BDA project report

Amanda Aarnio, Anni Niskanen, Antti Huttunen

2022-12-02

Contents

1 Introduction

Can the grade point averages (GPAs) of first-year college students be predicted based on high school GPAs, SAT scores and other factors? This report presents a solution to this problem by applying two statistical models to predict the first-year college GPA. These models are fitted in Stan and compared according to Bayesian data analysis concepts.

The report begins with a description of the data and the problem. After this, the models are introduced and their performances are analysed. Model improvements are then discussed and conclusions are made. Finally, self-reflection of the project is presented.

2 Data and Problem

The data contains information from a sample of 219 first year students at a Midwestern college in 1996. The data has 10 variables:

```
GPA          First-year college GPA on a 0.0 to 4.0 scale
HSGPA        High school GPA on a 0.0 to 4.0 scale
SATV         Verbal/critical reading SAT score
SATM         Math SAT score
Male         1= male, 0= female
HU           Number of credit hours earned in humanities courses in high school
SS           Number of credit hours earned in social science courses in high
             school
FirstGen     1= student is the first in her or his family to attend college,
             0=otherwise
White        1= white students, 0= others
CollegeBound 1=attended a high school where >=50% students intended to go on to
             college, 0=otherwise
```

The data can be obtained from: https://vincentarelbundock.github.io/Rdatasets/doc/Stat2Data/FirstYearGPA.html

This report solves the problem of predicting the GPA of first-year college students using this data. Our models use all numerical values (HSGPA, SATV, SATM, HU and SS) to predict the first-year college GPA (GPA). In addition, the hierarchical model uses categorical variables Male and White to group the data into four different groups and predicts the GPA with group-level predictors. The barplot below represents these four groups and their sizes.

```r
library(cmdstanr)
library(bayesplot)
library(ggplot2)
library(gridExtra)
library(loo)
library(Stat2Data)

data("FirstYearGPA")
```

```r
White <- FirstYearGPA$White
White <- gsub(1, "White", White)
White <- gsub(0, "Non-White", White)
Male <- FirstYearGPA$Male
Male <- gsub(1, "Male", Male)
Male <- gsub(0, "Female", Male)
counts <- data.frame(White=White, Male=Male)
ggplot(data = counts, aes(x = Male, fill=White)) + geom_bar() +
  xlab("Gender") + scale_fill_brewer(palette = "Paired", name = "")
```

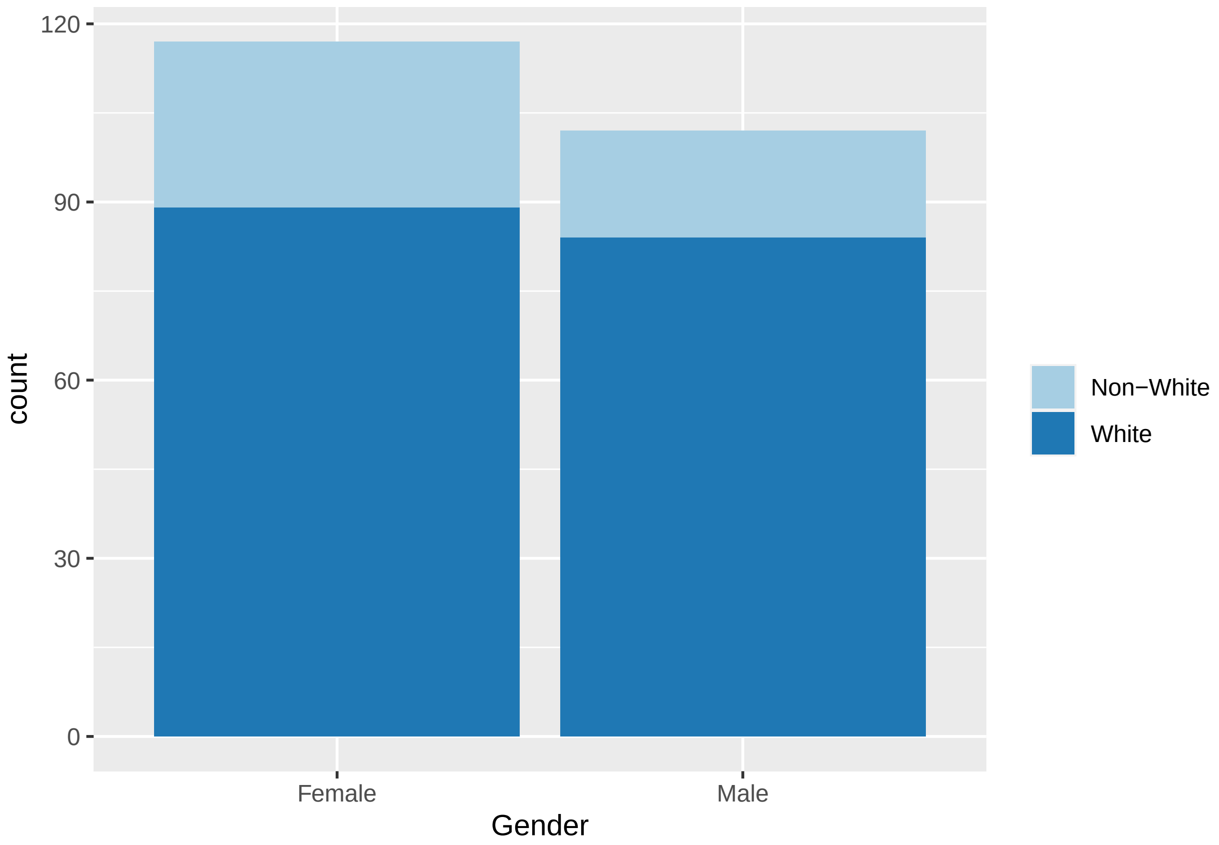


Figure 1: The division of the data in four different groups. The amounts of the data points in the groups can be seen in bars.

3 Models

In this project, two different models are utilised: pooled and hierarchical model. Both are are covered in following sections. The mathematical notations, Stan implementations, and stan model runs are included in the sections. The both of the models follow the linear Gaussian model whose expected values are constructed using linear function with variables found in data (HSGPA, SATV, SATM, HU, and SS) and parameters α, β_1, β_2, β_3, β_4, and β_5.

Weakly informative normal priors were used in both models. N(0,10) was the prior distribution for σ in both models. In the pooled model, N(0,100) was the prior for both α and all β_k. The hierarchical model had similar priors: N(0,100) for all means μ_α and μ_{β_k}, and N(0,10) for all standard deviations σ_α and σ_{β_k}. We thought these priors reasonable for multiple reasons. Firstly, they all center on 0, which was thought wisest as we have no information whether the intercept α or the slopes β_k should be positive or negative. Secondly, the standard deviations, 100 for the means and 10 for the standard deviations, were considered large enough to produce wide enough (but not too wide) prior distributions.

As can be seen from the above code lines, default values were used for running the MCMC chains. That is, 4 chains of 2000 iterations were run, and the first 1000 iterations of each chain were considered warm-up.

3.1 Pooled model

In pooled model, all the expected values are constructed using the common parameters α and βs with weakly informative priors. All the GPAs have common σ.

Mathematical notation

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

Stan code

```
data {
  int<lower=0> N;
  matrix[N,5] x;
  vector[N] y;
  real musigma;
  real sigmasigma;
}

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];
}

model {
  // priors
  alpha ~ normal(0, musigma);
  betas ~ normal(0, musigma);
  sigma ~ normal(0, sigmasigma);

  // likelihood
  y ~ normal(mu, sigma);
}

generated quantities {
  vector[N] ypred;
  vector[N] log_lik;

  // Generate predictive distributions for GPA
  for (i in 1:N)
    ypred[i] = normal_rng(mu[i], sigma);

  // log likelihoods
  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu[i], sigma);
}
```

Running the model

```
data_pooled <- list(N = nrow(FirstYearGPA),
                    x = subset(FirstYearGPA, select = c('HSGPA', 'SATV', 'SATM',
                                                        'HU','SS')),
                    y = FirstYearGPA$GPA,
                    musigma = 100,
                    sigmasigma = 10)
```

```
mod_pooled <- cmdstan_model("pooled.stan")
fit_pooled <- mod_pooled$sample(data_pooled, refresh = 0, seed = 03091900)
```

```
## Running MCMC with 4 sequential chains...
##
## Chain 1 finished in 15.7 seconds.
## Chain 2 finished in 16.4 seconds.
## Chain 3 finished in 17.7 seconds.
## Chain 4 finished in 14.8 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 16.2 seconds.
## Total execution time: 65.1 seconds.
```

3.2 Hierarchical model

In Hierarchical model, students have been divided in four different groups: white males, white females, non white males, and non white females. All students in all the groups have own parameters αs and βs which are constructed using common hyperparameters for all the different groups.

