Problem Sheet 4

```julia
begin
    using Distributions
    using PlutoUI
    using LinearAlgebra
    using DelimitedFiles
    using DataFrames
    using KernelFunctions
    using Plots
    using Random
    using StatsPlots
    default(;linewidth=3.0, legendfontsize=15.0)
end
```

Table of Contents

1. Gaussian process (GP) regression

For the GP regression problem, we assume that data are generated as

$$y_i = f(x_i) + \nu_i \qquad i = 1, \ldots, n$$

where the ν_i are independent, zero mean Gaussian noise variables within $E[\nu_i^2] = \sigma^2$ and $f(\cdot)$ has a GP prior with kernel $K(x, x')$.

(a) [MATH] Show that the Bayesian evidence is given by

$$p(\mathbf{y}) = \frac{1}{(2\pi)^{n/2}|\det(\mathbf{K}+\sigma^2\mathbf{I})|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}\mathbf{y}^T(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}\right]$$

where $\mathbf{y} = (y_1, \ldots, y_n)$ and the kernel matrix is defined by $\mathbf{K}_{ij} = K(x_i, x_j)$.

Tip

Calculate the joint density of $\mathbf{y}$ and use the fact that $f(x_j)$ and ν_i are independent Gaussian random variables. Hence you can add the respective covariance matrices.

Solution

The evidence can be computed via the joint distribution :

$$p(\mathbf{y}) = \int p(\mathbf{y},\mathbf{f})d\mathbf{f} = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f}$$

Where

$$\begin{aligned} p(\mathbf{y}|\mathbf{f}) &= \frac{1}{(2\pi)^{N/2}|\det(\sigma^2\mathbf{I})|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{f})^\top\sigma^{-2}\mathbf{I}(\mathbf{y}-\mathbf{f})\right] \\ p(\mathbf{f}) &= \frac{1}{(2\pi)^{N/2}|\det(\mathbf{K})|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f}\right] \end{aligned}$$

In the integration one can do it the hard way and reformulate

$$p(\mathbf{y}\mid\mathbf{f})p(\mathbf{f}) \equiv \mathcal{N}(y\mid 0, \sigma^2\mathbf{I}+\mathbf{K})\mathcal{N}(f\mid\overline{\mu},\overline{\Sigma})$$

