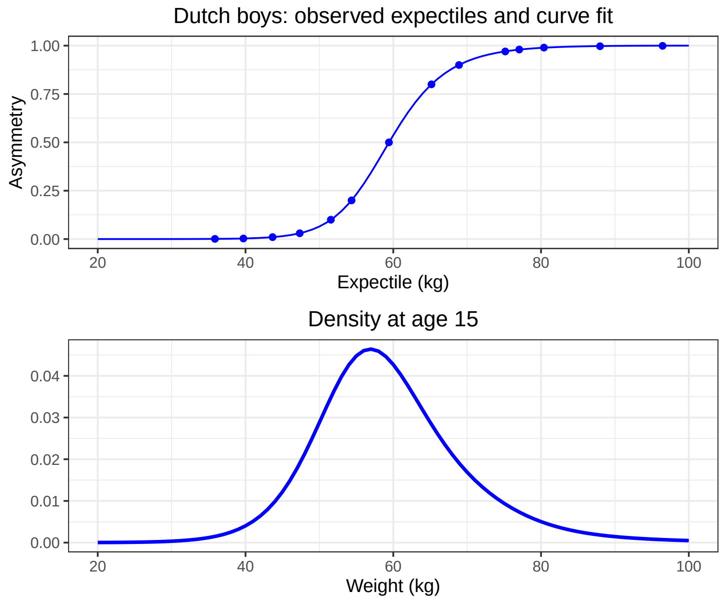
Estimate expectile curves and density at age 15 (Dutch boys data)



Boys' weights. Expectiles at age 15 were obtained from the individual expectile curves, shown in Figure 5.9. They are plotted in the upper panel. The right panel shows the density estimate obtained from the expectiles. R code in `f-exp-curves-1000-dens`

```
# Estimate expectile curves and density at age 15 (Dutch boys data)
# Paul Eilers and Brian Marx 2019
# A graph in "Practical Smoothing. The Joys of P-splines"

library(colorspace)
library(AGD)
library(ggplot2)
library(JOPS)
library(gridExtra)

# Get the data
data(boys7482)
Dat = subset(boys7482, age > 5, select = c(age, wgt))
Dat = na.omit(Dat)
m0 = nrow(Dat)
m = 1000
set.seed(2017)
sel = sample(1:m0, m)
x = (Dat$age[sel])
y = Dat$wgt[sel]

# P-spline paremeters
xl = min(x)
xr = max(x)
nsegx = 20
bdeg = 3
pp = c(0.001, 0.003, 0.01, 0.03, 0.1, 0.2, 0.5, 0.8, 0.9, 0.97,
       0.98, 0.99, 0.997, 0.999)
np= length(pp)

# Compute bases for curves
B = bbase(x, xl, xr, nsegx, bdeg)
nbx = ncol(B)
xg = seq(xl, xr, length = 100)
```

```
Bg = bbase(xg, xl, xr, nsegx, bdeg)

# Compute basis for expectile values at age 'xe'
xe = 15
Be = bbase(xe, xl, xr, nsegx, bdeg)

# Compute penalty matrix
D = diff(diag(nbx), diff = 2)
lambdax = 10
P = lambdax * t(D) %*% D

# Fit the expectile curves
Zg = NULL
ze = NULL
for (p in pp) {
  dzmax = 1e-3 * max(y)
  z = 0
  for (it in 1:10) {
    r = y - z
    w = ifelse(r > 0, p, 1- p)
    W = diag(c(w))
    Q = t(B) %*% W %*% B
    a = solve(Q + P, t(B) %*% (w * y))
    znew = B %*% a
    dz = sum(abs(z - znew))
    if (dz < dzmax) break
    z = znew
  }
  Zg = cbind(Zg, Bg %*% a)
  ze = c(ze, Be %*% a)
  cat(p, '\n')
}

# Set grid for density estimation
umin = 20
umax = 100
u = seq(umin, umax, by = 1)
nu = length(u)

# Prepare penalty for density
d = 2
D2 = diff(diag(nu), diff = d)
lambda2 = 1
P2 = lambda2 * t(D2) %*% D2

# Make model matrix A
A = matrix(0, np + 1, nu)
A[np + 1,] = 1
for(k in 1:np){
  a1 = (1 - pp[k]) * (u - ze[k]) * (u <= ze[k])
  a2 = pp[k] * (u - ze[k]) * (u > ze[k])
  A[k,] = a1 + a2
}

# Linear start for density
v = solve(t(A) %*% A + P2, t(A) %*% c(rep(0,np), 1))

# Model for log-density
q = c(rep(0, np), 1)
z = log(v - min(v) + 0.02 * max(v))
for (it2 in 1:1) {
  P2 = lambda2 * t(D2) %*% D2
  for (it in 1:20) {
    g = exp(z)
    r = q - A %*% g
    B = A * outer(rep(1, np + 1), as.vector(g))
    Q = t(B) %*% B
```

```
    znew = solve(Q + P2, t(B) %*%  r + Q %*% z)
    dz = max(abs(z - znew))
    z = znew
    if (dz < 1e-6) break
  }

  # Update lambda (HFS algorithm)
  G = solve(Q + P2, Q)
  ed = sum(diag(G))
  v1 = sum((D2 %*% z) ^ 2) / ed  + 1e-8
  v2 = sum(r ^ 2) / (length(r) - ed - d)
  lanew = v2 / v1
  cat(it2, lambda2, lanew, '\n')
  dla = abs(lambda2 - lanew)
  if (dla < 1e-4 * lambda2) break
  lambda2 = lanew
}

# Compute fitted expectile curve
sl = cumsum(g)
tl = cumsum(u * g)
sr = sl[nu] - sl
tr = tl[nu] - tl
bb = (tr - u * sr) / (u * sl - tl)
aa = 1  /  (1 + bb)

# Dataframes for ggplot
np = length(pp)
ng = length(xg)
DF1 = data.frame(tau = pp, ze =  ze)
DF2 = data.frame(u = u, g = g, aa = aa )

# Build the graphs
plt1 = ggplot(DF1) +
  geom_point(aes(x = ze, y = tau), col = 'blue') +
  geom_line(data = DF2, aes(y = aa, x = u), col = 'blue', lty = 1) +
  ylab('Asymmetry') + xlab('Expectile (kg)') +
  ggtitle('Dutch boys: observed expectiles and curve fit')  +
  JOPS_theme()

plt2 = ggplot(DF2) +
  geom_line(aes(x = u, y = g), col = 'blue', size = 1) +
  xlab('Weight (kg)') + ylab('') +
  ggtitle(paste('Density at age', xe)) +
  JOPS_theme()

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