

Akshata Patil

AA222: Engineering Design Optimization

Date of Submission: 05/26/2023

Project 3:

Part 1: Noiseless Gaussian Process Fitting:

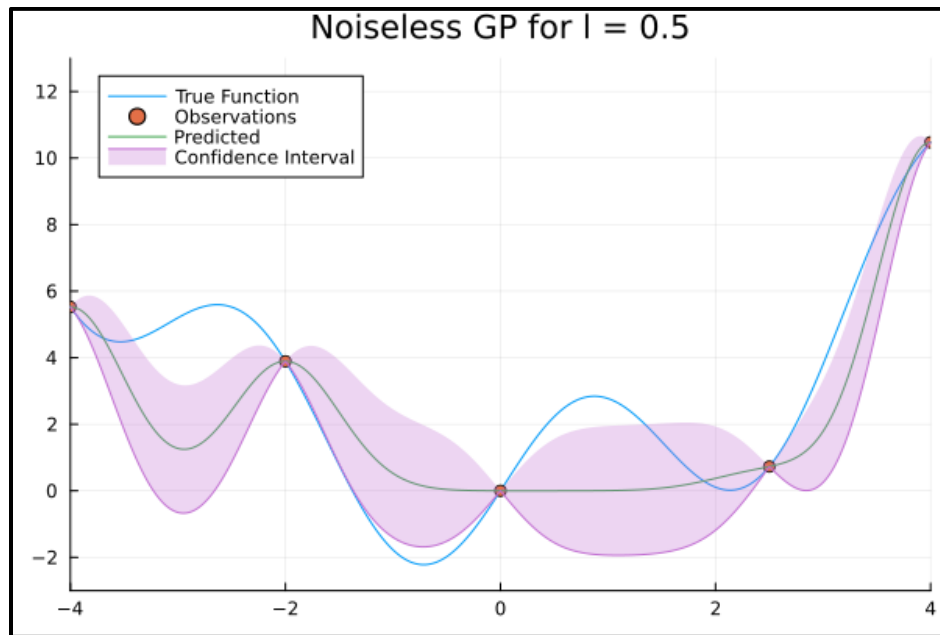


Figure 1: Noiseless Gaussian Process Fitting for characteristic length = 0.5

