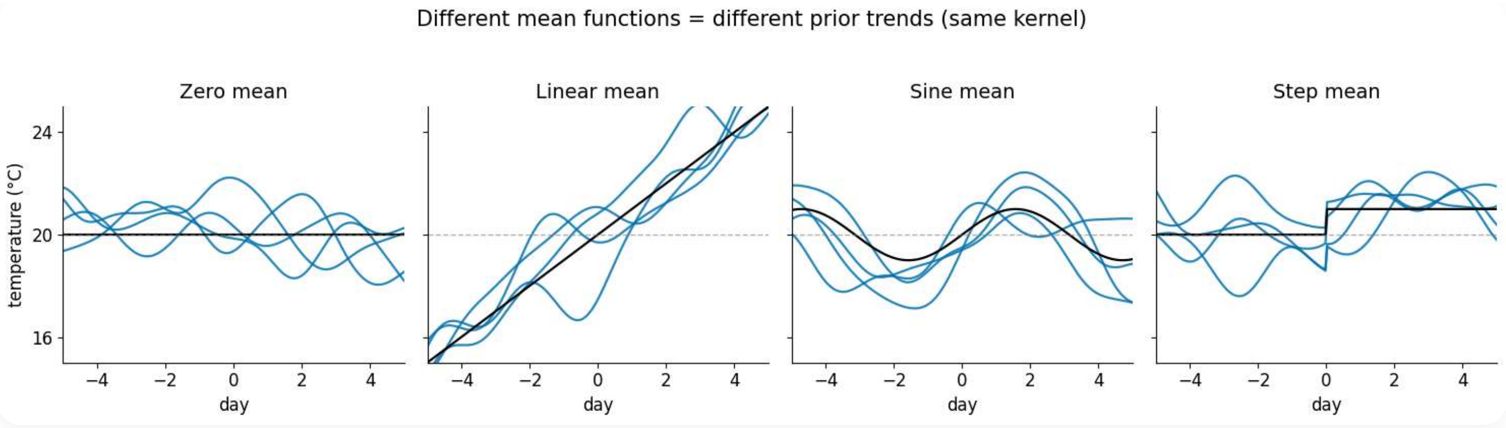
The Mean Function: expected trend

We use the mean function $m(x)$ to describe the average shape of functions we expect *before* seeing any data:

$$m(x) = \mathbb{E}[f(x)]$$

For example, we might assume no overall trend (a **zero mean**), a straight-line trend (a **linear mean**), or even a repeating pattern.

Below, we show how different mean functions change the prior trend (black line), while sample functions (blue) vary around it.



Different mean functions = different prior trends (same kernel)

Zero mean

Linear mean

Sine mean

Step mean

The Covariance Function (Kernel): wiggliness of functions

The second ingredient is the **covariance function**, or **kernel**. It measures how similar two inputs x and x' are, and therefore how much their function values should “move together.”

Formally:

