Section 4: Gaussian Processes

Natalia Zuniga-Garcia

3/26/2018

Exercise 4.1

What is the predictive distribution $p(f_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X})$? Note: this is very similar to questions we did in Section 1.

Reference: C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. c 2006 Massachusetts Institute of Technology. www.GaussianProcess.org/gpml

Solution

From the multivariate Gaussian - Equation 2.9 from Rasmussen & Williams (2006) - we know that,

$$p(f_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X}) \sim N(m, S)$$

Where,

$m = \frac{1}{\sigma_n^2}\mathbf{x}_*^T A^{-1} X\mathbf{y}, \mathbf{x}_*^T A^{-1}\mathbf{x}_*$

$S = \mathbf{x}_*^T A^{-1}\mathbf{x}_*$

$A = \sigma_n^{-2} X X^T + \Sigma_p^{-1}$

Since $f(\mathbf{x}) \equiv \phi(\mathbf{x})\beta$, we can approxiate $p(f_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X})$ for a random process - using Equation 2.11 from Rasmussen & Williams (2006) - as,

$$p(f_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X}) \sim N(m_*, S_*)$$

Where,

$m_* = \frac{1}{\sigma^2}\phi(\mathbf{x}_*)^T A^{-1}\phi(X)\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)$

$S_* = \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)$

$A = \sigma_n^{-2}\phi(X)\phi(X)^T + \Sigma_p^{-1}$

Exercise 4.2

Let $\phi(x) = (1, x, x^2, x^3)$. Using appropriate priors on β and σ^2, obtain a posterior distribution over $f := \phi(x)^T\beta$. Plot the function (with a 95% credible interval) by evaluating this on a grid of values.

Solution

Data Description

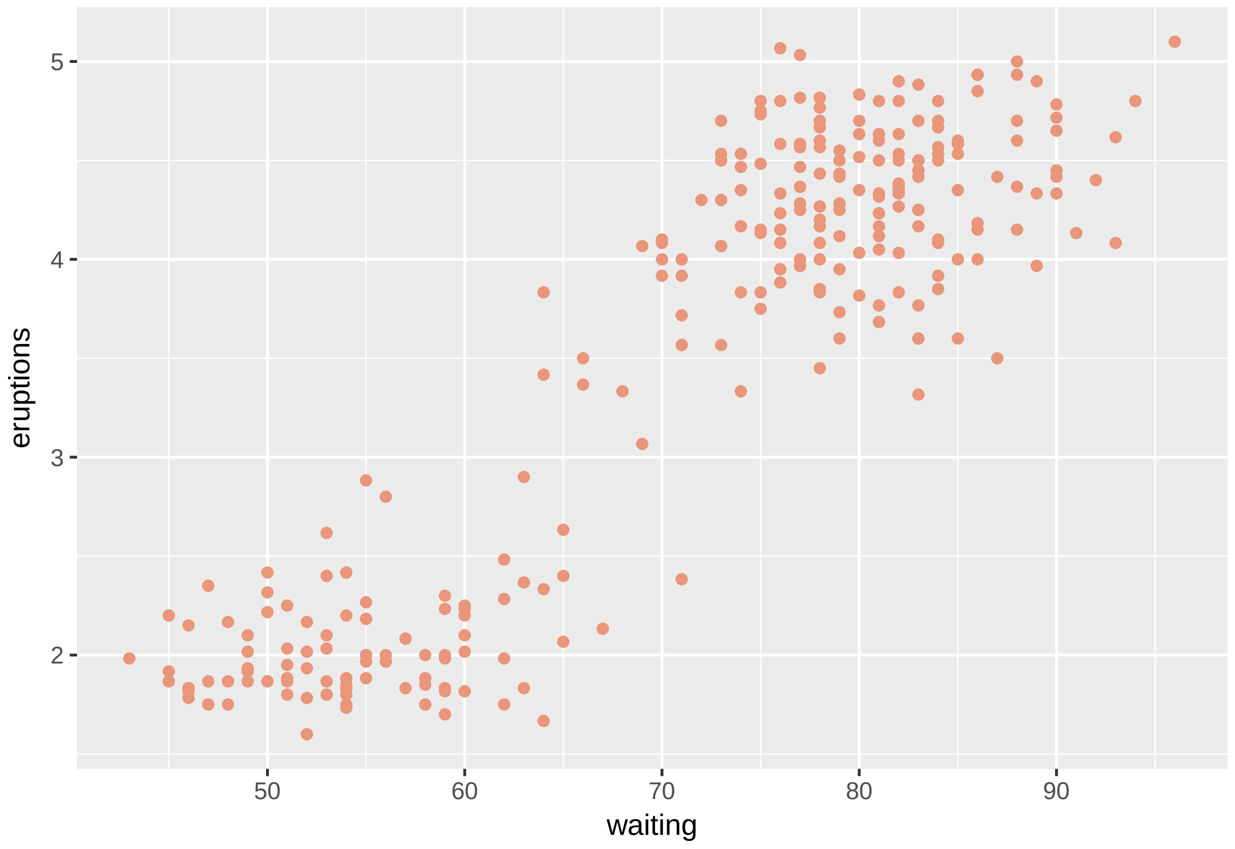
R description of the "faithful" data set: "Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA." The dataframe has two variables, eruptions and waiting, where eruptions is the duration of an eruption in minutes and waiting is the waiting time to the next eruption in minutes.

