Simple registration examples

Lars Lau Raket

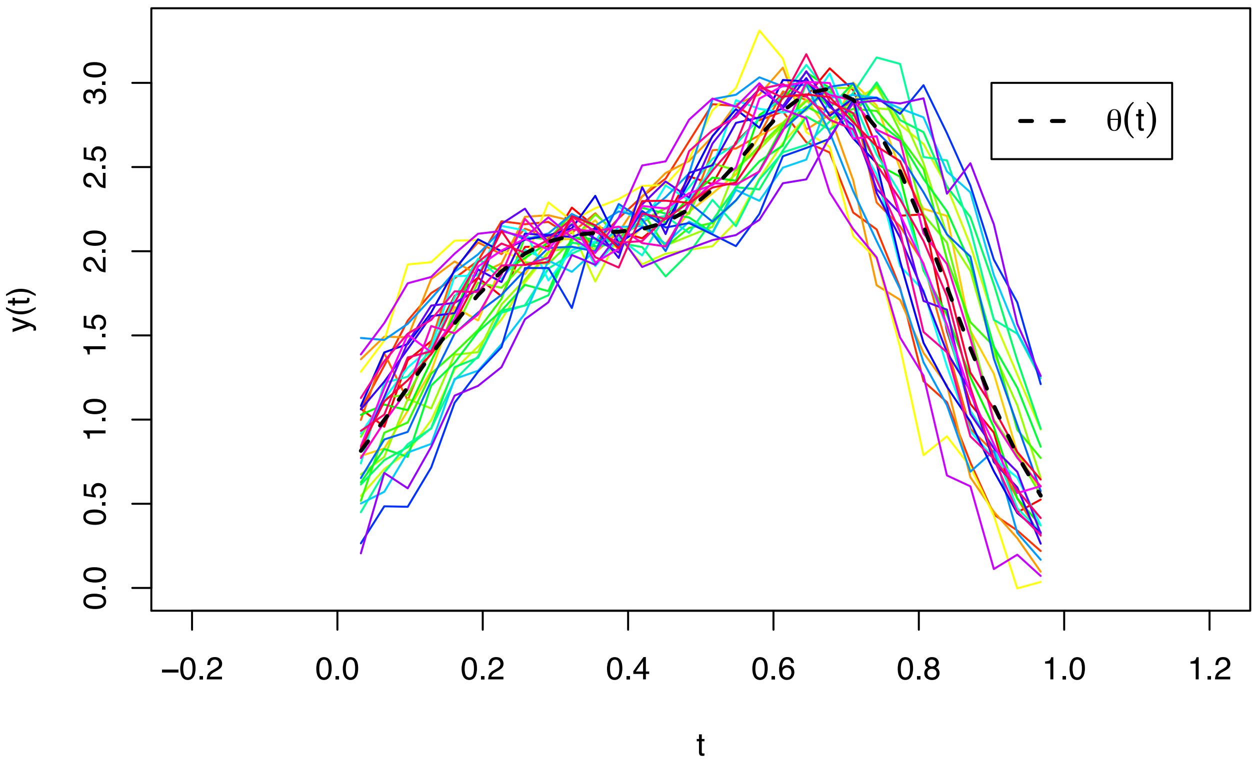
2016-01-08

In this vignette we will consider some simple examples of data that needs registration.

Example 1: Shifted curves

First, we generate a dataset with randomly shifted curves with noise

```
# Number of samples
n <- 30
# Number of observation points
m <- 30

# Observation points
t <- seq(0, 1, length = m + 2)[2:(m + 1)]

# Mean function
theta <- function(t) dnorm(t, mean = 0.3, sd = 0.2) + dnorm(t, mean = 0.7, sd = 0.15)

# Generate shifts
w <- runif(n, min = -0.1, max = 0.1)

# Generate data with random shifts
sigma <- 0.1
y <- lapply(w, function(w) {theta(t + w) + rnorm(m, sd = sigma)})
t <- lapply(1:n, function(x) t)

# Plot shifted curves
plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 't', ylab = 'y(t)')
legend(0.9, 3, legend = expression(theta(t)), lty = 2, lwd = 2)
for (i in 1:n) lines(t[[i]], y[[i]], col = rainbow(n)[i])
lines(t[[1]], theta(t[[1]]), lwd = 2, lty = 2)
```



