Thiomon Data Analysis Report

Long Nguyen and Amaanat Ali

Contents

1. Introduction

Thiopurines are widely used immunomodulators for the treatment of ulcerative colitis [UC] and Crohn's disease [CD] in inflammatory bowel disease [IBD] patients. Thiopurines can induce immune suppression, clinical and biological remission, and are steroid-sparing agents (1).

1.1. Motivation

The study uses a data set based on de-identified Laboratory Data on IBD Patients using Thiourines for at least 4 weeks and their eventual Remission/Active Status after at least 12 Weeks of therapy.

1.2. Problem Statement

We plan to apply the Bayesian data analysis technique to predict the Objective Remission (OR) in optimizing thiopurine therapy for inflammatory bowel disease by applying pooled, separate, and hierarchical models. Furthermore, we make a comparison between the 3 models and draw some observations.

2. Data: Thiomon Dataset

A dataset containing laboratory data and outcomes of IBD patients on Thiopurine therapy at the University of Michigan. Data on laboratory values for a complete blood count and chemistry panel at least 4 weeks after the start of thiopurine therapy in IBD patients (2). The University of Michigan Hospital is in Ann Arbor, USA. These data have been anonymized, and time-shifted. Age is reported in days of life.

2.1. Data description

Based on the blood parameters, we can categorize them into broad categories like Hematology, Chemistry, Electrolytes, Liver Function, and Inflammation/Other Markers.

Hematology Parameters (Related to blood cells, hemoglobin, and related components)

Platelet Count (plt)

Mean Platelet Volume (mpv)

White Blood Cell Count (wbc)

Hemoglobin (hgb)

Hematocrit (hct)

Red Blood Cell Count (rbc)

Mean Corpuscular Volume (MCV) (mcv)

Mean Corpuscular Hemoglobin (MCH) (mch)

Mean Corpuscular Hemoglobin Concentration (MCHC) (mchc)

Red Cell Distribution Width (RDW) (rdw)

Neutrophil Percent (neut_percent)

Lymphocyte Percent (lymph_percent)

Monocyte Percent (mono_percent)

Eosinophil Percent (eos_percent)

Basophil Percent (baso_percent)

Chemistry Parameters (General body chemistry markers)

Blood Urea Nitrogen (BUN) (un)

Sodium (sod)

Potassium (pot)

Chloride (chlor)

Bicarbonate (CO2) (co2)

Creatinine (creat)

Glucose (gluc)

Calcium (cal)

Protein (prot)

Albumin (alb)

Liver Function Parameters (Enzyme and bilirubin levels)

Aspartate Transaminase (AST) (ast)

Alanine Transaminase (ALT) (alt)

Alkaline Phosphatase (ALK) (alk)

Total Bilirubin (Tbil) (tbil)

Inflammation and Other Markers

Active Inflammation Despite Thiopurines for > 12 Weeks (active)

Remission of Inflammation After Thiopurines for > 12 Weeks (remission)

2.2. Data preprocessing

Load the data comprising approximately 5K rows of blood chemistry with remission and active status (each is 1 bit and complementary to each other).

```
# Take the ratio of hgb and hct which describes if the patient is anaemic or not
source_data$hbghctrat = round(thiomon$hgb/thiomon$hct,2)

# Group into 5 groups
source_data$hbghctrat <- cut(source_data$hbghctrat, breaks = seq(min(source_data$hbghctrat), max(source_

# Convert this to a character label
source_data$hbghctrat <- as.character(source_data$hbghctrat)
source_data$hgbratnum <- cut(source_data$hgb, breaks = seq(min(source_data$hgb), max(source_data$hgb),
source_data$hgbrat <- as.character(source_data$hgbratnum)

# Sample a subset of the 4500 rows to improve computation time
MAX_DATA_SAMPLES <- 4500
#Number of samples for the likelihood
NUM_DATA_SAMPLES <- 2000
source_data <- na.omit(source_data)
train_data_norm <- source_data[sample(MAX_DATA_SAMPLES, NUM_DATA_SAMPLES), ]

train_data <- train_data_norm
train_data <- na.omit(train_data)
test_data <- source_data[4501:nrow(source_data), ]
```

2.3 Identifying variables for our modeling

There are 30 blood parameters which are identified by the database. Using the MLE algorithm 'lm', we can quickly check that all of these variables together explain 66% of the variability in the data set. Unfortunately, using all of them is computationally inefficient and will take a long time to fit a Bayesian model. Hence we need to prune the list down.

```
#lm_formula <- remission ~ 0 + days_of_life + wbc + hgb + hct + plt + rbc + mcv + mch + #mchc + rdw + m
#residual sd = 0.44, R-Squared = 0.66

#residual sd = 0.47, R-Squared = 0.60
#lm_formula <- remission ~ 0 + hgb + hct + mchc

#From paper's recommendation
#residual sd = 0.46, R-Squared = 0.60
#lm_formula <- remission ~ 0 + hgb + lymph_percent + hct + neut_percent + plt + alb +
#ast

#lm1 <- lm(formula = lm_formula, data = train_data)
#display(lm1)
#install.packages("rstanarm")
#library(rstanarm)

#fitg <- stan_glm(formula = lm_formula, data = train_data, refresh=0)
#summary(fitg)

#fitg0 <- update(fitg, formula = remission ~ 1, QR=FALSE)

#(loog0 <- loo(fitg0))

#(loog <- loo(fitg, k_threshold=0.7))

#loo_compare(loog0, loog)

#fitg_cv <- cv_varsel(fitg, method='forward', cv_method='LOO', validate_search=FALSE)

#plot(fitg_cv, stats = c('elpd', 'rmse'))

#(nsel <- suggest_size(fitg_cv, alpha=0.1))

#(vsel <- solution_terms(fitg_cv)[1:nsel])
```

The model with covariates definitely has a better EPLD compared to the base model and hence having covariates in our Bayesian model definitely makes sense. Going through the exercise, the 'hgb' and 'lymph_percent' seem the dominant variables and they have 58% explanatory power (3).

```
#lm_formula <- remission ~ 0 + hgb + lymph_percent
#lm1 <- lm(formula = lm_formula, data = train_data)
#display(lm1)
```

We continue with this choice further in our model fitting and also remember that 'lymph_percent' adds only 1% more explanatory power overall, so we may also decide to drop it.

To allow for slope and intercept to be defined in the separate and hierarchical models, we need to distribute 'hgb' into N classes across the range of values. Furthermore, to ensure that the number of data samples are same in each class (otherwise the likelihood statistic will force a larger variance), we sample each class and

pick M samples. Hence the total number of samples for our study becomes M x N.

3. Models

3.1. Pooled model

Make a basic pooled model.

```
invisible({capture.output({
ref_thiomon_formulae_pooled <- bf(remission ~ 1, family = "bernoulli")

thiomon_priors_default_priors_pooled <- get_prior(ref_thiomon_formulae_pooled, data = train_data)

thiomon_set_priors_informative_pooled <- c(
  prior(
    normal(0,1),
    class = "Intercept"
  )
)

thiomon_data_model_pooled <- brm(
    formula = ref_thiomon_formulae_pooled,
    prior = thiomon_set_priors_informative_pooled,
    data = train_data,
    save_pars = save_pars(all = TRUE),
    iter = 2000,
    warmup = 1000,
    chains = 4,
    control = list(
      adapt_delta = 0.8,
      max_treedepth = 20
    ),
    threads = threading(2), cores = 8, backend = "cmdstanr"
)
})})
```

There are no divergent transactions reported. Let's check for some model-related diagnostics.

3.1.1 R-hat and Effective Sample Size (ESS)

First checking the model for any discrepancies in R-hat and ESS.

