Computer Lab 6

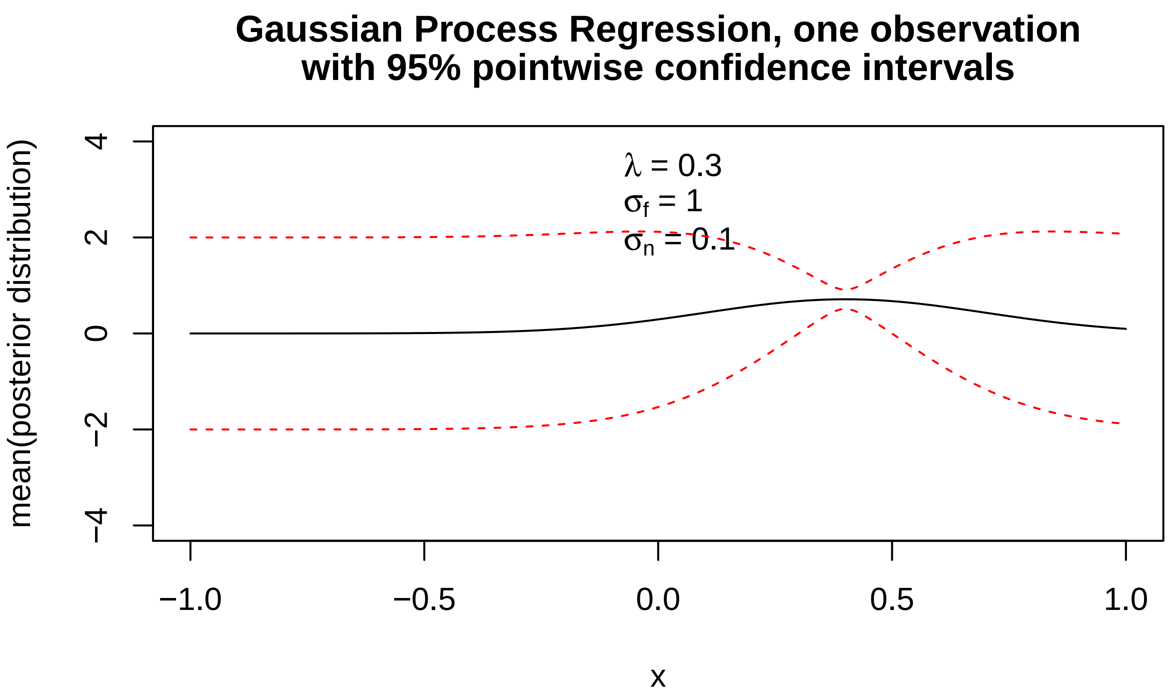
Thomas Zhang

2015-11-30

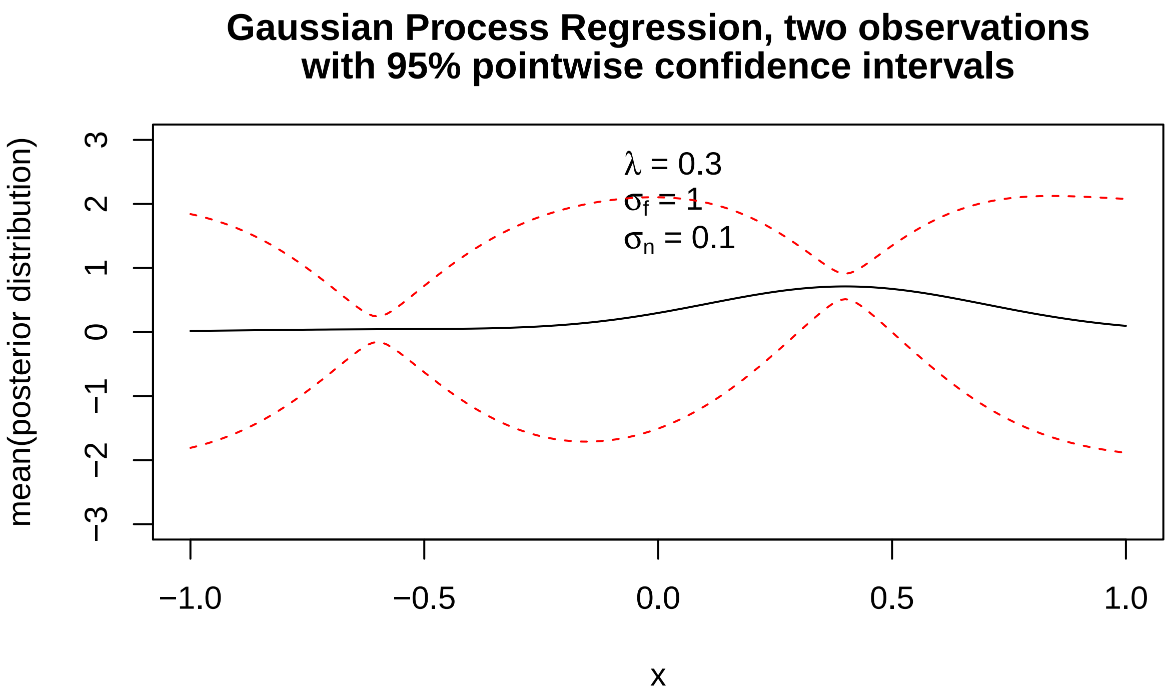
Assignment 1

We wish to perform Gaussian Process Regression starting with a constant zero function with normally distributed noise with variance σ_n^2.

We create a function for simulating from the posterior distribution of $f(x)$ using the squared exponential kernel with variance parameter σ_f^2 and bandwidth parameter λ. We update the zero mean prior with one observation, $(0.4, 0.719)$, and plot the posterior mean over the interval $(-1, 1)$ together with 95% pointwise probability bands for f.