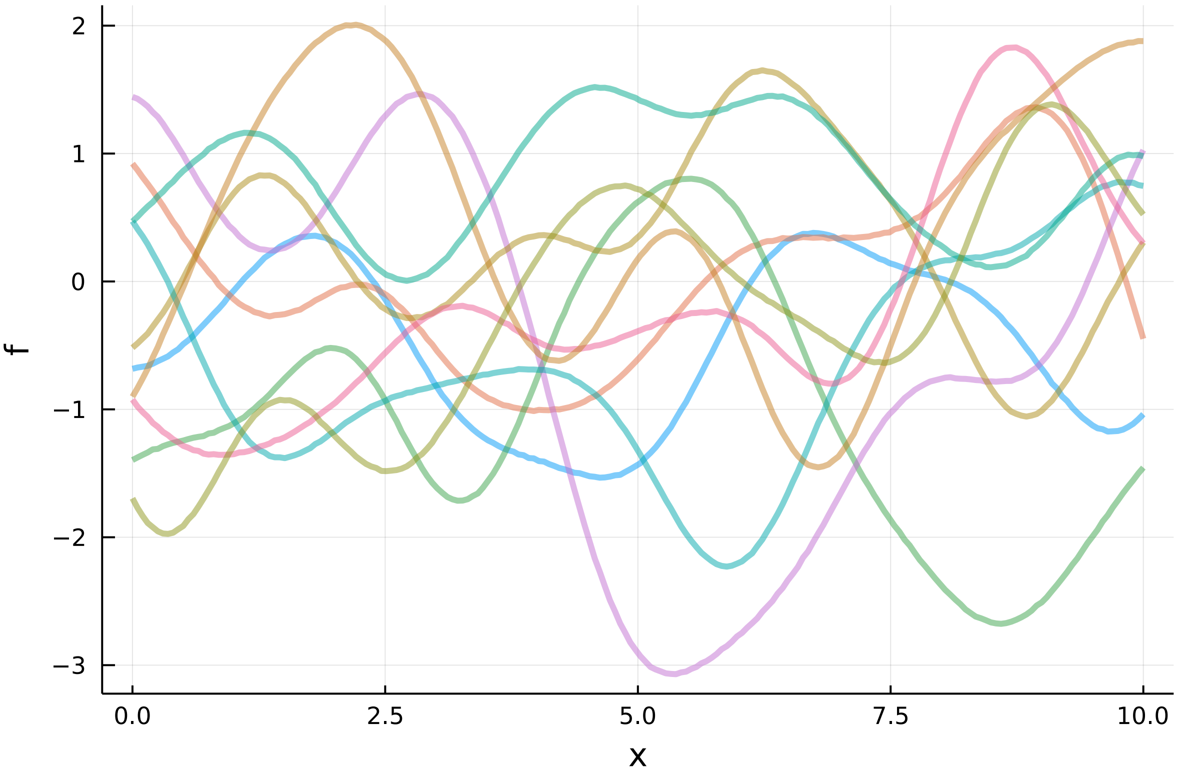
using the identity

$$\begin{aligned} \mathcal{N}(x\mid m_1,\Sigma_1)\cdot\mathcal{N}(x\mid m_2,\Sigma_2) =& \mathcal{N}(m_1\mid m_2,(\Sigma_1+\Sigma_2)\mathcal{N}(x\mid\overline{m},\overline{\Sigma}) \\ \overline{m} =& (\Sigma_1^{-1}+\Sigma_2^{-1})^{-1}(\Sigma_1^{-1}m_1+\Sigma_2^{-1}m_2) \\ \overline{\Sigma} =& (\Sigma_1^{-1}+\Sigma_2^{-1})^{-1} \end{aligned}$$

Replacing x by $\mathbf{f}$, the integral over $\mathbf{f}$ gives one and we recover the multivariate gaussian for $\mathbf{y}$. Or one can intuitively see that we are having a zero mean prior on the mean ($\mathbf{f}$), and therefore one can simply add the variances to get the final result.

(b) [CODE] Create a Gaussian prior with zero mean and a RBF Kernel and sample from it on a grid

```
begin
    x_test = range(0, 10, length = 200)
    k = SqExponentialKernel() # This computes k(x, x') = exp(-0.5||x - x'||^2)
    K = kernelmatrix(k, x_test) + 1e-5I
    prior_f = MvNormal(K)
    S = 10 # Number of GP samples
end;
```



(c) [MATH] Given a set of training data (X, y), compute the predictive distribution of some test data X_{test}

Solution

Given $f = f(X)$ and $f^* = f(X_{\text{test}})$, we know that the joint distribution $p(f, f^*) = \mathcal{N}(0, K)$, where K is equal to :

$$K = \begin{pmatrix} K_X & K_{X,X_{test}} \\ K_{X_{test},X} & K_{X_{test}} \end{pmatrix}$$

Using the properties of Gaussian distributions we get the exact conditional $p(f^*|f) = \mathcal{N}(\mu(f), \Sigma)$ where $\mu(f) = K_{X_{test},X}K_X^{-1}f$ and $\Sigma = K_{X_{test}} - K_{X_{test},X}K_X^{-1}K_{X,X_{test}}$.

Now if we have some data y we can compute the predictive distribution as :

$$p(f^*|y) = \int p(f^*|f)p(f|y)df = \mathcal{N}(f^*|\mu^*, \Sigma^*)$$

where $\mu^* = K_{X_{test},X}(K_X + \sigma^2 I)^{-1}y$ and $\Sigma^* = K_{X_{test}} - K_{X_{test},X}(K_X + \sigma^2 I)^{-1}K_{X,X_{test}}$

(d) [CODE] Implement the predictive distribution and plot the predictive mean along with one standard error