Mathematical notation

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

Stan code

```
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;
  real musigma;
  real sigmasigma;
}

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] + betas1[5]*x1[,5];
  vector[N2] mu2 = alpha2 + betas2[1]*x2[,1] + betas2[2]*x2[,2]
                   + betas2[3]*x2[,3] + betas2[4]*x2[,4] + betas2[5]*x2[,5];
  vector[N3] mu3 = alpha3 + betas3[1]*x3[,1] + betas3[2]*x3[,2]
                   + betas3[3]*x3[,3] + betas3[4]*x3[,4] + betas3[5]*x3[,5];
  vector[N4] mu4 = alpha4 + betas4[1]*x4[,1] + betas4[2]*x4[,2]
                   + betas4[3]*x4[,3] + betas4[4]*x4[,4] + betas4[5]*x4[,5];
}

model {
  // hyperpriors
  pmualpha ~ normal(0, musigma);
  psalpha ~ normal(0, sigmasigma);
  for (i in 1:5){
    pmubetas[i] ~ normal(0, musigma);
    psbetas[i] ~ normal(0, sigmasigma);
  }
  // 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, sigmasigma);

  // likelihoods
  y1 ~ normal(mu1, sigma);
  y2 ~ normal(mu2, sigma);
  y3 ~ normal(mu3, sigma);
  y4 ~ normal(mu4, sigma);
}

generated quantities{
  vector[N1] ypred1;
  vector[N2] ypred2;
  vector[N3] ypred3;
  vector[N4] ypred4;
  vector[N1+N2+N3+N4] log_lik;

  // Generate predictive distributions for GPA
  for (i in 1:N1)
    ypred1[i] = normal_rng(mu1[i], sigma);
  for (i in 1:N2)
    ypred2[i] = normal_rng(mu2[i], sigma);
  for (i in 1:N3)
    ypred3[i] = normal_rng(mu3[i], sigma);
  for (i in 1:N4)
    ypred4[i] = normal_rng(mu4[i], sigma);

  // log likelihoods
  for (i in 1:N1)
    log_lik[i] = normal_lpdf(y1[i] | mu1[i], sigma);
  for (i in 1:N2)
    log_lik[N1+i] = normal_lpdf(y2[i] | mu2[i], sigma);
  for (i in 1:N3)
    log_lik[N1+N2+i] = normal_lpdf(y3[i] | mu3[i], sigma);
  for (i in 1:N4)
    log_lik[N1+N2+N3+i] = normal_lpdf(y4[i] | mu4[i], sigma);
}
```

Running the model

```
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,
                            musigma = 100,
                            sigmasigma = 10)
```

```
mod_hierarchical <- cmdstan_model("hierarchical.stan")
fit_hierarchical <-mod_hierarchical$sample(data_hierarchical, refresh = 0,
                                           seed = 03091900)
```

```
## Running MCMC with 4 sequential chains...
##
## Chain 1 finished in 30.1 seconds.
## Chain 2 finished in 29.2 seconds.
## Chain 3 finished in 33.3 seconds.
## Chain 4 finished in 35.0 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 31.9 seconds.
## Total execution time: 128.2 seconds.

## Warning: 218 of 4000 (5.0%) transitions ended with a divergence.
## See https://mc-stan.org/misc/warnings for details.

## Warning: 4 of 4000 (0.0%) transitions hit the maximum treedepth limit of 10.
## See https://mc-stan.org/misc/warnings for details.
```

4 Analysis and Results

This section contains the analysis of the both models. The analysis contains convergence diagnostics, posterior predictive checks, and prior sensitivity analysis. This section also contains discussion about posterior distributions and model comparison in above order.

4.1 Converge diagnostics

4.1.1 Pooled model

```
summary_pooled <- fit_pooled$summary()
sum(summary_pooled$rhat > 1.01)
sum(summary_pooled$rhat < 0.99)
```

```
## [1] 0
## [1] 0
```

```
mean(summary_pooled$ess_tail)
sum(summary_pooled$ess_tail<400)
mean(summary_pooled$ess_bulk)
sum(summary_pooled$ess_bulk<400)
```

```
## [1] 3171.745
## [1] 0
## [1] 3207.508
## [1] 0
```

None of the $\hat{R}$-values for the pooled model exceeds 1.01 which indicates that there are no problems with the convergence of the set of simulated chains. The effective sample size (ESS) is over 400 (recommended threshold with four parallel chains, 100 per chain) for all of the ess_bulk and ess_tail values which indicates that the $\hat{R}$-estimate can be trusted to make decisions about convergence and the quality of the chains. In addition, the model does not observe divergence.

4.1.2 Hierarchical model

```
summary_hierarchical <- fit_hierarchical$summary()
sum(summary_hierarchical$rhat > 1.01)
sum(summary_hierarchical$rhat < 0.99)
```

```
## [1] 0
## [1] 0
```

```
mean(summary_hierarchical$ess_tail)
sum(summary_hierarchical$ess_tail<400)
mean(summary_hierarchical$ess_bulk)
sum(summary_hierarchical$ess_bulk<400)
```

```
## [1] 2984.828
## [1] 2
## [1] 2725.984
## [1] 4
```

Similarly to the pooled model, in the hierarchical model, no $\hat{R}$-values exceed 1.01, indicating that convergence is not a problem. A few ESS values are below 400 and the model gives a divergence of approximately 5%. These facts can indicate that there might be some problems with the convergence. On the other hand, these values do not differ significantly from the desired values the probability of having problems with convergence is small.

4.2 Posterior predictive checks

48 histograms of replications of the data (created based on the posterior predictive distributions) are shown for all the models. In addition, there is overlayed distributions and empirical cumulative distribution (ecdf) functions of 100 replications in the two with the original distribution and ecdf. These different plots of replication are compared to the plots of the original data to investigate, how good predictions fitted model can make. The closer the replications are original data, the better the predictions are.

4.2.1 Pooled model

As it can be seen in Figure 2, the histograms and overlayed distribution functions have almost same width as the original data, and they follow quite nicely the shape of the original data. This means that the replications are good, and follows well original data. Moreover, replicated ecdf functions follow the shape of the ecdf of the original data, and the variation around the original ecdf function is quite low. Finally, the model is good in posterior prediction.

```
y <- FirstYearGPA$GPA
ypred <- fit_pooled$draws("ypred", format = "matrix")

grid.arrange( ppc_hist(y, ypred[1:48,]),
              ppc_dens_overlay(y, ypred[1:100,]),
              ppc_ecdf_overlay(y, ypred[1:100,]),
              layout_matrix = matrix(c(1,2,1,3), nrow = 2), nrow=2)
```

