# Getting started with gcreg

*Joshua Bon*

*2017-11-30*

gcreg fits constrained regression models. The functionality and documentation of this package is still being developed. Constrained polynomial regression models can be fitted with the function cpm(). This vignette will demonstrate some examples.

## Monotonic Regression

Monotonicity is a constraint provided directly in cpm(). We will use some datasets from the fda package.

The onechild in fda (Tuddenham and Snyder, 1954):

```
library(fda)
```

```
## Loading required package: splines
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'fda'
```

```
## The following object is masked from 'package:graphics':
##
##     matplot
```

```
library(ggplot2)

g_oc <- ggplot(data = fda::onechild) + geom_point(aes(y = height, x = day), size = 0.5) + theme_bw()

print(g_oc)
```

We can fit a constrained polynomial model and an unconstrained model for comparison:

```
library(gcreg)
```

```
## Loading required package: polynom
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(dplyr)

deg <- 11

c_model <- cpm(height ~ day, data = fda::onechild, degree = deg,
```

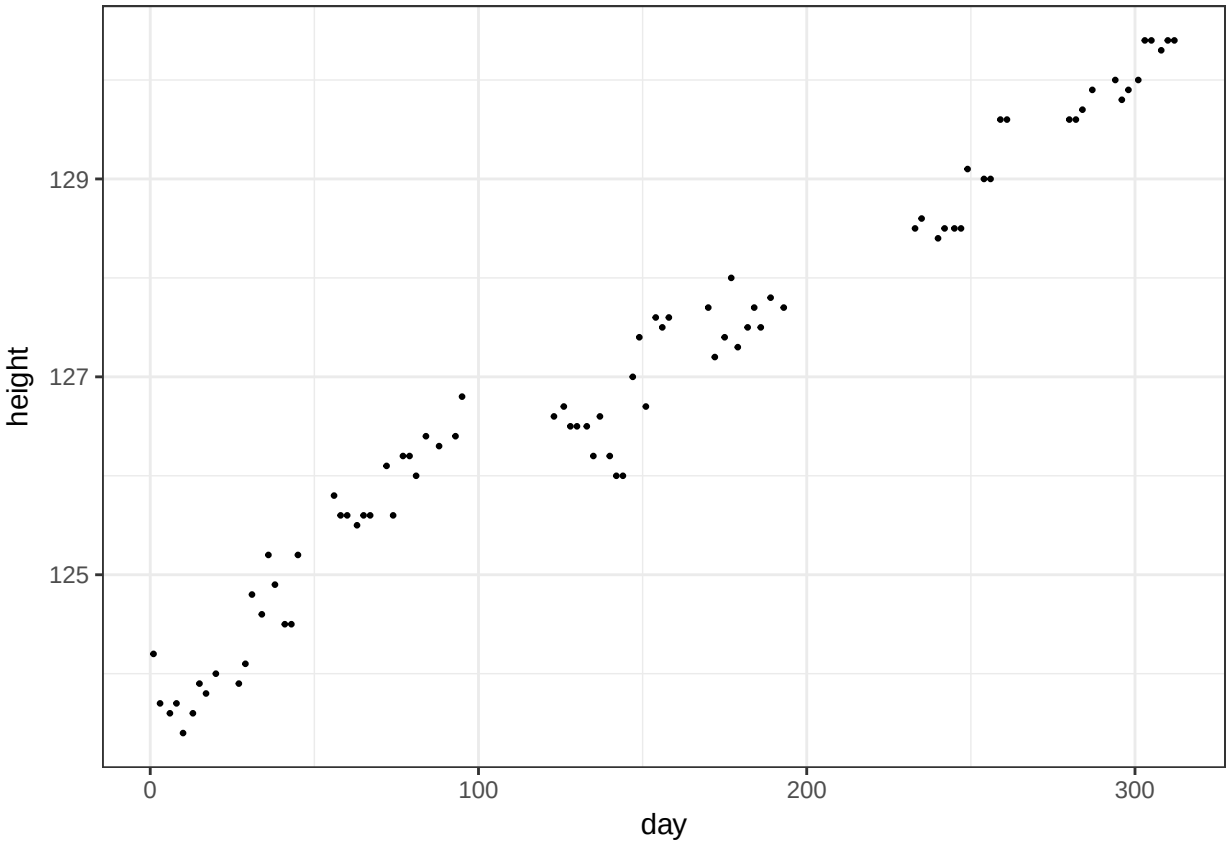



Figure 1: fda::onechild data

```
               constraint = "monotone", c_region = range(fda::onechild$day))

u_model <- lm(height ~ poly(day, degree = deg), data = fda::onechild)

# RSS
RSS <- list(
  c_model = sum(residuals(c_model)^2),
  u_model = sum(residuals(u_model)^2)
)

RSS
```

```
## $c_model
## [1] 6.556164
##
## $u_model
## [1] 5.595704
```

The fitted curves:

