Derivatives of GPs

Two things need understanding:

- how the GP representation of a derivative of an unknown function represented by a GP is related to the GP representation of the function,
- how to compare the alignment of two gradient fields.

1 How the GP representation of a derivative of an unknown function represented by a GP is related to the GP representation of the function

1.1 Fitting a GP in `R`

The following is taken from Keirstead (2012), a demo of Gaussian process regression with R by James Keirstead (5 April 2012). He writes:

> Chapter 2 of Rasmussen and Williams (2006) provides a detailed explanation of the math for Gaussian process regression. It doesn't provide much in the way of code though. This Gist is a brief demo of the basic elements of Gaussian process regression, as described on pages 13 to 16.

1.1.1 The GP prior

First for a (dense) set of 50 regularly-spaced input values (denoted x^*) we populate the covariance matrix using the chosen covariance function, or *kernel*, $k(\cdot,\cdot)$, in this case the squared exponential with length parameter $\Psi = 1$. One such covariance matrix $k(x^*, x^*)$ is printed below, but for only 10 regularly-spaced points, rather than the 50 used in the remaining of the work below:

$$\begin{bmatrix}
1 & 0.784 & 0.377 & 0.112 & 0.02 & 0.002 & 0 & 0 & 0 & 0 \\
0.784 & 1 & 0.784 & 0.377 & 0.112 & 0.02 & 0.002 & 0 & 0 & 0 \\
0.377 & 0.784 & 1 & 0.784 & 0.377 & 0.112 & 0.02 & 0.002 & 0 & 0 \\
0.112 & 0.377 & 0.784 & 1 & 0.784 & 0.377 & 0.112 & 0.02 & 0.002 & 0 \\
0.02 & 0.112 & 0.377 & 0.784 & 1 & 0.784 & 0.377 & 0.112 & 0.02 & 0.002 \\
0.002 & 0.02 & 0.112 & 0.377 & 0.784 & 1 & 0.784 & 0.377 & 0.112 & 0.02 \\
0 & 0.002 & 0.02 & 0.112 & 0.377 & 0.784 & 1 & 0.784 & 0.377 & 0.112 \\
0 & 0 & 0.002 & 0.02 & 0.112 & 0.377 & 0.784 & 1 & 0.784 & 0.377 \\
0 & 0 & 0 & 0.002 & 0.02 & 0.112 & 0.377 & 0.784 & 1 & 0.784 \\
0 & 0 & 0 & 0 & 0.002 & 0.02 & 0.112 & 0.377 & 0.784 & 1
\end{bmatrix}$$

Figure 1 plots some sample functions drawn from the the zero-mean Gaussian process prior with the aforementioned covariance matrix in order to give an idea of the type of functions it specifies.

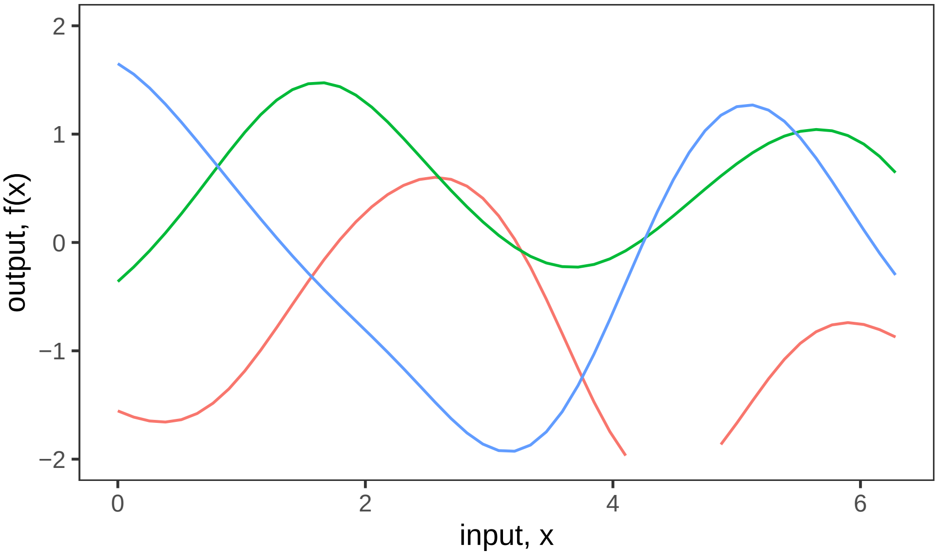


Figure 1: Three functions sampled from the GP prior distribution.