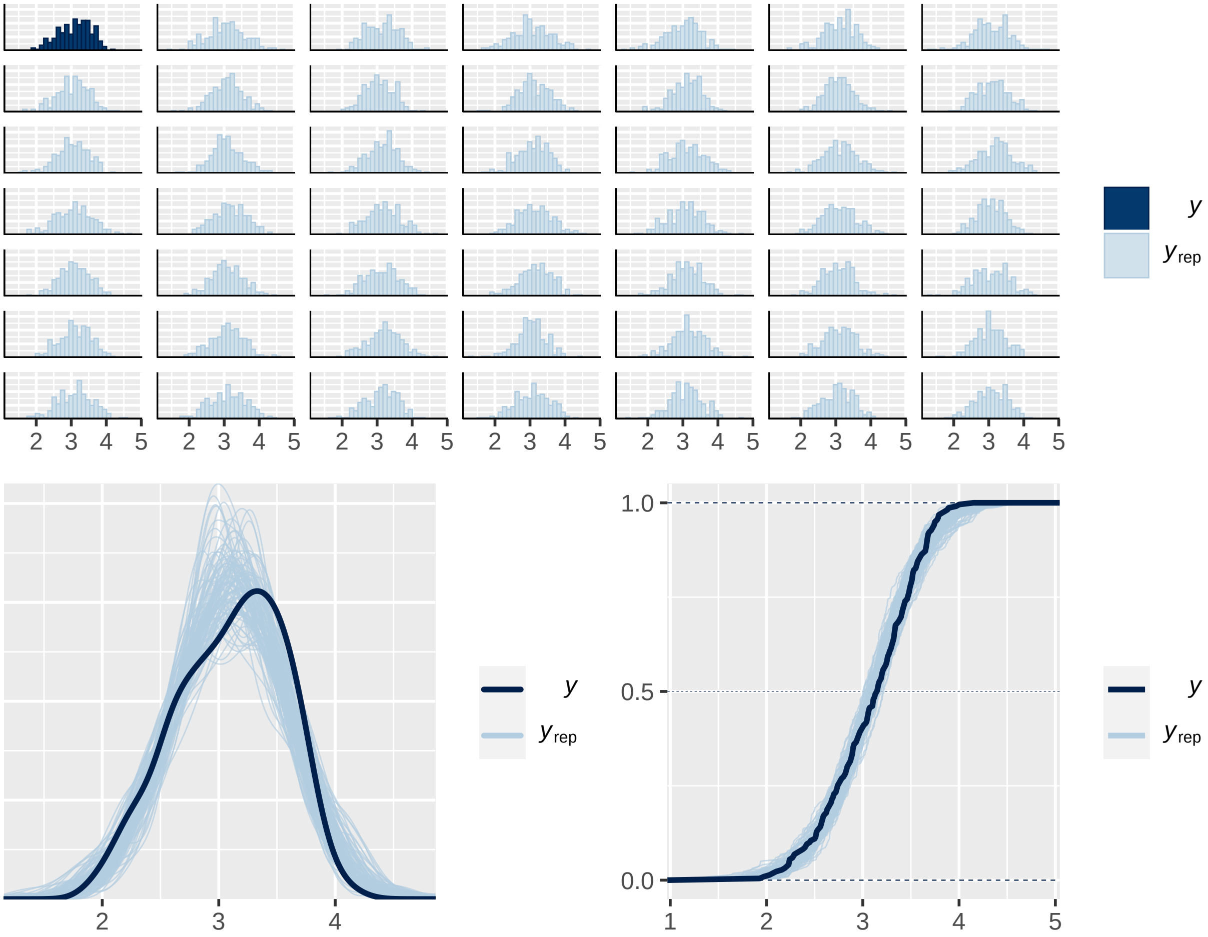


Figure 2: On the top of the figure 48 replications of the pooled model are shown using histograms. In addition the original data is shown in left top corner. Below the histograms, the 100 distributions of the replications are over layed with original data on left. On right 100 ecdf are over layed with original data.

4.2.2 Hierarchical model

As it can be seen when comparing the Figures 3, 4, 5, and 6, for white students the replications follow better the original data than non white students replications. The biggest reason for that is the tinier amount of the data points in non white groups. It can bee seen from Figures 5 and 6, that replications of non white groups have more width variations and more out layers than original data. In addition, edcfs have bigger variations in non white groups than in white groups.

In overall, the replications of the white group are quite good (not as good as in pooled model) and models in non white groups are not so good since replications in these groups don't follow the original data.

```
y1 <- data_hierarchical$y1
ypred1 <- fit_hierarchical$draws("ypred1", format = "matrix")

grid.arrange(ppc_hist(y1, ypred1[1:48,]),
ppc_dens_overlay(y1, ypred1[1:100,]),
ppc_ecdf_overlay(y1, ypred1[1:100,]),
             layout_matrix = matrix(c(1,2,1,3), nrow = 2), nrow=2)
```

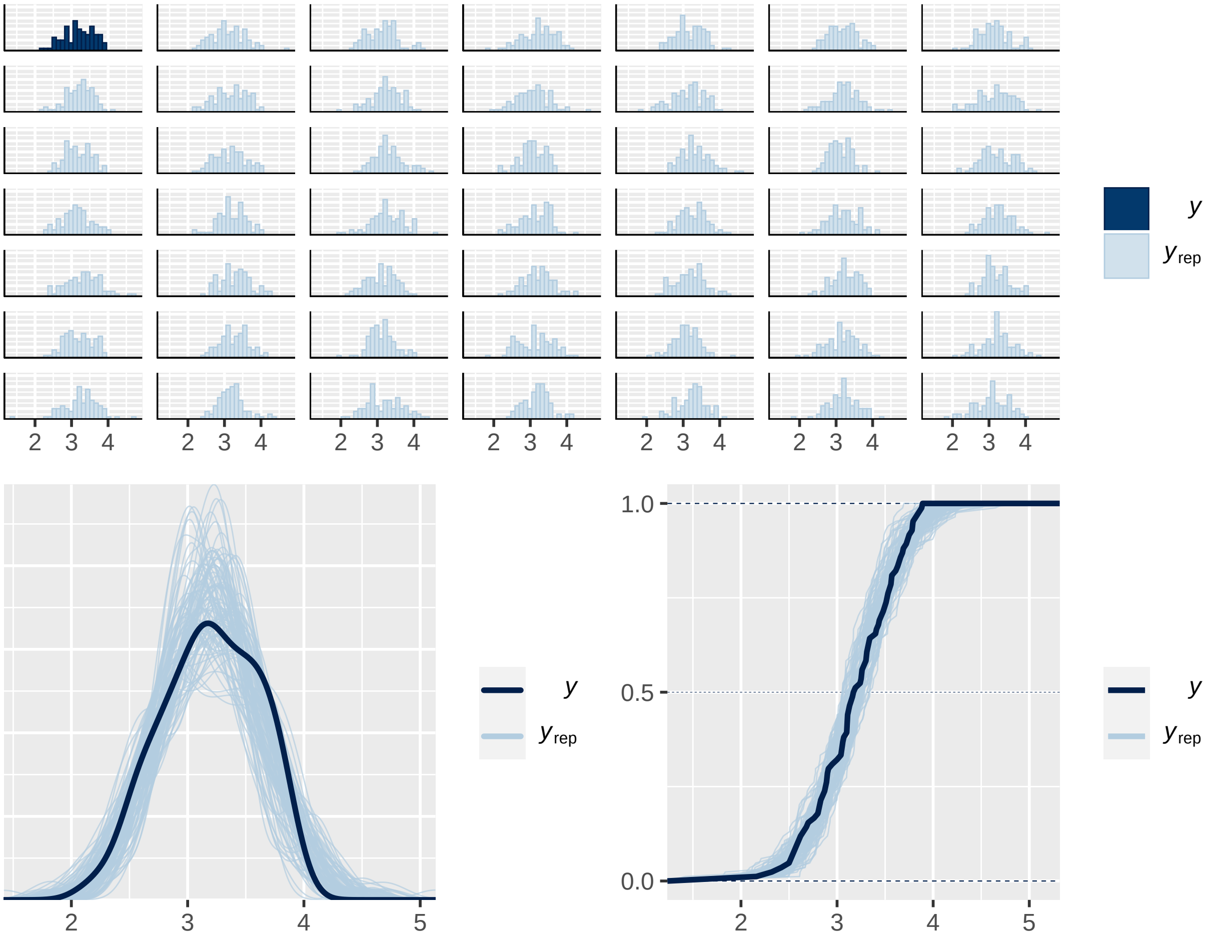


Figure 3: On the top of the figure 48 replications of the hierarchical model for data group of white male students are shown using histograms. In addition the original data is shown in left top corner. Below the histograms, the 100 distributions of the replications are over layed with original data on left. On right 100 ecdf are over layed with original data.

```
y2 <- data_hierarchical$y2
ypred2 <- fit_hierarchical$draws("ypred2", format = "matrix")

grid.arrange(ppc_hist(y2, ypred2[1:48,]),
ppc_dens_overlay(y2, ypred2[1:100,]),
ppc_ecdf_overlay(y2, ypred2[1:100,]),
              layout_matrix = matrix(c(1,2,1,3), nrow = 2), nrow=2)
```