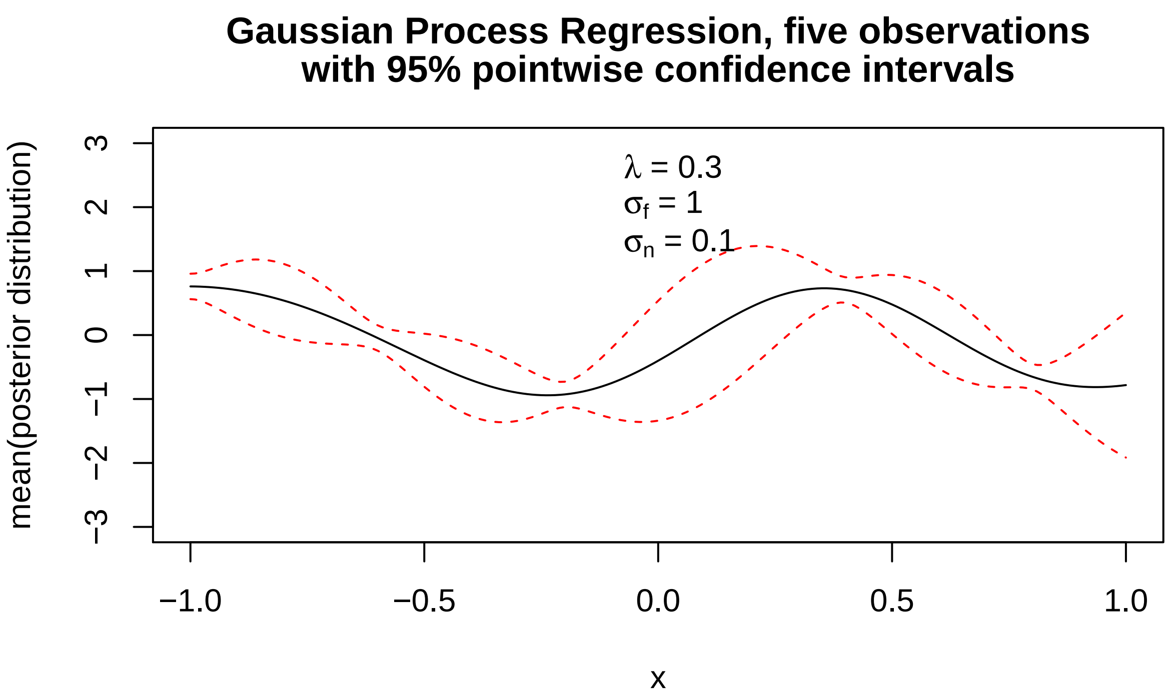


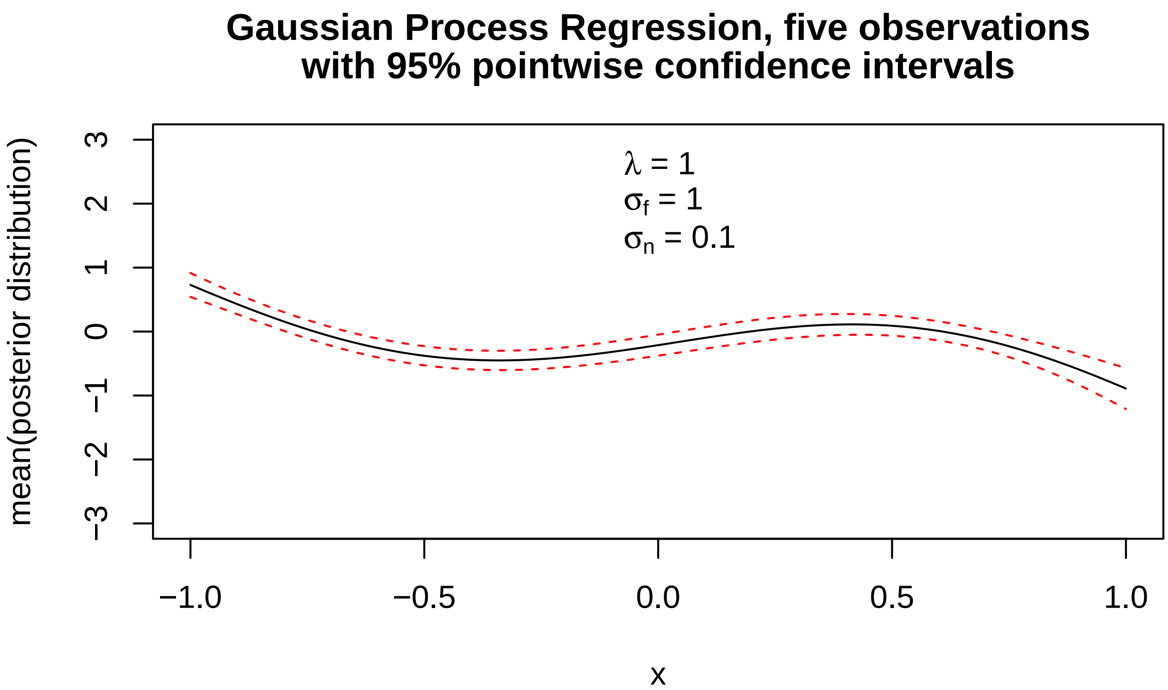
We now add one observation, (-0.6,-0.044) and run the Gaussian Process Regression again and plot the correspoinding out put again.



We see that the probability bands become narrow around the observations. This is characteristic of Gaussian process regression.

