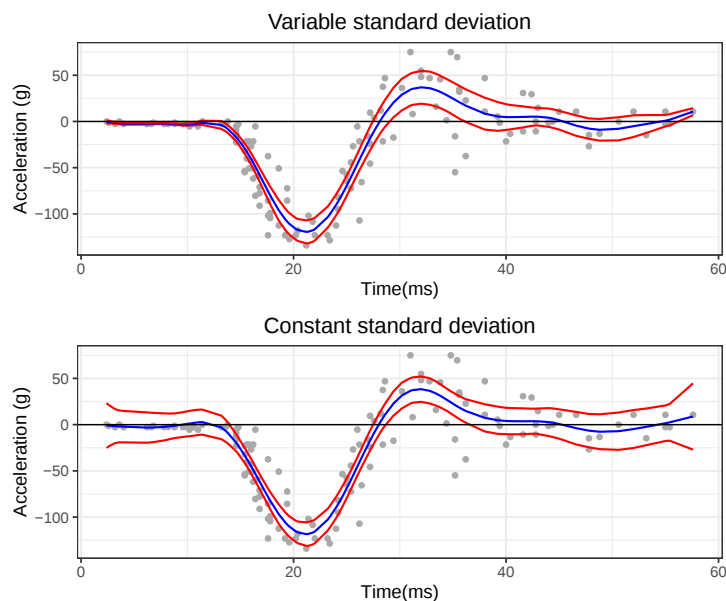


GAMLSS smoothing illustrating standard error bands (Motorcycle data)



Motorcycle impact data. Estimated mean curves and 95% confidence bands, for two assumptions for the standard deviation of the (normal) error distribution. Top panel: varying standard deviation. Bottom panel: fixed standard deviation. R code in `f-mot-gamlss2.R`

```
# GAMLSS smoothing illustrating standard error bands (Motorcycle data)
# A graph in the book "Practical Smoothing. The Joys of P-splines"
# Paul Eilers and Brian Marx, 2019

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

# Get the data
data(mcycle)
x = mcycle$times
y = mcycle$accel
ord = order(x)
dat = data.frame(x = x[ord], y = y[ord])
# Note: gamlss cannot compute standard errors on predictions
# Hence we compute predictions at the data points
# With sorted x we get decent curves

# Fit GAMLSS
gx = gamlss(y ~ pb(x), sigma.formula = ~ pb(x), data = dat)
g1 = gamlss(y ~ pb(x), sigma.formula = ~ 1, data = dat)

Ux = predict(gx, what = "mu", type = "response", se.fit = T, data = dat)
Vx = data.frame(dat$x, y1 = Ux$fit - 2 * Ux$se.fit, y2 = Ux$fit, y3 = Ux$fit + 2 * Ux$se.fit)
U1 = predict(g1, what = "mu", type = "response", se.fit = T, data = dat)
V1 = data.frame(dat$x, y1 = U1$fit - 2 * U1$se.fit, y2 = U1$fit, y3 = U1$fit + 2 * U1$se.fit)

# Build the graph
cols = c('red', 'blue', 'red')
plt1 = ggplot(data = Vx) +
  geom_point(data = dat, aes(x = x, y = y), size = 1.2, color = I("darkgrey")) +
  geom_line(aes(x = x, y = y1), color = cols[1], size = 0.6) +
```

```

geom_line(aes(x = x, y = y2), color = cols[2], size = 0.6) +
geom_line(aes(x = x, y = y3), color = cols[1], size = 0.6) +
ggtitle("Variable standard deviation") +
xlab("Time(ms)") + ylab("Acceleration (g)") +
geom_hline(yintercept = 0, size = 0.4) +
JOPS_theme()

plt2 = ggplot(data = V1) +
  geom_point(data = dat, aes(x = x, y = y), size = 1.2, color = I("darkgrey")) +
  geom_line(aes(x = x, y = y1), color = cols[1], size = 0.6) +
  geom_line(aes(x = x, y = y2), color = cols[2], size = 0.6) +
  geom_line(aes(x = x, y = y3), color = cols[1], size = 0.6) +
  ggtitle("Constant standard deviation") +
  xlab("Time(ms)") + ylab("Acceleration (g)") +
  geom_hline(yintercept = 0, size = 0.4) +
  JOPS_theme()

# Plot and save
grid.arrange(plt1, plt2, nrow = 2, ncol = 1)

```