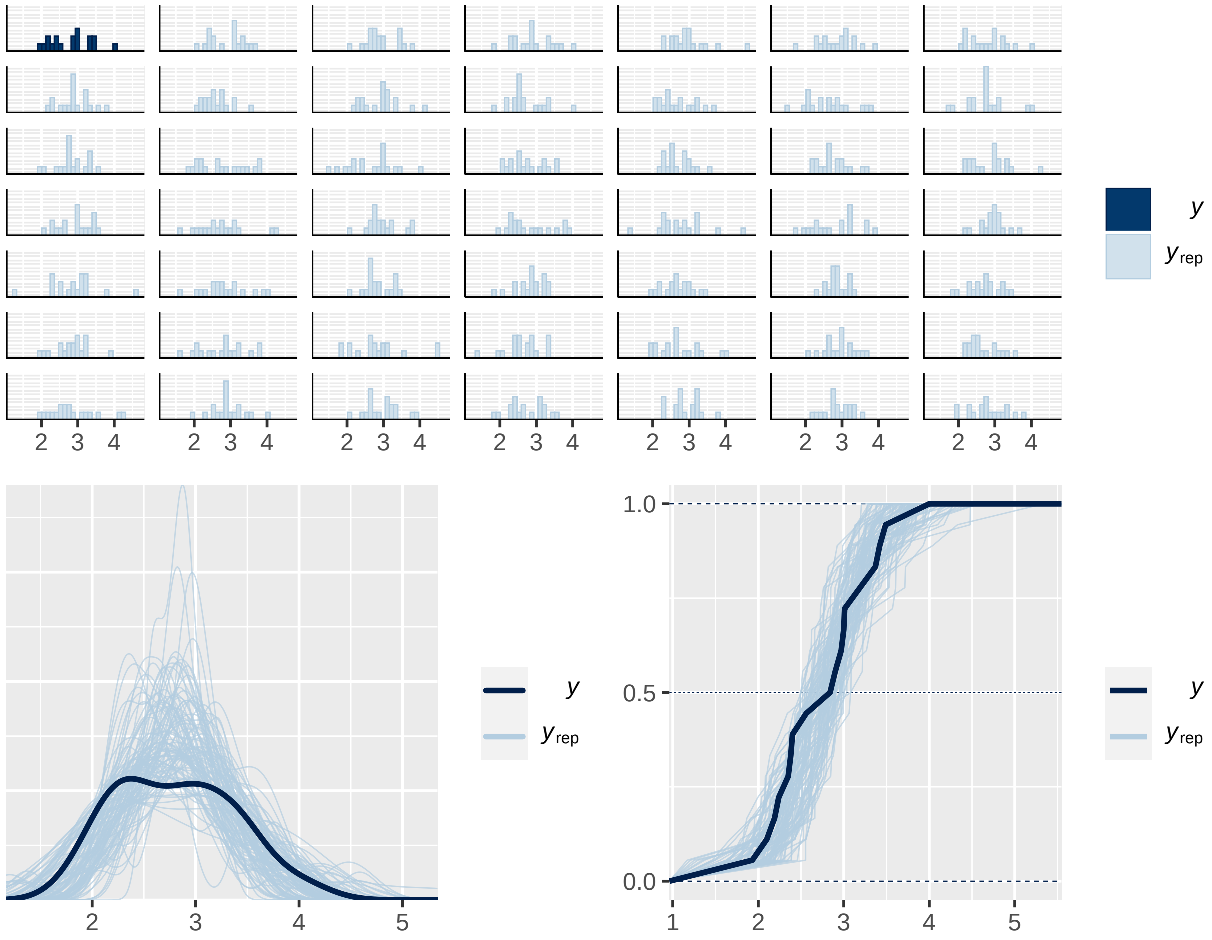


Figure 4: On the top of the figure 48 replications of the hierarchical model for data group of non white male students are shown using histograms. In addition the original data is shown in left top corner. Below the histograms, the 100 distributions of the replications are over layed with original data on left. On right 100 ecdf are over layed with original data.

```
y3 <- data_hierarchical$y3
ypred3 <- fit_hierarchical$draws("ypred3", format = "matrix")

grid.arrange(ppc_hist(y3, ypred3[1:48,]),
ppc_dens_overlay(y3, ypred3[1:100,]),
ppc_ecdf_overlay(y3, ypred3[1:100,]),
              layout_matrix = matrix(c(1,2,1,3), nrow = 2), nrow=2)
```

```
y4 <- data_hierarchical$y4
ypred4 <- fit_hierarchical$draws("ypred4", format = "matrix")

grid.arrange(ppc_hist(y4, ypred4[1:48,]),
ppc_dens_overlay(y4, ypred4[1:100,]),
```

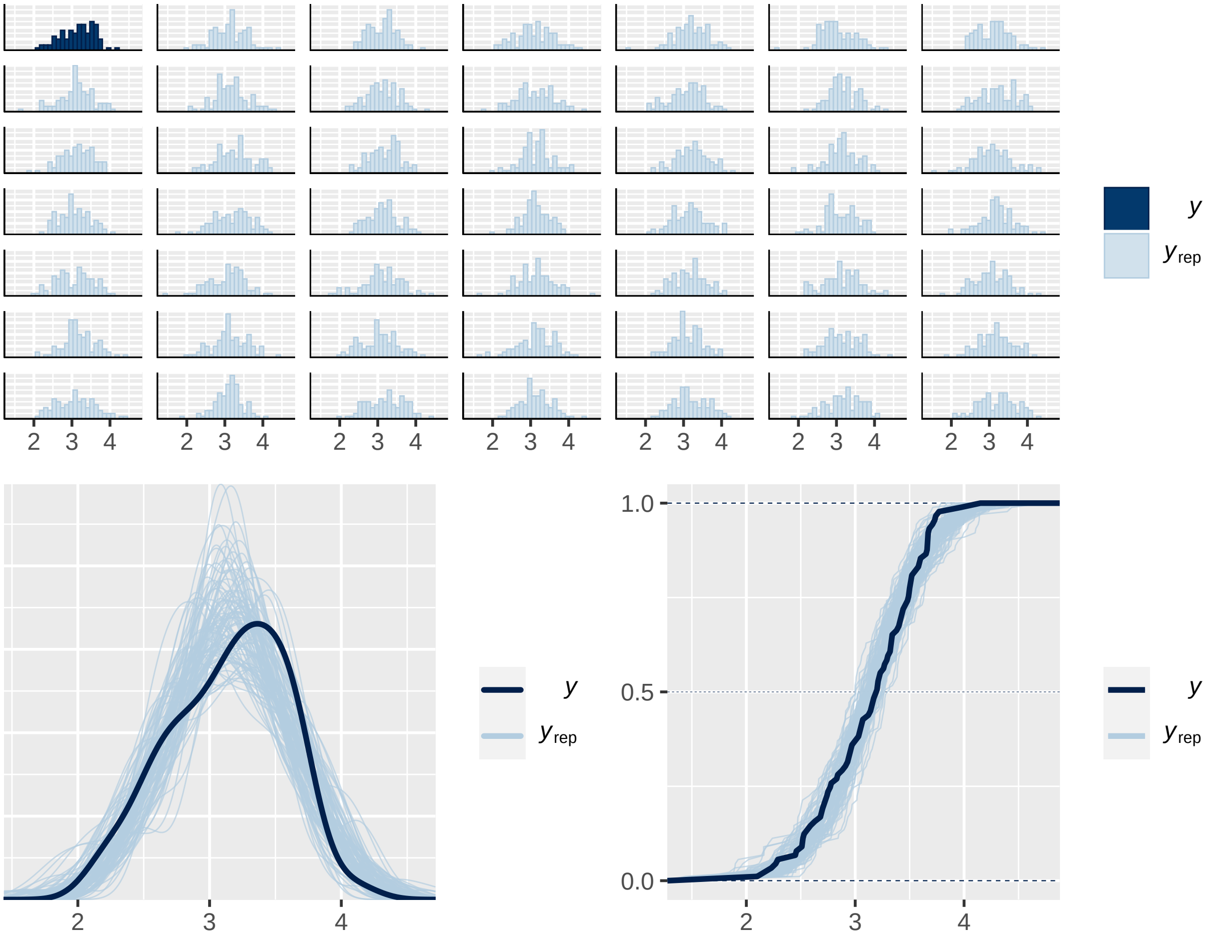


Figure 5: On the top of the figure 48 replications of the hierarchical model for data group of white female students are shown using histograms. In addition the original data is shown in left top corner. Below the histograms, the 100 distributions of the replications are over layed with original data on left. On right 100 ecdf are over layed with original data.

```
ppc_ecdf_overlay(y4, ypred4[1:100,]),
                layout_matrix = matrix(c(1,2,1,3), nrow = 2), nrow=2)
```