$$k(x,x') = \mathbb{E}\Big[(f(x) - m(x))(f(x') - m(x'))\Big]$$

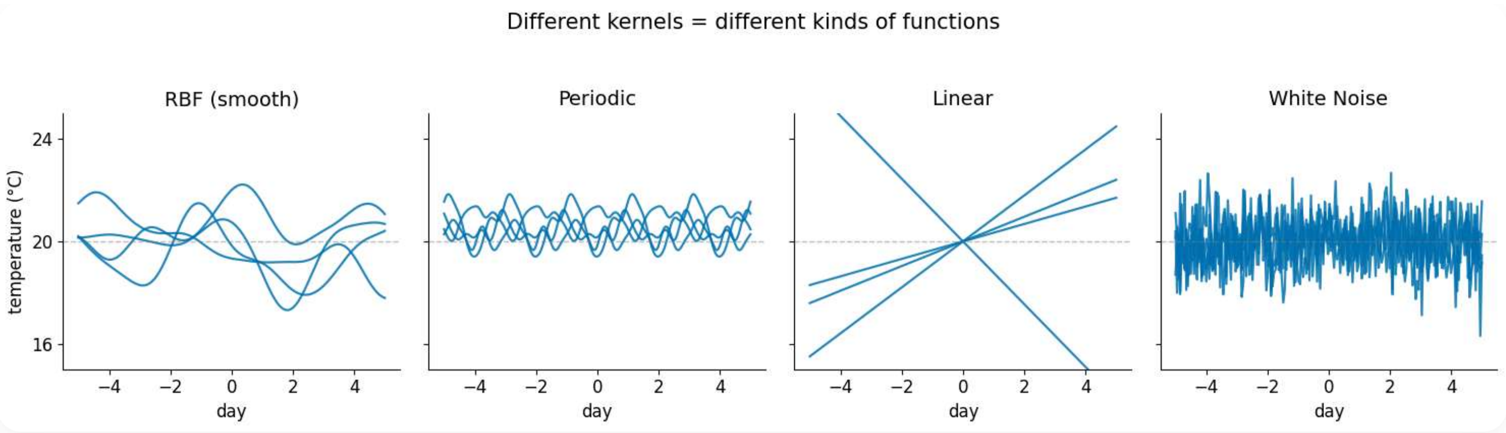
Intuitively, the kernel controls the *shape* of functions:

- how smooth or wiggly they are,
- whether they repeat (periodicity),
- and how far correlations extend.

A very popular choice is the **Radial-Basis-Function (RBF)** kernel which produces smooth functions: points that are close together are highly correlated, while far-apart points are only weakly correlated.

▶ Show the RBF formula

You can see in the next figure how different kernels (RBF, periodic, linear, white noise) lead to very different styles of functions.



4. Updating our model with data and getting predictions

So far, we’ve seen how a Gaussian Process is defined by a mean and a covariance function, which together describe our prior belief about what functions are likely. Once we observe data, we can update this belief and get predictions for new data points.

Conditioning as Bayes’ Rule

At its core, we use conditioning which relies on **Bayes’ rule**: We take what we believed before (the prior), combine it with what the data tells us (the likelihood), and get an updated belief (the posterior) which better reflects the reality.

$$\text{Posterior} \quad \propto \quad \text{Likelihood} \times \text{Prior}$$

(“$\propto$” means proportional to; the full version just includes a normalizing constant.)

- **Prior**: what we believe about functions before seeing data
- **Likelihood**: how likely the observed data is under a given function
- **Posterior**: our updated belief, combining both