First, I look at the data and plot the point. I estimate the linear regression.

```r
data("faithful",package="datasets")
summary(faithful)
```

```
##    eruptions        waiting
##  Min.   :1.600   Min.   :43.0
##  1st Qu.:2.163   1st Qu.:58.0
##  Median :4.000   Median :76.0
##  Mean   :3.488   Mean   :70.9
##  3rd Qu.:4.454   3rd Qu.:82.0
##  Max.   :5.100   Max.   :96.0
```

```r
# Show Data Points
ggplot(data = faithful)  +
  geom_point(aes(waiting, eruptions), colour = "darksalmon")
```

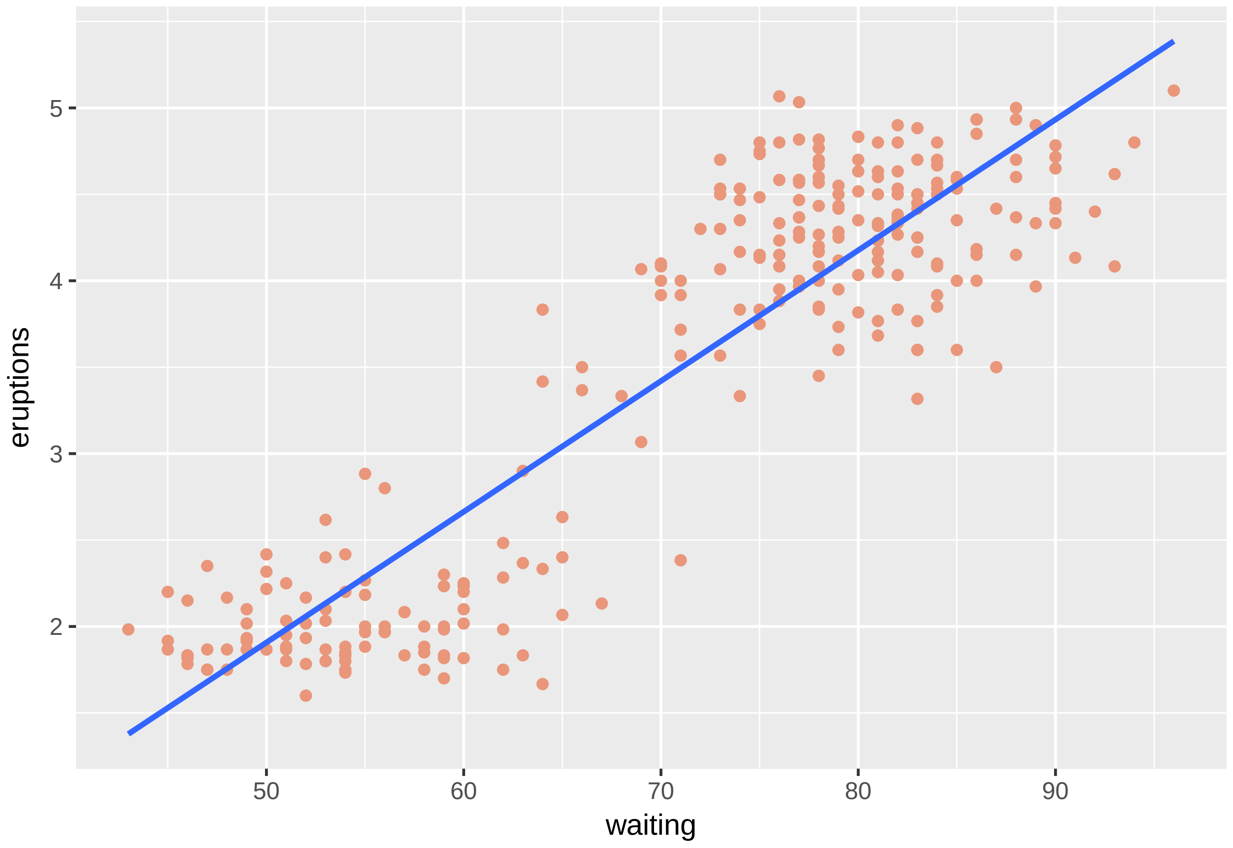


```r
# Linear Model
eruption.lm <- lm(eruptions ~ waiting, data=faithful)
summary(eruption.lm)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