1.1.2 GP posteriors

1.1.2.1 Noisy-observations but a noise-free GP posterior Next, we'll generate 5 data points using the $\sin(x)$ function in the range $x \in [0.2, 2\pi]$, imagining that that is the underlying real-like function we're interested in learning about. An observation error, randomly sampled from $\mathrm{N}(0, 0.1^2)$, is added to each term. The 'observed' values are denoted z and the associated input values x. The observations produced are shown in Figure 2, and the observation error added on visible by the fact that they tend not to sit exactly on the function.

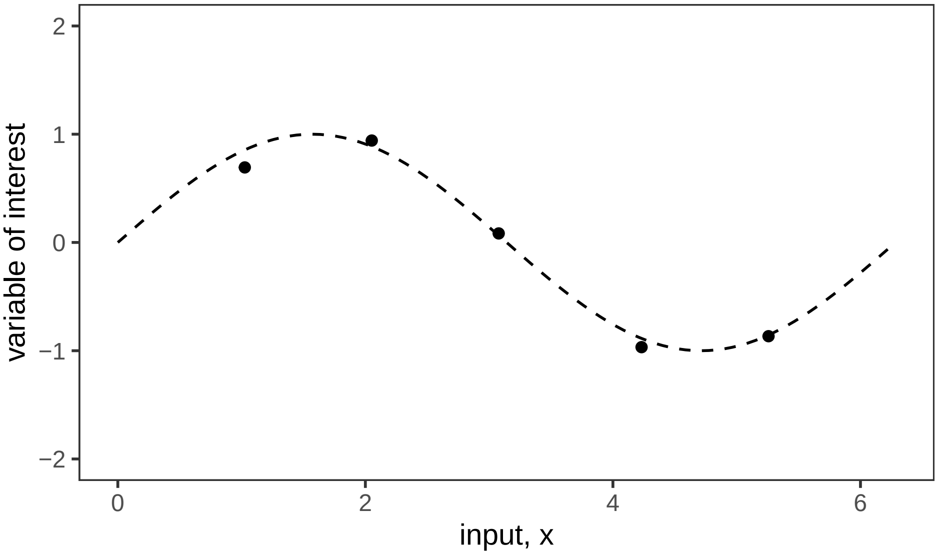


Figure 2: The true process, $\zeta(x)$, is indicated with the line, and five observations, z, by the points.

To specify the posterior GP distribution, we need three further covariance matrices: $k(x, x)$, $k(x, x^*)$ and $k(x^*, x)$ (recall that $k(\cdot, \cdot)$ was described in Section 1.1.1.). Using these four covaraince matrices, The posterior distribution is derived using Equation (2.19) in Rasmussen and Williams (2006):

$$\mathbf{f}^*|x^*, x, \mathbf{f} \sim N(k(x^*, x)k(x, x)^{-1}\mathbf{y},\\ k(x^*, x^*) - k(x^*, x)k(x, x)^{-1}k(x, x^*))$$

Next, we'll generate and plot 20 functions from the posterior distribution, along with the mean function and 95% confidence interval. Outside of the interval on which we have data, the mean function returns towards zero.