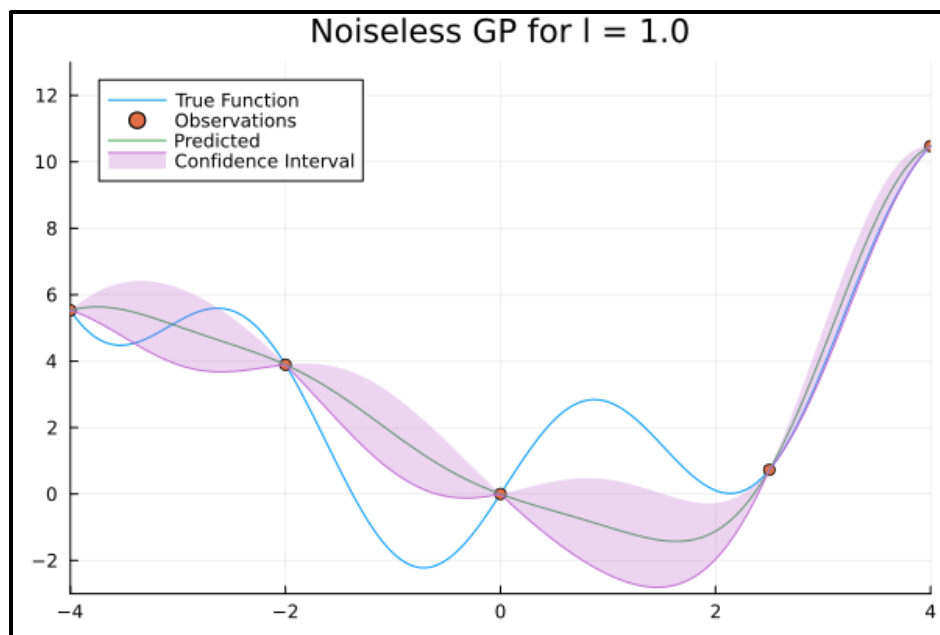


Figure 2: Noiseless Gaussian Process Fitting for characteristic length = 1.0

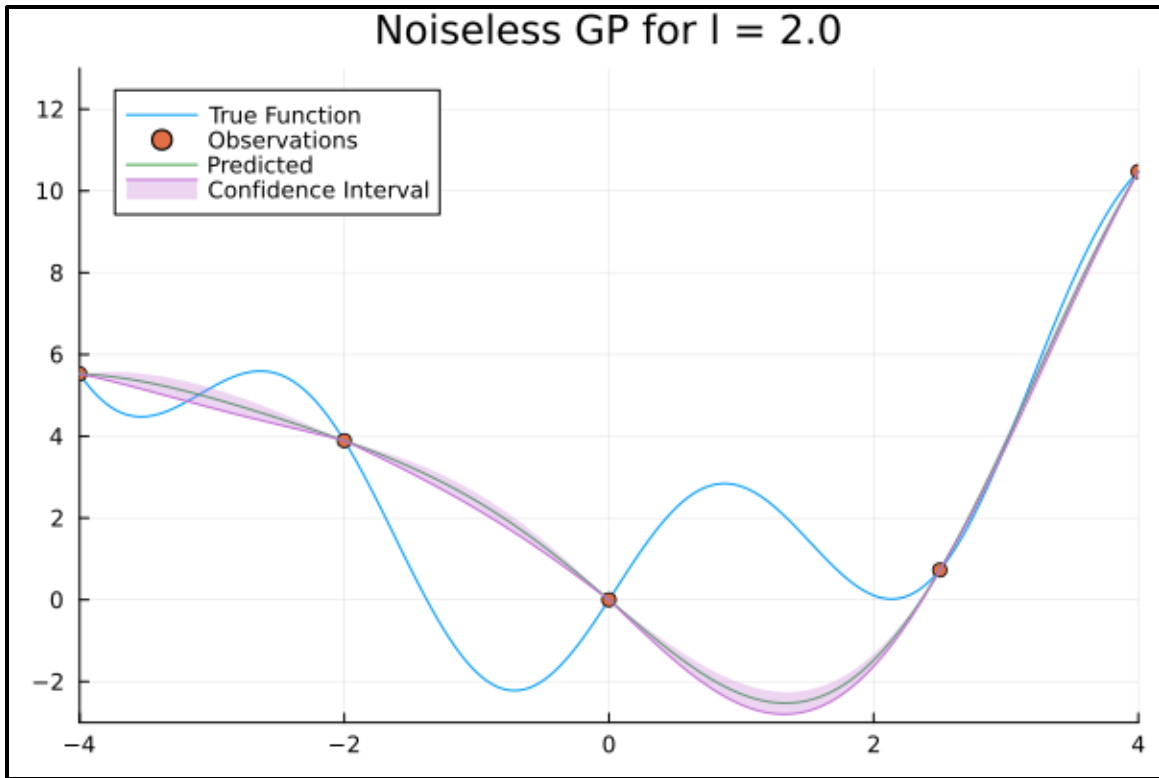


Figure 3: Noiseless Gaussian Process Fitting for characteristic length = 2.0

Part 2: Noisy Gaussian Fitting:

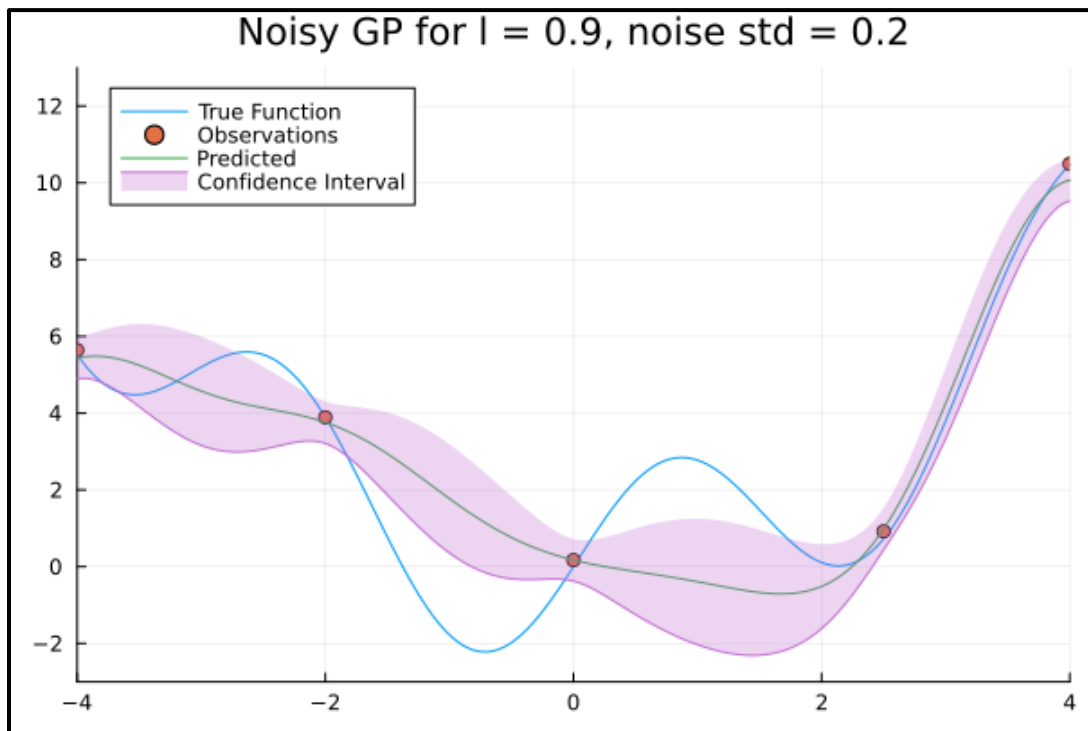


Figure 4: Noisy Gaussian Process Fitting for characteristic length = 0.9

Part 3: Exploration Strategies

Method	Design Point x	Predicted y
Prediction Based	1.67	-0.7080618717419538
Error Based	1.24	-0.5390502526770543
LCB Based	1.44	-0.6477921404833147
	-0.48	0.34463504852758514

Table 1: Function Predictions per different strategies

Comparison:

Various methods can guide an optimization process towards better design points, as demonstrated by the function predictions shown in Table 1. The Prediction-based exploration method selects the design point with the minimum value of the predicted surrogate function. This strategy aims to identify regions of the design space that are likely to have low values of the true function.

In contrast, the Error-based exploration method focuses on increasing confidence in the true function by sampling at a point where the standard deviation between the true function and the predicted function is maximized. By targeting areas with high uncertainty, this strategy aims to gain more information about the true behavior of the function.

The Lower Confidence Bound (LCB) based exploration method selects a design point that minimizes the Lower Confidence Bound of the objective function. This strategy strikes a balance between exploration and exploitation by seeking regions with both low predicted values and high uncertainty.

By employing these different exploration strategies, optimization algorithms can effectively navigate the design space and discover better solutions.