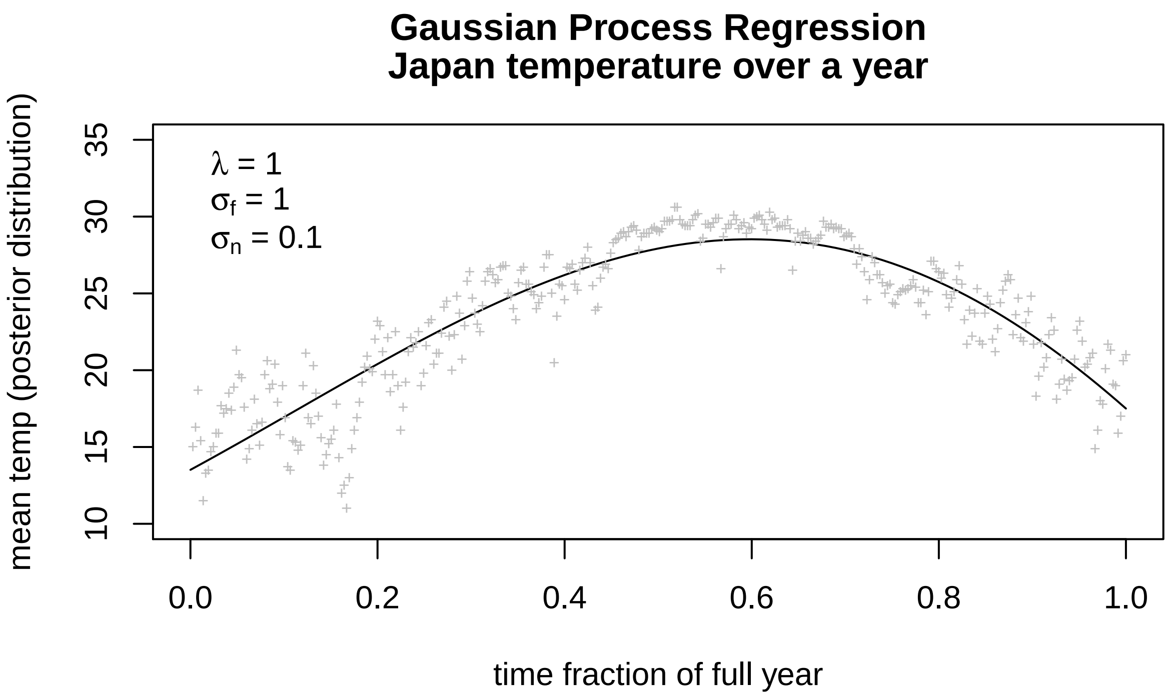
We now compute the posterior distribution for five points, given in the lab instructions. Let us do it one time for $\lambda = 0.3$ and one time for $\lambda = 1$.