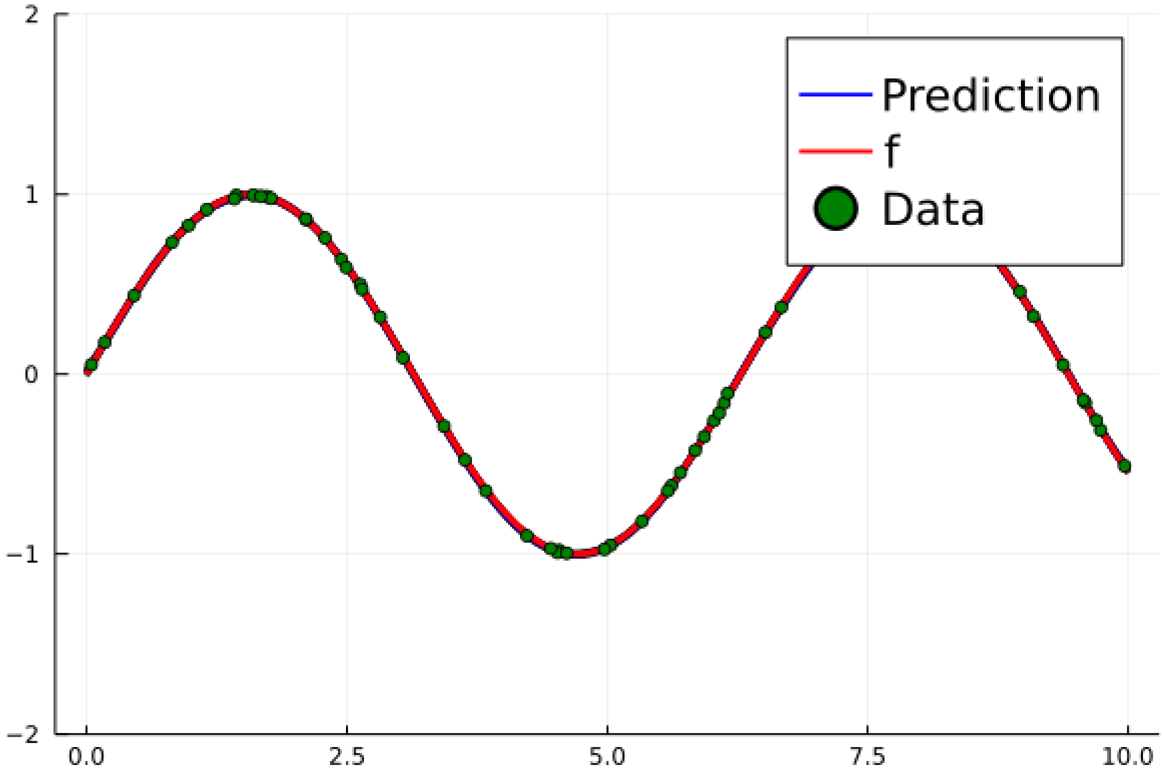
N 57

logσ -5

```
begin
    rng = Random.MersenneTwister(42)
    X = rand(rng, Uniform(0, 10), N)
    σ = exp(logσ)
    y = sin.(X) + randn(rng, N) * σ
end;
```

```
function pred_mean_and_cov(k, x_test, x, y)
    Kx = kernelmatrix(k, x)
    Σ = Kx + σ^2 * I
    Kxtest_x = kernelmatrix(k, x_test, x)
    Kxtest = kernelmatrix(k, x_test) + 1e-5I
    return Kxtest_x * inv(Σ) * y, Kxtest - Kxtest_x * inv(Σ) * Kxtest_x'
end;
```

```
m, C = pred_mean_and_cov(k, x_test, X, y);
```




```
begin
    plot(x_test, rand(MvNormal(m, Symmetric(C)), S * 10), color=:black, label="",
alpha=0.1, ylims=(-2, 2))
    plot!(x_test, m, ribbon = sqrt.(diag(C)), color=:blue, fillalpha=0.2, label =
"Prediction")
    plot!(x_test, sin.(x_test), color=:red, label="f")
    scatter!(X, y, color=:green, label = "Data")
end
```