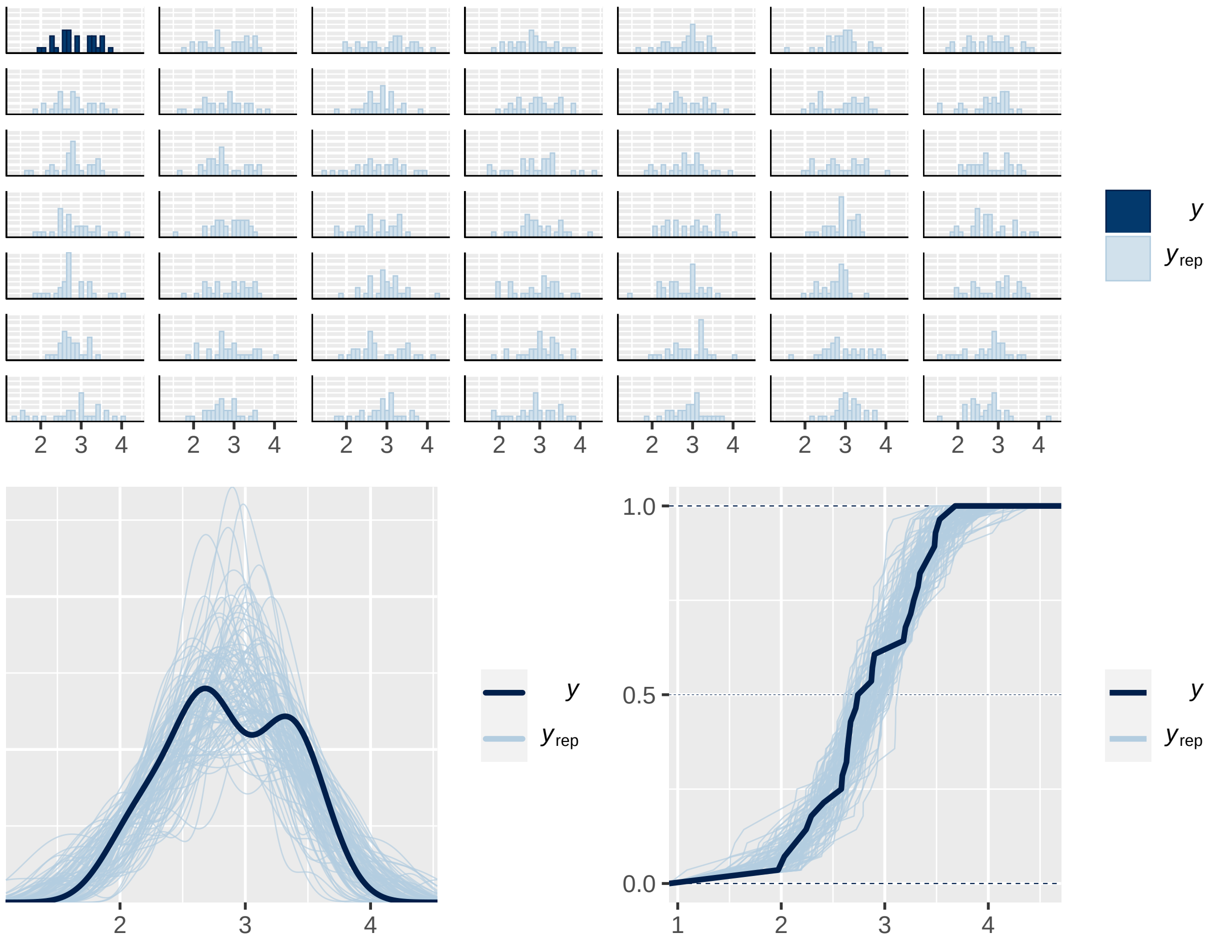


Figure 6: On the top of the figure 48 replications of the hierarchical model for data group of non white female students are shown using histograms. In addition the original data is shown in left top corner. Below the histograms, the 100 distributions of the replications are over layed with original data on left. On right 100 ecdf are over layed with original data.

Table 1: Posterior means

	1	2	3	4	5	6	7	8
alpha	0.3901930	0.3784617	0.3919734	0.3912547	0.3471881	0.3896688	0.3838581	0.3966175
beta1	0.4611466	0.4621786	0.4613767	0.4591259	0.4654118	0.4606663	0.4639042	0.4599203
beta2	0.0009451	0.0009459	0.0009381	0.0009445	0.0009521	0.0009358	0.0009462	0.0009378
beta3	0.0003677	0.0003771	0.0003667	0.0003791	0.0004012	0.0003740	0.0003640	0.0003718
beta4	0.0184211	0.0184577	0.0184912	0.0184173	0.0184428	0.0186155	0.0183185	0.0184142
beta5	0.0091861	0.0094709	0.0094099	0.0090907	0.0095479	0.0094306	0.0092477	0.0092374

Table 2: Posterior SDs

	1	2	3	4	5	6	7	8
alpha	0.3270551	0.3318461	0.3296851	0.3183283	0.3010347	0.3147375	0.3125549	0.3227091
beta1	0.0742112	0.0741120	0.0751797	0.0730227	0.0705110	0.0719253	0.0701911	0.0728267
beta2	0.0003850	0.0003910	0.0003827	0.0003757	0.0003819	0.0003773	0.0003999	0.0003716
beta3	0.0004237	0.0004315	0.0004293	0.0004152	0.0004136	0.0004094	0.0004302	0.0004172
beta4	0.0039195	0.0038783	0.0039419	0.0039258	0.0039068	0.0039219	0.0039229	0.0039180
beta5	0.0056140	0.0057777	0.0053882	0.0056051	0.0054916	0.0056848	0.0057847	0.0055735

4.3 Predictive performance assessment

Predictive performance assessment was not performed for the models. The reason for this is that the models are regression models, and no sensible metrics for examining the performance of regression models exist. Theoretically, a metric such as mean absolute error (MAE) or mean squared error (MSE) between the real and predicted values could be calculated for the data. This is not common practice, however, so predictive performance assessment is simply skipped here.

4.4 Prior sensitivity analysis

The original priors utilised were $\mathrm{N}(0, 100)$ for α and β or their means, and $\mathrm{N}(0, 10)$ for all standard deviations. For prior sensitivity analysis, the priors presented in the table below were tested to see how much the posteriors of α and β_k would differ from the original. Altogether, 8 prior combinations were tested, including the original priors for reference. The same prior combinations were used for testing both models. Narrower priors, e.g. $\mathrm{N}(0, 0.1)$, were excluded from the analysis, as they would not be weakly informative anymore.

alpha, beta OR mu_alpha, mu_beta	sigma, sigma_alpha, sigma_beta
N(0,100)	N(0,10)
N(0,1000)	N(0,10)
N(0,50)	N(0,10)
N(0,10)	N(0,10)
N(0,1)	N(0,10)
N(0,100)	N(0,100)
N(0,100)	N(0,50)
N(0,100)	N(0,1)

4.4.1 Pooled model

The above results show that regardless of the used priors, all posteriors for α and β_k are very similar to each other. Therefore it can be concluded that the pooled model is not sensitive to changes in the priors.

