BDA project report

Amanda Aarnio, Anni Niskanen, Antti Huttunen

2022-11-24

Introduction

```
set.seed(123)
library(cmdstanr)
library(Stat2Data)
data("FirstYearGPA")
```

Data and Problem

```
male_white <- FirstYearGPA[FirstYearGPA$Male==1 & FirstYearGPA$White==1,]
male_non_white <- FirstYearGPA[FirstYearGPA$Male==1 & FirstYearGPA$White==0,]
female_white <- FirstYearGPA[FirstYearGPA$Male==0 & FirstYearGPA$White==1,]
female_non_white <- FirstYearGPA[FirstYearGPA$Male==0 & FirstYearGPA$White==0,]
```

```
data_hierarchical <- list(N1 = nrow(male_white),
                          N2 = nrow(male_non_white),
                          N3 = nrow(female_white),
                          N4 = nrow(female_non_white),
                          x1 = subset(male_white, select = c('HSGPA', 'SATV', 'SATM','HU','SS')),
                          x2 = subset(male_non_white, select = c('HSGPA', 'SATV', 'SATM','HU','SS')),
                          x3 = subset(female_white, select = c('HSGPA', 'SATV', 'SATM','HU','SS')),
                          x4 = subset(female_non_white, select = c('HSGPA', 'SATV', 'SATM','HU','SS')),
                          y1 = male_white$GPA,
                          y2 = male_non_white$GPA,
                          y3 = female_white$GPA,
                          y4 = female_non_white$GPA)
data_pooled <- list(N = nrow(FirstYearGPA),
                    x = subset(FirstYearGPA, select = c('HSGPA', 'SATV', 'SATM','HU','SS')),
                    y = FirstYearGPA$GPA)
```

Models

Pooled model

Mathematical notation

$$
\begin{aligned}
GPA_i &\sim N(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 \cdot HSGPA_i + \beta_2 \cdot SATV_i + \beta_3 \cdot SATM_i + \beta_4 \cdot HU_i + \beta_5 \cdot SS_i \\
\sigma &\sim N(0, 10) \\
\alpha &\sim N(0, 100) \\
\beta_k &\sim N(0, 100)
\end{aligned}
$$

```
writeLines(readLines("pooled.stan"))
```

```
## // Pooled model.
## // Variables: HSGPA, SATM, SATV, HU, SS
## data {
##   int<lower=0> N;
##   matrix[N,5] x;
##   vector[N] y;
## }
##
## parameters {
##   real alpha;
##   vector[5] betas;
##   real<lower=0> sigma;
## }
##
## transformed parameters {
##   vector[N] mu;
##   mu = alpha + betas[1]*x[,1] + betas[2]*x[,2] + betas[3]*x[,3] + betas[4]*x[,4] + betas[5]*x[,5];
##   /*mu += alpha;
##   for (i in 1:5)
##     mu += betas[i]*x[,i];*/
## }
##
## model {
##   // priors
##   alpha ~ normal(0, 100);
##   betas ~ normal(0, 100);
##   sigma ~ normal(0, 10);
##
##   // likelihood
##   y ~ normal(mu , sigma);
## }
```

```
mod_pooled <- cmdstan_model("pooled.stan")
fit_pooled <- mod_pooled$sample(data_pooled, refresh = 2000)
```

```
## Running MCMC with 4 sequential chains...
##
## Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
```

```
## Chain 1 finished in 19.3 seconds.
## Chain 2 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2 finished in 19.6 seconds.
## Chain 3 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3 finished in 17.1 seconds.
## Chain 4 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4 finished in 20.9 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 19.2 seconds.
## Total execution time: 77.4 seconds.
```