In plain terms: **conditioning = learning from data.**

▶ Show full Bayes’ rule

▶ Show full conditioning formula

Getting predictions using Marginalisation

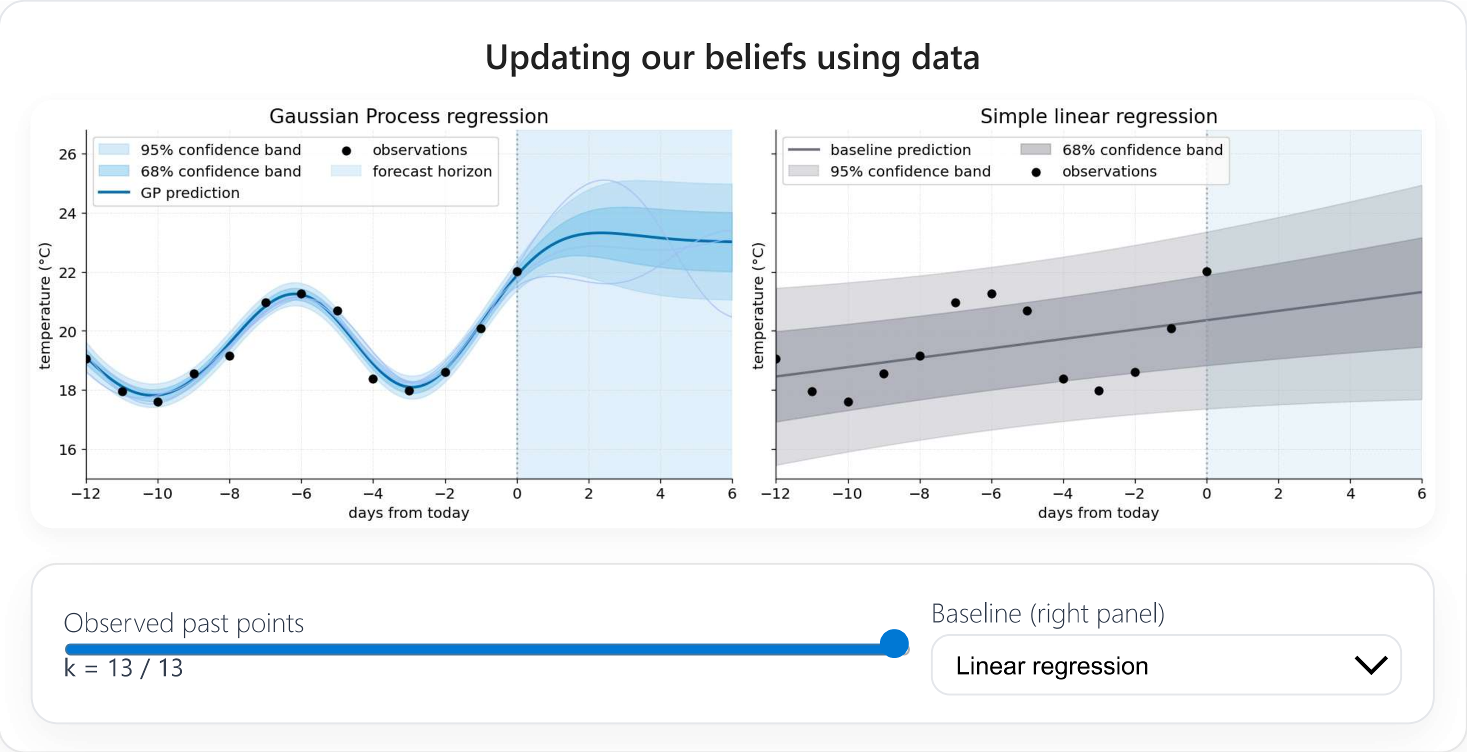
Once we updated the model the GP gives us predictions for new data points $x*$ through a process called marginalisation.

👉 In-depth: Marginalisation: Getting predictions at Specific Data Points

Example

Suppose we believe May in New York is usually 23 °C. Once we observe lower actual temperature values, the GP model gets updated and adapts its prediction downward. In the animation below we can see how our confidence changes after observing data:

- *Where we observed a lot of data, uncertainty shrinks, the predictions follow the data, and we are more confident about the forecast*
- *For far-away days with no data, we mainly rely on our prior beliefs and uncertainty grows*



Observed past points

k = 13 / 13

Baseline (right panel)

Linear regression

Move the slider to reveal more past observations. Use the dropdown to switch the baseline on the right.

In this way, GPs are a tool that lets you go from simple regression to something which captures your uncertainty in the future and gives you a fit which naturally reflects your beliefs.

5. Simple and Complex Models

Gaussian Processes are flexible: by choosing different settings, they can be made more simple or more complex.

- **Simple models** are very smooth — they miss details and underfit.
- **Complex models** are very wiggly — they follow the data too closely and overfit.

We can find a balance between simplicity and flexibility by changing the kernel settings, called **hyperparameter**. To control for example the smoothness and wiggliness of functions we set the lengthscale (ℓ):

- Short (ℓ): very wiggly functions — they follow the data closely but may overfit.
- Long (ℓ): very smooth functions — they miss details and underfit.
- Medium (ℓ): a good balance between the two.

Use the slider below to see how changing the lengthscale (ℓ) makes the GP simpler or more complex.

