Week 11: Splines

Kerry Hu

03/04/23

Overview

In this lab you'll be fitting a second-order P-Splines regression model to foster care entries by state in the US, projecting out to 2030.

```
library(tidyverse)
library(here)
library(rstan)
library(tidybayes)
source(here("getsplines.R"))
```

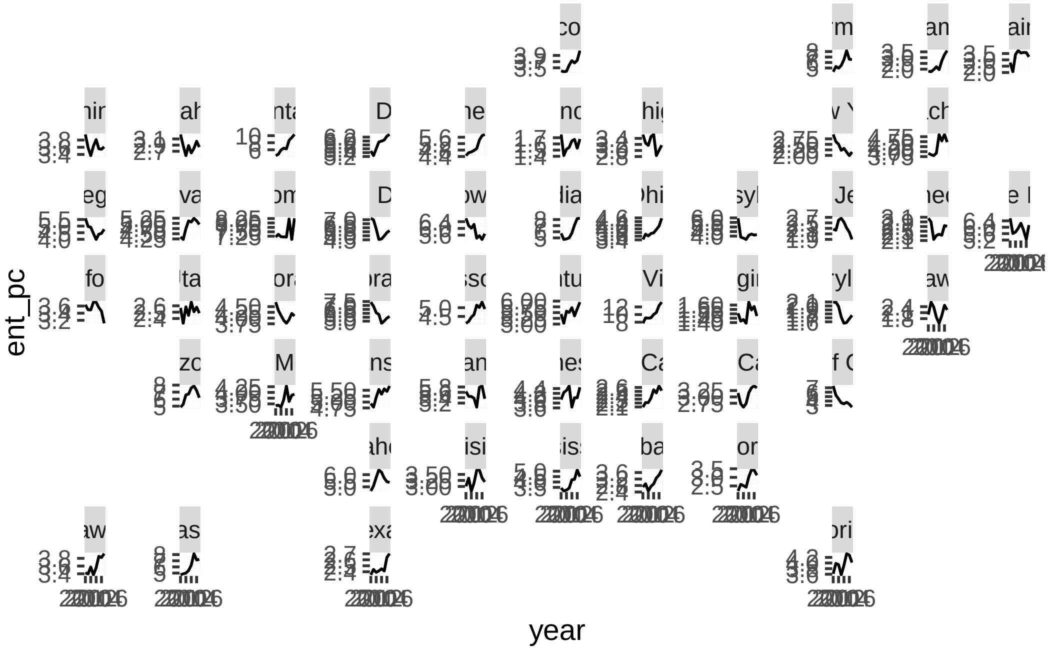
Here's the data

```
d <- read_csv(here("fc_entries.csv"))
```

Question 1

Make a plot highlighting trends over time by state. Might be a good opportunity to use `geofacet`. Describe what you see in a couple of sentences.

```
library(geofacet)
d |>
  ggplot(aes(year, ent_pc)) +
  geom_line()+
  facet_geo(~state, scales = "free_y")
```



We observed that there were some decreasing trends in some states such as North Dakota,New York,District of Columbia, New Jersey, and North Carolina.On the other hand, the trend of foster care entries per capita of other states increased across years. Moreover, some states showed no clear trend like Wyoming.

Question 2

Fit a hierarchical second-order P-Splines regression model to estimate the (logged) entries per capita over the period 2010-2017. The model you want to fit is

$$\begin{aligned} y_{st} &\sim N(\log \lambda_{st}, \sigma^2_{y,s}) \\ \log \lambda_{st} &= \alpha_k B_k(t) \\ \Delta^2 \alpha_k &\sim N(0, \sigma^2_{\alpha,s}) \\ \log \sigma_{\alpha,s} &\sim N(\mu_\sigma, \tau^2) \end{aligned}$$

Where $y_{s,t}$ is the logged entries per capita for state s in year t. Use cubic splines that have knots 2.5 years apart and are a constant shape at the boundaries. Put standard normal priors on standard deviations and hyperparameters.

```
years <- unique(d$year)
N <- length(years)
```

```
y <- log(d |>
  select(state, year, ent_pc) |>
  pivot_wider(names_from = "state", values_from = "ent_pc") |>
  select(-year) |>
  as.matrix())
res <- getsplines(years, 2.5)
B <- res$B.ik
K <- ncol(B)
stan_data <- list(N = N, y = y, K = K, S = length(unique(d$state)),
                  B = B)
mod <- stan(data = stan_data, file = "lab11.stan",
            refresh = 0,
            verbose = FALSE)
```