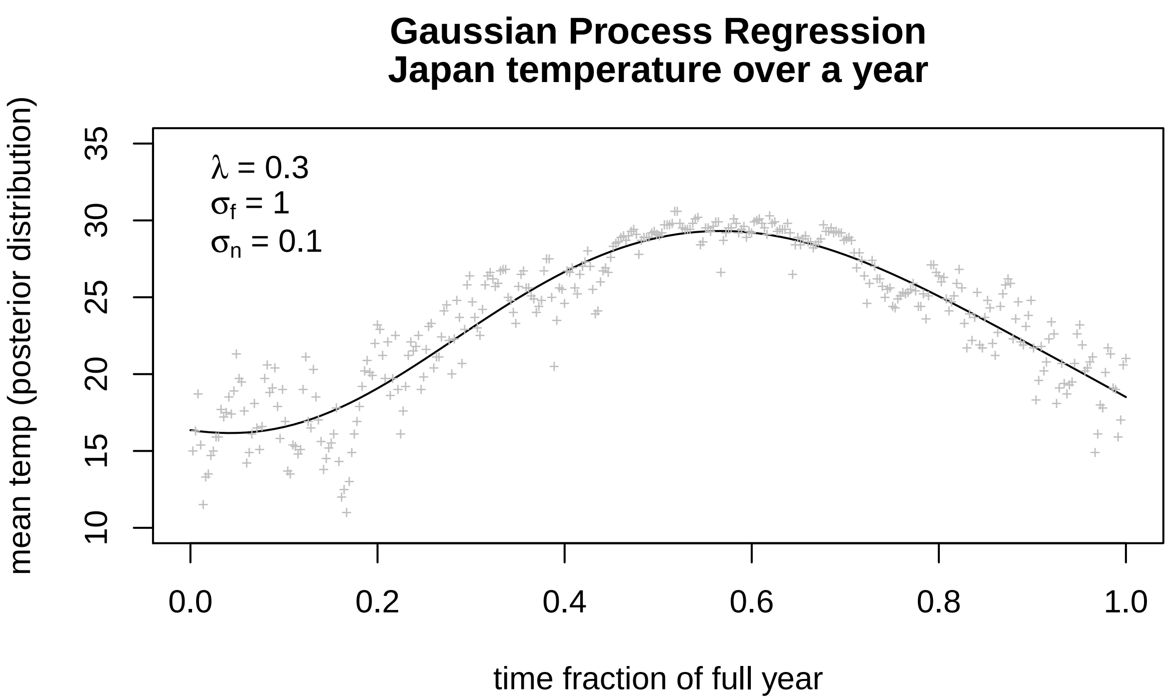


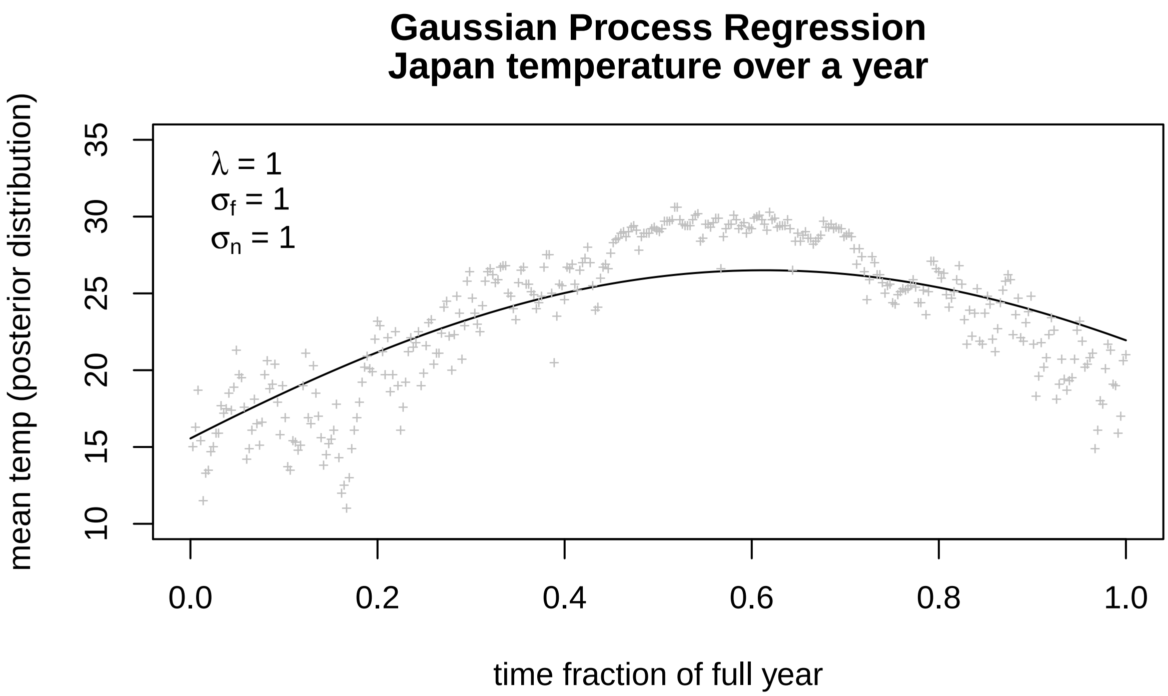


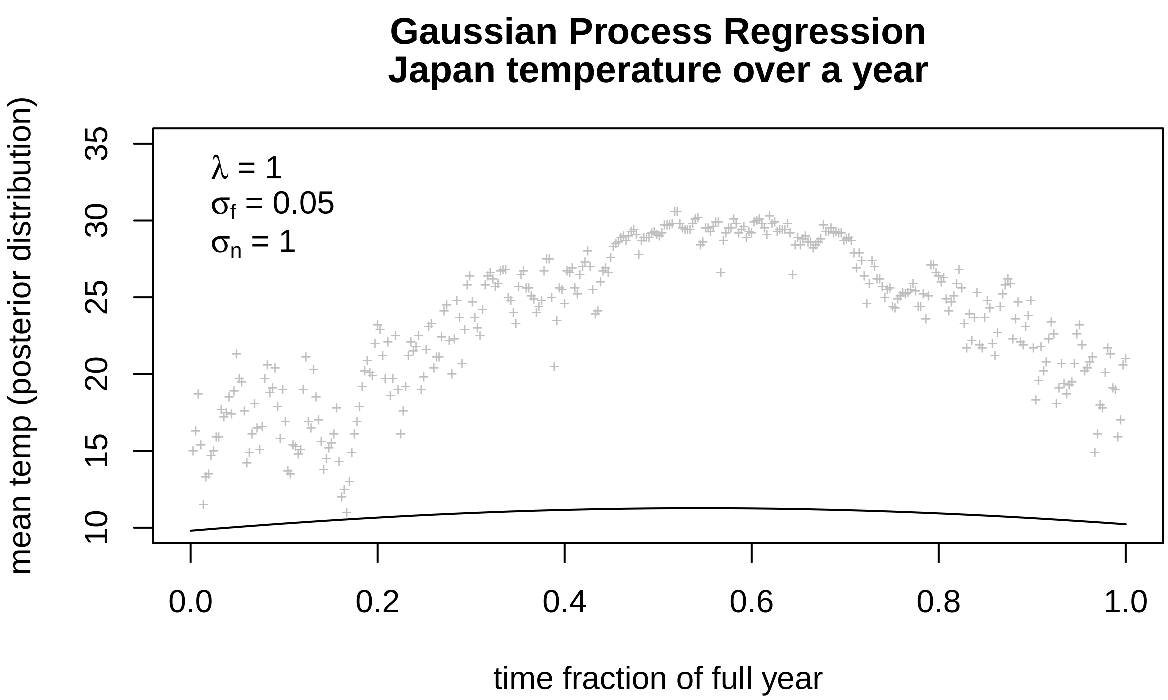
It is observed that a longer kernel bandwidth gives a smoother function and more narrow 95% pointwise probability bands for posterior mean of f.

Let us now try this Gaussian Process Regression on a data set containing daily temperatures somewhere in Japan over a year. The probability bands will not be plotted since they will almost coincide with the posterior mean due to the number of data points.


Gaussian Process Regression
Japan temperature over a year
λ = 1
σf = 1
σn = 0.1
mean temp (posterior distribution)
10
15
20
25
30
35
0.0
0.2
0.4
0.6
0.8
1.0
time fraction of full year


Gaussian Process Regression
Japan temperature over a year
λ = 0.3
σf = 1
σn = 0.1
mean temp (posterior distribution)
10
15
20
25
30
35
0.0
0.2
0.4
0.6
0.8
1.0
time fraction of full year





When σ_n^2 is increased, a less precise fit is obtained, since it is then assumed that the data is then more capable of being noisy. When λ is decreased, the fit becomes more overfitted to the data. This seems to be beneficial in this case. When σ_f^2 is decreased The function posterior mean becomes flatter, and more incapable of modeling the data.