2. Gibbs sampler for outlier detection

The file *outlier.dat* on the web page of the course contains a data set $D = (y_1, \ldots, y_N)$. Most of the observations have been drawn from a Gaussian probability distribution $\mathcal{N}(y_i; \mu, \sigma^2)$ with mean μ and variance σ^2. However, D contains some **outliers**, which occur with probability ϵ and are displaced by a random offset A_i. For the purpose of **outlier detection** the model is augmented with an indicator variable

$$\delta_i = \begin{cases} 1 & \text{if } y_i \text{ is an outlier,} \\ 0 & \text{if } y_i \text{ is a normal data point,} \end{cases}$$

for each observation. Assuming conjugate priors for the parameters yields the full stochastic model

$$\begin{array}{lllllllll} \mu & \sim & \mathcal{N}(\theta, v^2), & \sigma^{-2} & \sim & \text{Gamma}(\kappa, \lambda), & \epsilon & \sim & \text{Beta}(\alpha, \beta), \\ y_i & \sim & \mathcal{N}(\mu + \delta_i\, A_i, \sigma^2), & \delta_i & \sim & \text{Bernoulli}(\epsilon), & A_i & \sim & \mathcal{N}(0, \tau^2). \end{array}$$

We want to use a Gibbs sampler in order to draw samples from the posterior $p(\mu, \sigma^2, \epsilon, \boldsymbol{\delta}, \mathbf{A}|D)$ with $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)$ and $\mathbf{A} = (A_1, \ldots, A_N)$. Some conditional posteriors are given by

$$\begin{aligned} \mu \sim & \mathcal{N}\left(\frac{\sigma^2\theta + v^2 \sum_{i=1}^N (y_i - \delta_i\, A_i)}{\sigma^2 + Nv^2}, \frac{\sigma^2 v^2}{\sigma^2 + Nv^2}\right), \\ \sigma^{-2} \sim & \text{Gamma}\left(\kappa + \frac{N}{2}, \frac{2\lambda}{2 + \lambda \sum_{i=1}^N (y_i - \delta_i\, A_i - \mu)^2}\right). \end{aligned}$$

- ## <u>MATH</u> Show that the remaining conditional posteriors are given by

$$\begin{aligned} \delta_i \sim & \text{Bernoulli}\left(\frac{\epsilon}{\epsilon + (1-\epsilon)\exp(-A_i(y_i - A_i - \mu)/(2\sigma^2))}\right), \\ A_i \sim & \mathcal{N}\left(\frac{\tau^2\delta_i(y_i - \mu)}{\sigma^2 + \tau^2}, \frac{\sigma^2\tau^2}{\sigma^2 + \tau^2\delta_i}\right), \\ \epsilon \sim & \text{Beta}\left(\alpha + \sum_{i=1}^N \delta_i, \beta + \sum_{i=1}^N (1 - \delta_i)\right). \end{aligned}$$

Solution

- Joint probability distribution

$$\begin{aligned} p(\mu, \sigma^2, \epsilon, \boldsymbol{\delta}, \mathbf{A}, D) = & \frac{\Gamma(\alpha+\beta)\epsilon^{\alpha-1}(1-\epsilon)^{\beta-1}}{\sqrt{2\pi v^2}\Gamma(\kappa)\lambda^\kappa\Gamma(\alpha)\Gamma(\beta)} \sigma^{-2(\kappa-1)} e^{-\frac{\sigma^{-2}}{\lambda} - \frac{(\mu-\theta)^2}{2v^2}} \\ & \times \prod_{i=1}^N \frac{\epsilon^{\delta_i}(1-\epsilon)^{1-\delta_i}}{2\pi\sigma\tau} e^{-\frac{(y_i - \delta_i\, A_i - \mu)^2}{2\sigma^2} - \frac{A_i^2}{2\tau^2}} \end{aligned}$$