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: remission ~ 1
##    Data: train_data (Number of observations: 2000)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.17      0.04     0.08     0.26 1.00     1200     1862
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
```

```
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

R-hat and effective sample sizes look good. At convergence also R-hat value is 1 means the chains have converged well.

3.1.2 Pareto k-hat

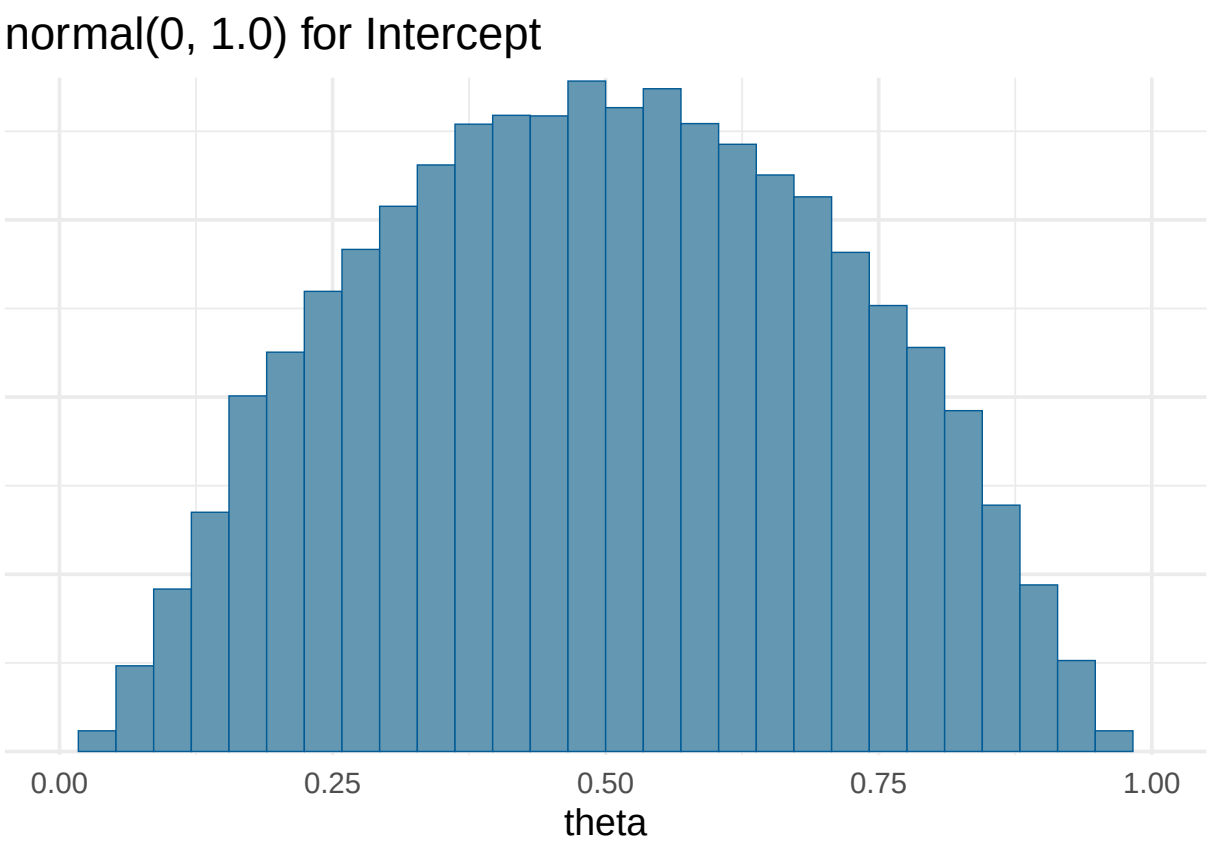
```
##
## Computed from 4000 by 2000 log-likelihood matrix.
##
##          Estimate  SE
## elpd_loo  -1379.9 3.8
## p_loo         1.0 0.0
## looic      2759.7 7.6
## ------
## MCSE of elpd_loo is 0.0.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.3, 0.3]).
