

Week 11: Splines

30/03/23

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

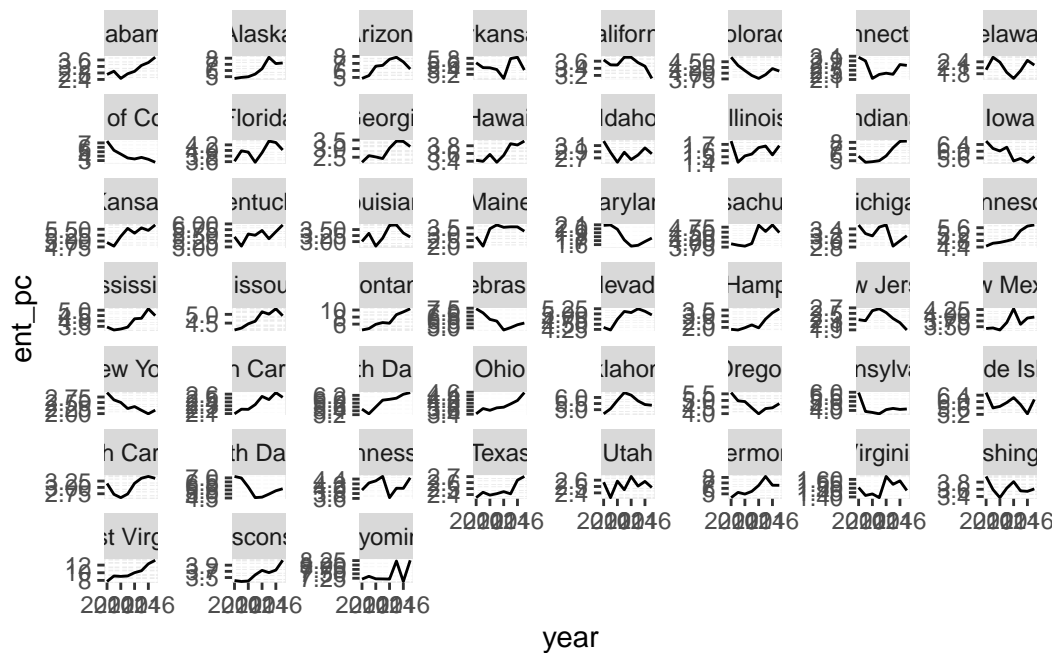
```
d <- read_csv(here("fc_entries.csv"))
```

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_wrap(~state, scales="free_y")
```



There are some states with an overall increasing trend and some states with an overall decreasing trend. There are also some states where the trend switches between increasing and decreasing, so the overall trend can be unclear.

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

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

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")

```

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),2.5)$B.ik
K <- ncol(B) # number of knots in sample
K_full <- ncol(B.ik_full) # number of knots over entire period
proj_steps <- K_full - K # number of projection steps
# get your posterior samples
alphas <- extract(mod)[["alpha"]]
sigmas <- extract(mod)[["sigma_alpha"]] # sigma_alpha
sigma_ys <- extract(mod)[["sigma_y"]]
nsims <- nrow(alphas)
# first, project the alphas
alphas_proj <- array(NA, c(nsims, proj_steps, length(unique(d$state))))
set.seed(1098)
# project the alphas
for(j in 1:length(unique(d$state))){
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])