- Conditional distributions

$$
\begin{aligned}
p(\mu|\ldots) \propto& e^{-\frac{(\mu-\theta)^2}{2v^2}-\sum_{i=1}^N \frac{(y_i-\delta_i A_i-\mu)^2}{2\sigma^2}} \\
\Rightarrow p(\mu|\ldots) =& \mathcal{N}\left(\mu|\left(\frac{1}{\nu^2}+\frac{N}{\sigma^2}\right)^{-1}\left(\frac{\theta^2}{\nu^2}+\frac{1}{\sigma^2}\sum_i^N y_i-\delta_i A_i)\right),\left(\frac{1}{\nu^2}+\frac{N}{\sigma^2}\right)^{-1}\right) \\
p(\sigma^{-2}|\ldots) \propto& \sigma^{-2(\kappa+N/2-1)} e^{-\sigma^{-2}\left(\frac{1}{\lambda}+\sum_{i=1}^N \frac{(y_i-\delta_i A_i-\mu)^2}{2}\right)} \\
\Rightarrow p(\sigma^{-2}|\ldots) =& \text{Gamma}\left(\sigma^{-2} \mid \kappa+\frac{N}{2}, \frac{2\lambda}{2+\lambda\sum_{i=1}^N (y_i-\delta_i\, A_i-\mu)^2}\right). \\
p(\delta_i|\ldots) \propto& \epsilon^{\delta_i}(1-\epsilon)^{1-\delta_i} e^{-\delta_i \frac{A_i(A_i+\mu-y_i)}{2\sigma^2}} \\
\Rightarrow p(\delta_i \mid \ldots) =& \text{Bernoulli}\left(\frac{\epsilon}{\epsilon+(1-\epsilon)\exp(-A_i(y_i-A_i-\mu)/(2\sigma^2))}\right)
\end{aligned}
$$

To get the new parameter of the Bernoulli distribution, compute the normalization constant by summing over $\delta_i = \{0, 1\}$:

$$
\begin{aligned}
p(\delta_i=0) \propto& (1-\epsilon), \quad p(\delta_i=1) \propto \epsilon e^{-\frac{A_i(A_i+\mu-y_i)}{2\sigma^2}} \\
\Rightarrow p(\delta_i=1) =& \frac{\epsilon e^{-\frac{A_i(A_i+\mu-y_i)}{2\sigma^2}}}{(1-\epsilon)+\epsilon e^{-\frac{A_i(A_i+\mu-y_i)}{2\sigma^2}}} = \frac{\epsilon}{(1-\epsilon)e^{\frac{A_i(A_i+\mu-y_i)}{2\sigma^2}}+\epsilon} \\
p(A_i|\ldots) \propto& e^{-\frac{(y_i-\delta_i\, A_i-\mu)^2}{2\sigma^2}-\frac{A_i^2}{2\tau^2}} \\
\Rightarrow p(A_i|\ldots) =& \mathcal{N}\left(A_i \mid \frac{\tau^2\delta_i(y_i-\mu)}{\sigma^2+\tau^2}, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2\delta_i}\right) \\
p(\epsilon|\ldots) \propto& \epsilon^{\alpha-1+\sum_{i=1}^N \delta_i}(1-\epsilon)^{\beta-1+\sum_{i=1}^N (1-\delta_i)} \\
\Rightarrow p(\epsilon|\ldots) =& \text{Beta}\left(\epsilon \mid \alpha+\sum_{i=1}^N \delta_i, \beta+\sum_{i=1}^N (1-\delta_i)\right).
\end{aligned}
$$

- **(b) [CODE] Write a program that implements the *Gibbs sampler*. Generate 10^3 samples from the posterior using the hyperparameters $\theta = 0$, $v^2 = 100$, $\kappa = 2$, $\lambda = 2$, $\alpha = 2$, $\beta = 20$, $\tau^2 = 100$. Plot histograms showing the marginal posteriors $p(\mu|D)$ and $p(\epsilon|D)$.**

Solution

```julia
function sample_μ(σ², θ, ν², y, δ, A, N)
    m = (σ² * θ  + ν² * sum(y - δ .* A)) / (σ² + N * ν²)
    c = σ² * ν² / (σ² + N * ν²)
    return rand(Normal(m, sqrt(c)))
end;
```

```julia
function sample_σ²(κ, N, λ, y, δ, A, μ)
    a = κ + N / 2
    b = 2λ / (2 + λ * sum(abs2, y - δ .* A .- μ))
    return inv(rand(Gamma(a, b)))
end;
```