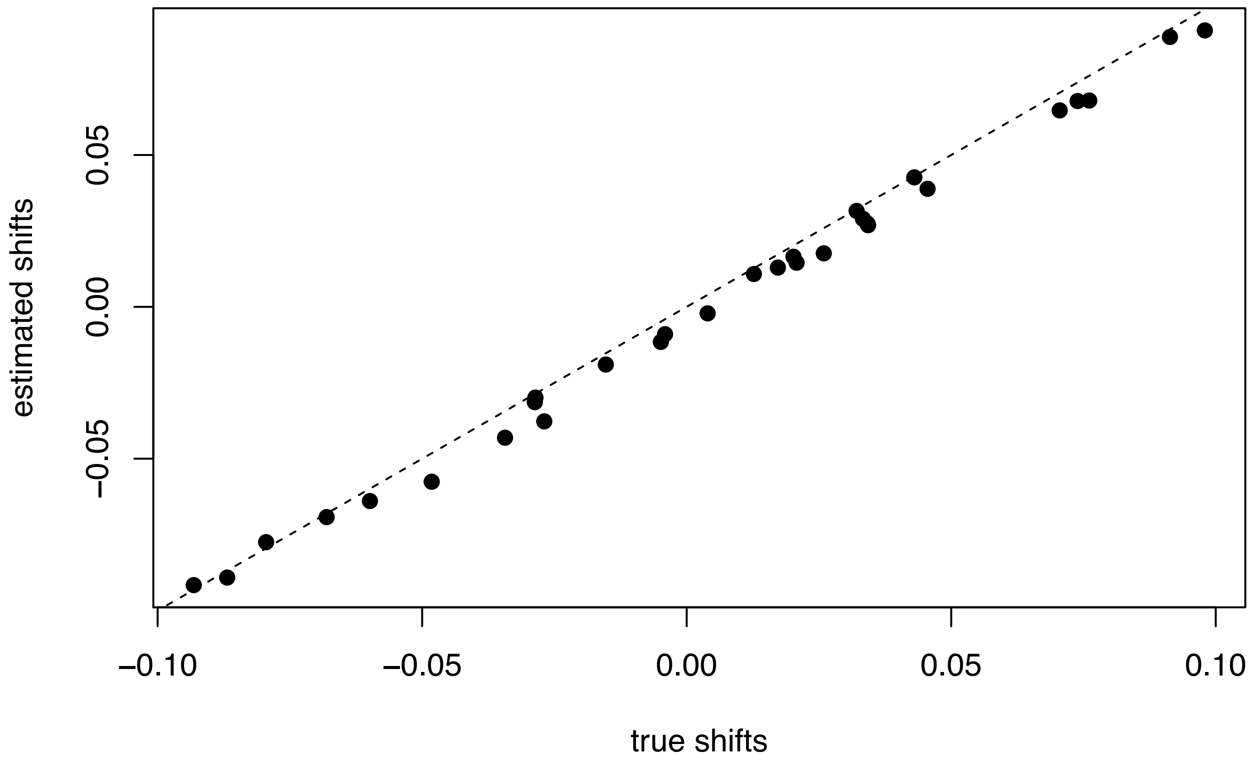
We now set up the pavpop model to estimate in the model

```
# Set up basis function
kts <- seq(-0.2, 1.2, length = 15)[2:14]
basis_fct <- make_basis_fct(kts = kts, intercept = TRUE, control = list(boundary = c(-0.2, 1.2)))

# Set up warp function
warp_fct <- make_warp_fct(type = 'shift')

# Estimate in the model
res <- pavpop(y, t, basis_fct, warp_fct, amp_cov = NULL, warp_cov = NULL, iter = c(1, 5))
```