Appendix

R code

```
library(MASS)

www = "D:/R_HW/ML-Lab-2/data/JapanTemp.dat"
data <- read.delim(www, header = TRUE, sep="")

Kernel <- function(x_0,x,lambda,sigmaf){
  res <- matrix(0,nrow=length(x_0),ncol=length(x))
  for(i in 1:length(x_0)){
    for(j in 1:length(x)){
      res[i,j] <- sigmaf^2 * exp(-1/2 * (abs(x_0[i] - x[j]) / lambda )^2)
    }
  }
  return(res)
}

posteriorGP <- function(x,y,xStar,hyperParam,sigmaNoise){

  meanfstar <- Kernel(xStar,x,hyperParam[2],hyperParam[1]) %*%
    solve(Kernel(x,x,hyperParam[2],hyperParam[1]) +
    sigmaNoise^2 * diag(length(x))) %*% y
  covfstar <- Kernel(xStar,xStar,hyperParam[2],hyperParam[1]) -
    Kernel(xStar,x,hyperParam[2],hyperParam[1]) %*%
    solve(Kernel(x,x,hyperParam[2],hyperParam[1]) +
    sigmaNoise^2 * diag(length(x))) %*%
    Kernel(x,xStar,hyperParam[2],hyperParam[1])
  posterior <- mvrnorm(1,meanfstar,covfstar)

  return(list(posteriorMean = meanfstar, posteriorCov= covfstar))
}

btask <- posteriorGP(0.4,0.719,seq(-1,1,0.01),c(1, 0.3),0.1)
uband <- btask$posteriorMean + 2 * sqrt(diag(btask$posteriorCov))
lband <- btask$posteriorMean - 2 * sqrt(diag(btask$posteriorCov))

plot(seq(-1,1,0.01),ylim=c(-4,4),btask$posteriorMean,type="l",
     main=c("Gaussian Process Regression, one observation",
            "with 95% pointwise confidence intervals"),ylab="mean(posterior distribution)",
     xlab="x")
legend("top",c(expression(paste(lambda," = 0.3")),
               expression(paste(sigma[f]," = 1")),
               expression(paste(sigma[n]," = 0.1"))),bty="n")

lines(seq(-1,1,0.01),uband,col="red",lty=2)
lines(seq(-1,1,0.01),lband,col="red",lty=2)
ctask <- posteriorGP(c(0.4,-0.6),c(0.719,0.044),seq(-1,1,0.01),c(1, 0.3),0.1)
uband2 <- ctask$posteriorMean + 2 * sqrt(diag(ctask$posteriorCov))
lband2 <- ctask$posteriorMean - 2 * sqrt(diag(ctask$posteriorCov))
plot(seq(-1,1,0.01),ylim=c(-3,3),ctask$posteriorMean,type="l",
     main=c("Gaussian Process Regression, two observations",
```

```r
            "with 95% pointwise confidence intervals"),ylab="mean(posterior distribution)",
      xlab="x")
legend("top",c(expression(paste(lambda," = 0.3")),
               expression(paste(sigma[f]," = 1")),
               expression(paste(sigma[n]," = 0.1"))),bty="n")
lines(seq(-1,1,0.01),uband2,col="red",lty=2)
lines(seq(-1,1,0.01),lband2,col="red",lty=2)
yvec <- c(0.768,-0.044,-0.940,0.719,-0.664)
xvec <- c(-1,-.6,-.2,.4,.8)
dtask <- posteriorGP(xvec,yvec,seq(-1,1,0.01),c(1, 0.3),0.1)
uband3 <- dtask$posteriorMean + 2 * sqrt(diag(dtask$posteriorCov))
lband3 <- dtask$posteriorMean - 2 * sqrt(diag(dtask$posteriorCov))
plot(seq(-1,1,0.01),ylim=c(-3,3),dtask$posteriorMean,type="l",
     main=c("Gaussian Process Regression, five observations",
            "with 95% pointwise confidence intervals"),ylab="mean(posterior distribution)",
     xlab="x")
legend("top",c(expression(paste(lambda," = 0.3")),
               expression(paste(sigma[f]," = 1")),
               expression(paste(sigma[n]," = 0.1"))),bty="n")
lines(seq(-1,1,0.01),uband3,col="red",lty=2)
lines(seq(-1,1,0.01),lband3,col="red",lty=2)

etask <- posteriorGP(xvec,yvec,seq(-1,1,0.01),c(1, 1),0.1)