```
ggplot()  +
  geom_point(data = faithful, aes(waiting, eruptions), color = "darksalmon") +
  stat_smooth(data = faithful, aes(waiting, eruptions), method = "lm", se = FALSE)
```



Use the Gaussian process

Based on the Solution of Exercise 4.1, first assign $\phi(x) = (1, x, x^2, x^3)$,

```
# Assign Values
y <- faithful$eruptions
x <- faithful$waiting

# Define the points at which we want to define the functions
x.star <- seq(min(x), max(x), len = 50)

# Assign Phi
X <- matrix(1,length(x),4)
X[,2] <- x
```

```r
X[,3] <- x^2
X[,4] <- x^3

# Assign Phi.star
X.star <- matrix(1,50,4)
X.star[,2] <- x.star
X.star[,3] <- x.star^2
X.star[,4] <- x.star^3
```

Now, estimate $A = \sigma_n^{-2}\phi(X)\phi(X)^T + \Sigma_p^{-1}$,

```r
Sigma = diag ( rep (1, 4))
sig2 = 1
A = (1 / sig2) * t(X) %*% X + solve(Sigma)
Ainv <- solve(A)
```

Finally, estimate m_* and S_*,

$m_* = \frac{1}{\sigma^2}\phi(\mathbf{x}_*)^T A^{-1}\phi(X)\mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)$

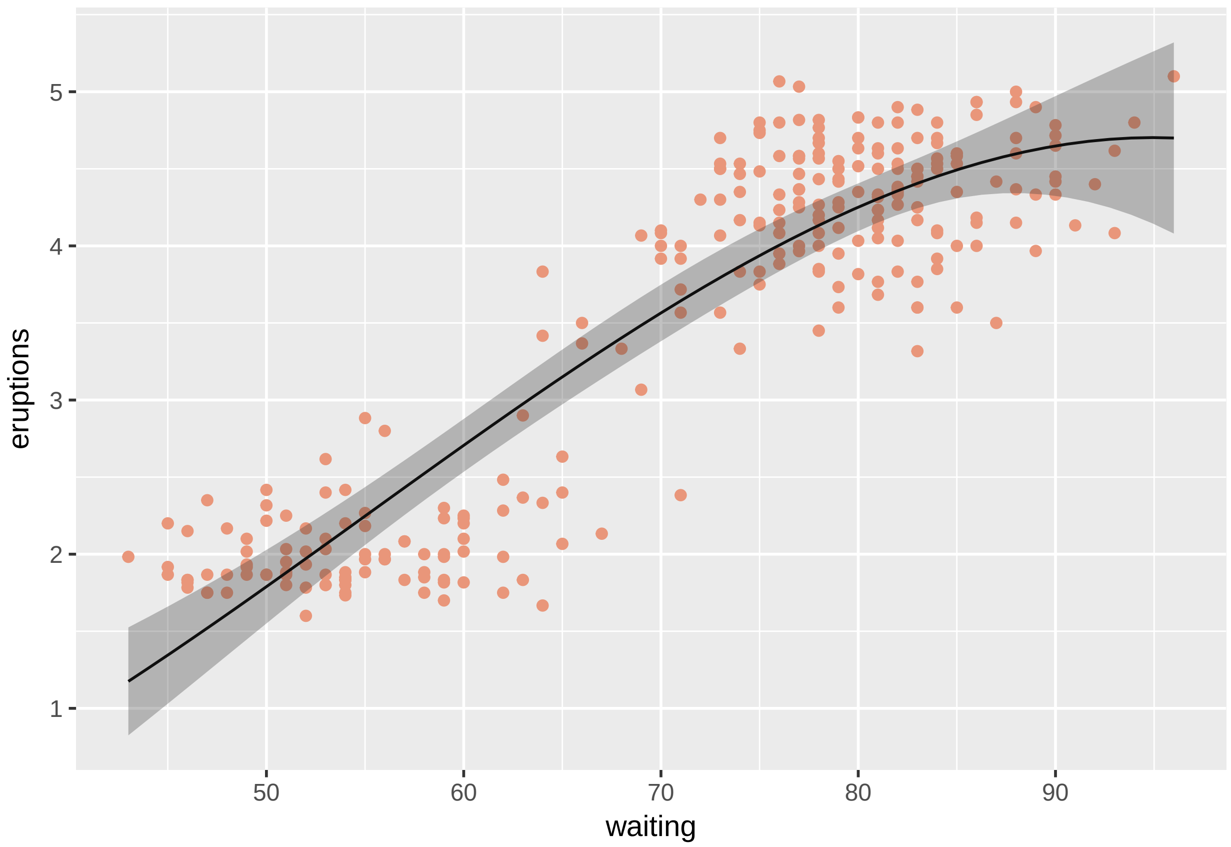
$S_* = \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)$

```r
m <- (1 / sig2) * Ainv %*% t(X) %*% y
fstar.mean = X.star %*% m
fstar.cov = X.star %*% Ainv %*% t(X.star)