Table 3: Posterior means

	1	2	3	4	5	6	7	8
alpha1	0.6750332	0.6448616	0.6662860	0.6821844	0.6395425	0.6810750	0.6555698	0.6463861
alpha2	0.4111187	0.3943184	0.3997457	0.3743995	0.3352635	0.3880243	0.2993572	0.4585604
alpha3	0.7128101	0.6752439	0.6866892	0.7155874	0.6722886	0.7127689	0.5965686	0.6715814
alpha4	0.3897968	0.3724202	0.3940690	0.3703460	0.3795197	0.3832656	0.3374951	0.4402399
beta1_1	0.4057357	0.4130092	0.4044361	0.3985098	0.4157612	0.4047242	0.4296213	0.4152957
beta2_1	0.0009161	0.0009118	0.0009252	0.0009230	0.0009069	0.0009258	0.0008901	0.0009170
beta3_1	0.0004205	0.0004240	0.0004258	0.0004422	0.0004380	0.0004113	0.0003669	0.0004089
beta4_1	0.0164358	0.0166801	0.0166881	0.0163857	0.0162979	0.0164801	0.0149326	0.0165747
beta5_1	0.0047366	0.0051141	0.0050221	0.0047960	0.0044168	0.0046604	0.0051837	0.0051053
beta1_2	0.4601736	0.4636770	0.4576573	0.4583448	0.4680847	0.4630411	0.4706504	0.4575245
beta2_2	0.0001425	0.0001986	0.0001855	0.0001215	0.0002062	0.0001723	0.0003439	0.0001760
beta3_2	0.0005267	0.0004937	0.0005069	0.0005949	0.0005201	0.0005060	0.0005070	0.0004494
beta4_2	0.0230482	0.0223456	0.0234314	0.0229716	0.0230432	0.0226863	0.0225060	0.0225580
beta5_2	0.0270770	0.0273333	0.0275369	0.0281800	0.0290554	0.0286187	0.0257224	0.0265067
beta1_3	0.5216420	0.5244947	0.5219051	0.5195021	0.5163522	0.5216265	0.5110154	0.5198326
beta2_3	0.0008183	0.0007855	0.0007992	0.0008151	0.0008167	0.0008194	0.0008061	0.0008268
beta3_3	-0.0002625	-0.0001973	-0.0002110	-0.0002491	-0.0001662	-0.0002560	-0.0000113	-0.0002065
beta4_3	0.0168655	0.0171582	0.0170607	0.0168341	0.0167791	0.0167281	0.0177735	0.0171218
beta5_3	0.0053349	0.0054389	0.0056359	0.0052374	0.0055388	0.0049167	0.0044065	0.0059370
beta1_4	0.5553511	0.5526324	0.5475788	0.5553802	0.5425339	0.5536779	0.5454140	0.5356532
beta2_4	0.0003289	0.0003561	0.0003578	0.0003302	0.0003655	0.0003498	0.0004755	0.0003622
beta3_4	0.0002268	0.0002409	0.0002282	0.0002645	0.0002901	0.0002344	0.0002496	0.0002182
beta4_4	0.0168398	0.0170680	0.0170811	0.0166111	0.0162914	0.0168015	0.0178228	0.0169723
beta5_4	0.0065033	0.0068282	0.0070543	0.0062833	0.0070126	0.0060944	0.0062873	0.0068158

Table 4: Posterior SDs

	1	2	3	4	5	6	7	8
alpha1	0.4894600	0.4922202	0.4704830	0.4836623	0.4672079	0.4866199	0.4268033	0.4625858
alpha2	0.5651351	0.5604468	0.5706474	0.6036677	0.5544141	0.5902998	0.5114067	0.4948603
alpha3	0.4263073	0.4427410	0.4251016	0.4266427	0.4130309	0.4475214	0.4187079	0.4293073
alpha4	0.4793913	0.4673857	0.4738151	0.4912799	0.4405367	0.4893630	0.4123352	0.4358221
beta1__1	0.1048547	0.1110763	0.1100664	0.1116394	0.1028123	0.1100622	0.1011167	0.1089618
beta2__1	0.0005564	0.0005602	0.0005633	0.0005713	0.0005610	0.0005781	0.0004905	0.0005490
beta3__1	0.0006127	0.0006181	0.0005967	0.0006240	0.0005908	0.0006132	0.0005280	0.0006142
beta4__1	0.0054367	0.0052319	0.0052774	0.0052427	0.0051809	0.0051911	0.0051674	0.0051220
beta5__1	0.0078472	0.0078153	0.0078175	0.0079875	0.0082797	0.0079119	0.0069903	0.0077567
beta1__2	0.1304401	0.1359752	0.1315170	0.1371966	0.1278564	0.1342254	0.1175676	0.1290665
beta2__2	0.0007335	0.0007175	0.0007115	0.0007558	0.0007223	0.0007198	0.0006918	0.0007081
beta3__2	0.0009388	0.0008958	0.0009124	0.0009657	0.0008936	0.0009169	0.0007789	0.0008503
beta4__2	0.0093742	0.0090260	0.0093723	0.0092454	0.0092051	0.0092826	0.0080377	0.0088471
beta5__2	0.0175087	0.0171382	0.0174081	0.0176670	0.0166751	0.0176711	0.0168265	0.0166206
beta1__3	0.0944189	0.0955468	0.0959415	0.0936899	0.0915493	0.0984502	0.0844680	0.0952295
beta2__3	0.0005504	0.0005441	0.0005571	0.0005516	0.0005492	0.0005610	0.0004794	0.0005400
beta3__3	0.0006075	0.0006104	0.0005916	0.0005983	0.0006134	0.0006136	0.0006411	0.0005943
beta4__3	0.0052600	0.0051185	0.0053043	0.0051537	0.0054579	0.0053101	0.0047129	0.0051888
beta5__3	0.0081323	0.0080232	0.0079123	0.0080326	0.0083201	0.0082822	0.0073729	0.0079228
beta1__4	0.1419090	0.1372708	0.1386515	0.1450786	0.1298481	0.1421431	0.1211416	0.1289271
beta2__4	0.0007072	0.0006832	0.0006751	0.0007083	0.0006743	0.0006881	0.0006229	0.0006716
beta3__4	0.0007758	0.0007401	0.0007390	0.0007706	0.0007220	0.0007324	0.0006342	0.0007365
beta4__4	0.0075732	0.0074678	0.0079579	0.0077700	0.0081422	0.0077223	0.0067485	0.0075312
beta5__4	0.0126388	0.0126928	0.0129125	0.0129774	0.0128282	0.0130516	0.0113618	0.0122682

4.4.2 Hierarchical model

There is slightly more variation in the posteriors of the hierarchical model when various priors are utilised, but the posteriors look quite similar nonetheless. It can be concluded that the hierarchical model, similarly to the pooled model, is not sensitive to changes in the priors.

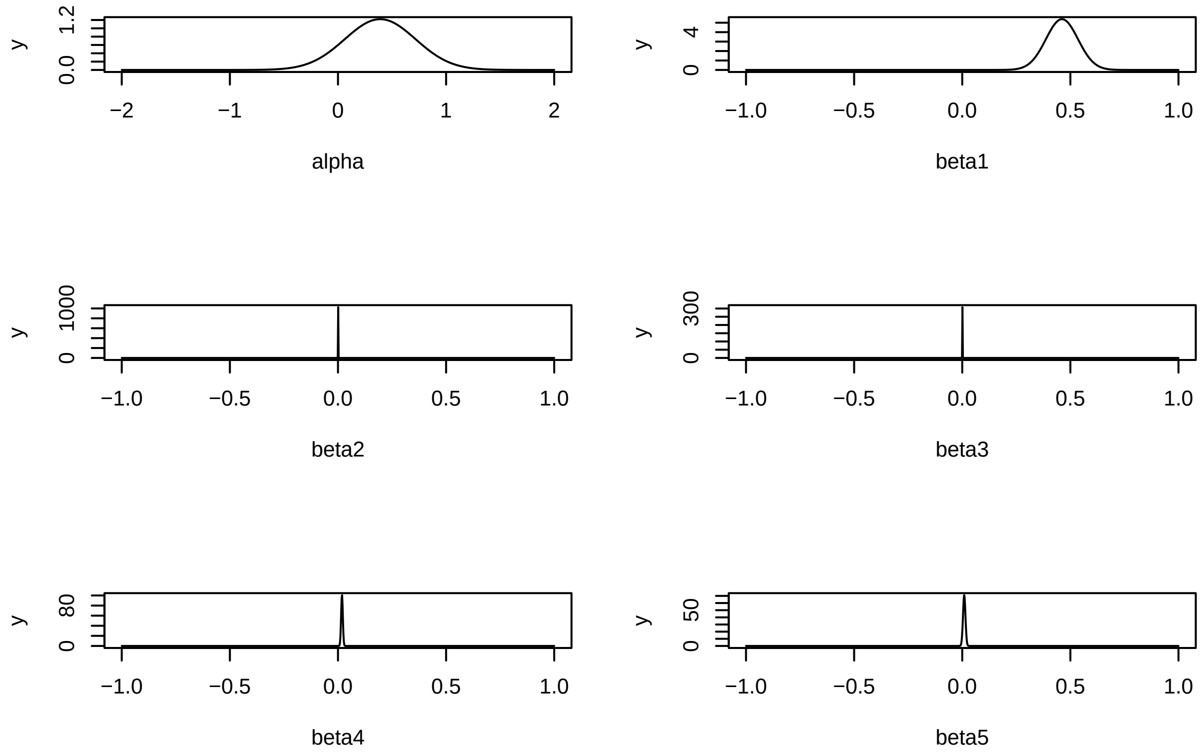
4.5 Posterior distributions

Next, let us inspect and visualise the posterior distributions of α and β_k. The table below contains the posterior distributions of the six variables for the pooled model and all 4 groups of the hierarchical model separately.