##
## All Pareto k estimates are good (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

There are no Pareto k-hat estimated greater than 0.7 which is what we desire.

3.1.3 Choosing the prior

Continuing with checks on prior, justification for the chosen prior is that the default prior 'student_t(3, 0, 2.5)' is almost equivalent to a 'normal(0, 1.5)' but we find that the underlying 'theta' of the distribution is '0.48' which allows us to make a narrower choice.

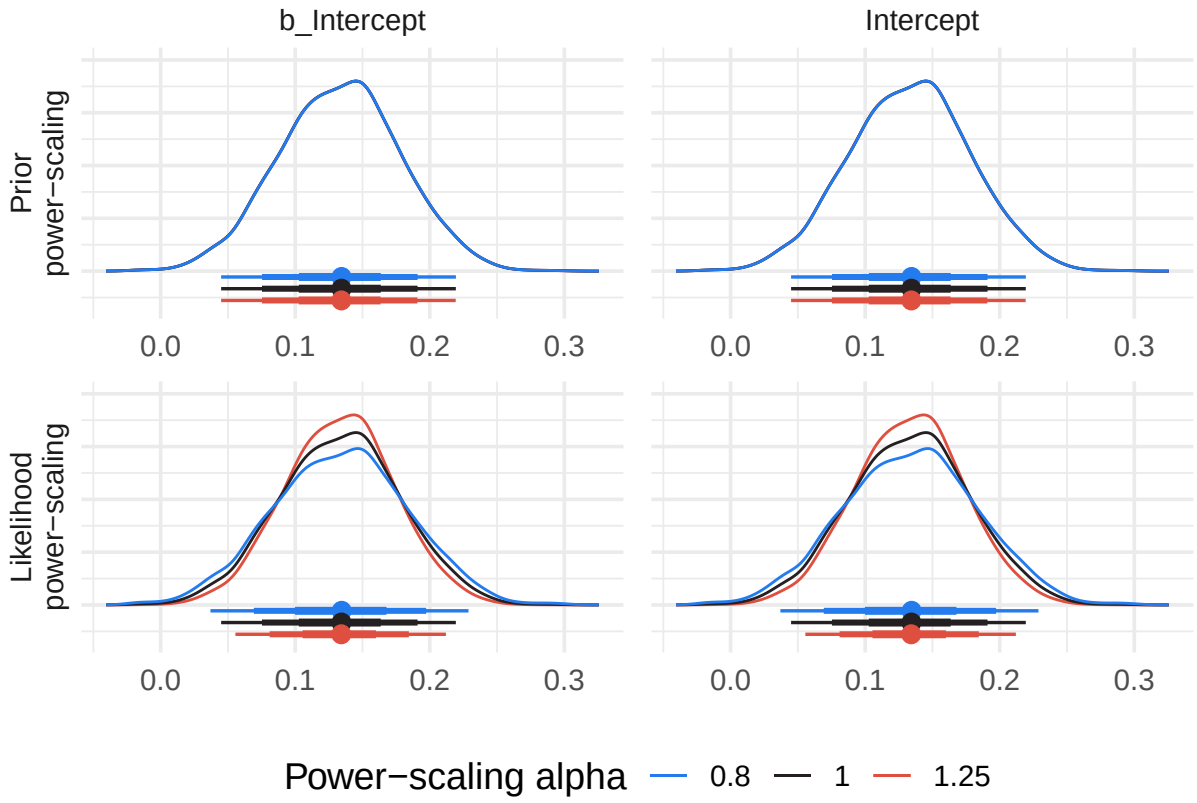
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



It seems it is a correct choice as there is no conflict revealed from the prior diagnostics.

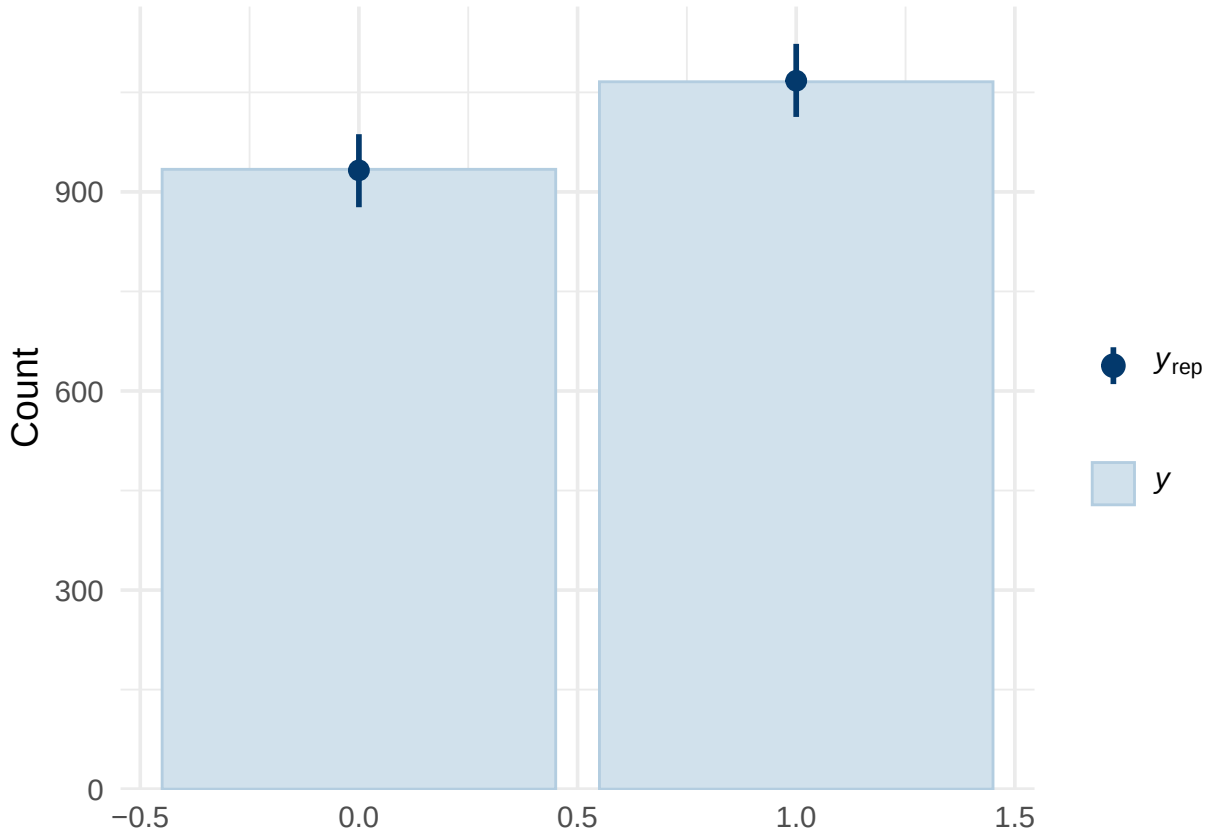
variable	prior	likelihood	diagnosis
$b_Intercept$	0.0012	0.082	-
Intercept	0.0012	0.082	-

3.1.4 Posterior prior sensitivity