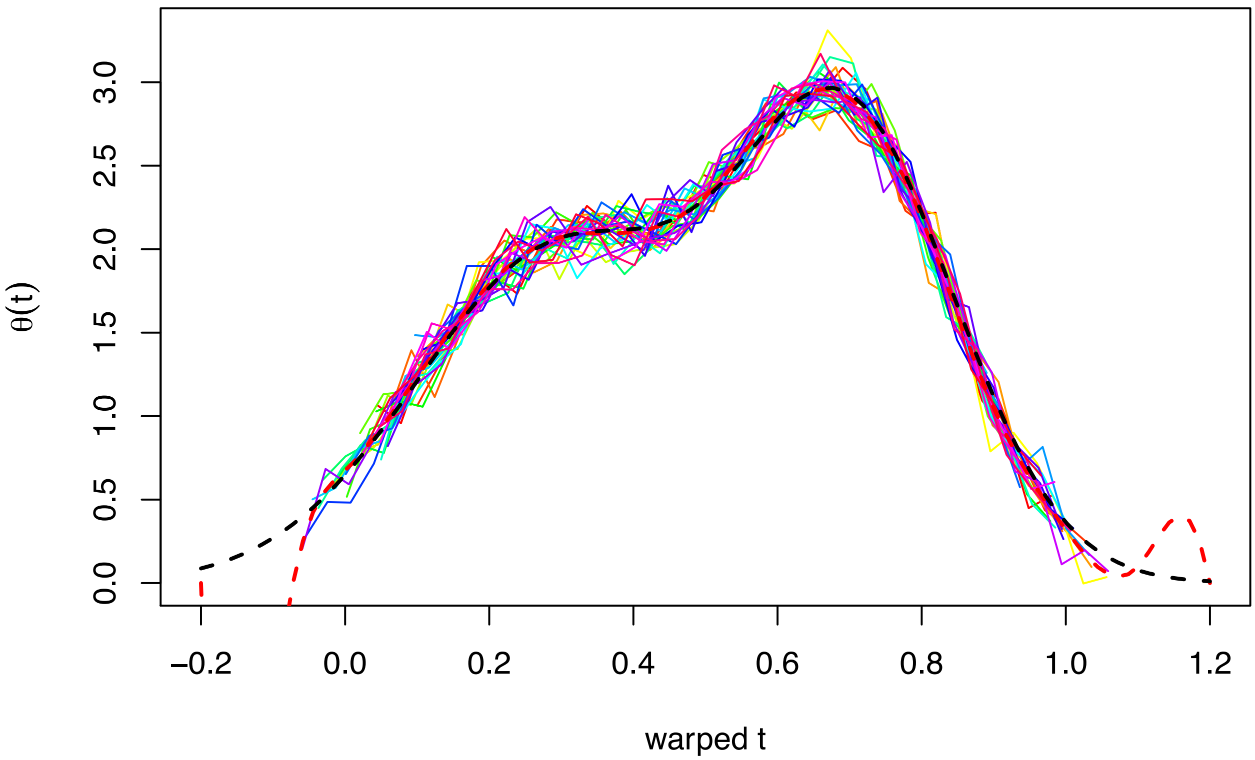
```
#> Outer    :   Inner   :   Estimates
#> 1    2
```

We can not plot the results using a coloring similar to before.

```
plot(w, res$w, xlab = 'true shifts', ylab = 'estimated shifts', pch = 19)
abline(0, 1, lty = 2)
```



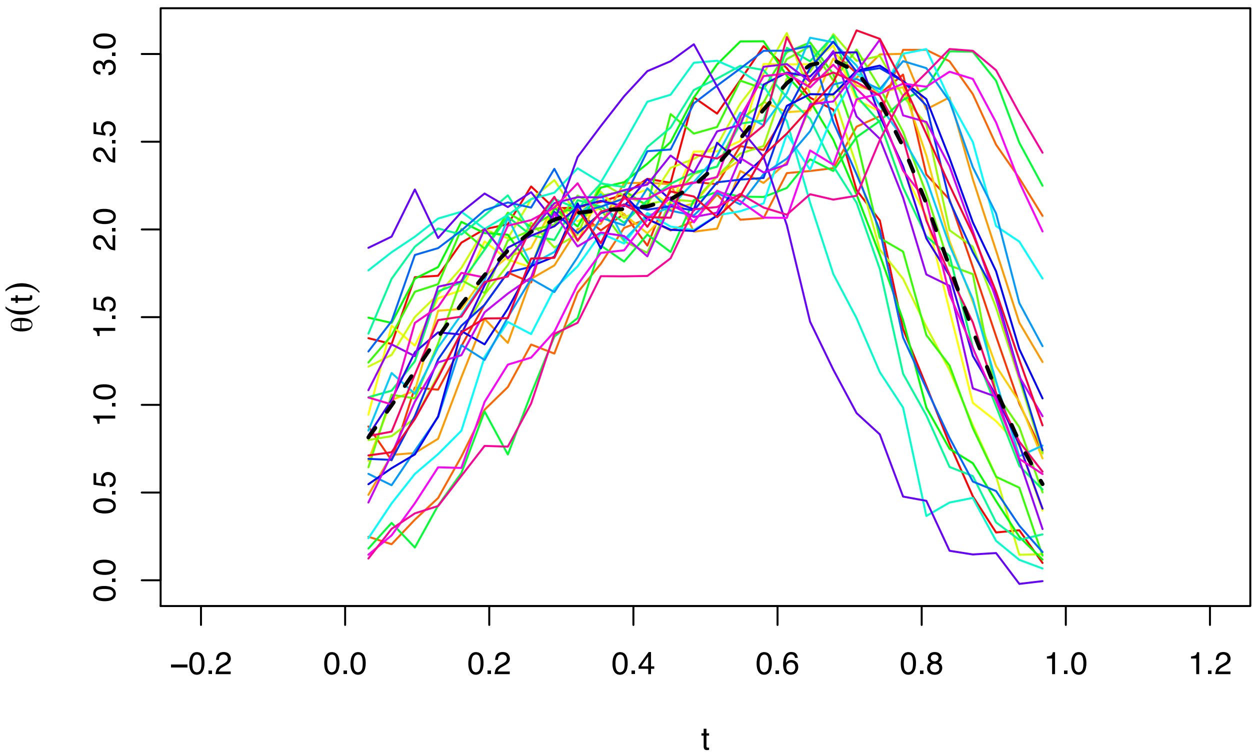

```r
t_plot <- seq(-0.2, 1.2, length = 100)
plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 'warped t', ylab = expression(theta(t)))
for (i in 1:n) lines(t[[i]] + res$w[i], y[[i]], col = rainbow(n)[i])
lines(t_plot, basis_fct(t_plot) %*% res$c, lwd = 2, lty = 2, col = 'red')
lines(t_plot, theta(t_plot), lwd = 2, lty = 2)
```