# Credible Intervals
interval <- 1.96 * sqrt(diag(fstar.cov))
l1 <- fstar.mean - interval
l2 <- fstar.mean + interval
```

Plot the results,

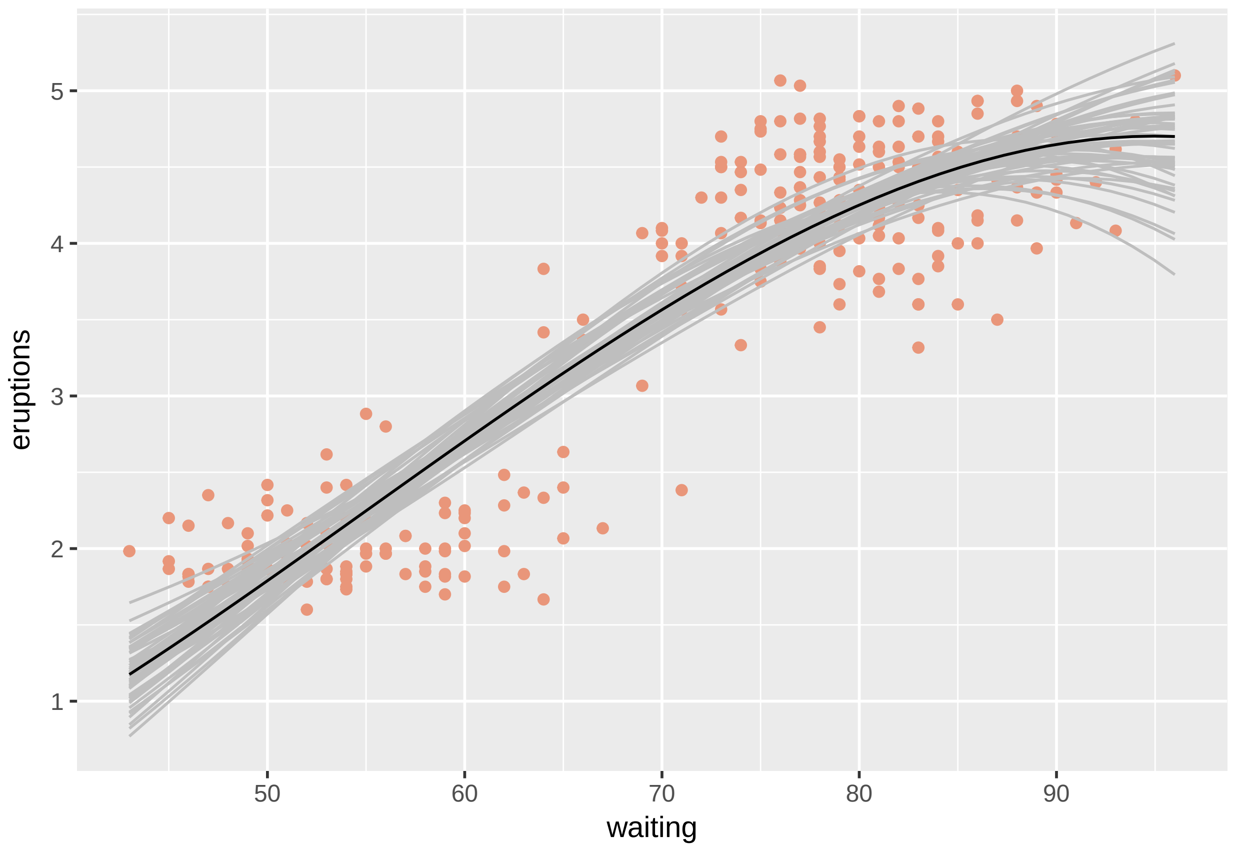
```r
ggplot()  +
  geom_point(data = faithful, aes(waiting, eruptions), color = "darksalmon") +
  geom_line(aes(x.star,fstar.mean)) +
  geom_ribbon(aes(x.star, ymin = l1, ymax = l2), alpha = 0.3)
```



Optionally, sample from the multivariate normal,

```
n.samples <- 50
values <- matrix(rep(0, length(x.star) * n.samples), ncol = n.samples)
for (i in 1:n.samples) {
  values[,i] <- mvrnorm(1, fstar.mean, fstar.cov)
}
values <- cbind(x = x.star, as.data.frame(values))
values <- melt(values, id ="x")

ggplot()  +
  geom_point(data = faithful, aes(waiting, eruptions), color = "darksalmon") +
  geom_line(data = values, aes(x = x, y = value, group = variable), colour="grey") +
  geom_line(aes(x.star,fstar.mean))
```



Exercise 4.3

Let's explore the resulting distribution over functions. Write some code to sample from a Gaussian process prior with squared exponential covariance function, evaluated on a grid of 200 inputs between 0 and 100. For $\ell = 1$, sample 5 functions and plot them on the same plot. Repeat for $\ell = 0.1$ and $\ell = 10$.

Solution

```r
rm(list = ls()) # Clear Environment

# I start writing a function to estimate the covariance
calcSigma <- function(X1, X2, l = 1, alpha = 1) {
  # Calculates the covariance matrix using the squared exponential covariance function
  Sigma <- matrix(rep(0, length(X1)*length(X2)), nrow=length(X1))