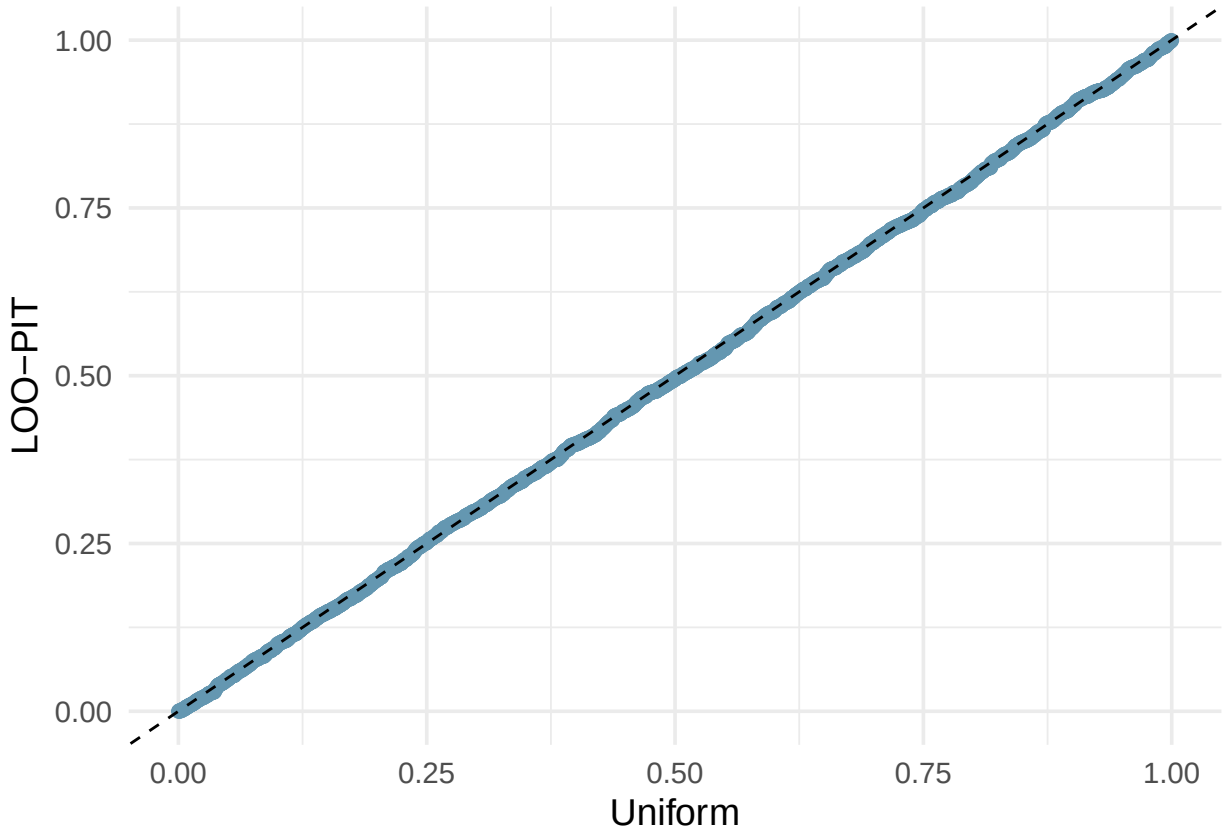
The values themselves are also very small.

Posterior predictive checks seem fine as well. Bars in the plot seem to fit well.



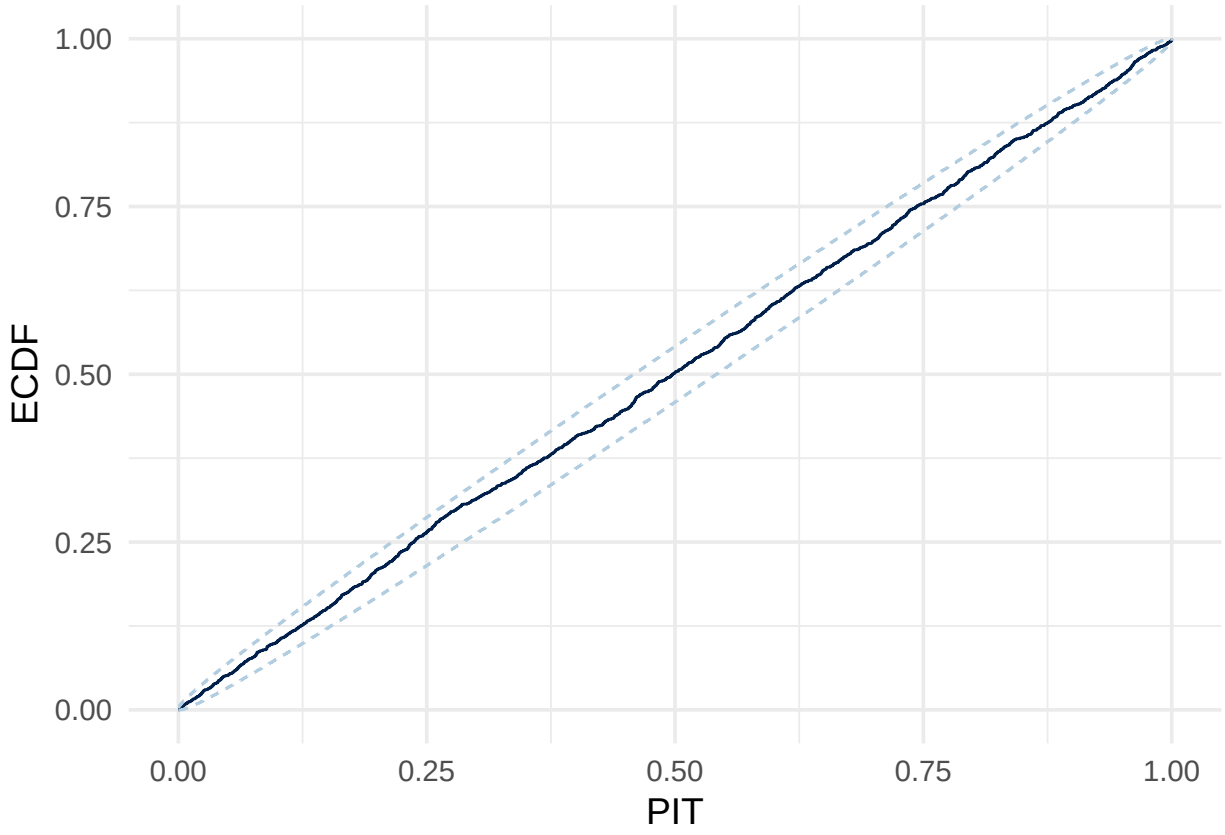
3.1.5 Posterior Predictive Checking

LOO-PIT check seems fine as well.



LOO-PIT ECDF is within the blue recommendation.

variable	mean	median	sd	mad	q5	q95	rhat	ess$_bulk$	ess$_tail$
b$_I$*ntercept*	0.17	0.17	0.044	0.044	0.098	0.24	1	1200	1862
Intercept	0.17	0.17	0.044	0.044	0.098	0.24	1	1200	1862



3.1.6 Model parameters

The value of the population intercept 'alpha' is estimated at 0.14. We exclude the likelihood parameters.

3.2. Separate model

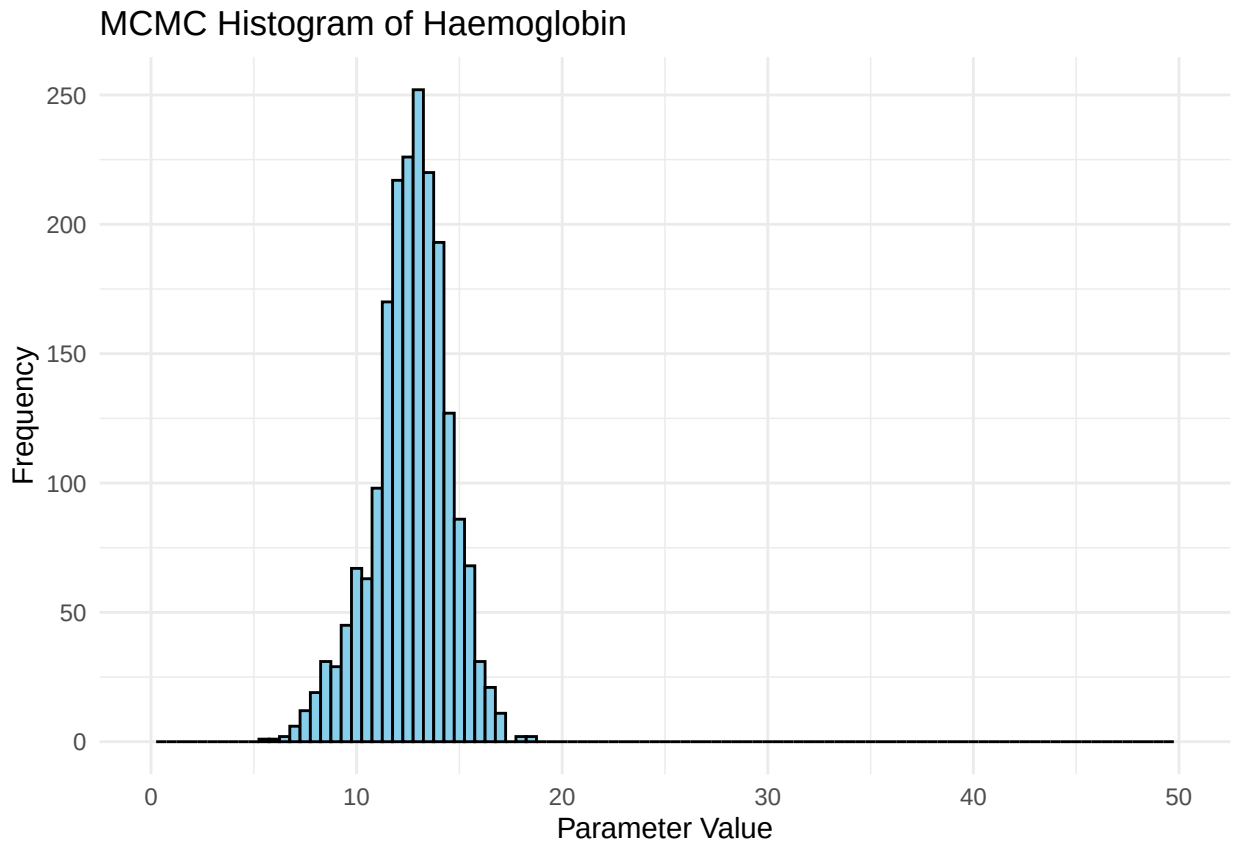
Try with a separate model. Let's try with the sample one below using the recommended parameters from the above section:

```
invisible({capture.output({
ref_thiomon_formulae_separate <- bf(remission ~ 0 + hgbrat + lymph_percent, family = "bernoulli")

thiomon_priors_default_priors_separate <- get_prior(ref_thiomon_formulae_separate, data = train_data)

(thiomon_set_priors_informative_separate <- c(
  prior(
    normal(-2,5),
    class = "b"
  ),

  prior(
```

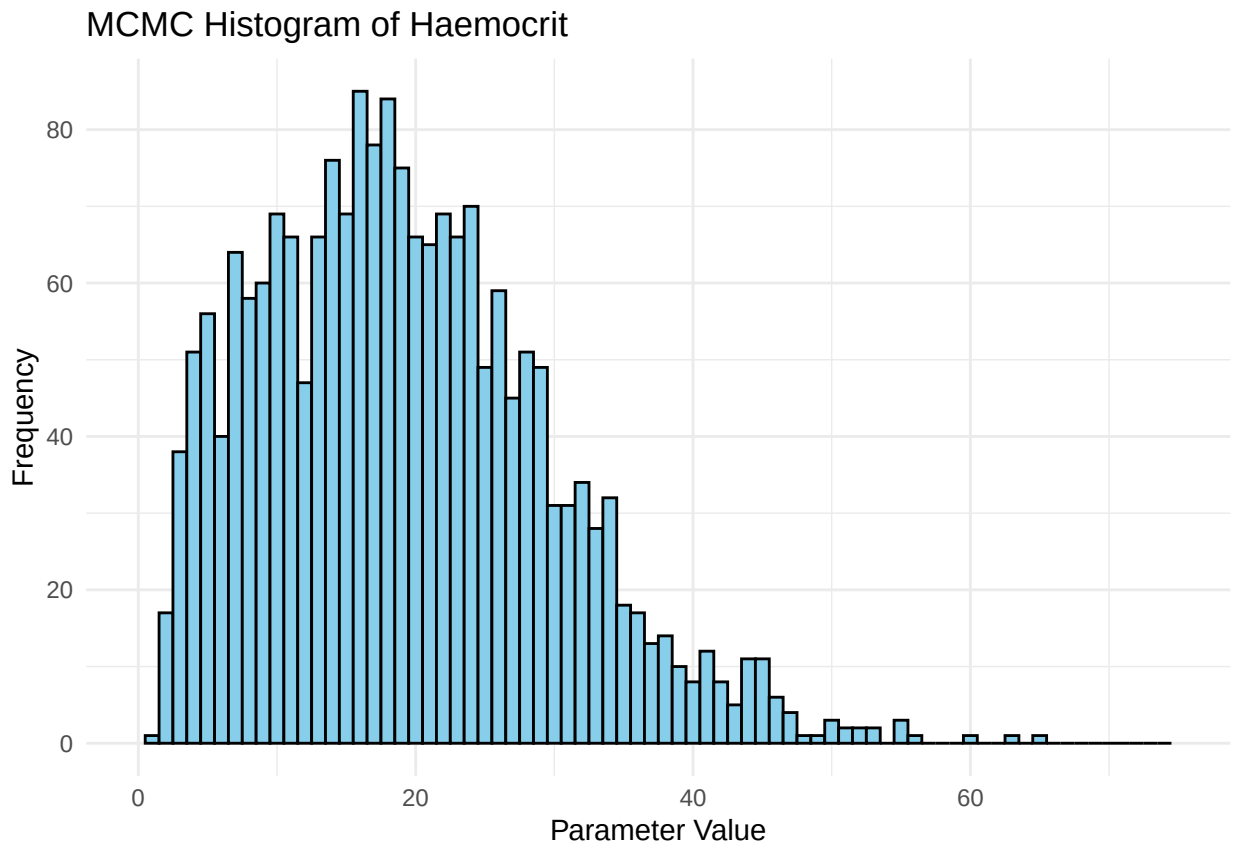
```
    normal(0,1),
    class = "b",
    coef = "lymph_percent"
  )
))

thiomon_data_model_separate <- brm(
    formula = ref_thiomon_formulae_separate,
    prior = thiomon_set_priors_informative_separate,
    data = train_data,
    iter = 2000,
    warmup = 1000,
    chains = 4,
    control = list(
      adapt_delta = 0.9,
      max_treedepth = 20
    ),
    threads = threading(2), cores = 8, backend = "cmdstanr"
)
})})
```

3.2.1 Checking if the covariates are normally distributed

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



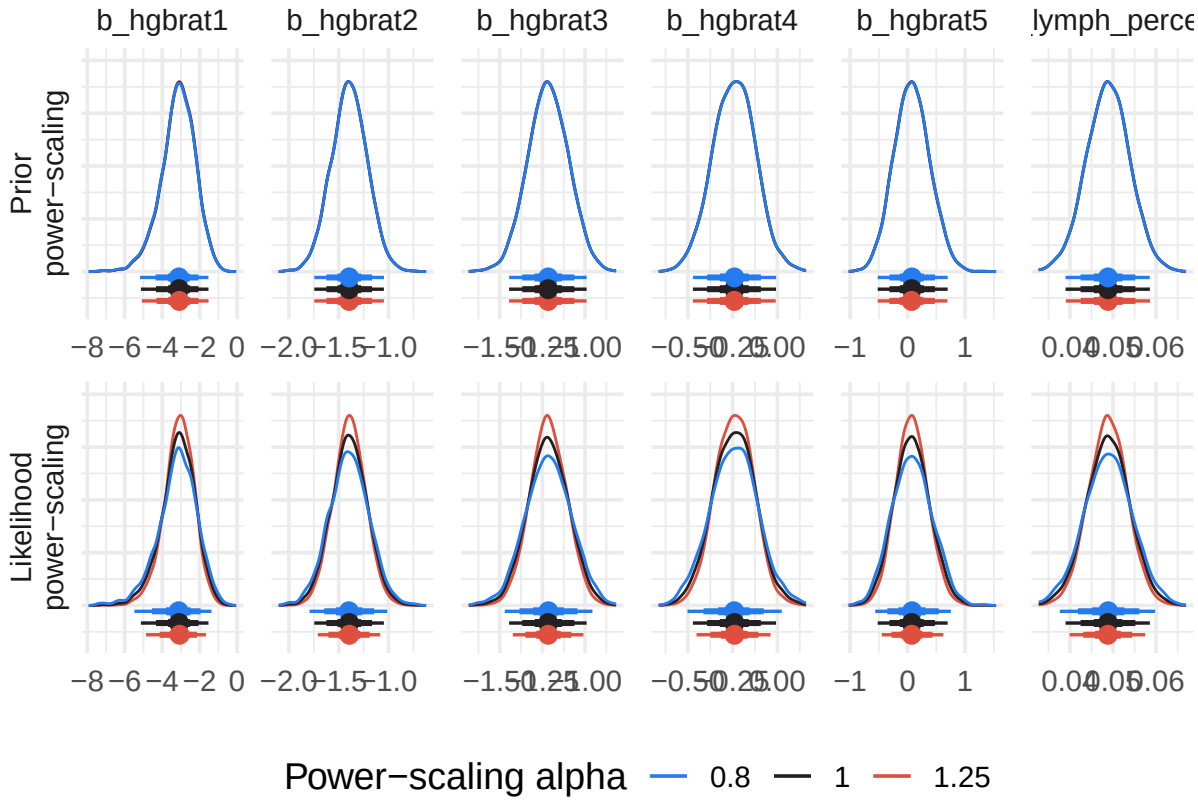
Both the parameters demonstrate a normal distribution around their respective means. We can approximate with normal distribution for all the parameters.

We don't use flat priors as they are uninformative and we have lots of data to give us good reference values. Also with flat priors, posterior prior checking tools due to input containing infinite or NA values as there is a constant tail. We notice also that the entire population parameters are assigned the same priors. We want to change it. Also, notice that we don't have a population level Intercept.

3.2.2 Checking prior sensitivity

With our chosen priors (we used the default priors first and then used the posterior of the slopes to judge what we need), we expect to see no prior sensitivity and we are right. Power-scaling with cumulative Jensen-Shannon distance diagnostic indicates no prior data conflict.

variable	prior	likelihood	diagnosis
$b_h gbrat1$	0.029	0.12	-
$b_h gbrat2$	0.0012	0.096	-
$b_h gbrat3$	0.00086	0.09	-
$b_h gbrat4$	0.0011	0.086	-
$b_h gbrat5$	0.005	0.081	-
$b_l ymph_p ercent$	0.00087	0.087	-



3.2.3 R-hat and Effective Sample Size (ESS)

First checking the model for any discrepancies in R-hat and ESS before proceeding any further.

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: remission ~ 0 + hgbrat + lymph_percent
##    Data: train_data (Number of observations: 2000)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## hgbrat1          -4.21      1.19    -6.92    -2.20 1.00     2515     1949
## hgbrat2          -1.22      0.18    -1.57    -0.87 1.00     2673     2610
```

```
## hgbrat3          -1.11      0.12    -1.33    -0.88 1.00     1703     1892
## hgbrat4          -0.12      0.12    -0.36     0.11 1.00     1756     1984
## hgbrat5           0.52      0.34    -0.12     1.19 1.00     2776     2579
## lymph_percent     0.05      0.00     0.04     0.06 1.00     1583     1947
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

R-hat and effective sample sizes look good. At convergence also R-hat value is 1 means the chains have converged well.