Hierarchical model

$$
\begin{aligned}
GPA_{ij} &\sim N(\mu_{ij}, \sigma) \\
\mu_{ij} &= \alpha_j + \beta_{1j} \cdot HSGPA_i + \beta_{2j} \cdot SATV_i + \beta_{3j} \cdot SATM_i + \beta_{4j} \cdot HU_i + \beta_{5j} \cdot SS_i \\
\sigma &\sim N(0, 10) \\
\alpha_j &\sim N(\mu_\alpha, \sigma_\alpha) \\
\beta_{1j} &\sim N(\mu_{\beta_1}, \sigma_{\beta_1}) \\
\beta_{2j} &\sim N(\mu_{\beta_2}, \sigma_{\beta_2}) \\
\beta_{3j} &\sim N(\mu_{\beta_3}, \sigma_{\beta_3}) \\
\beta_{4j} &\sim N(\mu_{\beta_4}, \sigma_{\beta_4}) \\
\beta_{5j} &\sim N(\mu_{\beta_5}, \sigma_{\beta_5}) \\
\mu_\alpha &\sim N(0, 100) \\
\sigma_\alpha &\sim N(0, 10) \\
\mu_{\beta_k} &\sim N(0, 100) \\
\sigma_{\beta_k} &\sim N(0, 10)
\end{aligned}
$$

```
writeLines(readLines("hierarchical.stan"))
```

```
## // Hierarchical model.
## // Betas in following order: HSGPA, SATM, SATV, HU, SS
## // Alpha: intercept
## data {
##   int<lower=0> N1;
##   int<lower=0> N2;
##   int<lower=0> N3;
##   int<lower=0> N4;
##   matrix[N1,5] x1;
##   matrix[N2,5] x2;
##   matrix[N3,5] x3;
##   matrix[N4,5] x4;
##   vector[N1] y1;
##   vector[N2] y2;
##   vector[N3] y3;
##   vector[N4] y4;
```

```
## }
##
## parameters {
##   // parameters
##   real alpha1;
##   real alpha2;
##   real alpha3;
##   real alpha4;
##   vector[5] betas1;
##   vector[5] betas2;
##   vector[5] betas3;
##   vector[5] betas4;
##   real<lower=0> sigma;
##
##   // hyperparameters
##   real pmualpha;
##   real<lower=0> psalpha;
##   vector[5] pmubetas;
##   vector<lower=0>[5] psbetas;
## }
##
## transformed parameters {
##   vector[N1] mu1 = alpha1 + betas1[1]*x1[,1] + betas1[2]*x1[,2] + betas1[3]*x1[,3] + betas1[4]*x1[,4]
##   vector[N2] mu2 = alpha2 + betas2[1]*x2[,1] + betas2[2]*x2[,2] + betas2[3]*x2[,3] + betas2[4]*x2[,4]
##   vector[N3] mu3 = alpha3 + betas3[1]*x3[,1] + betas3[2]*x3[,2] + betas3[3]*x3[,3] + betas3[4]*x3[,4]
##   vector[N4] mu4 = alpha4 + betas4[1]*x4[,1] + betas4[2]*x4[,2] + betas4[3]*x4[,3] + betas4[4]*x4[,4]
## }
##
## model {
##   // hyperpriors
##   pmualpha ~ normal(0, 100);
##   psalpha ~ normal(0, 10);
##   for (i in 1:5){
##     pmubetas[i] ~ normal(0, 100);
##     psbetas[i] ~ normal(0, 10);
##   }
##
##   // priors
##   alpha1 ~ normal(pmualpha, psalpha);
##   alpha2 ~ normal(pmualpha, psalpha);
##   alpha3 ~ normal(pmualpha, psalpha);
##   alpha4 ~ normal(pmualpha, psalpha);
##   betas1 ~ normal(pmubetas, psbetas);
##   betas2 ~ normal(pmubetas, psbetas);
##   betas3 ~ normal(pmubetas, psbetas);
##   betas4 ~ normal(pmubetas, psbetas);
##   sigma ~ normal(0, 10);
##
##   // likelihoods
##   y1 ~ normal(mu1, sigma);
##   y2 ~ normal(mu2, sigma);
##   y3 ~ normal(mu3, sigma);
##   y4 ~ normal(mu4, sigma);
## }
```

```r
mod_hierarchical <- cmdstan_model("hierarchical.stan")
fit_hierarchical <-mod_hierarchical$sample(data_hierarchical, refresh = 2000)
```

```
## Running MCMC with 4 sequential chains...
##
## Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1 finished in 37.0 seconds.
## Chain 2 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2 finished in 38.1 seconds.
## Chain 3 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3 finished in 35.6 seconds.
## Chain 4 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4 finished in 56.5 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 41.8 seconds.
## Total execution time: 167.6 seconds.
```

Analysis and Results

Converge diagnostics

Posterior predictive checks

Prior sensitivity analysis

Discussion

Conclusion

Self-reflection