  for (i in 1:nrow(Sigma)) {
    for (j in 1:ncol(Sigma)) {
      Sigma[i,j] <- alpha^2 * exp(-0.5*(abs(X1[i]-X2[j])/l)^2)
    }
  }
  return(Sigma)
}
```

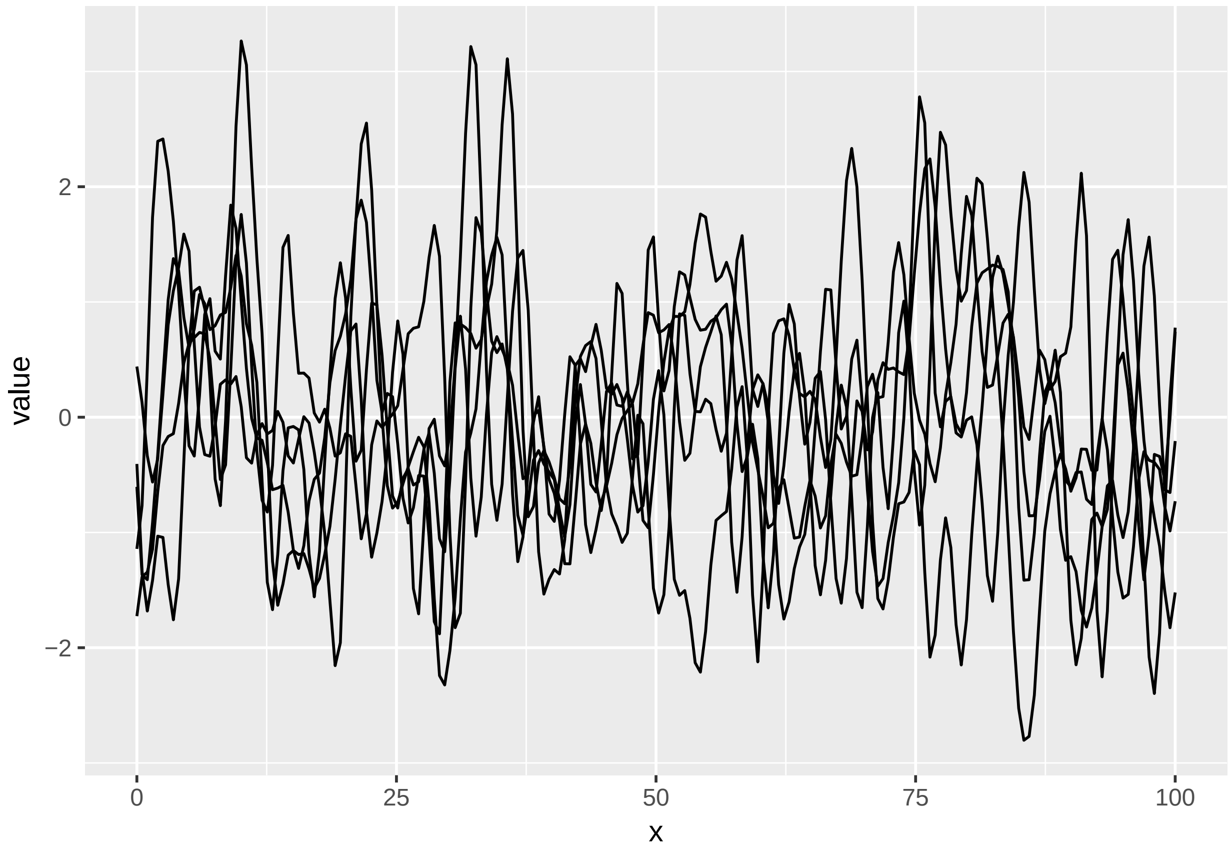
```r
# Generate a grid of 200 inputs
x.star <- seq(0, 100, len = 200)
```

Using $\ell = 1$,

```r
sigma <- calcSigma(x.star,x.star)

# Sample 5 functions
n.samples <- 5
values <- matrix(rep(0,length(x.star)*n.samples), ncol=n.samples)
for (i in 1:n.samples) {
  values[,i] <- mvrnorm(1, rep(0, length(x.star)), sigma)
}
values <- cbind(x = x.star, as.data.frame(values))
values <- melt(values, id = "x")