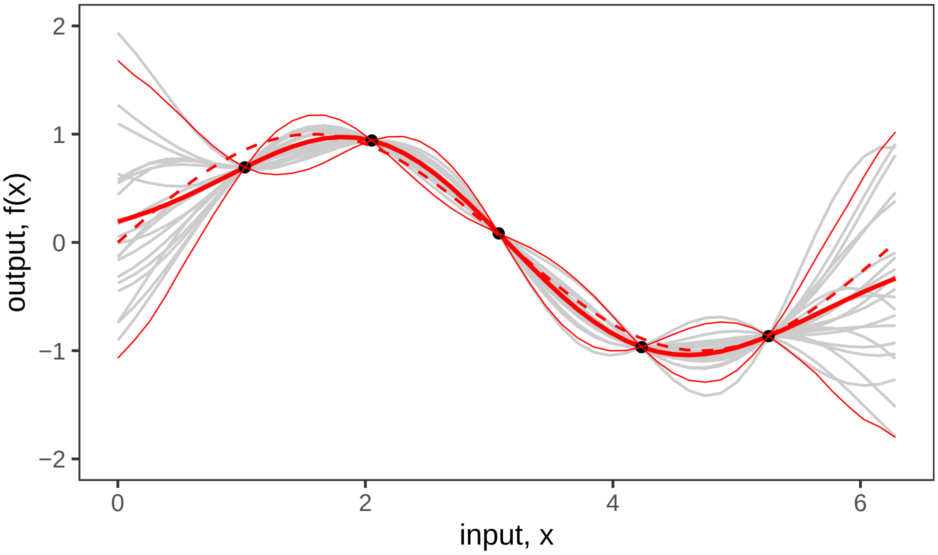


Figure 3: The posterior mean function (red solid line) along with 20 functions sampled from the posterior (grey) and 95% confidence intervals (red dashed line). The underlying real-life function is again a dashed black line.

1.1.2.2 Noisy-observations and a GP posterior with noise Since there is observation error, it might make more sense for the posterior draws *not* to pass through the observed points. This can be achieved by adding a constant (observation noise) term onto the diagonal of the covariance matrix. Recalling that normally distributed measurement errors sampled from N(0, 0.01) were added to the simulated values, we'll add 0.1 to the diagonal of the covariance matrix as observation noise.

Figure 4 replicates Figure 3 to incorporate this added noise, and now the mean function (thick red line) doesn't pass through the data points, and clearly the posterior uncertainty (thin solid red lines) has increased:

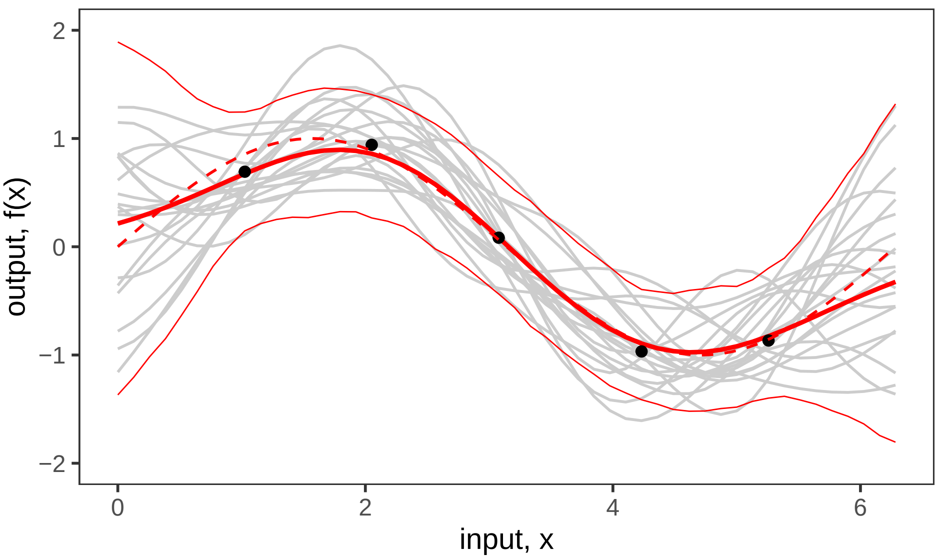


Figure 4: The posterior mean function (red solid line) along with 20 functions sampled from the posterior (grey) and 95% confidence intervals (red dashed line). The underlying real-life function is again a dashed black line.

