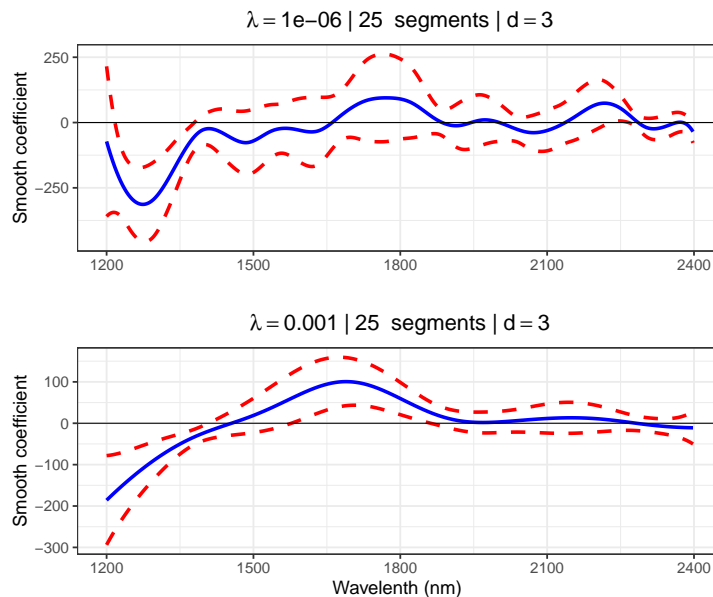# Penalized signal regression coefficient vectors (Biscuit data)



P-spline smooth coefficient vectors for biscuit data (including twice standard error bands) with tuning parameter $\lambda = 10^{-6}$ (top) and 0.001 (bottom) using 25 segments, and $d = 3$. R code in `f-psrcoef.R`

```
# Penalized signal regression coefficient vectors (Biscuit data)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

library(ggplot2)
library(gridExtra)
library(JOPS)

# Get the data
library(fds)
data(nirc)

# Spectra (X) and index of spectra (x)
iindex = nirc$x
X = nirc$y
sel = 50:650   #sel=1200 <= x & x<= 2400
X = X[sel, ]
iindex = iindex[sel]
dX = diff(X)
diindex = iindex[-1]
y = as.vector(labc[1, 1:40])  #fat
oout = 23  # Denoted as outlier by Brown et al. (2001)
dX = t(dX[, -oout])
y = y[-oout]

nseg = 25
pord = 3
lam1 = 1e-06
lam2 = 0.001
titl1 = bquote(lambda == .(lam1) ~ "|" ~ .(nseg) ~ " segments |" ~ d ==
    .(pord))
titl2 = bquote(lambda == .(lam2) ~ "|" ~ .(nseg) ~ " segments |" ~ d ==
    .(pord))

# Signal fits for plotting
```

```
fit1 = psSignal(y, dX, diindex, nseg = 25, lambda = lam1)
fit2 = psSignal(y, dX, diindex, nseg = 25, lambda = lam2)
lower1=fit1$beta-2*fit1$stdev_beta
upper1=fit1$beta+2*fit1$stdev_beta
lower2=fit2$beta-2*fit2$stdev_beta
upper2=fit2$beta+2*fit2$stdev_beta

# Dataframes for ggplot
F1=data.frame(beta=fit1$beta, lower=lower1, upper=upper1)
F2=data.frame(beta=fit2$beta, lower=lower2, upper=upper2)

# Make plots
plt1 = ggplot() +
  geom_line(aes(x = diindex, y = beta), data = F1, size = 1, colour = I("blue"), lty = 1) +
  geom_line(aes(x = diindex, y = upper), data = F1, size = 1, colour = I("red"), lty = 2) +
  geom_line(aes(x = diindex, y = lower), data = F1, size = 1, colour = I("red"), lty = 2) +
  geom_hline(yintercept = 0, size = 0.3) +
  xlab("") + ylab("Smooth coefficient") +
  ggtitle(titl1) +
  JOPS_theme()

plt2 = ggplot() +
  geom_line(aes(x = diindex, y = beta), data = F2, size = 1, colour = I("blue"), lty = 1) +
  geom_line(aes(x = diindex, y = upper), data = F2, size = 1, colour = I("red"), lty = 2) +
  geom_line(aes(x = diindex, y = lower), data = F2, size = 1, colour = I("red"), lty = 2) +
  geom_hline(yintercept = 0, size = 0.3) +
  xlab("Wavelenth (nm) ") + ylab("Smooth coefficient") +
  ggtitle(titl2) +
  JOPS_theme()

# Save plots
grid.arrange(plt1, plt2, ncol = 1, nrow = 2)
```