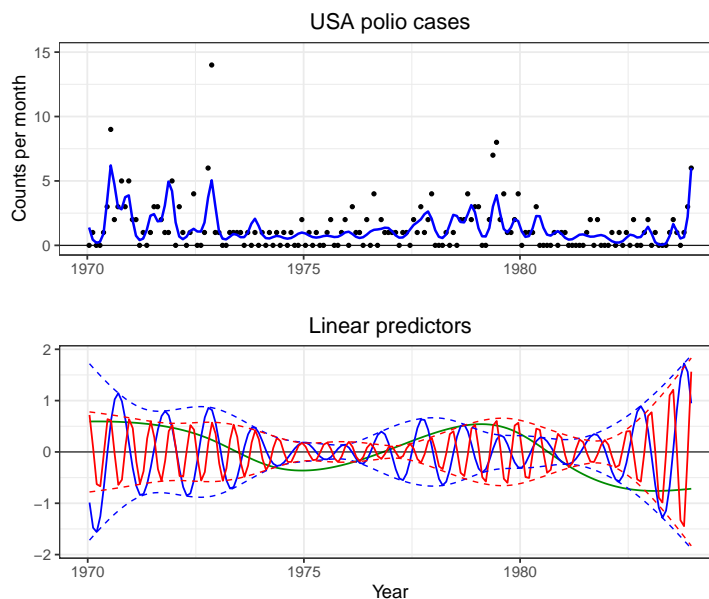


Varying-coefficient components with Poisson response (Polio data)



Polio data. Top: observed monthly counts and fitted values of the varying coefficient model. Bottom: the components of the linear predictor, a smooth trend (green) and modulated (co)sines with periods of 6 (red) and 12 (blue) months (bottom). The broken lines show the varying amplitudes. R code in `f-polio-vcM.R`

```
# Varying-coefficient components with Poisson response (Polio data)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

library(ggplot2)
library(gridExtra)
library(gamlss.data)
library(JOPS)

# Get the data
data(polio)
y = as.numeric(polio)
m = length(y)
x = 1970 + (1:m - 0.5)/12

# B-splines and penalty construction
B = bbase(x, nseg = 10)
n = ncol(B)
D = diff(diag(n), diff = 2)
P = t(D) %*% D

# Build VCM basis and block diag penalty matrix
om = 2 * pi * x
cs = cos(om)
sn = sin(om)
cs2 = cos(2 * om)
sn2 = sin(2 * om)
C = cbind(B, diag(cs) %*% B, diag(sn) %*% B, diag(cs2) %*% B, diag(sn2) %*% B)
K = diag(5 * n) * 1e-10

# Initialize
eta = log(y + 1)
a = 0
```

```

lambdas = rep(1, 5)

# Iterate for lambdas (Schall algorithm)
for (it2 in 1:20) {
  Q = kronecker(diag(lambdas), P) + K

  # Fit VCM
  a1 = a
  for (it in 1:10) {
    mu = exp(eta)
    z = y - mu + mu * eta
    W = diag(c(mu))
    S = t(C) %>% W %>% C
    anew = solve(S + Q, t(C) %>% z)
    da = max(abs(anew - a))
    if (da < 1e-4) break
    a = anew
    eta = C %>% a
  }

  # Compute effective dimensions and sums of squares of coefficients
  G = solve(S + Q, S)
  ssa = eds = rep(0, 5)
  g = diag(G)
  for (k in 1:5) {
    r = (k - 1) * n + (1:n)
    ssa[k] = sum(a[r] ^ 2)
    eds[k] = sum(g[r])
  }
  lam = lambdas
  lambdas = eds / ssa
  dla = max(abs(log10(lam) - log10(lambdas)))
  cat(it2, dla, log10(lambdas), '\n' )
}

# Compute
A = matrix(a, n, 5)
Fits = B %>% A

# Extract components for plotting
trend = Fits[, 1]
seas1 = Fits[, 2] * cs + Fits[, 3] * sn
seas2 = Fits[, 4] * cs2 + Fits[, 5] * sn2
amp1 = sqrt(Fits[, 2]^2 + Fits[, 3]^2)
amp2 = sqrt(Fits[, 4]^2 + Fits[, 5]^2)

Fdat = data.frame(x, y, mu, trend, seas1, seas2, amp1, amp2)

plt1 = ggplot(Fdat, aes(x = x, y = y)) +
  geom_point(size = 0.8) +
  geom_hline(yintercept = 0, size = 0.3) +
  geom_line(aes(x = x, y = mu), data = Fdat, size = 0.7, colour = "blue", lty = 1) +
  xlab("") + ylab("Counts per month") +
  ggtitle("USA polio cases") +
  ylim(c(0, 15)) +
  JOPS_theme()

ls1 = 0.5
ls2 = 0.4
plt2 = ggplot(Fdat, aes(x = x, y = trend)) +
  geom_line(col = "green4", size = ls1, lty = 1) +
  geom_hline(yintercept = 0, size = 0.3) +
  geom_line(aes(x = x, y = seas1), data = Fdat, size = ls1, colour = "blue", lty = 1) +
  geom_line(aes(x = x, y = seas2), data = Fdat, size = ls1, colour = "red", lty = 1) +
  geom_line(aes(x = x, y = amp1), data = Fdat, size = ls2, colour = "blue", lty = 2) +
  geom_line(aes(x = x, y = -amp1), data = Fdat, size = ls2, colour = "blue", lty = 2) +
  geom_line(aes(x = x, y = amp2), data = Fdat, size = ls2, colour = "red", lty = 2) +

```

```
geom_line(aes(x = x, y = -amp2), data = Fdat, size = ls2, colour = "red", lty = 2) +
xlab("Year") + ylab("") +
ggtitle("Linear predictors") +
JOPS_theme()

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