1.2 The GP representation of a derivative of an unknown function represented by a GP

Having fitted a GP to the data, and arrived at a posterior mean function that lies close to the real-life underlying function (Figure 4), we turn our attention to

The GP posterior with noise built in Section (2.2.2) will be used. To obtain the derivative of the posterior mean function, we first need an expression for the posterior mean function. Viewing the posterior mean function, $\bar{f}(x^*)$, as a linear combination of 5 kernel functions, each centered at one of the 5 training points, Equation (2.27) in Rasmussen and Williams (2006), states that, for any particular input value x^*,

$$\bar{f}(x^*) = \sum_{i=1}^{5} a_i k(x_i, x^*)$$

where $\boldsymbol{\alpha} = (k(x,x) + 0.1I_5)^{-1}\mathbf{y}$. As such, the posterior mean function here will be

$$\begin{aligned}\bar{f}(x^*) = \alpha_1 \exp\left(-\frac{1}{2}\left(\frac{x_1 - x^*}{l}\right)^2\right) + \alpha_2 \exp\left(-\frac{1}{2}\left(\frac{x_2 - x^*}{l}\right)^2\right) \\ + \alpha_3 \exp\left(-\frac{1}{2}\left(\frac{x_3 - x^*}{l}\right)^2\right) + \alpha_4 \exp\left(-\frac{1}{2}\left(\frac{x_4 - x^*}{l}\right)^2\right) + \alpha_5 \exp\left(-\frac{1}{2}\left(\frac{x_5 - x^*}{l}\right)^2\right), \quad (1)\end{aligned}$$

which is plotted as the solid red line in Figure 5.

From O'Hagan (1992), the derivatives of functions modelled by a Gaussian process with

$$E\left\{\eta(x)\right\} = \mathbf{h}^T(x)\boldsymbol{\beta}$$
$$Cov\left\{\eta(x), \eta(\mathbf{x}')\right\} = k(x,x)$$

can also be modelled by a Gaussian process, with

$$\begin{aligned} E\left\{\frac{\partial}{\partial x}\eta(x)\right\} &= \frac{\partial}{\partial x}\mathbf{h}^T(x)\boldsymbol{\beta} \\ Cov\left\{\frac{\partial}{\partial x}\eta(x), \frac{\partial}{\partial x}\eta(x')\right\} &= \frac{\partial^2}{\partial x \partial x'}k(x, x'). \end{aligned} \tag{2}$$

Led by Equation (2), differentiating the posterior mean function in Equation (1) gives

$$\frac{\partial}{\partial x^*}\bar{f}(x^*) = \frac{1}{l^2}\left[\alpha_1(x_1 - x^*)\exp\left(-\frac{1}{2}\left(\frac{x_1 - x^*}{l}\right)^2\right) + \cdots + \alpha_5(x_5 - x^*)\exp\left(-\frac{1}{2}\left(\frac{x_5 - x^*}{l}\right)^2\right)\right],$$

which is also plotted in Figure 5. The real, underlying function $\sin(x)$, and its derivative $\cos(x)$ are included in Figure 5 too (dashed lines) for comparison.

