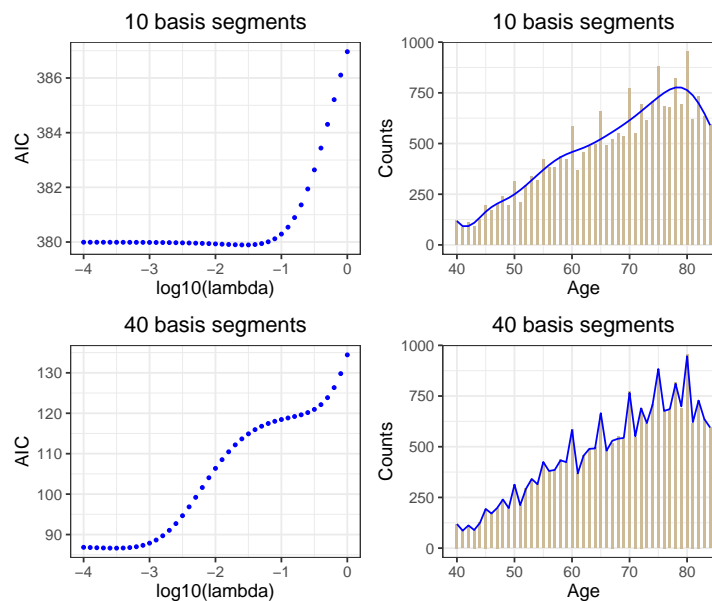


## Fit overdispersed counts with different size bases (Greece deaths data)



Counts of male deaths in Greece, in 1960, versus age. In the top row a cubic basis with 10 segments was used. The upper left panel shows the profile of AIC and the upper right panel the data and the minimum AIC fit. Bottom row: ditto, using 40 segments in the basis. R code in `f-Greece-AIC.R`

```
# Fit overdispersed counts with bases of different size (Greece deaths 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)

# Read the data (Greek number of deaths)
data(Greece_deaths)
Dat = subset(Greece_deaths, 40 <= Age & Age < 85) # Get rid of grouped data above 84
sel = 'Males'
x = Dat$Age
y = Dat[, sel]
m = length(x)

nsegs = c(10, 40)
lambdas = 10 ^ seq(-4, 0, by = 0.1)
M = A = NULL

for (nseg in nsegs) {
  # Create basis and penalty matrix
  B = bbase(x, nseg = nseg)
  n = ncol(B)
  D = diff(diag(n), diff = 2)
  P = t(D) %*% D

  # Compute P-spline Poisson fit
  aics = devs = eds = NULL
  Mus = NULL
  for (lambda in lambdas) {
    eta = log(y + 1)
    a = rep(0, n)
    mon = F
  }
}
```

```

for (it in 1:10) {
  mu = exp(eta)
  z = y - mu + mu * eta
  G = t(B) %*% diag(c(mu)) %*% B
  anew = solve(G + lambda * P, t(B) %*% z)
  da = max(abs(anew - a))
  if (mon) cat(it, lambda, da, '\n')
  a = anew
  eta = B %*% a
  if (da < 1e-6) break
}
# Save fit and AIC
Mus = cbind(Mus, mu)
H = solve(G + lambda * P, G)
hd = diag(H)
ed = sum(hd)
eds = c(eds, ed)
dev = 2 * sum(y * log(y / mu))
devs = c(devs, dev)
aic = dev + 2 * ed
aics = c(aics, aic)
}

# Select minimum of AIC
k = which.min(aics)
lambda = lambdas[k]
dev = devs[k]
ed = eds[k]
M = cbind(M, Mus[, k])
A = cbind(A, aics)
cat(nseg, devs[k], eds[k], aics[k], log10(lambdas[k]), '\n')
}

F1 = data.frame(lambdas = lambdas, aic1 = A[, 1], aic2 = A[, 2])
F2 = data.frame(x = x, y = y, mu1 = M[, 1], mu2 = M[, 2])

plt1 = ggplot(F1) +
  geom_point(aes(x = log10(lambdas), y = aic1), size = 0.7, color = 'blue') +
  ggtitle(paste(nsegs[1], 'basis segments')) +
  xlab('log10(lambda)') + ylab('AIC') +
  JOPS_theme()

plt2 = ggplot(F2) +
  geom_bar(aes(x, y), stat = 'identity', fill = 'wheat3', width = 0.5) +
  geom_line(aes(x, mu1), col = 'blue') +
  xlab('Age') + ylab('Counts') +
  ggtitle(paste(nsegs[1], 'basis segments')) +
  JOPS_theme()

plt3 = ggplot(F1) +
  geom_point(aes(x = log10(lambdas), y = aic2), size = 0.7, color = 'blue') +
  ggtitle(paste(nsegs[2], 'basis segments')) +
  xlab('log10(lambda)') + ylab('AIC') +
  JOPS_theme()

plt4 = ggplot(F2) +
  geom_bar(aes(x, y), stat = 'identity', fill = 'wheat3', width = 0.5) +
  geom_line(aes(x, mu2), col = 'blue') +
  xlab('Age') + ylab('Counts') +
  ggtitle(paste(nsegs[2], 'basis segments')) +
  JOPS_theme()

# Make and save graph
grid.arrange(plt1, plt2, plt3, plt4, nrow = 2, ncol = 2)

```

---