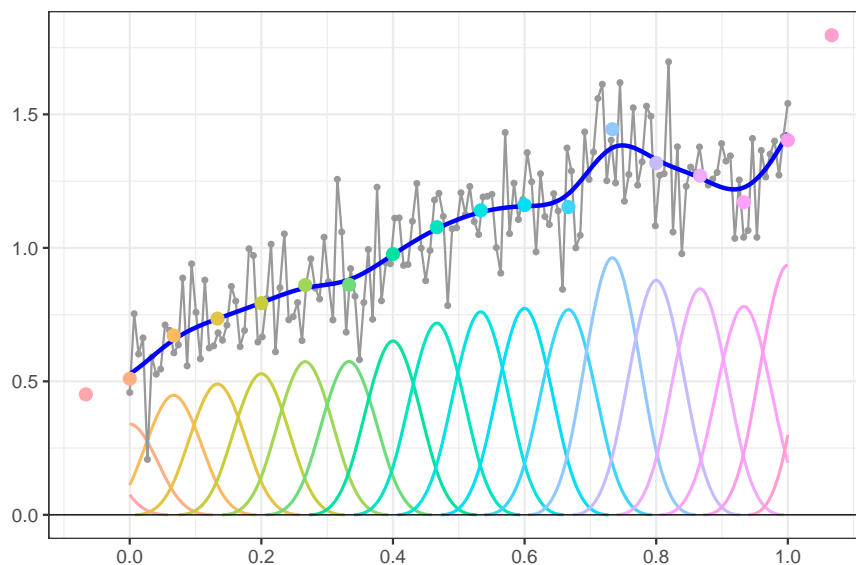


Show the essence of P-splines



The core idea of P-splines: a sum of B-spline basis functions, with gradually changing heights. The small connected grey dots show simulated data. The blue curve shows the P-spline fit, and the large dots show the B-spline coefficients (they have the same colours as the splines). R code in `f-ps-show.R`

```
# Show the essence of P-splines
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

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

# Simulate data
n = 150
set.seed(2016)
x = seq(0, 1, length = n)
y = 0.3 + sin(1.2 * x + 0.3) + rnorm(n) * 0.15
Data = data.frame(x, y, id = as.factor(15))

# Make a matrix containing the B-spline basis
ndx = 15
deg = 3
B = bbase(x, 0, 1, nseg = ndx, bdeg = deg)
nb = ncol(B)

# A basis for plotting the fit on the grid xg
ng = 500
xg = seq(0, 1, length = ng)
Bg = bbase(xg, 0, 1, nseg = ndx, bdeg = deg)

# Positions of the peaks of the B-splines
dk = 1/ndx
xa = (1:(ndx + deg)) * dk - (deg + 1) * dk/2

# Estimate the coefficients and compute the fit on the grid
D = diff(diag(nb), diff = 2)
lambda = 0.1
```

```

a = solve(t(B) %*% B + lambda * t(D) %*% D, t(B) %*% y)
z = Bg %*% a

# Make a matrix with B-splines scaled by coefficients
Bsc = Bg %*% diag(c(a))

# Create data frames for ggplot
Zf = data.frame(x = xg, y = z, id = as.factor(1))
C = data.frame(x = xa, y = a, id = as.factor(1))
Bf = data.frame(x = rep(xg, nb), y = as.vector(Bsc), id = as.factor(rep(1:nb,
  each = ng)))
Bf$y[abs(Bf$y) < 1e-04] = NA
Bf = na.omit(Bf)

# Build the graph
pal = rainbow_hcl(nb, c = 80, l = 80, start = 10, end = 350)
plt = ggplot(Bf, aes(x = x, y = y, group = id, colour = id)) +
  geom_line(size = 0.7) +
  geom_hline(yintercept = 0, size = 0.3) +
  geom_line(data = Data, color = I("grey60")) +
  geom_point(data = Data, color = I("grey60"), size = 0.9) +
  geom_line(data = Zf, size = 1, color = I('blue')) +
  geom_point(data = C, color = pal, size = 2.5) +
  xlab("") + ylab("") +
  JOPS_theme() +
  theme(legend.position = "none") +
  scale_x_continuous(breaks = seq(0, 1, by = 0.2)) +
  scale_color_manual(values = pal)

# Show the graphs on the screen
grid.arrange(plt, ncol = 1, nrow = 1)

```