Example 2: Normally distributed shifts

We use the same data as before, but now we generate random shifts from a normal distribution

```r
# Generate shifts
scale <- 0.5
w <- rnorm(n, sd = sigma * sqrt(scale))

# Generate data with random shifts
y <- lapply(1:n, function(i) {theta(t[[i]] + w[i]) + rnorm(m, sd = sigma)})

# Plot shifted curves
plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 't', ylab = expression(theta(t)))
for (i in 1:n) lines(t[[i]], y[[i]], col = rainbow(n)[i])
lines(t[[1]], theta(t[[1]]), lwd = 2, lty = 2)
```



Let us first ignore the uncertainty and use the same model as in Example 1.

```
res_no_uncert <- pavpop(y, t, basis_fct, warp_fct, amp_cov = NULL, warp_cov = NULL,
                        iter = c(1, 5))
```

```
#> Outer    :   Inner   :   Estimates
#> 1    2   3
```

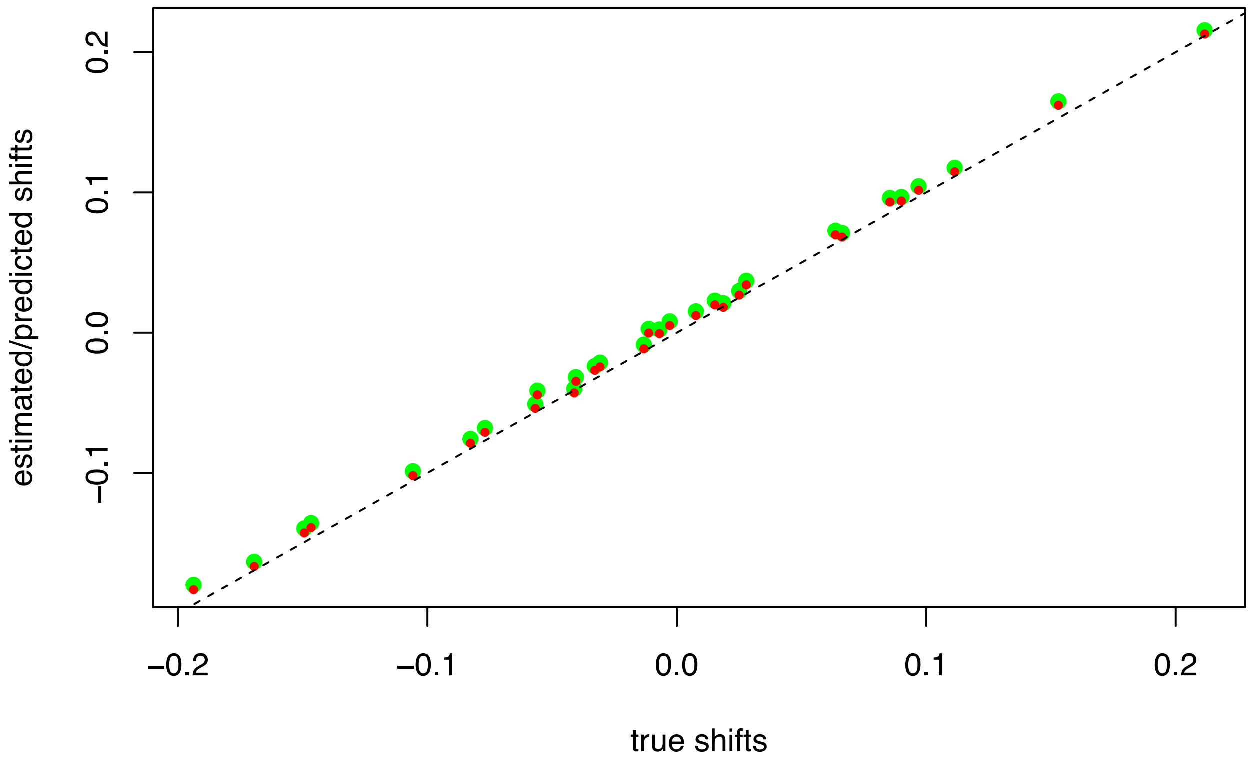
Correct specification of the model where the shifts are a normal random effect can be done as follows

```
warp_cov <- make_cov_fct(id_cov, noise = FALSE)
res <- pavpop(y, t, basis_fct, warp_fct, amp_cov = NULL, warp_cov = warp_cov, iter = c(5, 5))
```