# Plot the 5 functions
ggplot(values, aes(x,value)) +
  geom_line(aes(group=variable))
```

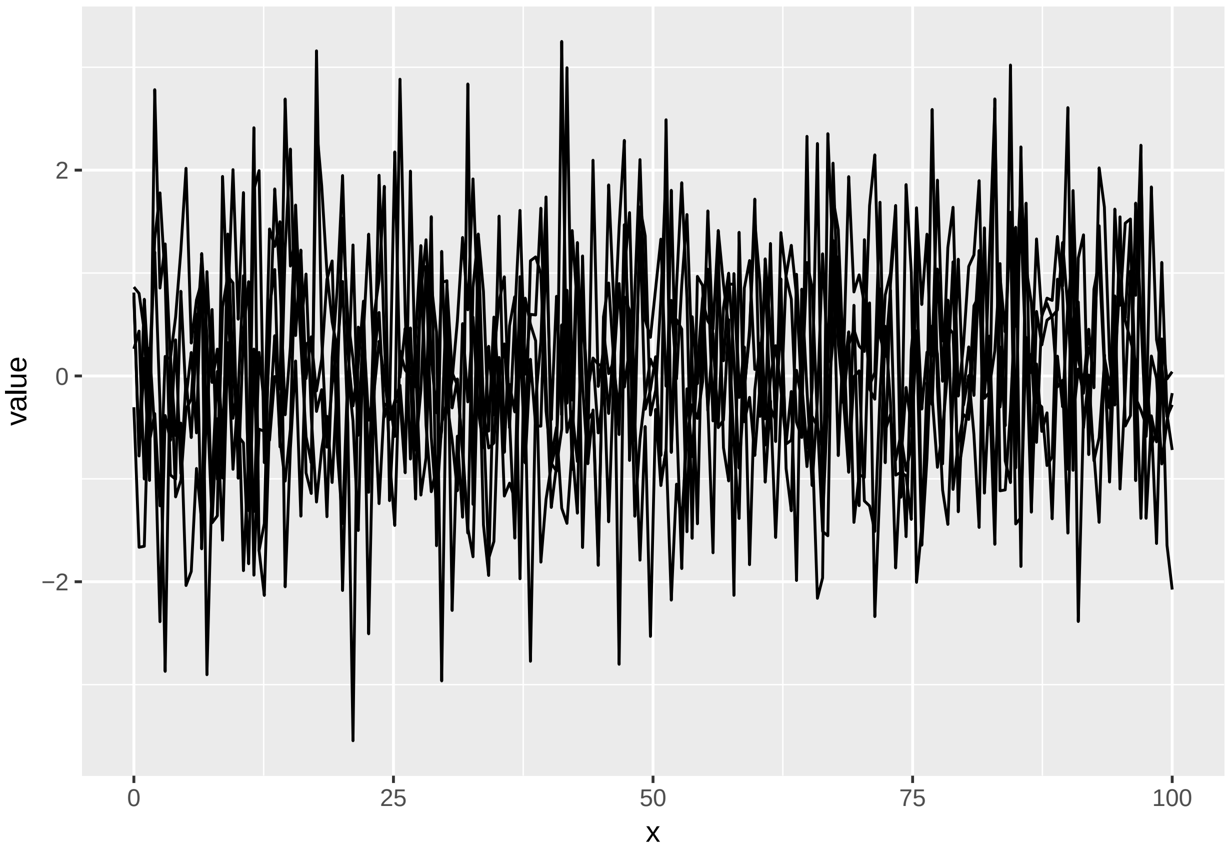


Using $\ell = 0.1$,

```r
sigma <- calcSigma(x.star,x.star, l = 0.1)

# Sample 5 functions
n.samples <- 5
values <- matrix(rep(0,length(x.star)*n.samples), ncol=n.samples)
for (i in 1:n.samples) {
  values[,i] <- mvrnorm(1, rep(0, length(x.star)), sigma)
}
values <- cbind(x = x.star, as.data.frame(values))
values <- melt(values, id = "x")