```
##
## Computed from 4000 by 2000 log-likelihood matrix.
##
##          Estimate   SE
## elpd_loo  -1258.3 15.7
## p_loo         6.6  1.5
## looic      2516.5 31.4
## ------
## MCSE of elpd_loo is 0.1.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.4, 1.4]).
##
## All Pareto k estimates are good (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

3.2.4 Issue with Pareto K-hat and Solution

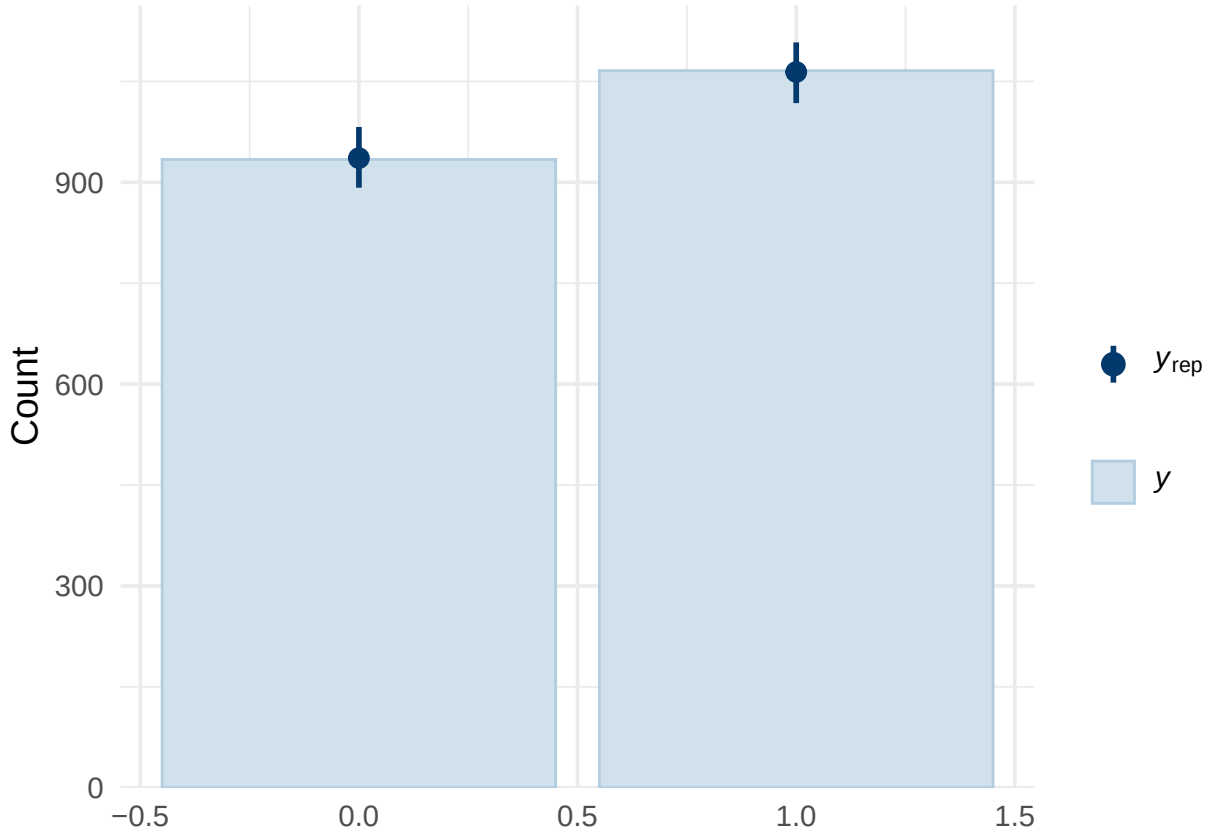
```
##
## Computed from 4000 by 2000 log-likelihood matrix.
##
##          Estimate   SE
## elpd_loo  -1258.3 15.7
## p_loo         6.6  1.5
## looic      2516.5 31.4
## ------
## MCSE of elpd_loo is 0.1.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.4, 1.4]).
##
## All Pareto k estimates are good (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

In case there are complaints about Pareto k-estimates, we use the following command and refit the model by removing the observations causing the conflict. Here the model was refit 1 times as there was 1 value with a value between 0.7 and 1.