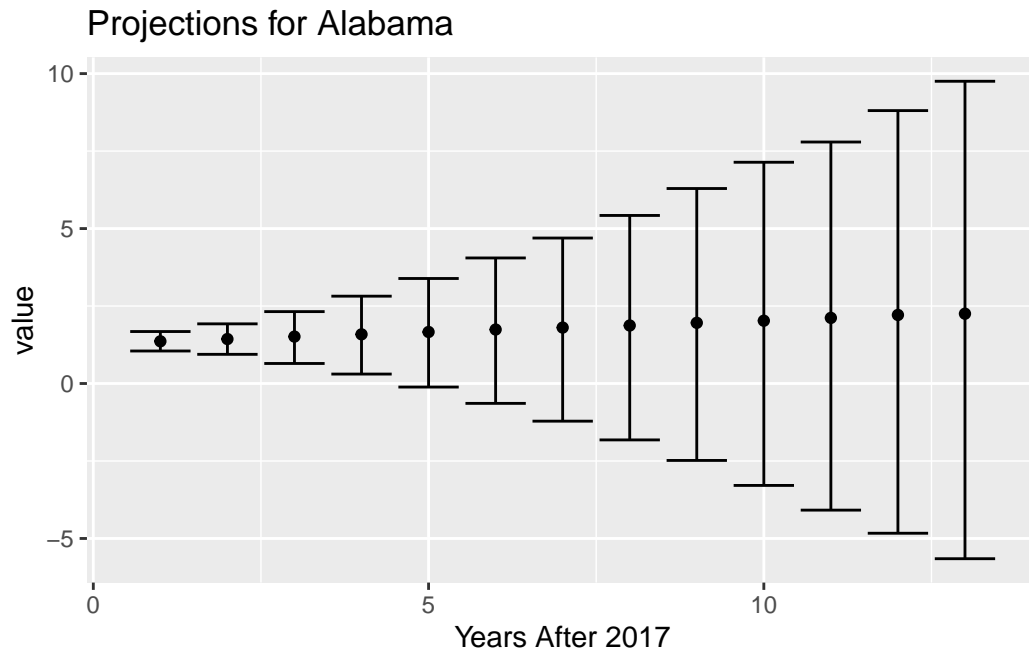
```

```

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(unique(d$state))))
for(i in 1:length(proj_years)){ # now over years
for(j in 1:length(unique(d$state))){
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])
}
}

res=as_tibble(y_proj[, ,1]) %>% median_qi()
value=res[seq_len(42)%%3==1]
value=value[1:13]%>% pivot_longer(V1:V13)
upper=res[seq_len(42)%%3==0]
upper=upper[1:13] %>% pivot_longer(V1.upper:V13.upper)
lower=res[seq_len(42)%%3==2]
lower=lower[1:13] %>% pivot_longer(V1.lower:V13.lower)
value %>% ggplot(aes(x=1:13,y=value))+geom_point()+
  geom_errorbar(aes(ymin = lower$value, ymax = upper$value))+
  ggtitle("Projections for Alabama")+
  xlab("Years After 2017")

```

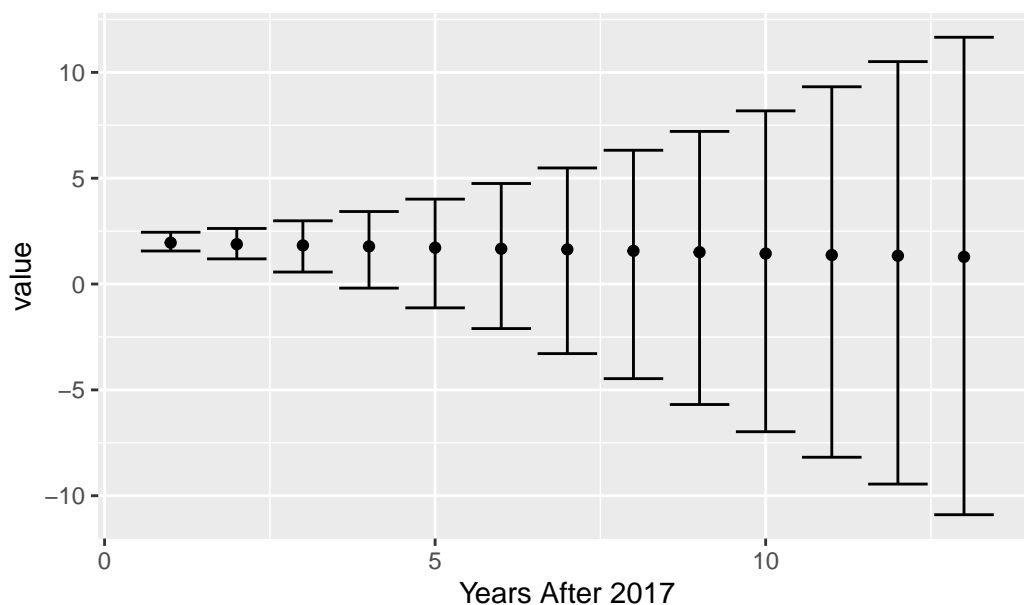


