Bayesian mortality estimation with principal component models

Monica Alexander

Table of contents

1 Overview

This lab illustrates how to fit Bayesian hierarchical 'principal component' regression models to estimate mortality in a context where geographic areas are small and/or deaths are rare. The example is estimating underlying age-specific mortality curves for Californian counties in 2022.

Let's load in the packages we'll need for this lab:

```
library(rstan)
library(tidyverse)
library(tidybayes)
library(janitor)
```

2 Read in data

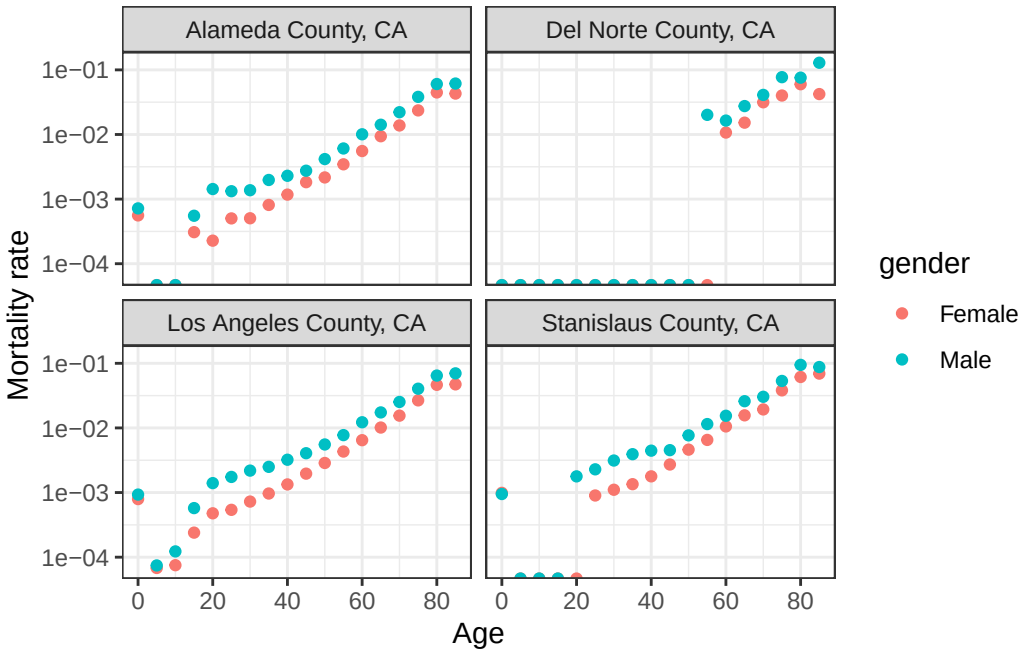
2.1 Californian counties

Firstly, let's read in the observe death counts and population counts by age and sex for Californian counties in 2022. The deaths data were sourced from CDC Wonder, and the population counts came from the US Census Bureau.

```
d_ca <- read_csv("../data/CA_mortality.csv")
```

Select a few different counties and plot. Notice the difference in data availability: most counties have at least some of the age group death counts missing (too low to report). In contrast, LA county has observations at each age (note that LA county has a population of around 10 million!)

```
d_ca |>
  mutate(rate = deaths/pop) |>
  filter(county %in% c("Alameda County, CA", "Los Angeles County, CA",
                       "Stanislaus County, CA", "Del Norte County, CA")) |>
  ggplot(aes(age, rate, color = gender)) +
  geom_point()+
  scale_y_log10()+
  facet_wrap(~county)+
  theme_bw()+
  labs("Mortality rates for four Californian counties, 2022",
       y = "Mortality rate", x = "Age")
```