# Plot the 5 functions
ggplot(values, aes(x,value)) +
  geom_line(aes(group=variable))
```

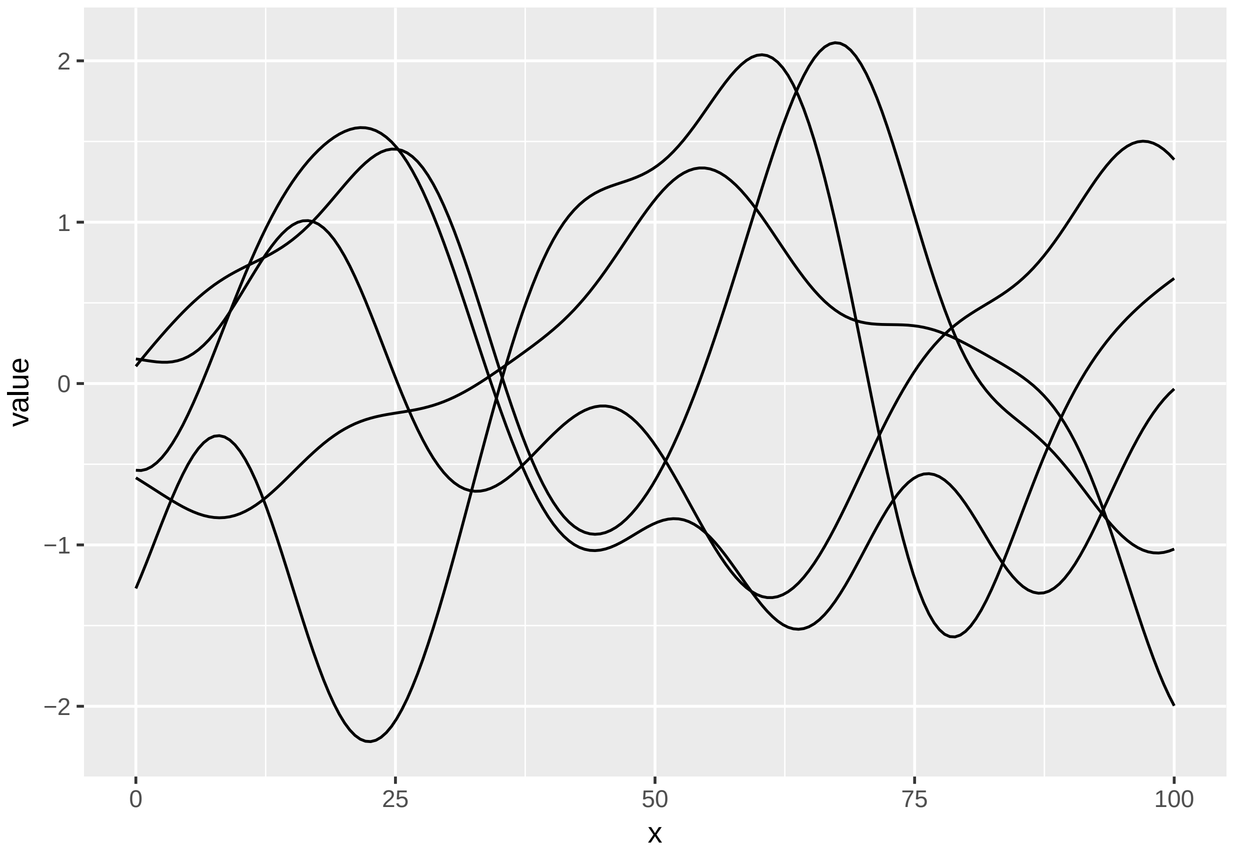


Using $\ell = 10$,

```
sigma <- calcSigma(x.star,x.star, l = 10)

# Sample 5 functions
n.samples <- 5
values <- matrix(rep(0,length(x.star)*n.samples), ncol=n.samples)
for (i in 1:n.samples) {
  values[,i] <- mvrnorm(1, rep(0, length(x.star)), sigma)
}
values <- cbind(x = x.star, as.data.frame(values))
values <- melt(values, id = "x")

# Plot the 5 functions
ggplot(values, aes(x,value)) +
  geom_line(aes(group=variable))
```



Why do we call ℓ the lengthscale of the kernel?

Because it helps to scale the kernel by moderating the covariance.

Exercise 4.4

Let $\mathbf{f}_* := f(\mathbf{X}_*)$ be the function f evaluated at test covariate locations $\mathbf{X}_*$. Derive the posterior distribution $p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y})$, where $\mathbf{y}$ and $\mathbf{X}$ comprise our training set. (You can start from the answer to Exercise 1 if you'd like).

Solution