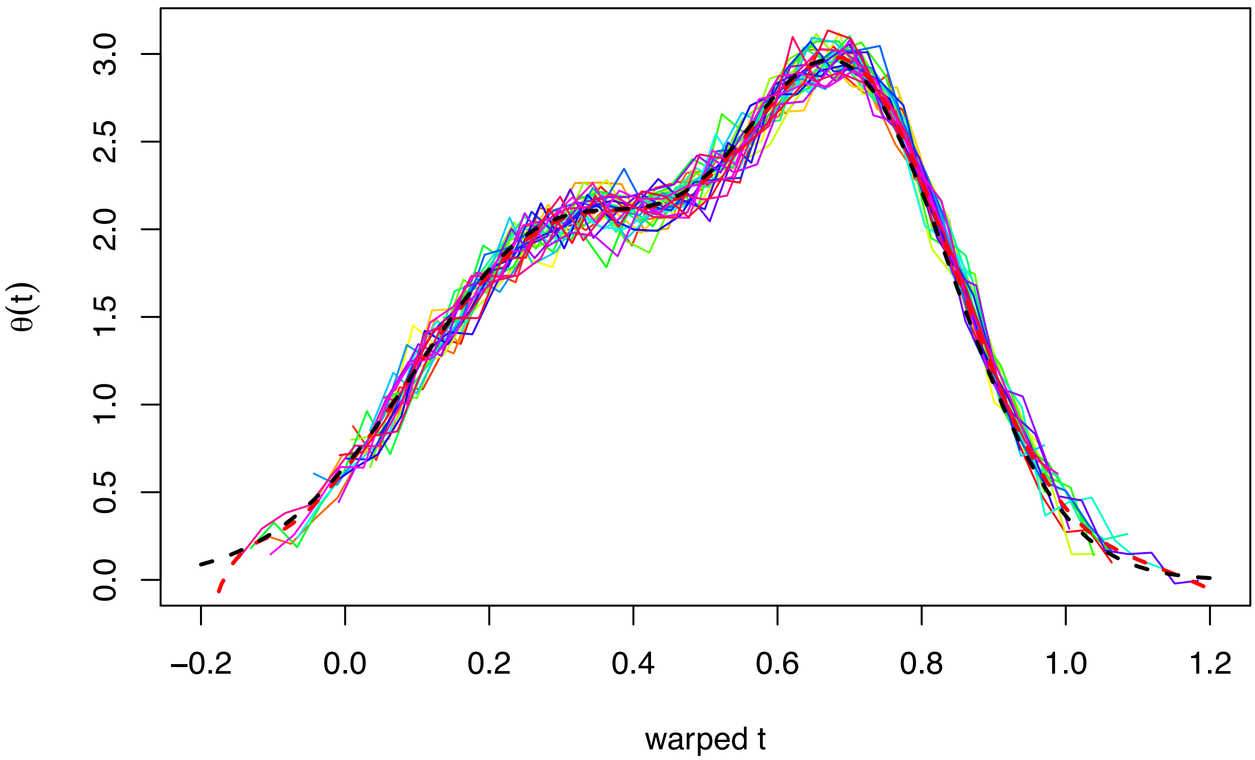
```
#> Outer    :   Inner   :   Estimates
#> 1    :   1   2   3   :    0.8920735
#> Linearized likelihood:    -3985.253
#> 2    :   1   .
#> Likelihood not improved, returning best likelihood estimates.
```

We can not plot the results, where the predicted shifts are green and the estimated shifts from the model with warps as fixed effects are red.

```
plot(w, res$w, xlab = 'true shifts', ylab = 'estimated/predicted shifts',
     pch = 19, col = 'green')
points(w, res_no_uncert$w, pch = 19, col = 'red', cex = 0.5)
abline(0, 1, lty = 2)
```




```
t_plot <- seq(-0.2, 1.2, length = 100)
plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 'warped t', ylab = expression(theta(t)))
for (i in 1:n) lines(t[[i]] + res$w[i], y[[i]], col = rainbow(n)[i])
lines(t_plot, basis_fct(t_plot) %*% res$c, lwd = 2, lty = 2, col = 'red')
lines(t_plot, theta(t_plot), lwd = 2, lty = 2)
```




```r
# Compare estimated variance parameters of warps

# True standard deviation
sigma * sqrt(scale)
```

```
#> [1] 0.07071068
```

```r
# Standard deviation of true shifts
sd(w)
```

```
#> [1] 0.09467399
```

```r
# Estimated standard deviation
res$sigma * sqrt(res$warp_cov_par)
```

```
#>      scale
#> 0.09225188
```