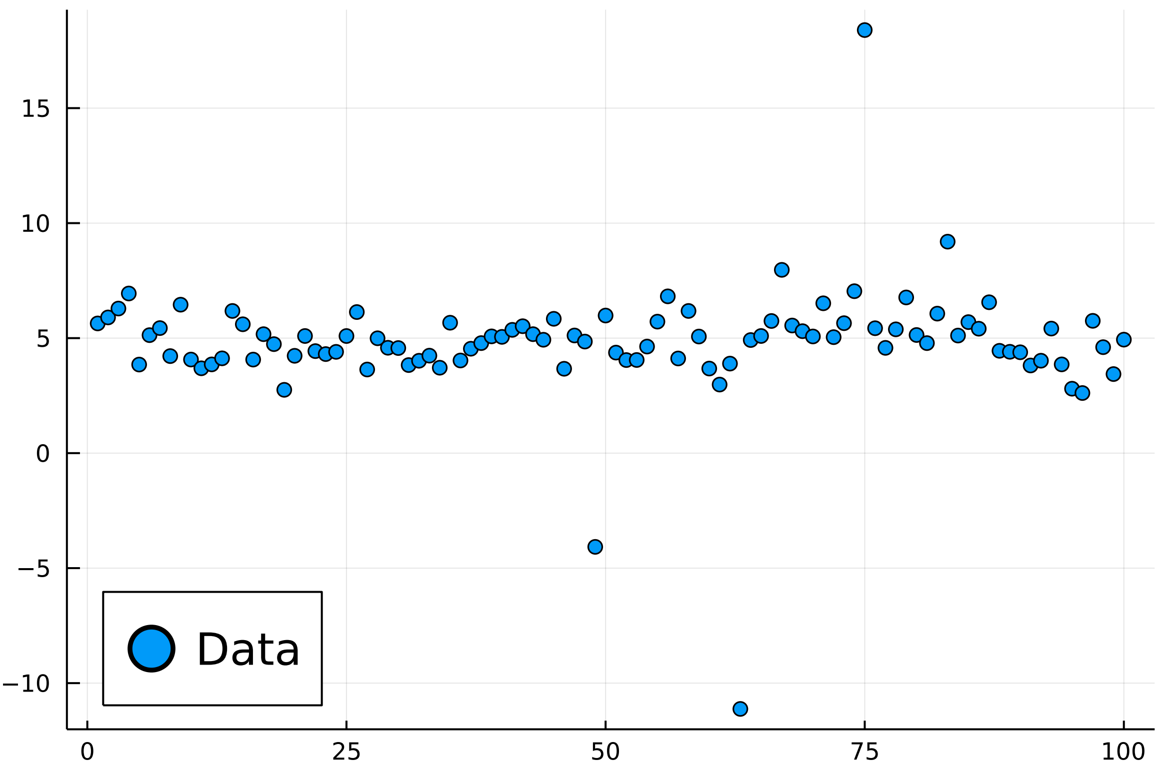
```julia
function sample_δ(ϵ, A, y, μ, σ²)
    θ = ϵ ./ (ϵ .+ (1 - ϵ) * exp.(-A .* (y .- A .- μ) ./ (2σ²)))
    return rand.(Bernoulli.(θ))
end;
```

```julia
function sample_A(τ², δ, y, μ, σ²)
    m = τ² * δ .* (y .- μ) / (σ² + τ²)
    c = σ² * τ² ./ (σ² .+ τ² .* δ)
    rand.(Normal.(m, sqrt.(c)))
end;
```

```julia
function sample_ϵ(α, δ, β)
   rand(Beta(α + sum(δ), β + sum(1 .- δ)))
end;
```