Question 3

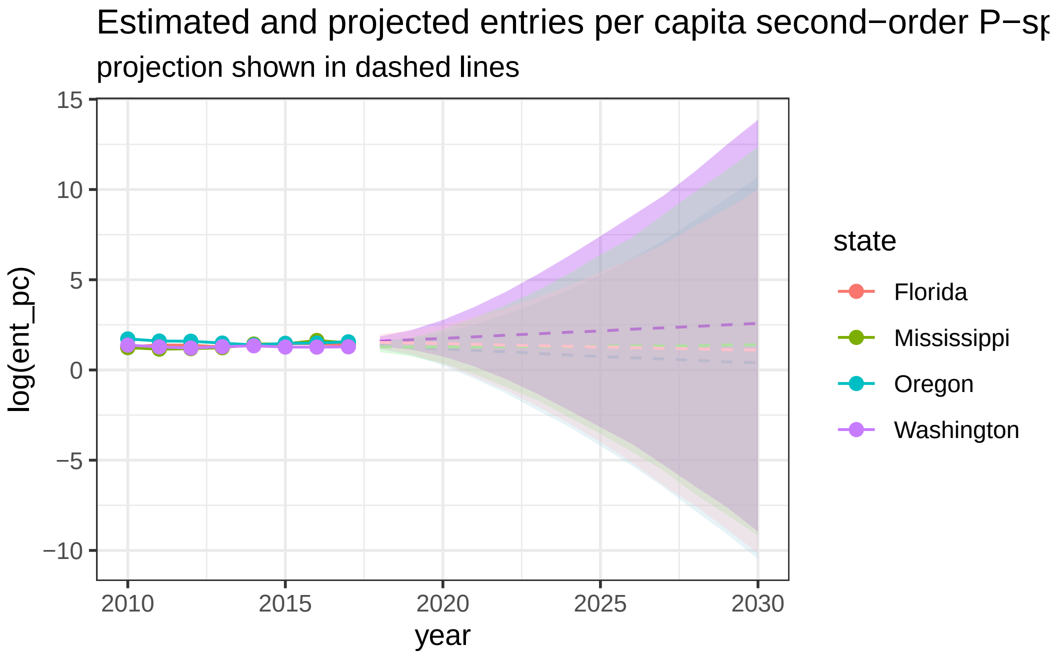
Project forward entries per capita to 2030. Pick 4 states and plot the results (with 95% CIs). Note the code to do this in R is in the lecture slides.

```
proj_years <- 2018:2030
# Note: B.ik are splines for in-sample period
# has dimensions i (number of years) x k (number of knots)
# need splines for whole period
B.ik_full <- getsplines(c(years, proj_years),I=2.5)$B.ik
K <- ncol(res$B.ik) # number of knots in sample
K_full <- ncol(B.ik_full) # number of knots over entire period
proj_steps <- K_full - K # number of projection steps
# get your posterior samples
alphas <- extract(mod)[["alpha"]]
sigmas <- extract(mod)[["sigma_alpha"]] # sigma_alpha
sigma_ys <- extract(mod)[["sigma_y"]]
nsims <- nrow(alphas)

# first, project the alphas
states <- unique(d$state)
alphas_proj <- array(NA, c(nsims, proj_steps, length(states)))
set.seed(1098)
# project the alphas
for(j in 1:length(states)){
  first_next_alpha <- rnorm(n = nsims,
                            mean = 2*alphas[,K,j]-alphas[,K-1,j],
                            sd = sigmas[,j])
  second_next_alpha <- rnorm(n=nsims,
                             mean = 2*first_next_alpha-alphas[,K,j],
                             sd = sigmas[,j])
  alphas_proj[,1,j] <- first_next_alpha
  alphas_proj[,2,j] <- second_next_alpha
  # now project the rest
  for(i in 3:proj_steps){ #!!! not over years but over knots
  alphas_proj[,i,j] <- rnorm(n = nsims,
                            mean = 2*alphas_proj[,i-1,j] - alphas_proj[,i-2,j],
                            sd = sigmas[,j])
  }
}
# now use these to get y's
y_proj <- array(NA, c(nsims, length(proj_years), length(states)))
for(i in 1:length(proj_years)){ # now over years
for(j in 1:length(states)){
all_alphas <- cbind(alphas[,,j], alphas_proj[,,j] )
this_lambda <- all_alphas %*% as.matrix(B.ik_full[length(years)+i, ])
y_proj[,i,j] <- rnorm(n = nsims, mean = this_lambda, sd = sigma_ys[,j])
  }
}
# then proceed as normal to get medians, quantiles etc
```