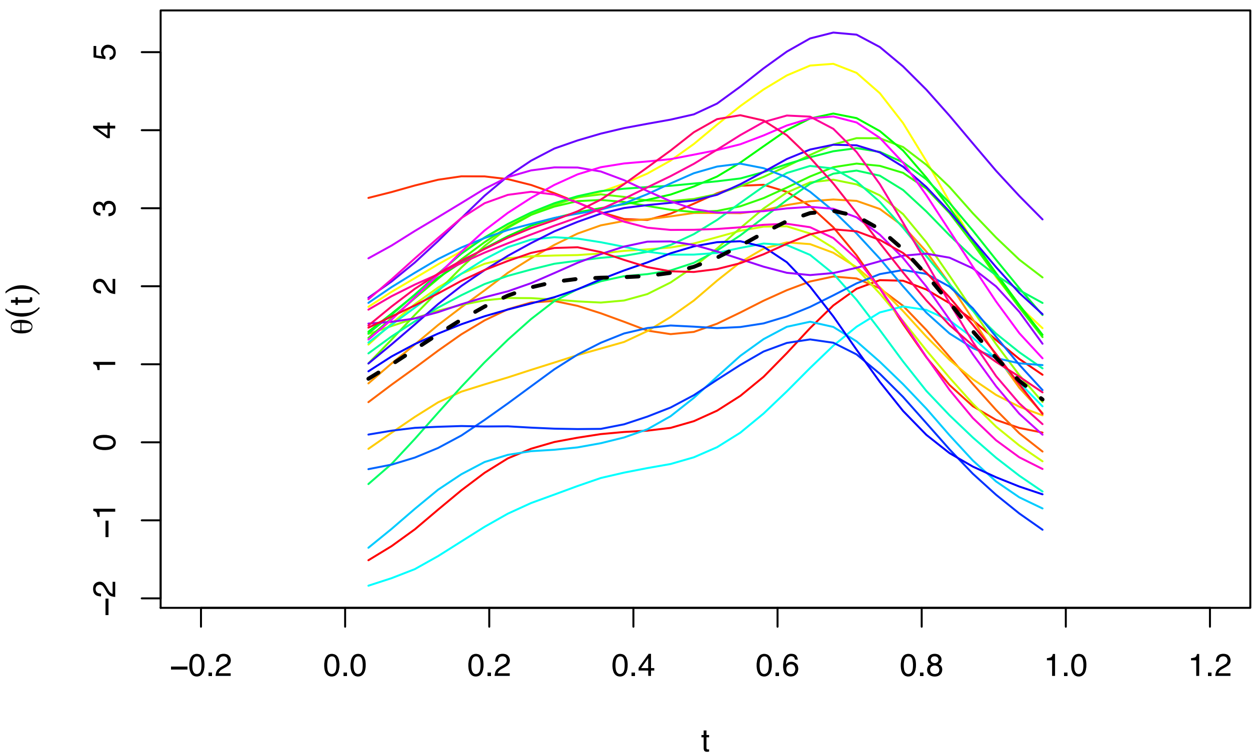
```r
# Standard deviation of predicted warps
sd(res$w)
```

```
#> [1] 0.09392661
```

Example 3: Serial correlation and normally distributed shifts

We use the same data as in Example 2, but instad of iid normally distributed noise we now add a Matérn process to the observations.

```
# Generate shifts
scale <- 0.5
w <- rnorm(n, sd = sigma * sqrt(scale))

amp_cov <- make_cov_fct(Matern, noise = FALSE)


# Generate data with random shifts
gen_dat <- function(i) {
  theta(t[[i]] + w[i]) + sigma * t(chol(amp_cov(t[[i]], c(100, 0.3, 2)))) %*% rnorm(m)
}
y <- lapply(1:n, gen_dat)

# Plot shifted curves
plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 't', ylab = expression(theta(t)))
for (i in 1:n) lines(t[[i]], y[[i]], col = rainbow(n)[i])
lines(t[[1]], theta(t[[1]]), lwd = 2, lty = 2)
```



Let us first ignore the uncertainty and use the same model as in Example 1.

```
res_no_uncert <- pavpop(y, t, basis_fct, warp_fct, amp_cov = NULL, warp_cov = NULL, iter = c(1, 10))
```

```
#> Outer    :   Inner   :   Estimates
#> 1    2   3   4   5   6   7   8   9   10
```