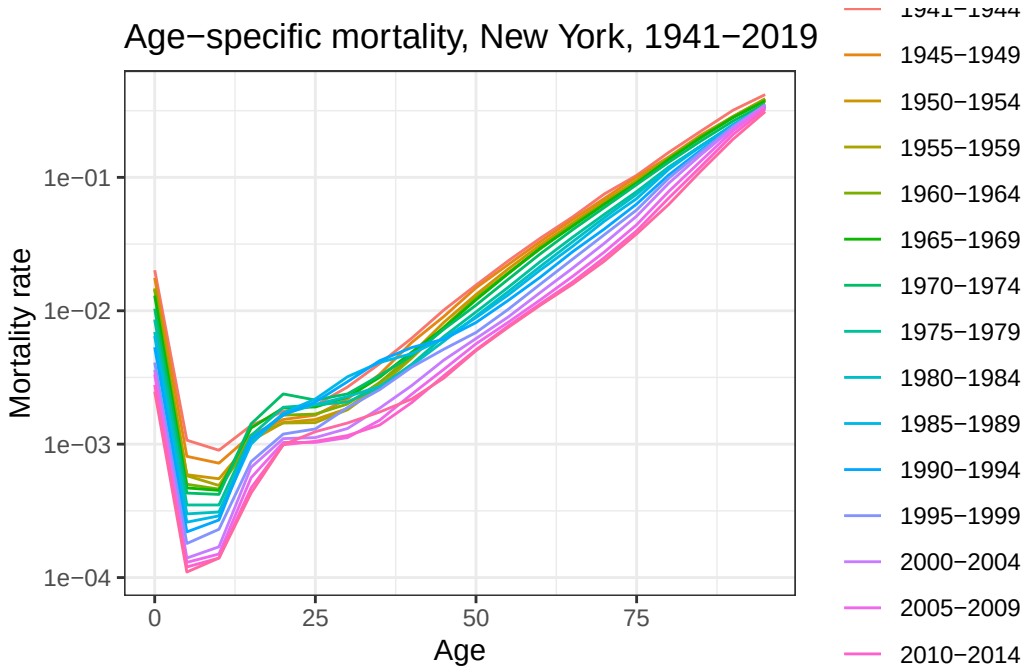
2.2 US state estimates to extract principal components

Now let's read in some US state-level data that will be used to construct principal components. These data come from the US states mortality database, which is a free resource containing life table estimates for all US states, regions, and divisions.

```
lt <- read_csv("../data/lt.csv")
```

Let's plot the male mortality curves for New York over time:

```
lt |>
  filter(pop_name=="NY") |>
  filter(gender == "Male") |>
  ggplot(aes(age, mx, color = year))+
  geom_line()+
  scale_y_log10()+
  labs(title = "Age-specific mortality, New York, 1941-2019",
       y = "Mortality rate", x = "Age")+
  theme_bw()
```



2.3 Calculate the principal components

Let's restrict to male mortality only:

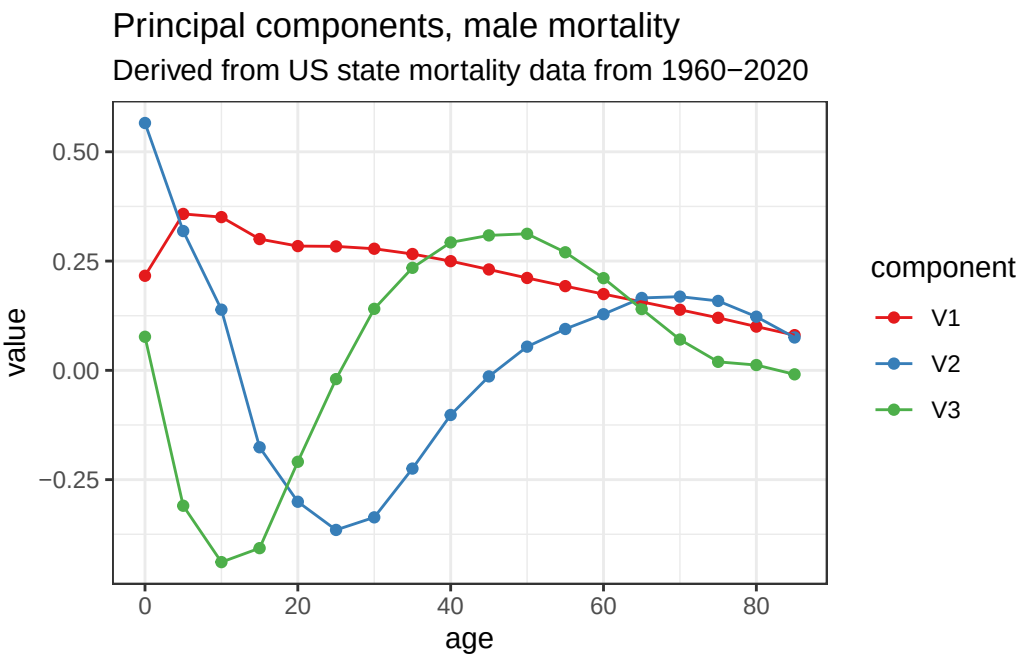
```
lt_male <- lt |>
  filter(gender=="Male")
```

Do a singular value decomposition on the logged male mortality rates over time for all states, and pull out the first 3 right singular vectors.

```
m_ga <- lt_male |>
  select(-gender) |>
  filter(age<90) |>
  pivot_wider(names_from = "age", values_from = "mx") |>
  select(-year, -pop_name) |>
  as.matrix()

log_m_ga <- log(m_ga)
ages <- seq(0, 85, by = 5)

pcs <- svd(log_m_ga)$v[,1:3]
```

