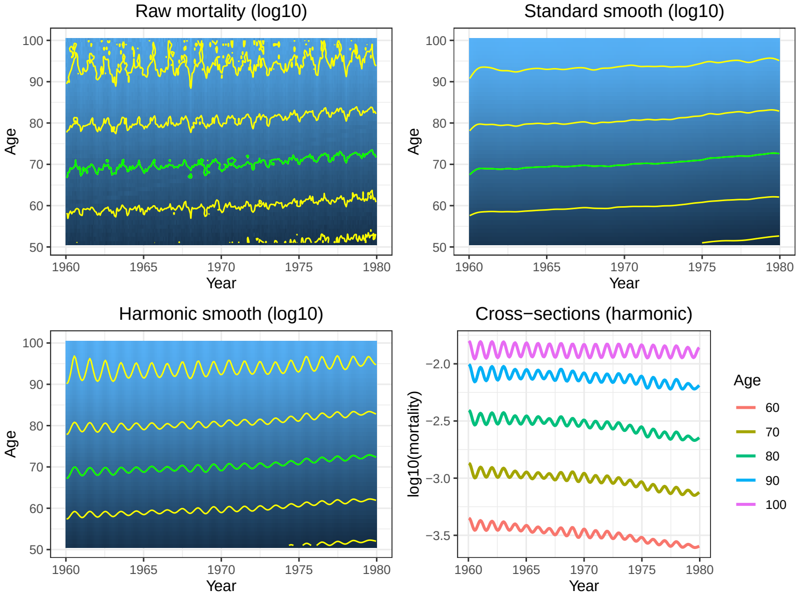
Smoothing of life table, monthly data (US Women cardiovascular data)



Life table smoothing with monthly data on cardiovascular deaths of women in the USA, using tensor products P-splines. The penalty on age is the standard one, using second order differences; the one on time is harmonic with period one year. The distance between the contour levels in the images is 0.5 and the green contour is at level -3. R code in `f-mortality.R`

```
# Smoothing of a life table with monthly data (US Women cardiovascular data)
# Paul Eilers and Brian Marx 2019
# A graph in "Practical Smoothing. The Joys of P-splines"

library(JOPS)
library(fields)
library(ggplot2)
library(reshape2)
library(gridExtra)

# Get the data
data(CardioData)
Expall = CardioData$Exposed[, 313:780]
Dthall = CardioData$Deaths
ages <- 51:100
ix = 1:240
Y = Dthall[ages + 1, 12 + ix]
Exp = Expall[ages, ix ]
ma = nrow(Y)
mx = ncol(Y)
x = 1960 + ((1:mx) - 0.5) / 12

# Compute the basis matrices
nsegx = 40
nsega = 10
da = dx = 2
lambdax = 1000
lambdaa = 10
kappa = 1e-4
tol = 1e-5


Bx = bbase(x, nseg = nsegx)
```

```
Ba = bbase(ages, nseg = nsega)
nx = ncol(Bx)
na = ncol(Ba)
Tx = rowtens(Bx)
Ta = rowtens(Ba)

# Compute standard penalty matrices
Dx = diff(diag(nx), diff = dx)
Da = diff(diag(na), diff = da)
Px = lambdax * t(Dx) %*% Dx
Pa = lambdaa * t(Da) %*% Da
P = kronecker(Px, diag(na)) + kronecker(diag(nx), Pa)
P = P + kappa * diag(nrow(P))

for (harm in c(F, T)) {

  # Harmonic penalty
  if (harm) {
    dk = (max(x) - min(x)) / nsegx
    per = 1
    phi = 2 * pi * dk / per
    dp = c(1, -2 * cos(phi), 1)
    Dx = matrix(0, nx - 2, nx)
    for (j in 1:(nx - 2)) Dx[j, j:(j + 2)] = dp
    Dx = diff(Dx)
    Px = lambdax * t(Dx) %*% Dx
    P = kronecker(Px, diag(na)) + kronecker(diag(nx), Pa)
    P = P + kappa * diag(nrow(P))
  }

  # Initialize
  Z = log((Y + 1) / (Exp + 2))

  # Iterate, using the array algorithm
  for (it in 1:10) {
    Mu = exp(Z) * Exp
    U = Y - Mu + Mu * Z
    Q = t(Ta) %*% Mu %*% Tx
    dim(Q) = c(na, na, nx, nx)
    Q = aperm(Q, c(1, 3, 2, 4))
    dim(Q) = c(nx * na, nx * na)
    r = t(Ba) %*% U %*% Bx
    dim(r) = c(nx * na, 1)
    A = solve(Q + P, r)
    a = A
    dim(A) = c(na, nx)
    Znew = Ba %*% A %*% t(Bx)
    dz = sum(abs(Z - Znew))
    cat(it, dz, ’\n’)
    if (dz < tol) break
    Z = Znew
  }

  # Save linear predictors
  if (harm == F) Z0 = Z
  if (harm) Zh= Z
}

# Data frames for ggplot
Z10 = log10(exp(Z0))
Zh10 = log10(exp(Zh))
R = log10(Y / Exp)
DF = data.frame(r = as.vector(R), z0 = as.vector(Z10), zh = as.vector(Zh10),
                y = rep(ages, mx), x = rep(x, each = ma))
sel = seq(10, 50, by = 10)
asel = ages[sel]
Zsel = t(Zh10[sel, ])
```

```
nz = length(sel)
DF2 = data.frame(x = rep(x, nz), y = as.vector(Zsel),
                 Age = as.factor(rep(asel, each = mx)))

# Parameters for the plots
leg = F
nbin = 6
brk = seq(-5, -1, by = 0.5)
brk2 = -3
ccols = c('yellow', 'green')

# Image of raw data
plt1 = ggplot(DF, aes(x, y, fill = r)) +
  geom_raster(show.legend = leg) +
  geom_contour(aes(z = r), color = ccols[1], show.legend = T, breaks = brk) +
  geom_contour(aes(z = r), color = ccols[2], show.legend = T, breaks = brk2) +
  xlab('Year') + ylab('Age') +
  ggtitle('Raw mortality (log10)') +
  JOPS_theme()

# Image of standard smooth
plt2 = ggplot(DF, aes(x, y, fill = z0)) +
  geom_raster(show.legend = leg) +
  geom_contour(aes(z = z0), color = ccols[1], show.legend = T, breaks = brk) +
  geom_contour(aes(z = z0), color = ccols[2], show.legend = T, breaks = brk2) +
  xlab('Year') + ylab('Age') +
  ggtitle('Standard smooth (log10)') +
  JOPS_theme()

# Image of harmonic smooth
plt3 = ggplot(DF, aes(x, y, fill = zh)) +
  geom_raster(show.legend = leg) +
  geom_contour(aes(z = zh), color = ccols[1], show.legend = T, breaks = brk) +
  geom_contour(aes(z = zh), color = ccols[2], show.legend = T, breaks = brk2) +
  xlab('Year') + ylab('Age') +
  ggtitle('Harmonic smooth (log10)') +
  JOPS_theme()

# Time series of harmonic smooth at some ages
plt4 = ggplot(DF2) +
  geom_line(aes(x = x, y = y, group = Age, color = Age), size = 1) +
  ggtitle('Cross-sections (harmonic)') +
  xlab('Year') + ylab('log10(mortality)') +
  JOPS_theme()

grid.arrange(plt1, plt2, plt3, plt4,  nrow = 2, ncol = 2)
```