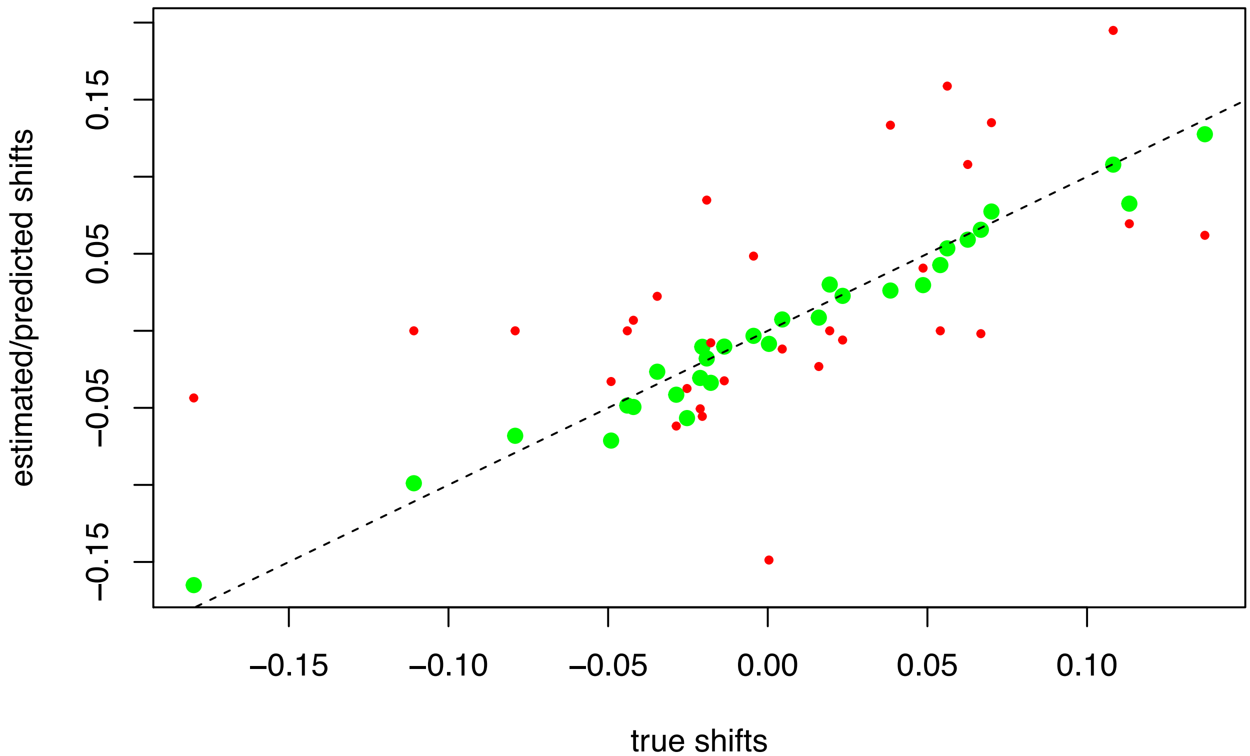
And now compare to the correct specification of the model

```r
res <- pavpop(y, t, basis_fct, warp_fct, amp_cov = amp_cov, warp_cov = warp_cov,
              iter = c(10, 5))
```