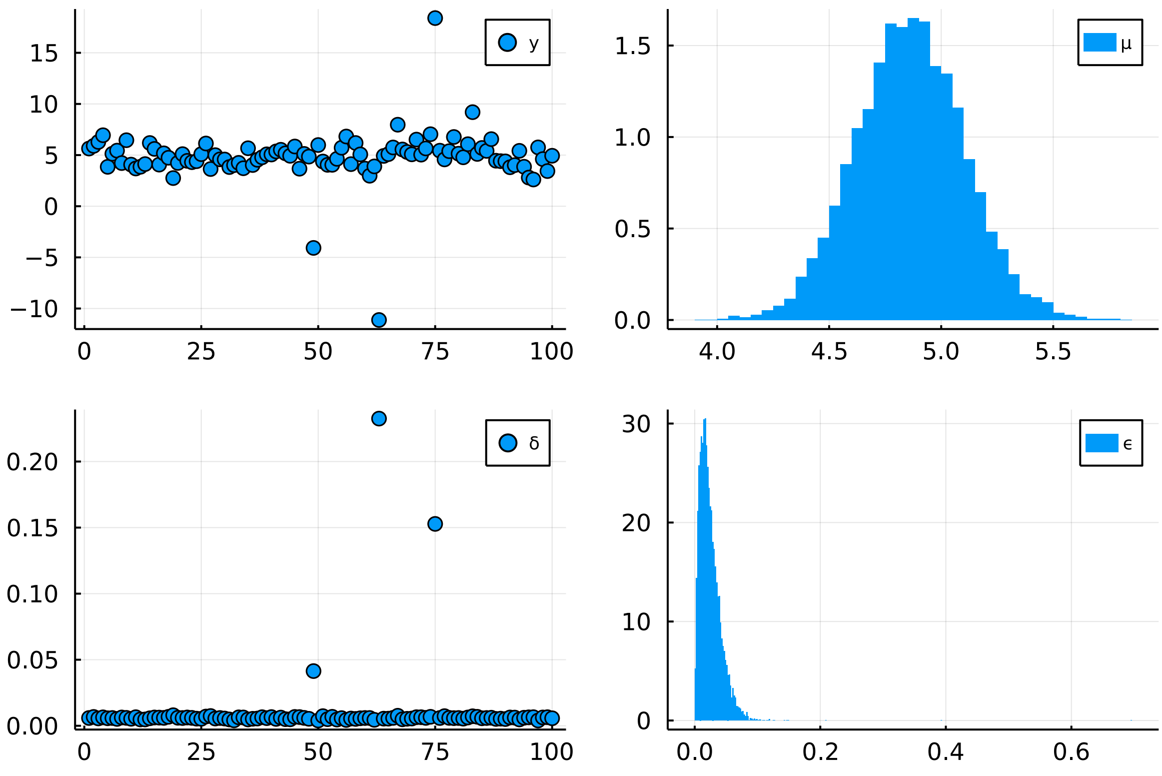
100

```julia
begin # We load the data
    y_outlier = vec(readdlm("outlier.dat"))
    Ny = length(y_outlier)
end
```



```julia
begin # We select multiple hyperparameters
    T = 10000
    θ = 0.0
    ν² = 100
    κ = 2
    λ = 2
    α = 2
    β = 20
    τ² = 100
end;
```

```julia
begin
    # We initialize the random variables and preallocate storage
    A = randn(Ny); As = zeros(Ny, T)
    δ = rand(0:1, Ny); δs = zeros(Ny, T)
    ϵ = rand(); ϵs = zeros(T)
    σ² = rand(); σ²s = zeros(T)
    μ = randn(); μs = zeros(T)
    for i in 1:T
        μ = sample_μ(σ², θ, ν², y_outlier, δ, A, Ny); μs[i] = μ
        σ² = sample_σ²(κ, Ny, λ, y_outlier, δ, A, μ); σ²s[i] = σ²
        δ = sample_δ(ϵ, A, y_outlier, μ, σ²); δs[:, i] = δ
        A = sample_A(τ², δ, y_outlier, μ, σ²); As[:, i] = A
        ϵ = sample_ϵ(α, δ, β); ϵs[i] = ϵ
    end
end
```




```julia
begin
    p1 = scatter(1:Ny, y_outlier, label = "y")
    p2 = histogram(μs, label = "μ", normalize = true, lw = 0.0)
    p3 = scatter(1:Ny, vec(mean(δs, dims = 2)), label = "δ")
    p4 = histogram(ϵs, label = "ϵ", normalize = true, lw = 0.0)
    plot(p1, p2, p3, p4, legendfontsize=6.0)
end
```

(c) Which data points in the file *outlier.dat* are outliers? Use the samples generated in part (b) and the condition $p(\delta_i|D) \geq 0.02$ in order to identify them.

	y	pδ
1	-4.07584	0.0414
2	-11.1217	0.2325
3	18.3938	0.1528

```
begin
    df = DataFrame([y_outlier[:], vec(mean(δs, dims = 2))], [:y, :pδ])
    df[df.pδ .>= 0.02, :]
end
```