Exercise 1: Implementation of a Gaussian Process for regression

In this exercise we will implement basic functionalities of a Gaussian Process. We will start by handling data points and in a second step we will implement a service for prediction. You can fill the provided code skeleton.

(a) What is a Gaussian process? How is a Gaussian Process specified? Deduce the parameters needed to create a Gaussian Process. And implement such a initializer.

(b) When a data point is observed, what should be done to update the Gaussian process? We propose the following message type:

```c
float64[] x
float64 t
```

Write the callback function for this message.

(c) In order to use our Gaussian process to do prediction, we propose the following service type:

```c
float64[] x
---
float64 mean
float64 std_dev
```

Write the handler for this service.

(d) We want to test our node with some data (data.csv provided in the archive). The format is \texttt{t,x} where \texttt{x} can be multidimensional. Write a node that reads this file and publish its content to the \texttt{/observation} topic.
(e) How could you display the resulting Gaussian Process? Write a script that display the data points, the mean function and the $2\sigma$ confidence interval.

(f) Change the parameters of the kernel and observe the differences.

**Exercise 2: Sampling from a Gaussian Process**

Recall that Gaussian distributions have several properties that make them very practical in probabilistic computations. Among them, the marginal of a multi-variate Gaussian distribution is a Gaussian distribution. Also, the image of a Gaussian distribution by a linear function remains a Gaussian distribution. Furthermore, the convolution of a Gaussian distribution by another Gaussian distribution is also Gaussian. Mathematically, if:

1. $p(x) = \mathcal{N}(x|\mu, \Lambda^{-1})$,
2. $p(y \mid x) = \mathcal{N}(y|Ax + b, L^{-1})$,

then:

1. $p(y) = \mathcal{N}(y|A\mu + b, L^{-1} + AA^{-1}A^T)$,
2. $p(x \mid y) = \mathcal{N}\left(x\mid \begin{bmatrix} \Sigma & \begin{bmatrix} A^T L(y - b) + \Lambda \mu \end{bmatrix} \end{bmatrix}, \Sigma \right)$ with $\Sigma = (\Lambda + A^T L A)^{-1}$.

We have a Gaussian Process $y(x)$ with given mean $\mu(x)$ and kernel $k(x, x')$, symmetric definite positive. The aim of this exercise is to draw samples from this kernel.

(a) For $x_1 \neq x_2$ two real values, what are the distributions $p(y(x_1))$, $p(y(x_2))$, and $p(y(x_1), y(x_2))$.

(b) Let’s introduce a second Gaussian process $u(x)$ with mean 0 and kernel $k'(x, x') = \delta_{x,x'}$ (1 if $x=x'$, 0 otherwise). What are the distributions $p(u(x_1))$, $p(u(x_2))$, and $p(u(x_1), u(x_2))$? What does that mean for the relation between $u(x_1)$ and $u(x_2)$?, for the Gaussian process $u$ at large?

(c) What is the Gram matrix $K$ for points $x_1$ and $x_2$?

(d) Let $L$ be a lower triangular matrix such that $K = LL^T$ (Cholesky factorization). We define $z(x) = \mu(x) + Lu(x)$. What is the shape of the distribution $p(z(x_1))$? What are the mean for $p(z(x_1))$, $p(z(x_2))$, and $p(z(x_1), z(x_2))$? What is the covariance of $p(z(x_1), z(x_2))$?

(e) Deduce a general approach for sampling a Gaussian Process. Implement it in the previous code, using the following service type:
float64[] x

---

float64[] y