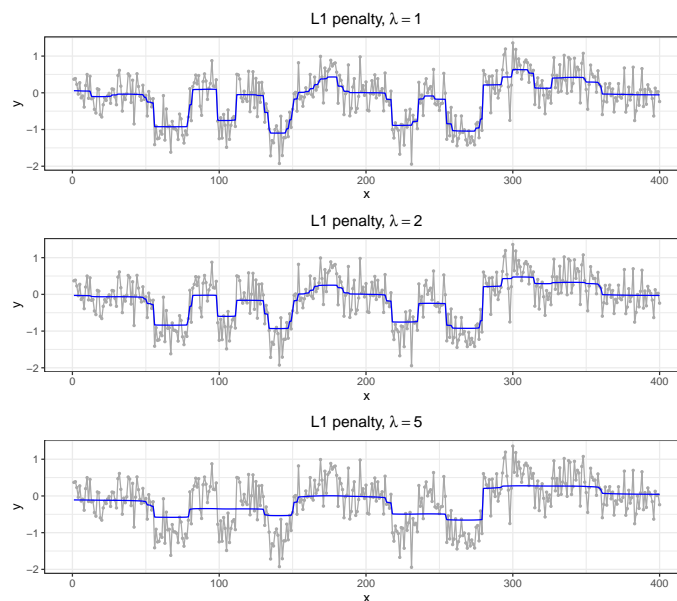


## Smoothing with the $L_1$ norm of differences in the penalty

---



Smoothing of simulated data with the sum of absolute values of differences in the penalty, for different values of  $\lambda$ , as indicated in the titles of the panels. R code in `f-cgh-L1.R`

---

```
# Smoothing with the sum of absolute values of differences in the penalty (Simulated data)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019
```

```
library(spam)
library(ggplot2)
library(gridExtra)
library(JOPS)
```

```
# Get the data
y = CGHsim$y
x = CGHsim$x
```

```
# Prepare for smoothing
m = length(y)
E = diag.spam(m)
D = diff(E)
beta = 0.001
nit = 200
z = 0
lambdas = c(1, 2, 5)
Z = NULL
```

```
# Do the smoothing
for (lambda in lambdas) {
  v = rep(1, m - 1)
  for (it in 1:nit) {
    V = diag.spam(as.vector(v))
    P = lambda * t(D) %*% V %*% D
    znew = solve(E + P, y)
    dz = max(abs(znew - z))
    z = znew
    if (dz < 1e-04) break
    g = diff(z)
    v = 1 / (g ^ 2 + beta ^ 2)
  }
}
```

```
    v = sqrt(v)
  }
  Z = cbind(Z, z)
  cat(lambda, "\n")
}

# Plot results
pp = list()
Dxy = data.frame(x = x, y = y)

for (k in 1:3) {
  Dz = data.frame(x = x, z = Z[, k])
  plt = ggplot(Dxy, aes (x = x, y = y)) +
    geom_point(col = 'darkgrey', size = 0.6) +
    geom_line(col = 'darkgrey') +
    geom_line(data = Dz, aes(x = x, y = z), color = "blue") +
    ggtitle(bquote("L1 penalty," ~ lambda == .(lambdas[k])) +
    JOPS_theme()
  pp[[k]] = plt
}

# Make and save graph
grid.arrange(grobs = pp, nrow = 3, ncol = 1)
```

---