```
## No problematic observations found. Returning the original 'loo' object.
```

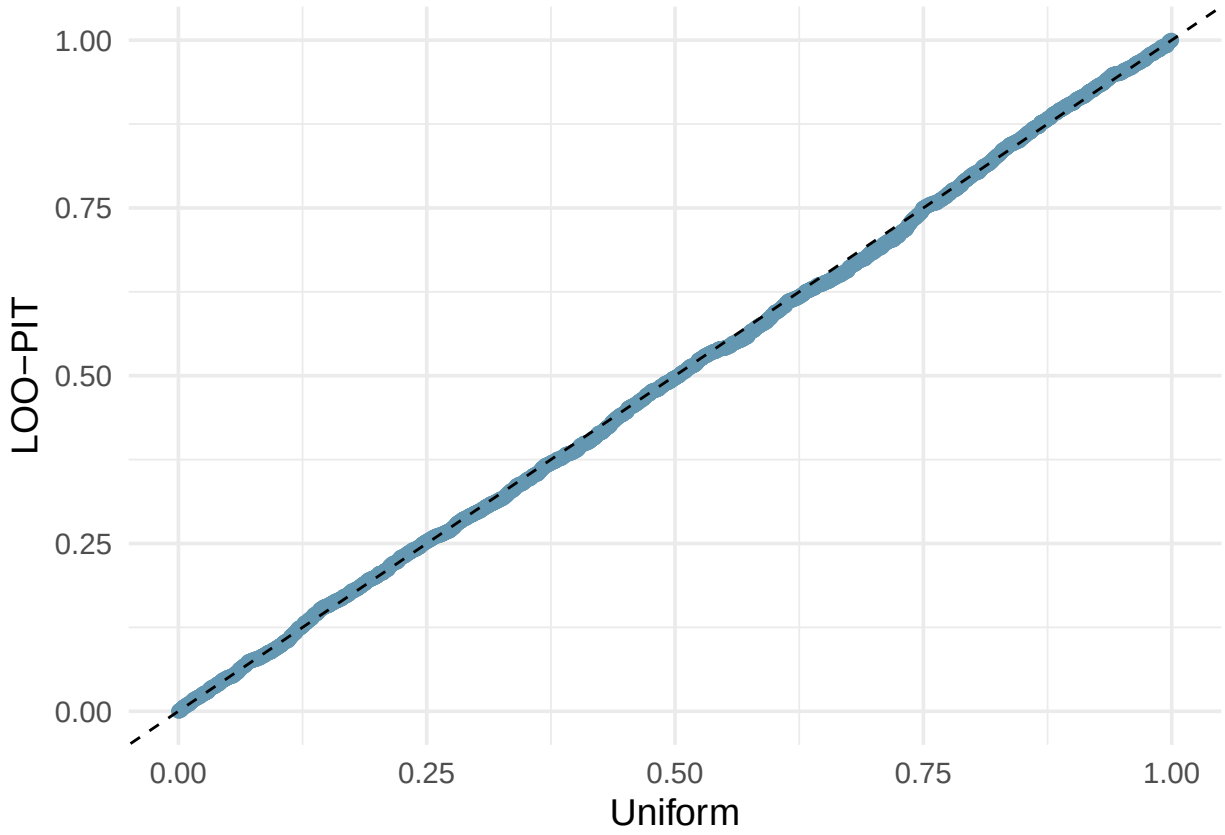
The Pareto k-estimates indicate there are no observations that have k $>$ 0.7. Posterior predictive checks seem fine as well. Bars plots seem to fit well.

3.2.5 Posterior Predictive Checking



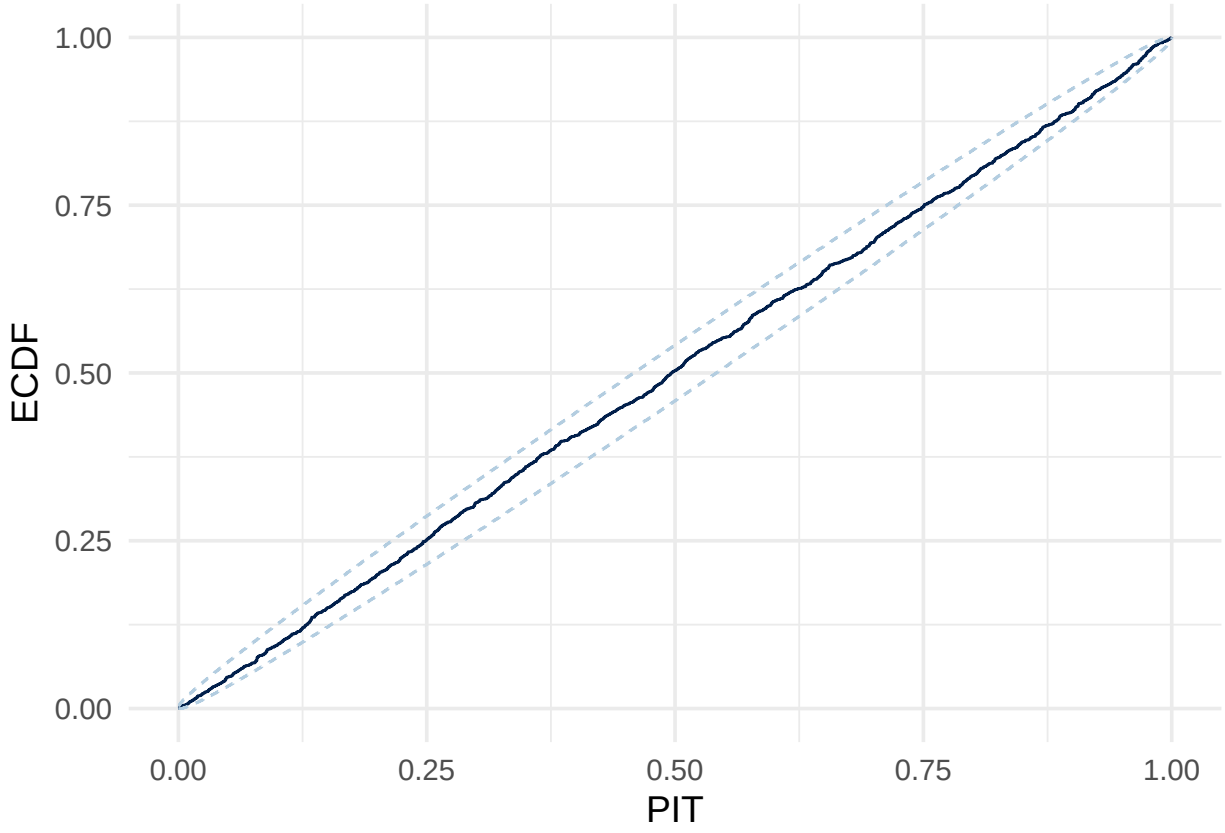
LOO-PIT check seems fine as well.

```
## Recomputing 'loo' for model '.x1'
```



LOO-PIT ECDF is within the recommendation.

variable	mean	median	sd	mad	q5	q95	rhat	ess$_b$*ulk*	ess$_t$*ail*
b$_h$*gbrat*1	-4.2	-4.1	1.2	1.1	-6.4	-2.5	1	2515	1949
b$_h$*gbrat*2	-1.2	-1.2	0.18	0.17	-1.5	-0.92	1	2673	2610
b$_h$*gbrat*3	-1.1	-1.1	0.12	0.12	-1.3	-0.91	1	1703	1892
b$_h$*gbrat*4	-0.12	-0.13	0.12	0.12	-0.32	0.076	1	1756	1984
b$_h$*gbrat*5	0.52	0.51	0.34	0.33	-0.027	1.1	1	2776	2579
b$_l$*ymph*$_p$*ercent*	0.045	0.045	0.005	0.0051	0.037	0.053	1	1583	1947



3.2.6 Model parameters

The slopes for the covariates can be listed using the following.

3.3. Hierarchical model

Let's try a hierarchical model. In 'hier_1' there is no global intercept but in 'hier_2' there is.

3.3.1. Hierarchical model without global concept

```
invisible({capture.output({
ref_thiomon_formulae_hier_1 <- bf(remission ~ 0 + lymph_percent + (1 | hgbrat), family = "bernoulli")

thiomon_priors_default_priors_hier_1 <- get_prior(ref_thiomon_formulae_hier_1, data = train_data)
```

```
thiomon_set_priors_informative_hier_1 <- c(

  prior(
    normal(2.5,4),
    class = "sd"
  ),

  prior(
    normal(0,1),
    class = "b",
    coef = "lymph_percent"
  )
)

thiomon_data_model_hier_1 <- brm(
    formula = ref_thiomon_formulae_hier_1,
    prior = thiomon_set_priors_informative_hier_1,
    data = train_data,
    iter = 2000,
    warmup = 1000,
    chains = 4,
    control = list(
      adapt_delta = 0.95,
      max_treedepth = 20
    ),
    threads = threading(2), cores = 8, backend = "cmdstanr"
)
})})
```

3.3.2. Hierarchical model with global concept

Note we have also set an informative prior on the Intercept as flat priors cause problems in loo checking due to Pareto K-hat values.

```
invisible({capture.output({
ref_thiomon_formulae_hier_2 <- bf(remission ~ 1 + lymph_percent + (1 | hgbrat), family = "bernoulli")

thiomon_priors_default_priors_hier_2 <- get_prior(ref_thiomon_formulae_hier_2, data = train_data)

thiomon_set_priors_informative_hier_2 <- c(

  prior(
    normal(-1.5,2),
    class = "Intercept"
  ),

  prior(
    normal(1.5,3),
    class = "sd"
  ),

  prior(
    normal(0,1),
    class = "b",
```

```
    coef = "lymph_percent"
  )
)

thiomon_data_model_hier_2 <- brm(
    formula = ref_thiomon_formulae_hier_2,
    prior = thiomon_set_priors_informative_hier_2,
    data = train_data,
    iter = 2000,
    warmup = 1000,
    chains = 4,                          # Number of chains
    control = list(
      adapt_delta = 0.95,                # Increase acceptance rate target if needed
      max_treedepth = 20                 # Increase the tree depth if needed
    ),
    threads = threading(2), cores = 8, backend = "cmdstanr"
)
})})
```

3.3.3 R-hat and Effective Sample Size (ESS)

R-hat and ESS values look good. Convergence at the end of the run indicates R-hat is 1.0 which means samples are reliable and can be used for inference. Also, posterior prior sensitivity is well adjusted.