	pooled	h. group 1	h. group 2	h. group 3	h. group 4
alpha	N(0.39, 0.327)	N(0.675, 0.489)	N(0.411, 0.565)	N(0.713, 0.426)	N(0.39, 0.479)
beta1	N(0.461, 0.074)	N(0.406, 0.105)	N(0.46, 0.13)	N(0.522, 0.094)	N(0.555, 0.142)
beta2	N(0.001, 0)	N(0.001, 0.001)	N(0, 0.001)	N(0.001, 0.001)	N(0, 0.001)
beta3	N(0, 0)	N(0, 0.001)	N(0.001, 0.001)	N(0, 0.001)	N(0, 0.001)
beta4	N(0.018, 0.004)	N(0.016, 0.005)	N(0.023, 0.009)	N(0.017, 0.005)	N(0.017, 0.008)
beta5	N(0.009, 0.006)	N(0.005, 0.008)	N(0.027, 0.018)	N(0.005, 0.008)	N(0.007, 0.013)

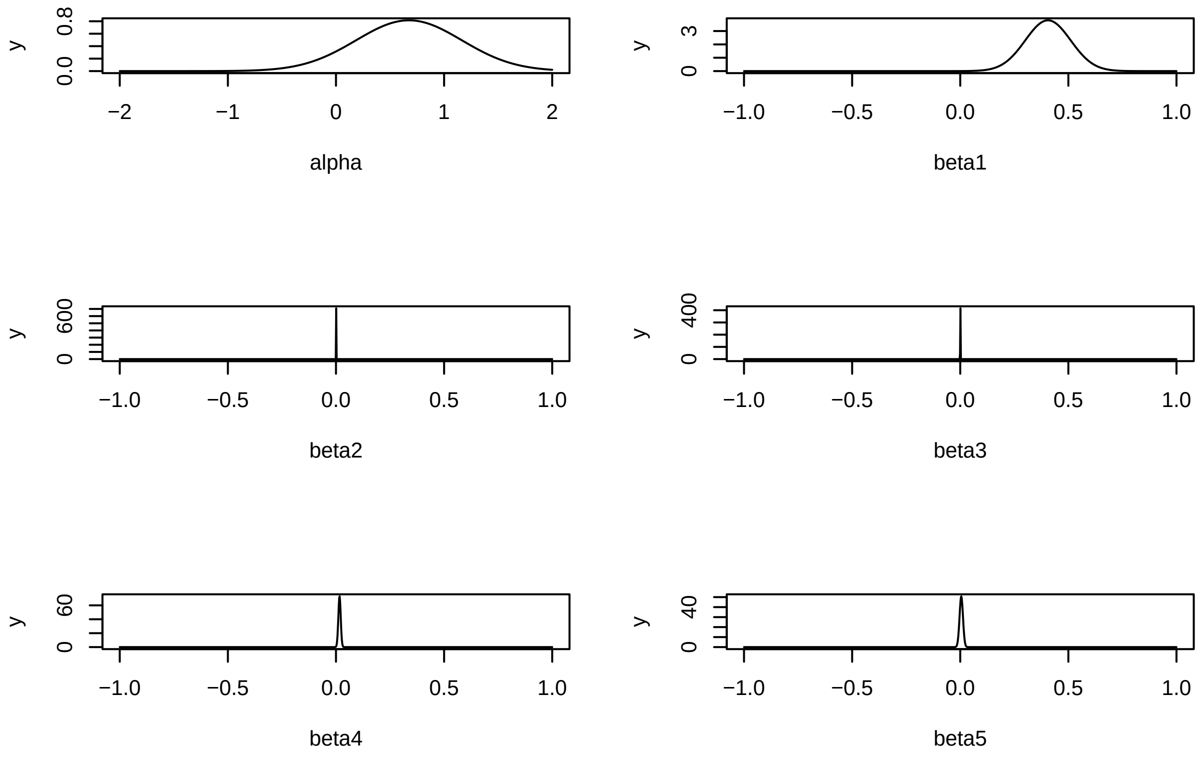
Let us also plot the posterior distributions.