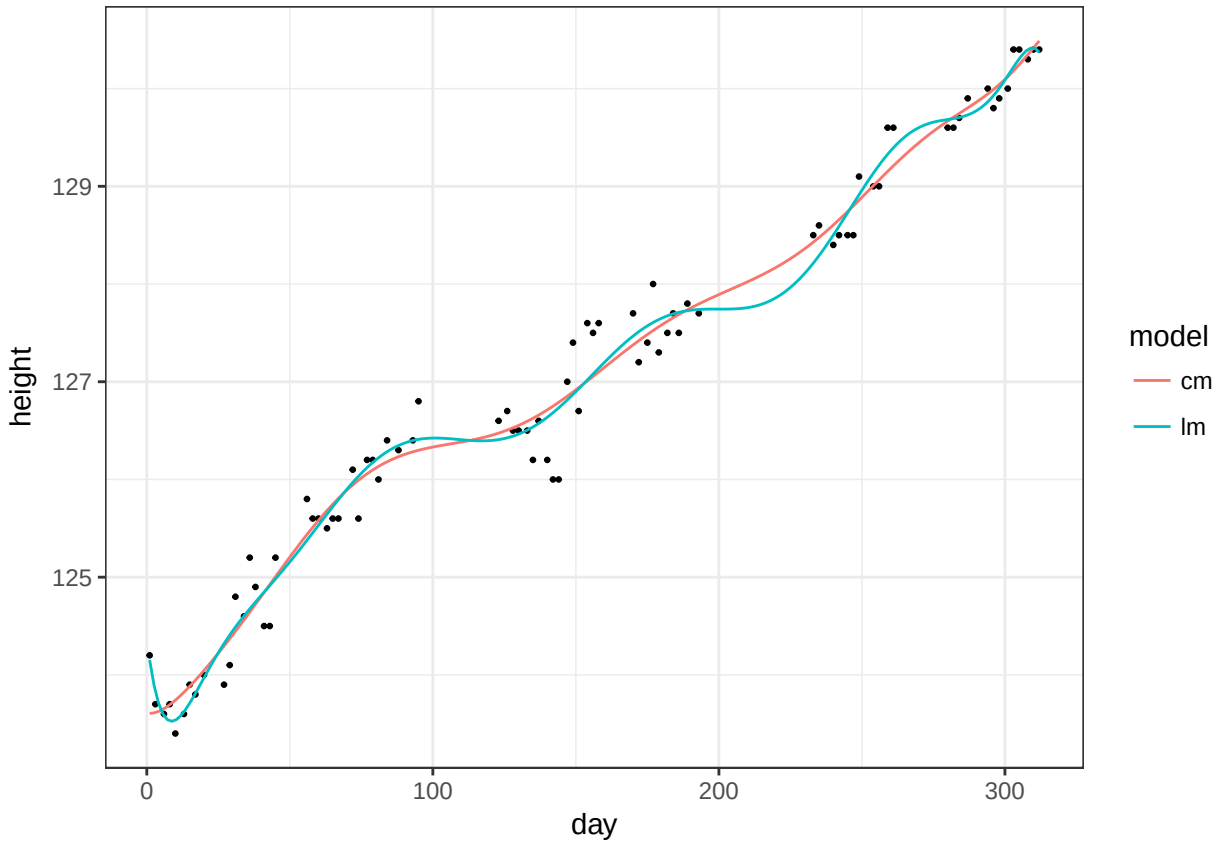
```
plot_dat <- with(fda::onechild, data.frame(day = seq(from = min(day), to = max(day), length.out = 201)))

c_plot_dat <- plot_dat %>% mutate(model = "cm", height = predict(c_model, newdata = plot_dat$day)[,1])

u_plot_dat <- plot_dat %>% mutate(model = "lm", height = predict(u_model, newdata = list(day=day)))

plot_dat <- rbind(c_plot_dat, u_plot_dat)

g_oc + geom_line(data = plot_dat, aes(y = height, x = day, colour = model))
```




## User-defined constraints

Use the make_oracle() function to define your own constraints. This is still under development - speed may vary.

```
 # Constraint: T if beta[3] >= 1, F otherwise
b3_g1 <- function(b){
  b[3,1] >= 1
}

deg <- 7

# Initial value
start_beta <- runif(n = deg + 1)
start_beta[3] <- 3

orc_f <- make_oracle(oracle = b3_g1)

c_model <- cpm(height ~ day, data = fda::onechild, degree = deg,
               oracle = orc_f, start = start_beta, step_start = 0.99, method = "down-walk")

c_model
```

```
##
## Constrained regression model
## Call:
```

```
## cpm(formula = height ~ day, data = fda::onechild, degree = deg,
##     oracle = orc_f, start = start_beta, step_start = 0.99, method = "down-walk")
##
## Coefficients:
##  Intercept         x^1         x^2         x^3         x^4         x^5
##  1.085e+03  -6.712e+01   1.000e+00  -5.109e-03   8.703e-06  -2.535e-09
##        x^6         x^7
##  5.454e-12  -4.668e-15
```