Next I will pick up 4 states to make a new graphs:

```
library(data.table)
set.seed(100)
N_states<-sample(length(unique(d$state)), 4)
y_projs <- y_proj[,,N_states]
state_names <- unique(d$state)[N_states]
new_state_names <- vector("list", length(state_names))
for (k in seq_along(state_names)) {
  new_state_names[[k]] <- paste0(unlist(as.name(state_names[k])),"_proj")

}

name <- list("var1", "var2", "var3","var4")

for (j in 1:4){
```

```r
name[[j]]<-as.name(new_state_names[[j]])
}

for (i in 1:4){
name[[i]]<- transpose(as.data.frame(y_projs[,,i]))
name[[i]]$median <- apply(name[[i]], 1, median)
name[[i]]$lower <- apply(name[[i]], 1, quantile, probs = c(0.025))
name[[i]]$upper <- apply(name[[i]], 1, quantile, probs = c(0.975))
name[[i]]$year=2018:2030

}


d |>
  filter(state %in% state_names) |>
  ggplot(aes(year, log(ent_pc)))+
  geom_line(aes(color=state), lty=1.5)+
  geom_point(aes(color=state),size=2)+
  geom_line(data=name[[1]], aes(x=year, y=median), linetype="dashed", color="lightblue")+
  geom_ribbon(data = name[[1]], aes(x=year, y = median, ymin = lower, ymax = upper), alpha
    geom_line(data=name[[2]], aes(x=year, y=median), linetype="dashed", color="purple")+
  geom_ribbon(data = name[[2]], aes(x=year, y = median, ymin = lower, ymax = upper), alpha
   geom_line(data=name[[3]], aes(x=year, y=median), linetype="dashed", color="lightgreen")
  geom_ribbon(data = name[[3]], aes(x=year, y = median, ymin = lower, ymax = upper), alpha
   geom_line(data=name[[4]], aes(x=year, y=median), linetype="dashed", color="pink")+
  geom_ribbon(data = name[[4]], aes(x=year, y = median, ymin = lower, ymax = upper), alpha
  labs(title = "Estimated and projected entries per capita second-order P-splines of 4 sta
       subtitle = "projection shown in dashed lines")+
  theme_bw(base_size=11)
```



Question 4 (bonus)

P-Splines are quite useful in structural time series models, when you are using a model of the form

$$f(y_t) = \text{systematic part} + \text{time-specific deviations}$$

where the systematic part is model with a set of covariates for example, and P-splines are used to smooth data-driven deviations over time. Consider adding covariates to the model you ran above. What are some potential issues that may happen in estimation? Can you think of an additional constraint to add to the model that would overcome these issues?

There are so many issues coused by adding covariates as following:

1) Overfitting: Too many covariates would overfit the model and made the model complexed.

2) Multicollinearity: There were some high correlationship among the added covariates.This could lead to numerical instability and difficulty in interpreting the coefficients of the model.

3) Misspecification: More added covariates could also mask the relationship between the covariates and the response variable.This would result in biased estimates and incorrect inference.

One way to solve the above problems is to introduce regularization through a penalty term on the coefficients of the P-spline basis functions. This can help address the 1) and 2) issues by shrinking the estimated coefficients towards zero.

Regularization can help to mitigate the issue of misspecification by providing some flexibility in the model without overfitting to the noise in the data as well.

Additionally, incorporating previous knowledge about the added covariates and their relationship with the response variable, such as through Bayesian priors, could aid for solving issues 3).