uband4 <- etask$posteriorMean + 2 * sqrt(diag(etask$posteriorCov))
lband4 <- etask$posteriorMean - 2 * sqrt(diag(etask$posteriorCov))
plot(seq(-1,1,0.01),ylim=c(-3,3),etask$posteriorMean,type="l",
     main=c("Gaussian Process Regression, five observations",
            "with 95% pointwise confidence intervals"),ylab="mean(posterior distribution)",
     xlab="x")
legend("top",c(expression(paste(lambda," = 1")),
               expression(paste(sigma[f]," = 1")),
               expression(paste(sigma[n]," = 0.1"))),bty="n")
lines(seq(-1,1,0.01),uband4,col="red",lty=2)
lines(seq(-1,1,0.01),lband4,col="red",lty=2)

jtask <- posteriorGP(data$time,data$temp,seq(0,1,0.01),c(1, 1),0.1)
ubandj <- jtask$posteriorMean + 2 * sqrt(diag(jtask$posteriorCov))
lbandj <- jtask$posteriorMean - 2 * sqrt(diag(jtask$posteriorCov))

plot(seq(0,1,0.01),jtask$posteriorMean,type="l", ylim = c(10,35),
     main=c("Gaussian Process Regression","Japan temperature over a year"
            ),ylab="mean temp (posterior distribution)",
     xlab="time fraction of full year")
points(data$time,data$temp,pch="+",col="gray",cex=0.6)
legend("topleft",c(expression(paste(lambda," = 1")),
               expression(paste(sigma[f]," = 1")),
               expression(paste(sigma[n]," = 0.1"))),bty="n")


jtask <- posteriorGP(data$time,data$temp,seq(0,1,0.01),c(1, 0.3),0.1)
ubandj <- jtask$posteriorMean + 2 * sqrt(diag(jtask$posteriorCov))
lbandj <- jtask$posteriorMean - 2 * sqrt(diag(jtask$posteriorCov))
```

```r
plot(seq(0,1,0.01),jtask$posteriorMean,type="l", ylim = c(10,35),
     main=c("Gaussian Process Regression","Japan temperature over a year"
     ),ylab="mean temp (posterior distribution)",
     xlab="time fraction of full year")
points(data$time,data$temp,pch="+",col="gray",cex=0.6)
legend("topleft",c(expression(paste(lambda," = 0.3")),
                   expression(paste(sigma[f]," = 1")),
                   expression(paste(sigma[n]," = 0.1"))),bty="n")

jtask <- posteriorGP(data$time,data$temp,seq(0,1,0.01),c(1, 1),1)
ubandj <- jtask$posteriorMean + 2 * sqrt(diag(jtask$posteriorCov))
lbandj <- jtask$posteriorMean - 2 * sqrt(diag(jtask$posteriorCov))

plot(seq(0,1,0.01),jtask$posteriorMean,type="l", ylim = c(10,35),
     main=c("Gaussian Process Regression","Japan temperature over a year"
     ),ylab="mean temp (posterior distribution)",
     xlab="time fraction of full year")
points(data$time,data$temp,pch="+",col="gray",cex=0.6)
legend("topleft",c(expression(paste(lambda," = 1")),
                   expression(paste(sigma[f]," = 1")),
                   expression(paste(sigma[n]," = 1"))),bty="n")

jtask <- posteriorGP(data$time,data$temp,seq(0,1,0.01),c(0.05, 1),1)

plot(seq(0,1,0.01),jtask$posteriorMean,type="l", ylim = c(10,35),
     main=c("Gaussian Process Regression","Japan temperature over a year"
     ),ylab="mean temp (posterior distribution)",
     xlab="time fraction of full year")
points(data$time,data$temp,pch="+",col="gray",cex=0.6)
legend("topleft",c(expression(paste(lambda," = 1")),
                   expression(paste(sigma[f]," = 0.05")),
                   expression(paste(sigma[n]," = 1"))),bty="n")

## NA
```