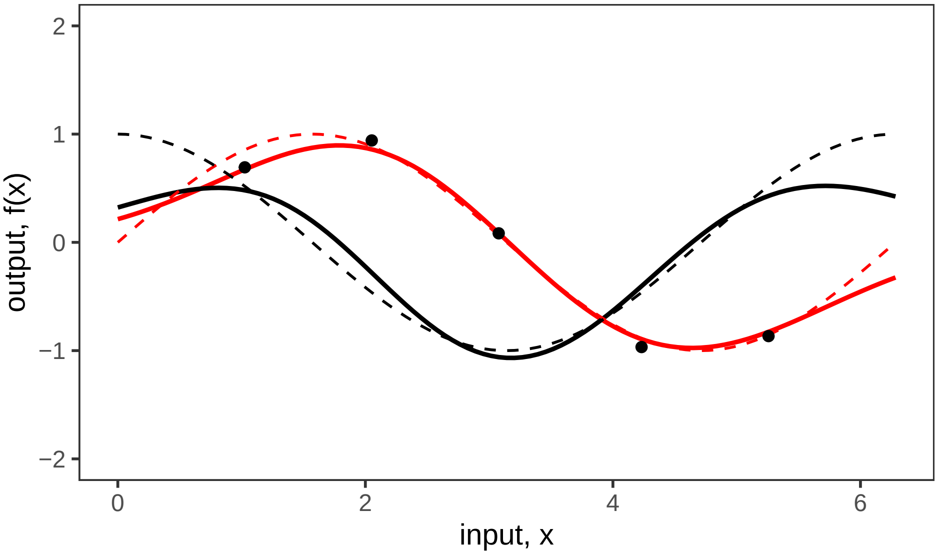


Figure 5: The posterior mean function (red solid line) and the true real-life process function, $\sin(x)$ (red dashed line). Also shown is the derivative of the posterior mean function (black solid line) and the derivative of the true real-life process function, $\cos(x)$ (black dashed line). Observations are shown as black dots.

