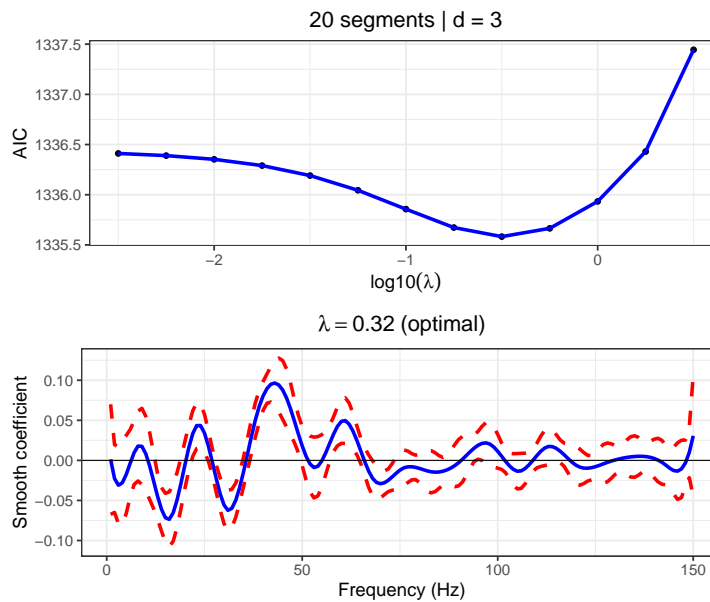# Binomial signal regression optimally tuned (Phoneme data)

Tracking AIC as a function of $\log_{10}(\lambda)$ for the phoneme data (top), and the optimal smooth coefficient vector (with twice standard error bands) corresponding to minimum AIC (bottom). R code in `f-vowelopt.R`

```
# Binomial signal regression optimally tuned (Phoneme 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(ElemStatLearn)
data(phoneme)
X = as.matrix(phoneme[, 1:150])
y = as.vector(phoneme[, 257])

# Select the 'aa' and 'ao' only
inn = which(y == "aa" | y == "ao")
X = X[inn, ]
y = y[inn]

# Change y to 0/1 ('aa' is success)
y[y == "aa"] = 1
y[y == "ao"] = 0
y = as.numeric(y)
iindex = 1:ncol(X)
pord = 3
nseg = 20

# Search for min AIC
llamb = seq(from = -2.5, to = 0.5, by = 0.25)
lambin = 10^llamb
r = length(llamb)
cv_ = rep(-999, r)
for (i in 1:r) {
    psrloop = psSignal(y, X, iindex, nseg = nseg, lambda = lambin[i],
```

```
                   pord = pord, family = "binomial")
     cv_[i] = psrloop$aic
}
plot(llamb, cv_, xlab = expression(log(lambda)), ylab = "AIC", type = "b")
lamopt = lambin[which(cv_ == min(cv_))]
fit2 = psSignal(y, X, iindex, nseg = nseg, lambda = lamopt,
                   pord = pord, family = "binomial")
lower2=fit2$beta-2*fit2$stdev_beta
upper2=fit2$beta+2*fit2$stdev_beta

# Data frames for ggplot
F1=data.frame(x=llamb,y=cv_)
F2=data.frame(beta=fit2$beta, lower=lower2, upper=upper2)

#titl1 = bquote(~nsegs==.(nseg)~"|"~pord==.(pord))
titl1 = "20 segments | d = 3"
titl2 = bquote(lambda==.(round(lamopt,2))~ "(optimal)")
plt1 = ggplot(F1, aes(x = x, y = y)) +
  geom_point(size = 1.2)  +
  geom_line(aes(x = x, y = y), data = F1, size = 1, colour = I("blue"), lty = 1) +
  labs(x=expression(paste("log10"(lambda))),  y="AIC")+
  ggtitle(titl1) +
  JOPS_theme()

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

# Plot and save pdf
grid.arrange(plt1, plt2, ncol = 1, nrow = 2)
```