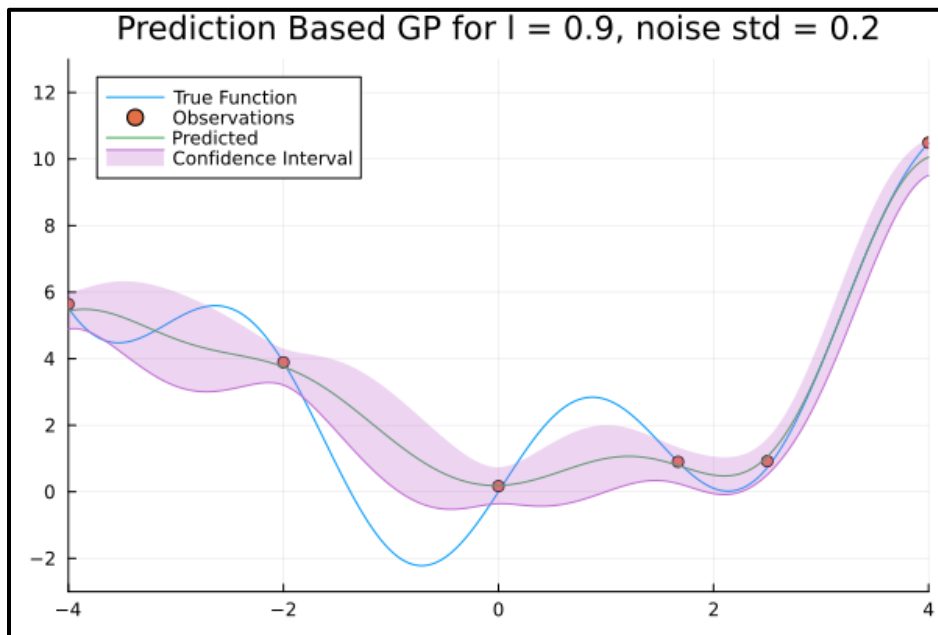


Figure 5: Updated Gaussian Plot from Prediction Based Exploration

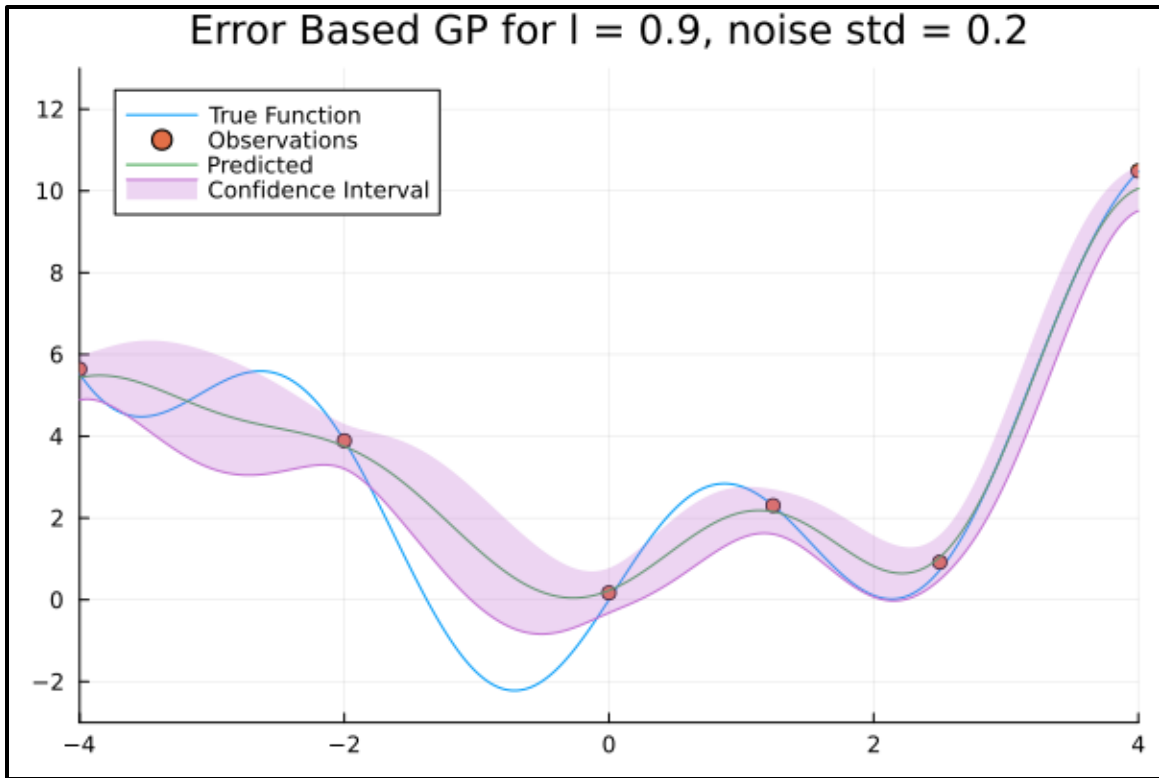


Figure 6: Updated Gaussian Plot from Error Based Exploration

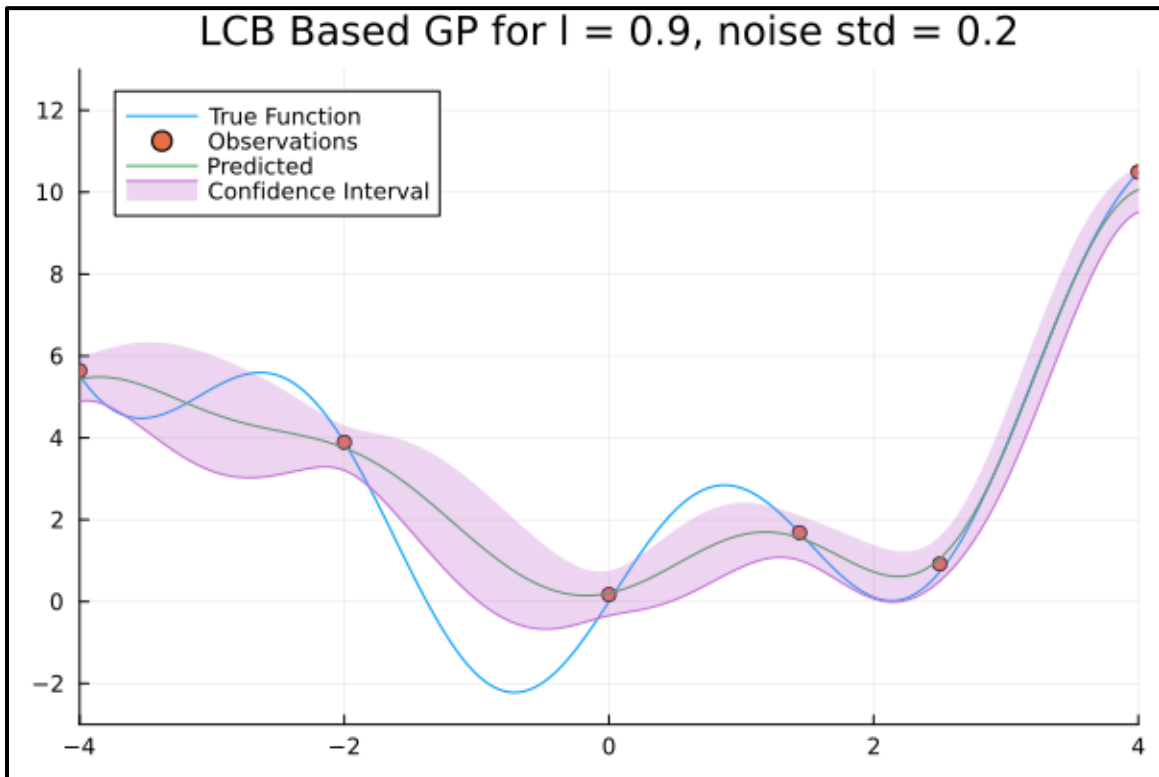


Figure 7: Updated Gaussian Plot from LCB Based Exploration

Code:

```
project3.jl
1 using GaussianProcesses, Plots, Random
2
3 # Defining the function:
4 f(x) = ((x^2)+(5*sin(2*x))) / 2
5 x_obs = [-4.0, -2.0, 0.0, 2.5, 4.0]
6 # Known Evals:
7 y_obs = f.(x_obs)
8 # True Evals:
9 x_all = collect(-4.0:0.01:4.0)
10 y_all = f.(x_all)
11 # characteristic lengths:
12 ls_values = [0.5, 1.0, 2.0]
13 # Gaussian Process:
14 for ls in ls_values
15     m = MeanZero()
16     kernel = SE(log(ls), 0.0)
17     gp = GP(x_obs, y_obs, m, kernel, -Inf)
18     μ, σ² = predict_y(gp, x_all)
19     std95 = 1.96 * sqrt.(σ²)
20
21     # Plotting:
22     plt = plot(x_all, y_all, label = "True Function")
23     scatter!(x_obs, y_obs, label = "Observations")
24     plot!(x_all, μ, label = "Predicted")
25     plot!(x_all, μ - std95, fillrange = μ + std95, fillalpha = 0.3, label = "Confidence Interval")
26     # Setting axes limits:
27     xlims!(-4, 4)
28     ylims!(-3, 13)
29     # Title:
30     title!("Noiseless GP for l = $ls")
31
32     # Saving the Plot:
33     savefig("GaussianProcesses_l_$ls.png")
34 end
35
36 # Noise:
37 f(x) = ((x^2)+(5*sin(2*x))) / 2
38 # known Evals
39 x_obs = [-4.0, -2.0, 0.0, 2.5, 4.0]
40 y_obs = [5.64, 3.89, 0.17, 0.92, 10.49]