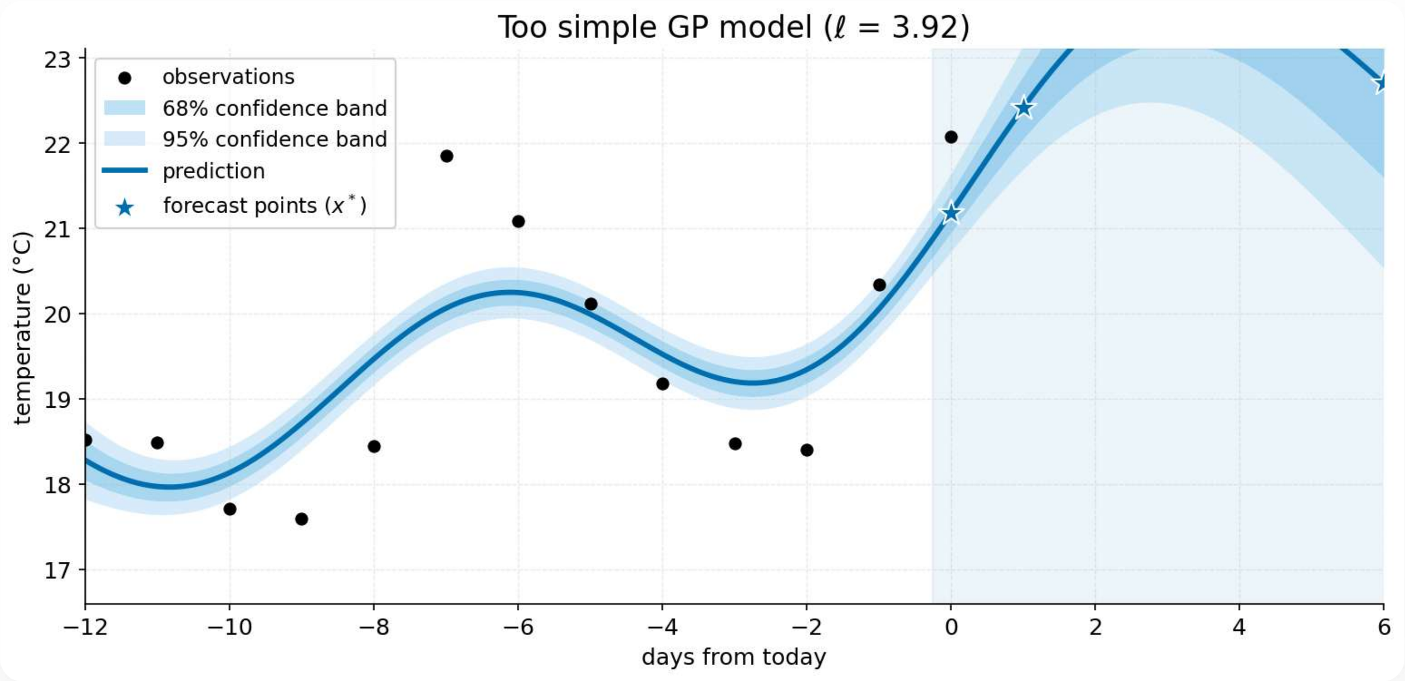
Model Settings

Lengthscale ℓ

ℓ = 3.92 · **Just right**

Zero mean × RBF kernel

Choosing between simple and complex models



Wrapping-up: Why Gaussian Processes?

Most regression models give you one best-fit function. That’s useful, but also risky: it hides how uncertain the model really is.

Gaussian Processes (GPs) are different:

- They are more data efficient: GPs can capture patterns from just a handful of points.
- They let you encode prior assumptions about data patterns and relations with covariance and mean functions.
- They provide uncertainty estimates for every prediction, so you know what the model doesn’t know.

That’s why we want GPs. And now you’ve seen the whole picture — in just 15 minutes.

Main Takeaway: A Gaussian Process is a distribution over functions, defined by a mean and a covariance. With two simple operations — conditioning on data and marginalising at test points — GPs combine flexibility, interpretability, and principled uncertainty in a way most standard regression models cannot.

Further Resources

For a deeper dive into Gaussian Processes, you can consider exploring the following resources:

Book *Gaussian Processes for Machine Learning* by Carl Edward Rasmussen and Christopher K. I. Williams.

Video Lecture: Richard Turner's Lecture on Gaussian Processes (November 23, 2016)

Online Tutorials:

- Visual Exploration of Gaussian Processes
- Gaussian Process Tutorial by Peter Roelants
- Visualization of Gaussian Processes