2 How to compare the alignment of two gradient fields

Assume an underlying function

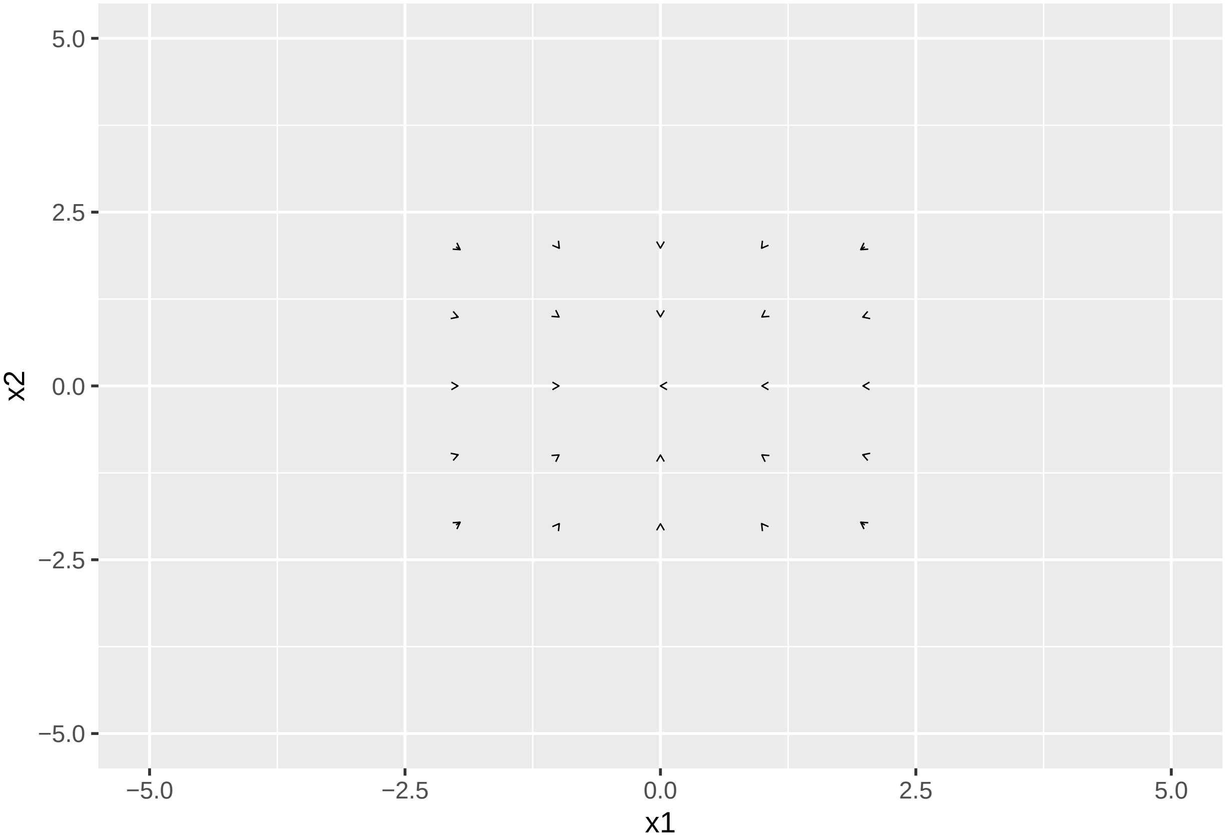
$$f(x) = x_1^2 x_2$$

with partial derivatives

$$\frac{d}{d\mathbf{x}} f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \end{bmatrix} = \begin{bmatrix} 2x_1 x_2 \\ x_1^2 \end{bmatrix}.$$

On an 11×11 grid with x_1 and x_2 values of $\{-5, -4, \dots, 5\}$, this returns the following gradient vectors:

```
## Warning: Removed 96 rows containing missing values or values outside the scale range
## (`geom_segment()`).
```



To define the comparison measure, we'll work through the steps outlined in the correspondence between Jonathan and Manolis. Step 1. requires the computation of partial derivatives w.r.t. the parameters used the test the case.

First we'll look at those pairs of fields which are an acceptable pairing, then those which aren't.

