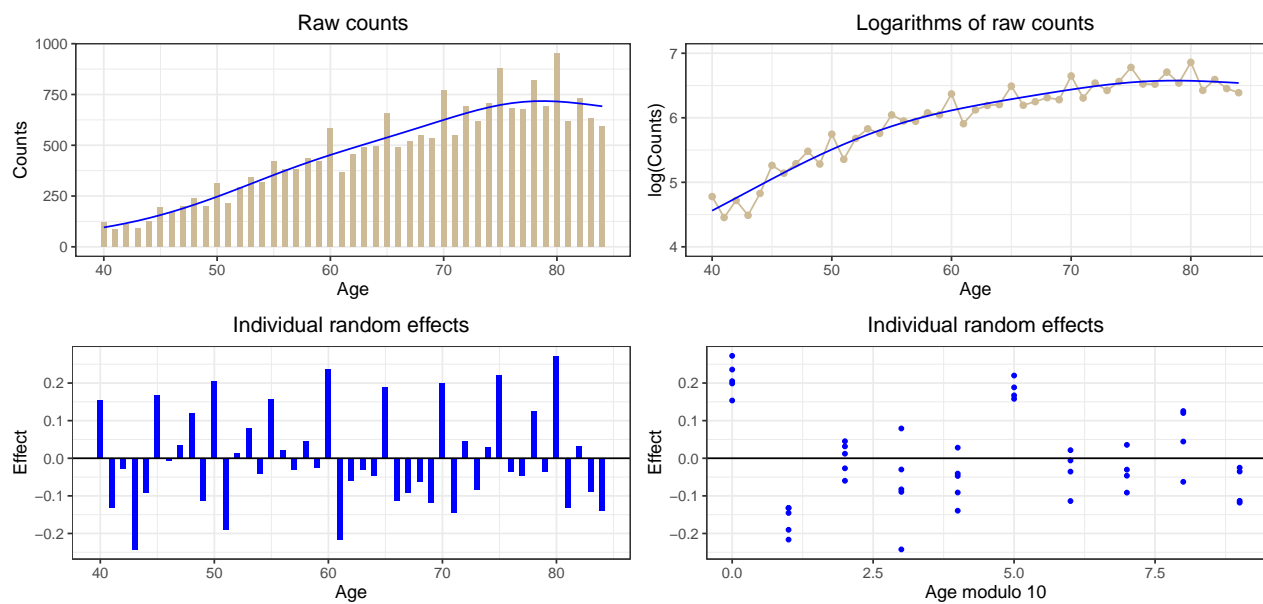


P-splines with PRIDE and Schall updates (Greece deaths data)



Counts of male deaths in Greece, in 1960, versus age, fitted by a model with a smooth trend plus individual random effects. Top left: raw counts and the trend. Top right: logarithms of raw counts and fitted trend. Bottom left: the individual random effects versus age. Bottom right: the individual random effects versus age modulo 10, to emphasize the repeating pattern of digit preference. R code in `f-Greece-PRIDE.R`

```
# P-splines with PRIDE and Schall updates (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)

# Create basis and penalty matrix
B = bbase(x, nseg = 20)
Bb = cbind(B, diag(m))
n = ncol(B)
D = diff(diag(n), diff = 2)
Pd = t(D) %>% D
Pr = diag(m)

# Pure P-spline Poisson fit
lambda = 100
kappa = 10
P = diag(n + m)
P[1:n, 1:n] = lambda * Pd
rr = (1:m) + n
P[rr, rr] = kappa * Pr
eta = log(y + 1)
```

```

a = rep(0, n + m)
mon = T
for (it in 1:40) {
  # Estimate coefficients
  mu = exp(eta)
  z = y - mu + mu * eta
  G = t(Bb) %%% diag(c(mu)) %%% Bb
  anew = solve(G + P, t(Bb) %%% z)
  da = max(abs(anew - a))
  if (mon) cat(it, lambda, kappa, da, '\n')
  a = anew
  eta = Bb %%% a

  # Update penalty parameters
  H = solve(G + P, G)
  hd = diag(H)
  edt = sum(hd[1:n])
  edr = sum(hd[rr])
  atrend = a[1:n]
  apride = a[rr]
  lambda = edt / sum((D %%% atrend) ^ 2)
  kappa = edr / sum(apride ^ 2)
  P[1:n, 1:n] = lambda * Pd
  P[rr, rr] = kappa * Pr
  if (da < 1e-6) break
}
trend = exp(B %%% atrend)

# Diagnostics
ed = edt + edr
ok = which(y > 0)
dev = 2 * sum(y * log(y / mu))
aic = dev + 2 * ed
cat('lambda, kappa, aic, ed:', lambda, kappa, aic, ed, edt, edr, '\n')

F2 = data.frame(x = x, y = y, trend = trend, apride = apride)

plt1 = ggplot(F2) +
  geom_bar(aes(x, y), stat = 'identity', fill = 'wheat3', width = 0.5) +
  geom_line(aes(x, trend), col = 'blue') +
  xlab('Age') + ylab('Counts') +
  ggtitle('Raw counts') +
  JOPS_theme()

plt2 = ggplot(F2) +
  geom_point(aes(x, log(y)), col = 'wheat3') +
  geom_line(aes(x, log(y)), col = 'wheat3') +
  geom_line(aes(x, log(trend)), col = 'blue') +
  ylim(4, 7) +
  xlab('Age') + ylab('log(Counts)') +
  ggtitle('Logarithms of raw counts') +
  JOPS_theme()

plt3 = ggplot(F2) +
  geom_bar(aes(x, apride), stat = 'identity', fill = 'blue', width = 0.5) +
  xlab('Age') + ylab('Effect') +
  geom_hline(yintercept = 0) +
  ggtitle('Individual random effects') +
  JOPS_theme()

plt4 = ggplot(F2) +
  geom_point(aes(x = x %% 10, y = apride), color = 'blue', size = 1) +
  xlab('Age modulo 10') + ylab('Effect') +
  geom_hline(yintercept = 0) +
  ggtitle('Individual random effects') +
  JOPS_theme()

```

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