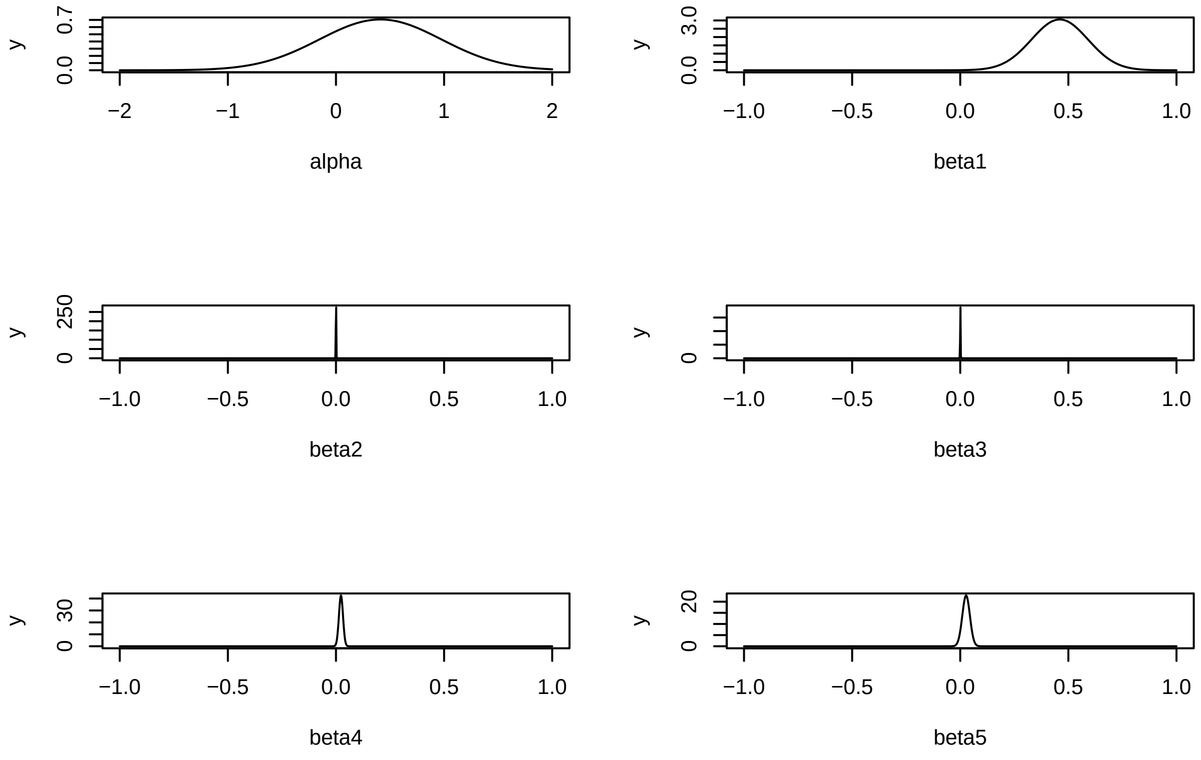
4.5.1 Pooled model

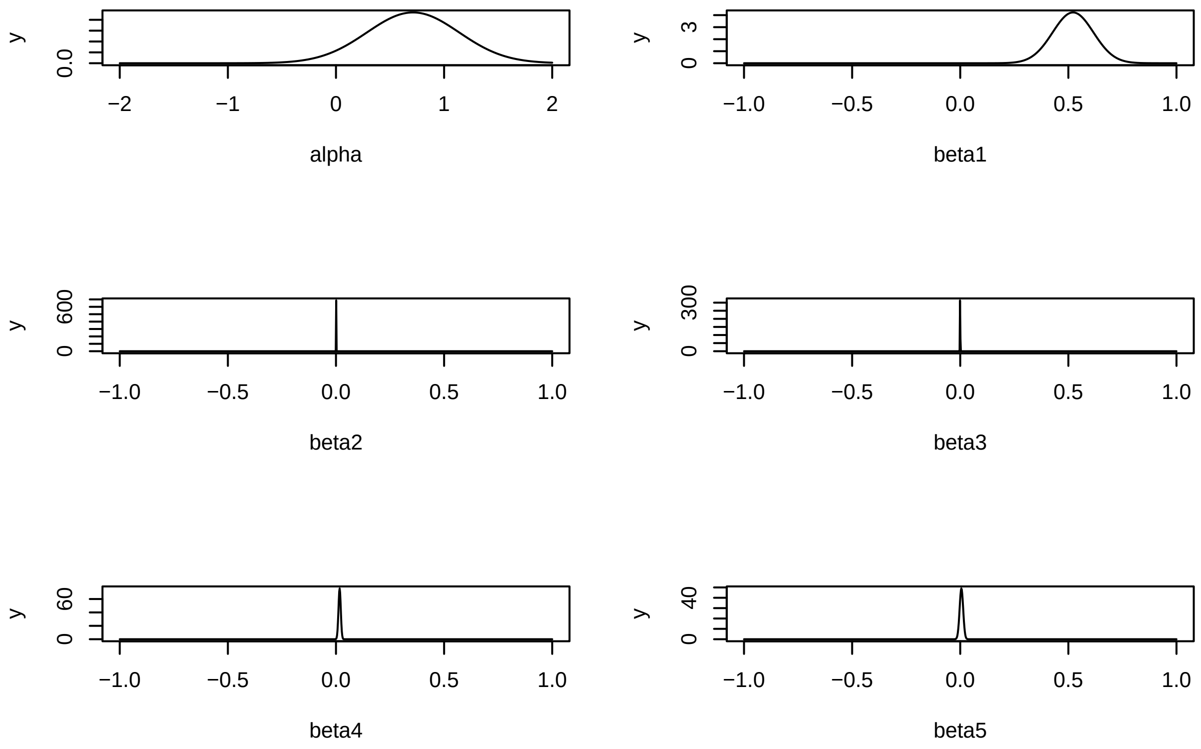


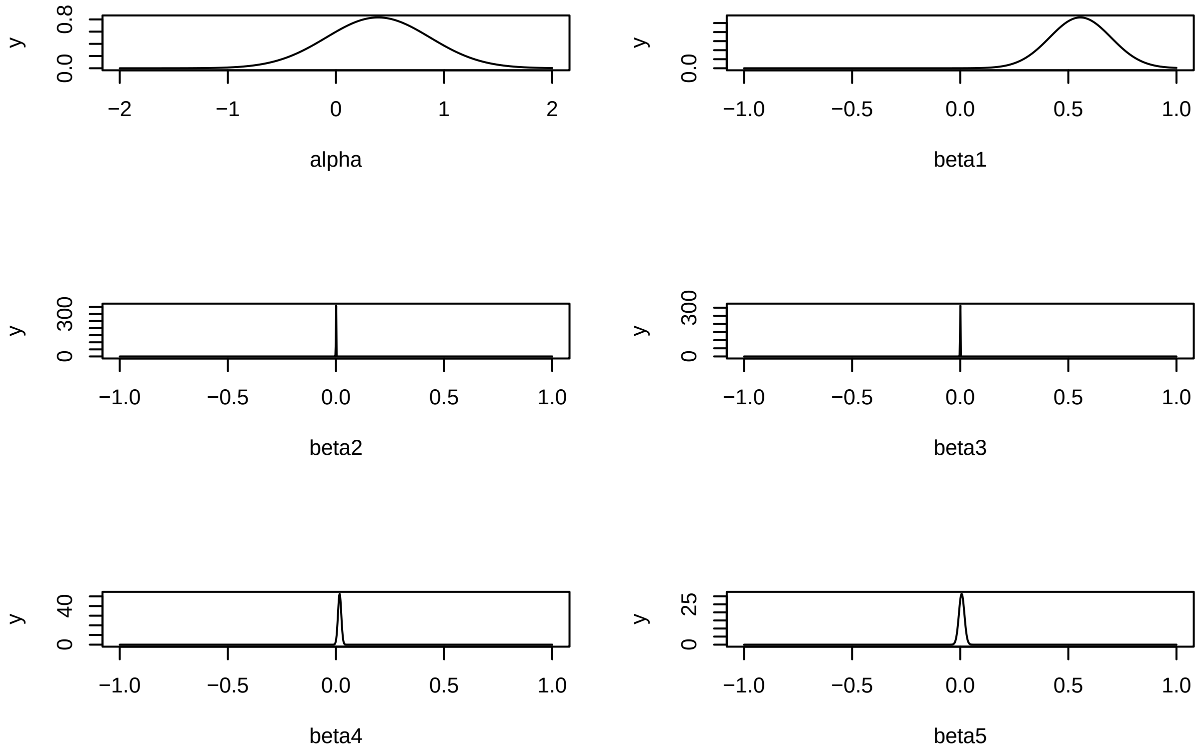
The plot shows that the slope parameter β_1, multiplying the variable for high school GPA, likely differs from 0 and has the most effect on college GPA, our target variable. The other βs have very narrow distributions close to 0, though, implying their effect on college GPA is very small or non-existent.

4.5.2 Hierarchical model






y
0.0
−2
−1
0
1
2
alpha
y
0
3
−1.0
−0.5
0.0
0.5
1.0
beta1
y
0
600
−1.0
−0.5
0.0
0.5
1.0
beta2
y
0
300
−1.0
−0.5
0.0
0.5
1.0
beta3
y
0
60
−1.0
−0.5
0.0
0.5
1.0
beta4
y
0
40
−1.0
−0.5
0.0
0.5
1.0
beta5



The posteriors of the hierarchical model are quite similar to the pooled model. The intercept terms α of groups 1 and 3, and of groups 2 and 4, are very similar to each other, implying that the ethnicity of the student governs the intercept term α. Furthermore, there is slight variation in the slope terms β_k, especially β_5: for groups with less data, the distributions are narrower, and for groups with more data, wider.

4.6 Model comparison

The log likelihoods of the data have been calculated in the “generated quantities” blocks of the models. They are given to the “loo” function of the “loo” package, and the individual results and model comparison are printed.

```r
log_lik_pooled <- fit_pooled$draws("log_lik", format = "matrix")
log_lik_hierarchical <- fit_hierarchical$draws("log_lik", format = "matrix")

loo_pooled <- loo(log_lik_pooled)
loo_hierarchical <- loo(log_lik_hierarchical)

loo_pooled
```

```
##
## Computed from 4000 by 219 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -108.0  9.5
## p_loo         6.9  0.7
## looic       216.0 19.0
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_hierarchical
```

```
##
## Computed from 4000 by 219 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -109.7  9.5
## p_loo        18.4  2.4
## looic       219.4 19.0
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     214   97.7%   877
##  (0.5, 0.7]   (ok)         4    1.8%   900
##    (0.7, 1]   (bad)        1    0.5%   21
##    (1, Inf)   (very bad)   0    0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

```
loo_compare(list("pooled" = loo_pooled, "hierarchical" = loo_hierarchical))
```

```
##              elpd_diff se_diff
## pooled        0.0       0.0
## hierarchical -1.7      14.0
```

The results show that the PSIS-LOO estimate for the pooled model can be considered reliable, as all values $\hat{k} \lesssim 0.7$. The same cannot be said of the hierarchical model, and the PSIS-LOO estimate for it should be considered, or at least suspected, biased and overly optimistic.

Comparing the two models reveals that the pooled model seems to fit to the data better than the hierarchical model (likely due to the lack of training data for the latter one). Therefore for any future predictions, the pooled model should be utilised.

5 Discussion about problems and improvements

The overall problem in the project was that there was not enough data. For example in hierarchical model, some of the data groups were quite small: th group of non white groups only contained 18 and 28 data, and the rest of the 173 data points were distributed between white groups. In addition, there was big variation in the magnitudes of the variables (HSGPA etc.). This could have been solved by using normalisation.

There were divergences in the hierarchical model. If there was more time, the divergences could have been solved by analysing more the running of the model; The default values (the number of the chains, the number of the iterations etc.) were utilised during the model fitting. Furthermore, testing out different priors to solve the divergence issue.

6 Conclusion

Based on the analysis, the pooled model describes the data better than the hierarchical model. However, both models performed well. The hierarchical model describes the white and non white students differently. So according to the hierarchical model, there were more differences between white and non-white students than between male and female students. This can be seen most clearly from the posterior distributions of the αs.

7 Self-reflection

During the project, we noticed that finding the data is difficult and time consuming. Moreover, the preprocessing of the data and deciding the suitable data groups for the hierarchical model was surprisingly time consuming. We also noticed the importance of the amount of the data: more data, better model.

We learned to make more complicated stan models.