41 # True Evals:
42 x_all = collect(-4.0:0.01:4.0)
43 y_all = f.(x_all)
44 ls = 0.9
45 noise_std = 0.2 # Standard deviation of the additive noise
46 # Gaussian Process:
47 m = MeanZero()
48 kernel = SE(log(ls), 0.0)
49 gp = GP(x_obs, y_obs, m, kernel, log(noise_std))
50 μ, σ² = predict_y(gp, x_all)
51 std95 = 1.96 * sqrt.(σ²)
52 # Plotting:
53 plt = plot(x_all, y_all, label = "True Function")
54 scatter!(x_obs, y_obs, label = "Observations")
55 plot!(x_all, μ, label = "Predicted")
56 plot!(x_all, μ - std95, fillrange = μ + std95, fillalpha = 0.3, label = "Confidence Interval")
57 # Setting axes limits:
58 xlims!(-4, 4)
59 ylims!(-3, 13)
60 # Title:
61 title!("Noisy GP for l = $ls, noise std = $noise_std")
62 # Saving the Plot:
63 savefig("NoisyGaussianProcesses_l_$ls.png")
64
65 # Exploration Strategies:
66
67 # Prediction Based Exploration:
68 x_pred_index = argmin(μ)
69 x_pred = x_all[x_pred_index]
70 f_pred = μ[x_pred_index]
71 f_model_pred = f.(x_pred)
72 println("Next point for evaluation, Prediction Based: x = $x_pred, f(x) = $f_pred")
73 # Updating Samples:
74 x_obs_pred = copy(x_obs)
```

```

75 push!(x_obs_pred, x_pred)
76 y_obs_pred = copy(y_obs)
77 push!(y_obs_pred, f_model_pred)
78 # Refitting the Gaussian Model:
79 m = MeanZero()
80 kernel = SE(log(l), 0.0)
81 gp = GP(x_obs_pred, y_obs_pred, m, kernel, log(noise_std))
82  $\mu_{pred}$ ,  $\sigma^2_{pred}$  = predict_y(gp, x_all)
83 std95 = 1.96 * sqrt( $\sigma^2_{pred}$ )
84 # Plotting:
85 plt = plot(x_all, y_all, label = "True Function")
86 scatter!(x_obs_pred, y_obs_pred, label = "Observations")
87 plot!(x_all,  $\mu_{pred}$ , label = "Predicted")
88 plot!(x_all,  $\mu_{pred}$  - std95, fillrange =  $\mu_{pred}$  + std95, fillalpha = 0.3, label = "Confidence Interval")
89 # Setting axes limits:std95
90 xlims!(-4, 4)
91 ylims!(-3, 13)
92 # Title:
93 title!("Prediction Based GP for l = $ls, noise std = $noise_std")
94 # Saving the Plot:
95 savefig("PredictionBased_l_$ls.png")
96
97 # Error Based Exploration:
98 x_err_index = argmax(sqrt( $\sigma^2$ ))
99 x_err = x_all[x_err_index]
100 f_err =  $\mu$ [x_err_index]
101 f_model_err = f(x_err)
102 println("Next point for evaluation, Error Based: x = $x_err, f(x) = $f_err")
103 # Updating Samples:
104 x_obs_err = copy(x_obs)
105 push!(x_obs_err, x_err)
106 y_obs_err = copy(y_obs)
107 push!(y_obs_err, f_model_err)
108 # Refitting the Gaussian Model:
109 m = MeanZero()
110 kernel = SE(log(l), 0.0)
111 gp = GP(x_obs_err, y_obs_err, m, kernel, log(noise_std))
112  $\mu_{err}$ ,  $\sigma^2_{err}$  = predict_y(gp, x_all)
113 std95 = 1.96 * sqrt( $\sigma^2_{err}$ )
114 # Plotting:
115 plt = plot(x_all, y_all, label = "True Function")
116 scatter!(x_obs_err, y_obs_err, label = "Observations")
117 plot!(x_all,  $\mu_{err}$ , label = "Predicted")
118 plot!(x_all,  $\mu_{err}$  - std95, fillrange =  $\mu_{err}$  + std95, fillalpha = 0.3, label = "Confidence Interval")
119 # Setting axes limits:
120 xlims!(-4, 4)
121 ylims!(-3, 13)
122 # Title:
123 title!("Error Based GP for l = $ls, noise std = $noise_std")
124 # Saving the Plot:
125 savefig("ErrorBased_l_$ls.png")
126
127 # Low confidence Bound:
128  $\alpha$  = 1.96
129 LB_index = argmin( $\mu - \alpha * \sqrt{\sigma^2}$ )
130 x_LB = x_all[LB_index]
131 f_LB =  $\mu$ [LB_index]
132 f_model_LB = f(x_LB)
133 println("Sixth point for evaluation, Lower Bound Based: x = $x_LB, f(x) = $f_LB")
134 # Updating Samples:
135 x_obs_LB = copy(x_obs)
136 push!(x_obs_LB, x_LB)
137 y_obs_LB = copy(y_obs)
138 push!(y_obs_LB, f_model_LB)
139 # Refitting the Gaussian Model:
140 m = MeanZero()
141 kernel = SE(log(l), 0.0)
142 gp = GP(x_obs_LB, y_obs_LB, m, kernel, log(noise_std))
143  $\mu_{LB}$ ,  $\sigma^2_{LB}$  = predict_y(gp, x_all)
144 std95 = 1.96 * sqrt( $\sigma^2_{LB}$ )
145 # Plotting:
146 plt = plot(x_all, y_all, label = "True Function")
147 scatter!(x_obs_LB, y_obs_LB, label = "Observations")
148 plot!(x_all,  $\mu_{LB}$ , label = "Predicted")

```

```

149 plot!(x_all, μ_LB - std95, fillrange = μ_LB + std95, fillalpha = 0.3, label = "Confidence Interval")
150 # Setting axes limits:
151 xlims!(-4, 4)
152 ylims!(-3, 13)
153 # Title:
154 title!("LCB Based GP for l = $ls, noise std = $noise_std")
155 # Saving the Plot:
156 savefig("LCBBased_l_$ls.png")
157 # Printing the 7th Point:
158 LB_index = argmin(μ_LB - d * sqrt.(d²_LB))
159 x_LB = x_all[LB_index]
160 f_LB = μ_LB[LB_index]
161 println("Seventh point for evaluation, Lower Bound Based: x = $x_LB, f(x) = $f_LB")

```