```

res2=as_tibble(y_proj[,2]) %>% median_qi()
value=res2[seq_len(42)%%3==1]
value=value[1:13]%>% pivot_longer(V1:V13)
upper=res2[seq_len(42)%%3==0]
upper=upper[1:13] %>% pivot_longer(V1.upper:V13.upper)
lower=res2[seq_len(42)%%3==2]
lower=lower[1:13] %>% pivot_longer(V1.lower:V13.lower)
value %>% ggplot(aes(x=1:13,y=value))+geom_point()+
  geom_errorbar(aes(ymin = lower$value, ymax = upper$value))+
  ggtitle("Projections for Alaska")+
  xlab("Years After 2017")

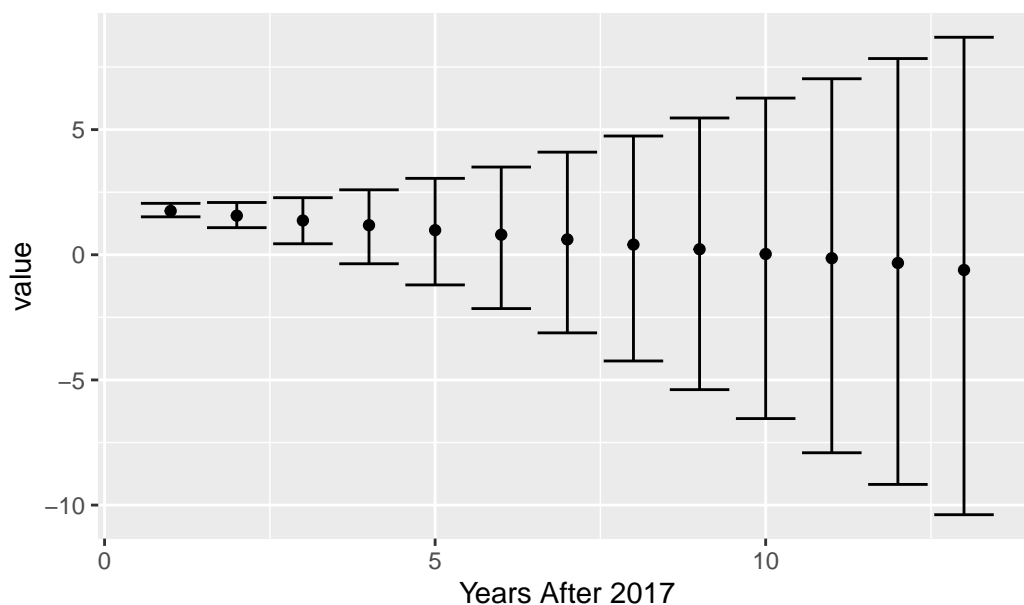
```

Projections for Alaska



```
res3=as_tibble(y_proj[, ,3]) %>% median_qi()
value=res3[seq_len(42)%%3==1]
value=value[1:13]%>% pivot_longer(V1:V13)
upper=res3[seq_len(42)%%3==0]
upper=upper[1:13] %>% pivot_longer(V1.upper:V13.upper)
lower=res3[seq_len(42)%%3==2]
lower=lower[1:13] %>% pivot_longer(V1.lower:V13.lower)
value %>% ggplot(aes(x=1:13,y=value))+geom_point()+
  geom_errorbar(aes(ymin = lower$value, ymax = upper$value))+
  ggtitle("Projections for Arizona")+
  xlab("Years After 2017")
```

Projections for Arizona



```
res4=as_tibble(y_proj[,4]) %>% median_qi()
value=res4[seq_len(42)%%3==1]
value=value[1:13]%>% pivot_longer(V1:V13)
upper=res4[seq_len(42)%%3==0]
upper=upper[1:13] %>% pivot_longer(V1.upper:V13.upper)
lower=res4[seq_len(42)%%3==2]
lower=lower[1:13] %>% pivot_longer(V1.lower:V13.lower)
value %>% ggplot(aes(x=1:13,y=value))+geom_point()+
  geom_errorbar(aes(ymin = lower$value, ymax = upper$value))+
  ggtitle("Projections for Arkansas")+
  xlab("Years After 2017")
```

Projections for Arkansas