Let's plot the first three components to see what they look like. The first shows the classic 'J' shape, the second allows for higher infant and old age mortality, and the third picks up the 'accident hump'.

```
pcs <- as_tibble(pcs)

pcs <- pcs |>
  mutate (age = ages)

pcs |>
  pivot_longer(-age) |>
  ggplot(aes(age, value, color = name))+
  geom_point()+
  geom_line()+
  theme_bw()+
  labs(title = "Principal components, male mortality",
       subtitle = "Derived from US state mortality data from 1960-2020")+
  scale_color_brewer(name = "component", palette = "Set1")
```



3 Fit hierarchical principal components regression model

We're going to use the components above as the basis of a Bayesian hierarchical regression model to estimate male mortality rates for all Californian counties in 2022.

3.1 Model description

The model is:

$$y_i|\mu_{a[i],c[i]} \sim \text{Poisson}(\mu_{a[i],c[i]} \cdot P_i)$$

where y_i is the number of deaths observed for observation $i = 1, \ldots, n$, P_i is the population size, and $\mu_{a,c}$ is the mortality rate for age group a and county c. We will further assume the log mortality rates $\mu_{a,c}$ are modeled as:

$$\log \mu_{a,c} = \beta_{1,c}X_{1,a} + \beta_{2,c}X_{2,a} + \beta_{3,c}X_{3,a} + \varepsilon_{a,c}$$

where $X_{p,a}$ refers to the value of the pth principal component (derived above) for age a. The principal components are modeled hierarchically, for example:

$$\beta_{1,c} \sim N(\mu_{\beta_1}, \sigma^2_{\beta_1})$$

The over dispersion term $\varepsilon_{a,c}$ is modeled allowing for a different variance at each age, which accounts for the fact that we usually have more uncertainty at younger ages:

$$\varepsilon_{a,c} \sim N(0, \sigma^2_a)$$

See the Stan model for full specification.

3.2 Run the model

Getting the data in Stan format:

```
d_ca_male <- d_ca |>
  filter(gender == "Male") |>
  mutate(deaths = ifelse(deaths==0, NA, deaths))

d_ca_male_non_missing <- d_ca_male |>
  drop_na(deaths)

counties <- unique(d_ca_male_non_missing$county)
y <- d_ca_male_non_missing$deaths
pop <- d_ca_male_non_missing$pop
age <- match(d_ca_male_non_missing$age, ages)
```

```
county <- match(d_ca_male_non_missing$county, counties)
X1 <- pcs$V1
X2 <- pcs$V3
X3 <- pcs$V3
X4 <- pcs$V4

stan_data <- list(N = length(y),
                  J = length(counties),
                  A = length(ages),
                  y = y,
                  pop = pop,
                  county = county,
                  age = age,
                  X1 = X1,
                  X2 = X2,
                  X3 = X3)
```

Code to run the model and save the output is below. Note that I've set this code chunk to not evaluate because it takes a while to run.

```
mod <- stan(data = stan_data,
            file = "../models/svd.stan",
            seed = 95,
            iter = 4000,
            thin = 4,
            control = list(adapt_delta = 0.99))
summary(mod)$summary[which(summary(mod)$summary[, "Rhat"]>1.05),]
write_rds(mod, "../output/svd_mod.rds")
```

3.3 Look at results

Let's read in the output from above and calculate the median estimates and 95% uncertainty bounds for the estimated (log) mortality rates.

```
mod <- read_rds("../output/svd_mod.rds")
res_log_mu <- mod |>
  gather_draws(log_mu[a,c]) |>
  median_qi() |>
  mutate(age = ages[a]) |>
  mutate(county = counties[c])
```

And plot the estimates for few selected counties from the start of the lab. Notice the difference in uncertainty based on size of the county and data availability.

```
res_log_mu |>
  left_join(d_ca_male_non_missing) |>
  mutate(log_mx_obs = log(deaths/pop)) |>
  filter(county %in% c("Alameda County, CA", "Los Angeles County, CA",
                       "Stanislaus County, CA", "Del Norte County, CA")) |>
  ggplot(aes(age, .value))+
  geom_line(aes(color = "fit"))+
  geom_point(aes(age, log_mx_obs, color = "data", fill = "data"))+
  facet_wrap(~county)+
  geom_ribbon(aes(age, ymin = .lower, ymax = .upper, fill = "fit"), alpha = 0.2)+
  scale_color_brewer(name = "", palette = "Set1")+
  scale_fill_brewer(name = "", palette = "Set1")+
  theme_bw()+
  labs(title = "Data and fitted male mortality rates for four Californian counties, 2022",
       x = "Age", y = "Log mortality rate")
```