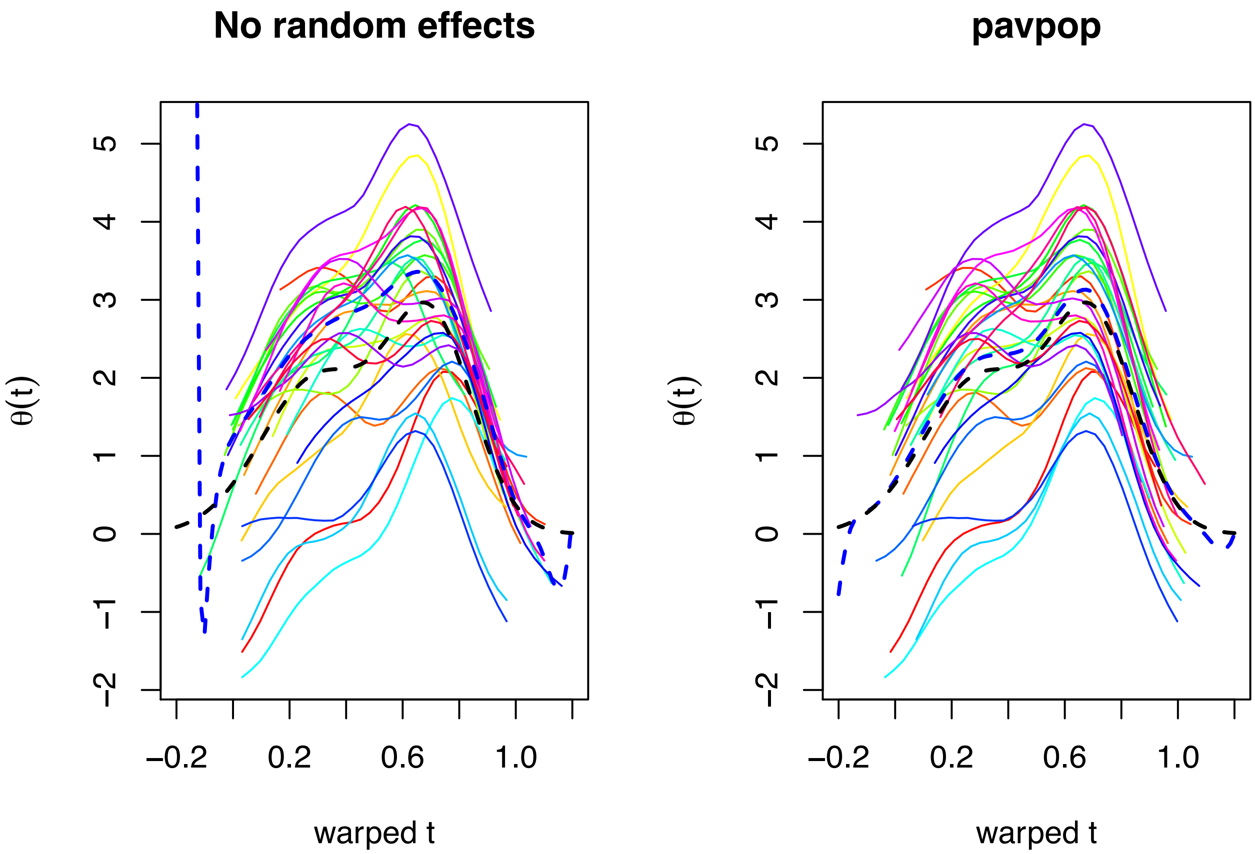
```
#> Outer    :   Inner   :   Estimates
#> 1    :   1   2   3   4   5   :    8.439427 0.32059 2.080244 0.01063026
#> Linearized likelihood:    -6897.458
#> 2    :   1   2   3   :    8.439254 0.2986459 1.989506 0.03548758
#> Linearized likelihood:    -7006.725
#> 3    :   1   2   :    8.439252 0.2999925 1.986209 0.03613124
#> Linearized likelihood:    -7007.115
#> 4    :   1   :    8.439251 0.2996842 1.98714 0.03634373
#> Linearized likelihood:    -7007.163
#> 5    :   1   :    8.43925 0.2994102 1.987947 0.03643408
#> Linearized likelihood:    -7007.199
#> 6    :   1   :    8.43925 0.2992597 1.988412 0.03649115
#> Linearized likelihood:    -7007.227
#> 7    :   1   :    8.43925 0.2992935 1.988464 0.03651716
#> Linearized likelihood:    -7007.248
#> 8    :   1   :    8.43925 0.2989943 1.98923 0.03654209
#> Linearized likelihood:    -7007.265
#> 9    :   1   :    8.43925 0.2990159 1.989267 0.036555
#> Linearized likelihood:    -7007.276
#> 10   :   1   :    8.43925 0.2988045 1.989809 0.03656657
#> Linearized likelihood:    -7007.285
```

We can not plot the results

```r
plot(w, res$w, xlab = 'true shifts', ylab = 'estimated/predicted shifts',
     pch = 19, col = 'green', ylim = range(res$w, res_no_uncert$w))
points(w, res_no_uncert$w, pch = 19, col = 'red', cex = 0.5)
abline(0, 1, lty = 2)
```




```
par(mfrow = c(1, 2))
plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 'warped t', ylab = expression(theta(t)), main = 'No random effects')
for (i in 1:n) lines(t[[i]] + res_no_uncert$w[i], y[[i]], col = rainbow(n)[i])
lines(t_plot, basis_fct(t_plot) %*% res_no_uncert$c, lwd = 2, lty = 2, col = 'blue')
lines(t_plot, theta(t_plot), lwd = 2, lty = 2)

plot(0, 0, xlim = c(-0.2, 1.2), ylim = range(y), type = 'n',
     xlab = 'warped t', ylab = expression(theta(t)), main = 'pavpop')
for (i in 1:n) lines(t[[i]] + res$w[i], y[[i]], col = rainbow(n)[i])
lines(t_plot, basis_fct(t_plot) %*% res$c, lwd = 2, lty = 2, col = 'blue')
lines(t_plot, theta(t_plot), lwd = 2, lty = 2)
```




```r
# Compare estimated variance parameters of warps

# True standard deviation
sigma * sqrt(scale)
```

```
#> [1] 0.07071068
```

```r
# Standard deviation of true shifts
sd(w)
```

```
#> [1] 0.06657387
```

```r
# Estimated standard deviation
res$sigma * sqrt(res$warp_cov_par)
```

```
#>      scale
#> 0.06287189
```

```r
# Standard deviation of predicted warps
sd(res$w)
```

```
#> [1] 0.06328579
```