2.1 Acceptable pairings

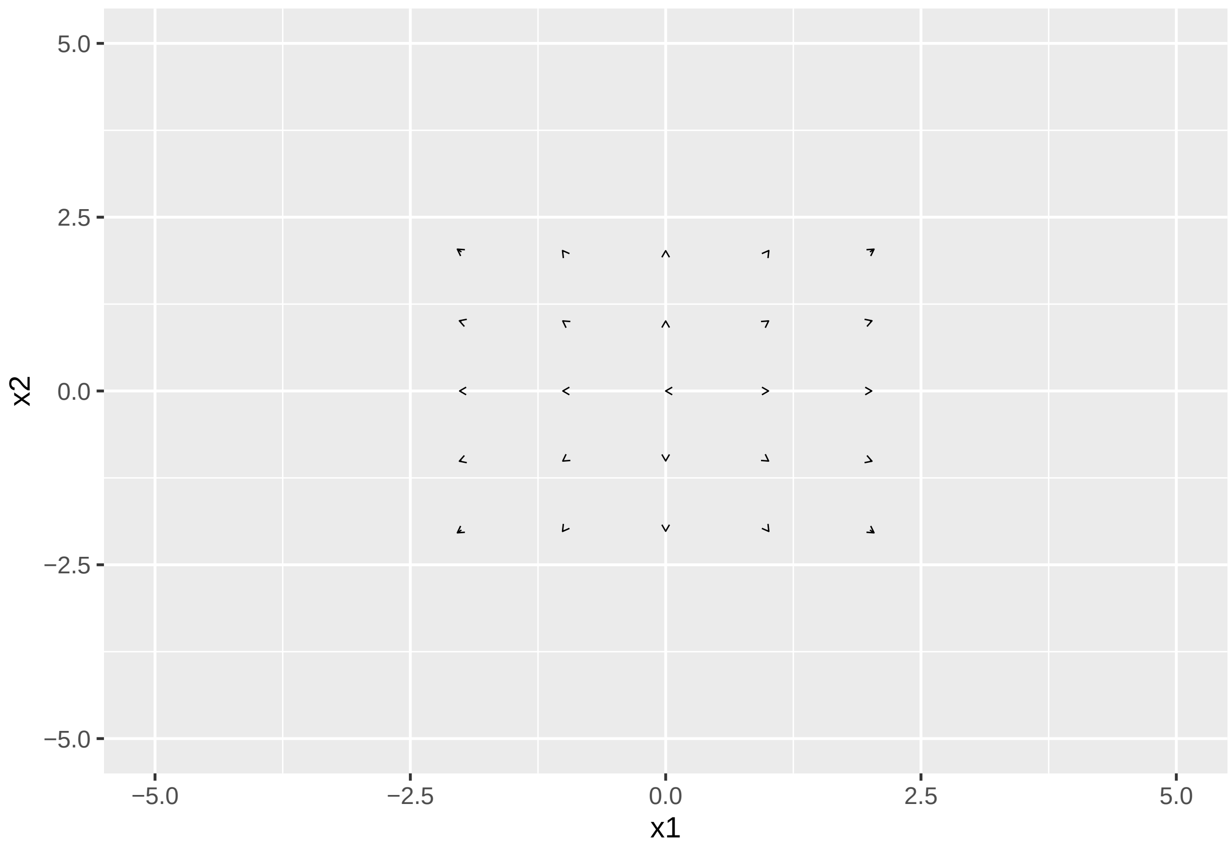
2.1.1 Identical fields

1. Done.
2. Done.
3. Normalise vectors of partial derivatives:

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Contrasting that with a field with exactly the opposite sign to this one:

```
## Warning: Removed 96 rows containing missing values or values outside the scale range
## (`geom_segment()`).
```



These two fields have perfect alignment, and differ only by the direction of the slope. We would want any measure to recognise the matching alignment here.

```
vector_field_1$vx1 * vector_field_2$vx1
```

```
##   [1]        NaN        NaN        NaN        NaN        NaN        NaN
##   [7]        NaN        NaN        NaN        NaN        NaN        NaN
##  [13]        NaN        NaN        NaN        NaN        NaN        NaN
##  [19]        NaN        NaN        NaN        NaN        NaN        NaN
##  [25]        NaN        NaN        NaN        NaN        NaN        NaN
##  [31]        NaN        NaN        NaN        NaN        NaN        NaN
##  [37] -4.0000000 -0.2500000  0.0000000 -0.2500000 -4.0000000        NaN
##  [43]        NaN        NaN        NaN        NaN        NaN -1.0000000
##  [49] -0.1428571  0.0000000 -0.1428571 -1.0000000        NaN        NaN
##  [55]        NaN        NaN        NaN       -Inf -0.8000000 -0.1250000
##  [61]  0.0000000 -0.1250000 -0.8000000       -Inf        NaN        NaN
##  [67]        NaN        NaN        NaN -1.0000000 -0.1428571  0.0000000
##  [73] -0.1428571 -1.0000000        NaN        NaN        NaN        NaN
##  [79]        NaN        NaN -4.0000000 -0.2500000  0.0000000 -0.2500000
##  [85] -4.0000000        NaN        NaN        NaN        NaN        NaN
##  [91]        NaN        NaN        NaN        NaN        NaN        NaN
##  [97]        NaN        NaN        NaN        NaN        NaN        NaN
## [103]        NaN        NaN        NaN        NaN        NaN        NaN
## [109]        NaN        NaN        NaN        NaN        NaN        NaN
## [115]        NaN        NaN        NaN        NaN        NaN        NaN
## [121]        NaN
```

Citations

Keirstead, James. 2012. "Gaussian Process Regression with r: R-Bloggers." *R*. https://www.r-bloggers.com/2012/04/gaussian-process-regression-with-r.

O'Hagan, A. 1992. "Some Bayesian Numerical Analysis." In *Bayesian Statistics 4: Proceedings of the Fourth Valencia International Meeting*. Oxford University Press. https://doi.org/10.1093/oso/9780198522669.003.0019.

Rasmussen, C. E., and C. K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.