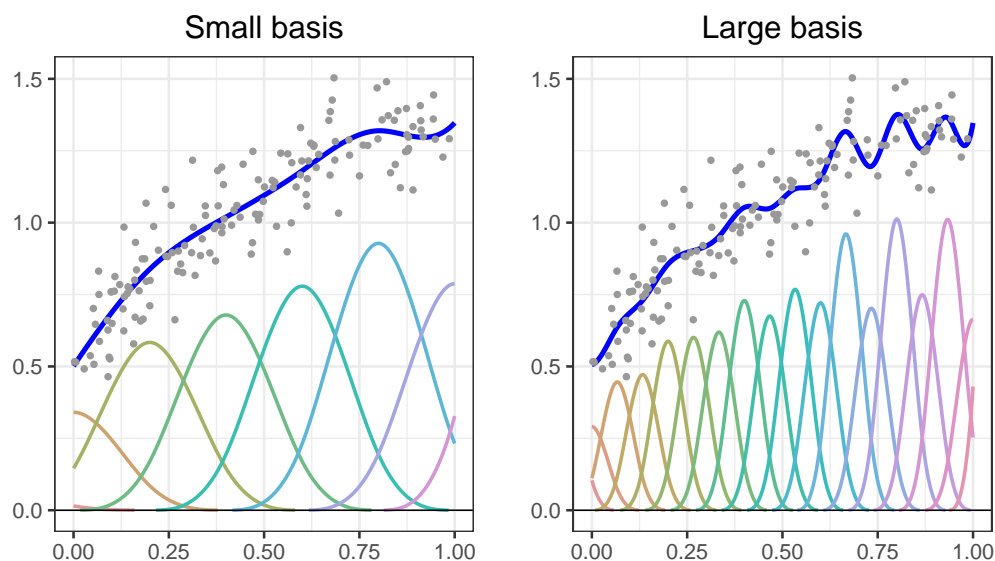


Illustration B-splines differing number of segments (simulated data)



Two cubic B-spline fits to the same simulated data, with a small basis (left) and with a larger one (right). R code in `f-b-size.R`

```
# Illustration of B-spline fits with varying basis size
# 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 = runif(n)
y = 0.3 + sin(1.2 * x + 0.3) + rnorm(n) * 0.1
Data = data.frame(x, y, id = as.factor(5))
```

```
# Make a matrix containing the small B-spline basis
ndx = 5
deg = 3
B = bbase(x, 0, 1, nseg = ndx, bdeg = deg)
nb1 = 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)
```

```
# Estimate the coefficients and compute the fit on the grid
a = solve(t(B) %*% B, t(B) %*% y)
z = Bg %*% a
```

```
# Make a matrix with B-splines scaled by coefficients
Bsc1 = Bg %*% diag(c(a))
```

```
# Create data frames for ggplot
```

```

Zf1 = data.frame(x = xg, y = z, id = as.factor(1))
Bf1 = data.frame(x = rep(xg, nb1), y = as.vector(Bsc1),
                 id = as.factor(rep(1:nb1, each = ng)))
Bf1$y[abs(Bf1$y) < 0.0001] = NA
Bf1 = na.omit(Bf1)

# Build the graphs
plt1 = ggplot(Bf1, aes(x = x, y = y, group = id, colour = id)) +
  geom_line(size = 0.7) +
  ggtitle("Small basis") +
  geom_hline(yintercept = 0, size = 0.3) +
  geom_line(data = Zf1, size = 1, colour = "blue") +
  geom_point(data = Data, color = "grey60", size = 0.8) +
  xlab("") + ylab("") +
  JOPS_theme() +
  theme(legend.position = "none") +
  scale_color_manual(values = rainbow_hcl(nb1 + 1, start = 10, end = 350))

# Make a matrix containing the large 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)

# Estimate the coefficients and compute the fit on the grid
a = solve(t(B) %>% B, 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))
Bf = data.frame(x = rep(xg, nb), y = as.vector(Bsc),
                 id = as.factor(rep(1:nb, each = ng)))
Bf$y[abs(Bf$y) < 0.0001] = NA
Bf = na.omit(Bf)

# Build the graphs
plt2 = ggplot(Bf, aes(x = x, y = y, group = id, colour = id)) +
  geom_line(size = 0.7) +
  ggtitle("Large basis") +
  geom_hline(yintercept = 0, size = 0.3) +
  geom_line(data = Zf, size = 1, color = 'blue') +
  geom_point(data = Data, color = "grey60", size = 0.8) +
  xlab("") + ylab("") +
  JOPS_theme() +
  theme(legend.position = "none") +
  scale_color_manual(values=rainbow_hcl(nb, start = 10, end = 350))

# Show the graphs on the screen and save them
grid.arrange(plt1, plt2, ncol = 2, nrow = 1)

```