```
## Sensitivity based on cjs_dist:
## # A tibble: 7 x 4
##   variable                 prior likelihood diagnosis
##   <chr>                    <dbl>      <dbl> <chr>
## 1 b_lymph_percent       0.00331      0.107  -
## 2 sd_hgbrat__Intercept  0.0366       0.0575 -
## 3 r_hgbrat[1,Intercept] 0.00246      0.117  -
## 4 r_hgbrat[2,Intercept] 0.00133      0.0956 -
## 5 r_hgbrat[3,Intercept] 0.00203      0.0996 -
## 6 r_hgbrat[4,Intercept] 0.00205      0.115  -
## 7 r_hgbrat[5,Intercept] 0.000787     0.0803 -
```

The prior sensitivity caused some issues for us. There were a few divergent transactions reported (less than 10) but R-hat values were <= 1.01 for all the parameters. Also, the model reported that R-hat values converged at the end of the simulation. When there are those many divergent transactions, we could have fixed them by increasing the 'adapt_delta' but we chose to check the priors and see if they could be tuned. When tuned too tight the diagnostic would reveal a prior-data conflict and when too wide it would complain about likelihood and prior conflict. We had to iterate to get the best balance of reducing the divergent transactions and the prior conflict. The values -1.5 for Intercept and 1.5 for sd are a testament to this iterative checking.

The following commands are generic for both the hierarchical model choices. We continue with hier_2 as both the models have almost the same EPLD value (negligible difference as expected).

```
## Sensitivity based on cjs_dist:
## # A tibble: 9 x 4
##   variable                 prior likelihood diagnosis
##   <chr>                    <dbl>      <dbl> <chr>
## 1 b_Intercept           0.0369       0.0208 -
## 2 b_lymph_percent       0.00169      0.0968 -
## 3 sd_hgbrat__Intercept  0.0453       0.0411 -
## 4 Intercept             0.0378       0.0214 -
## 5 r_hgbrat[1,Intercept] 0.0284       0.0527 -
```

```
## 6 r_hgbrat[2,Intercept] 0.0368      0.0208 -
## 7 r_hgbrat[3,Intercept] 0.0380      0.0209 -
## 8 r_hgbrat[4,Intercept] 0.0375      0.0217 -
## 9 r_hgbrat[5,Intercept] 0.0325      0.0269 -
```

3.3.4 Issue with Pareto K-hat and Solution

```
##
## Computed from 4000 by 2000 log-likelihood matrix.
##
##          Estimate   SE
## elpd_loo  -1258.1 15.2
## p_loo         6.1  1.0
## looic      2516.2 30.4
## ------
## MCSE of elpd_loo is 0.1.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.3, 1.2]).
##
## All Pareto k estimates are good (k < 0.7).
## See help('pareto-k-diagnostic') for details.
```

```
## Warning: Found 1 observations with a pareto_k > 0.7 in model
## 'thiomon_data_model_hier_2'. We recommend to set 'moment_match = TRUE' in order
## to perform moment matching for problematic observations.
```

```
##
## Computed from 4000 by 2000 log-likelihood matrix.
##
##          Estimate   SE
## elpd_loo  -1258.3 15.3
## p_loo         6.4  1.1
## looic      2516.6 30.7
## ------
## MCSE of elpd_loo is NA.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.5, 1.1]).
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1999  100.0%  1995
##    (0.7, 1]   (bad)         1    0.0%  <NA>
##    (1, Inf)   (very bad)    0    0.0%  <NA>
## See help('pareto-k-diagnostic') for details.
```

Since there were no values with problematic K-hat the following is not required.

```
thiomon_data_model_hier_1 <- add_criterion(thiomon_data_model_hier_1, criterion='loo', reloo=TRUE)
```

```
## No problematic observations found. Returning the original 'loo' object.
```

```
thiomon_data_model_hier_22 <- add_criterion(thiomon_data_model_hier_2, criterion='loo', reloo=TRUE)
```

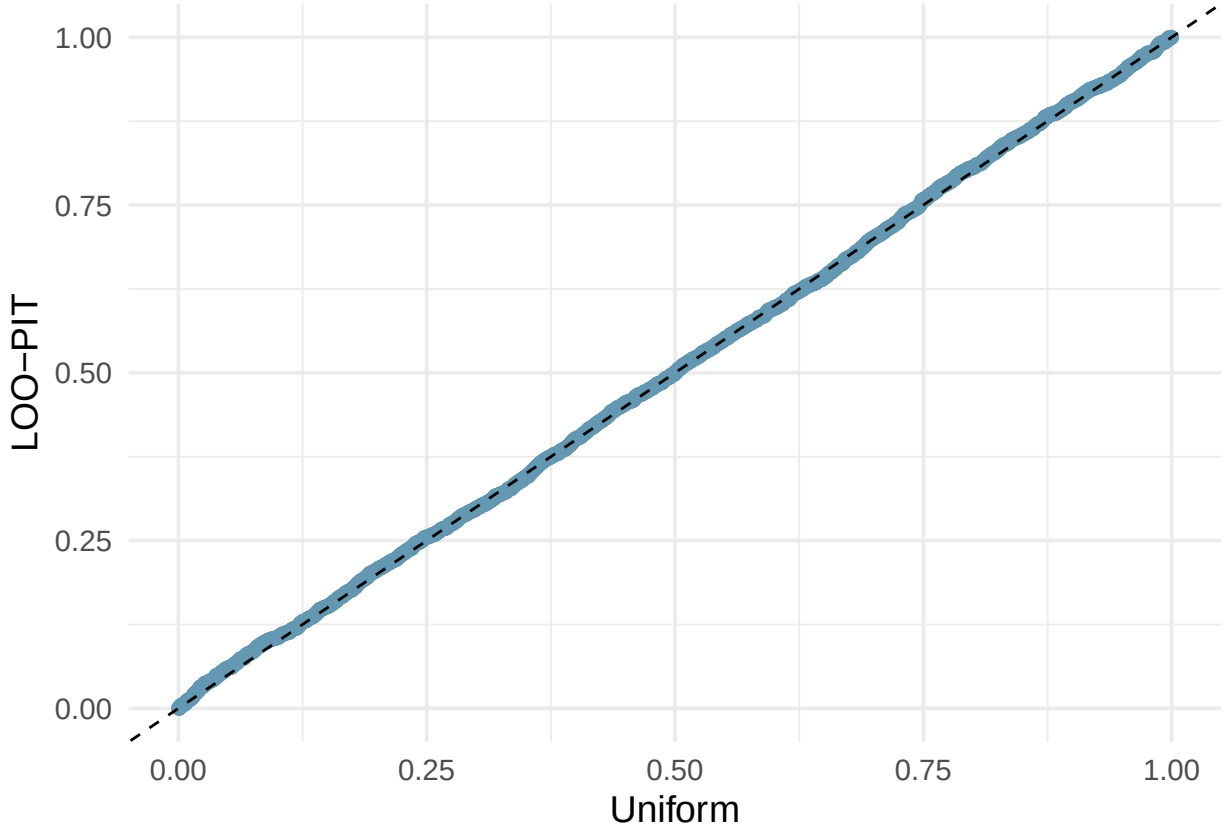
```
## 1 problematic observation(s) found.
## The model will be refit 1 times.
```

```
## Running MCMC with 4 sequential chains, with 2 thread(s) per chain...
##
## Chain 1 finished in 40.8 seconds.
## Chain 2 finished in 40.0 seconds.
```

```
## Chain 3 finished in 41.7 seconds.
## Chain 4 finished in 50.3 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 43.2 seconds.
## Total execution time: 174.5 seconds.
##
## Fitting model 1 out of 1 (leaving out observation 868)
## Start sampling
```

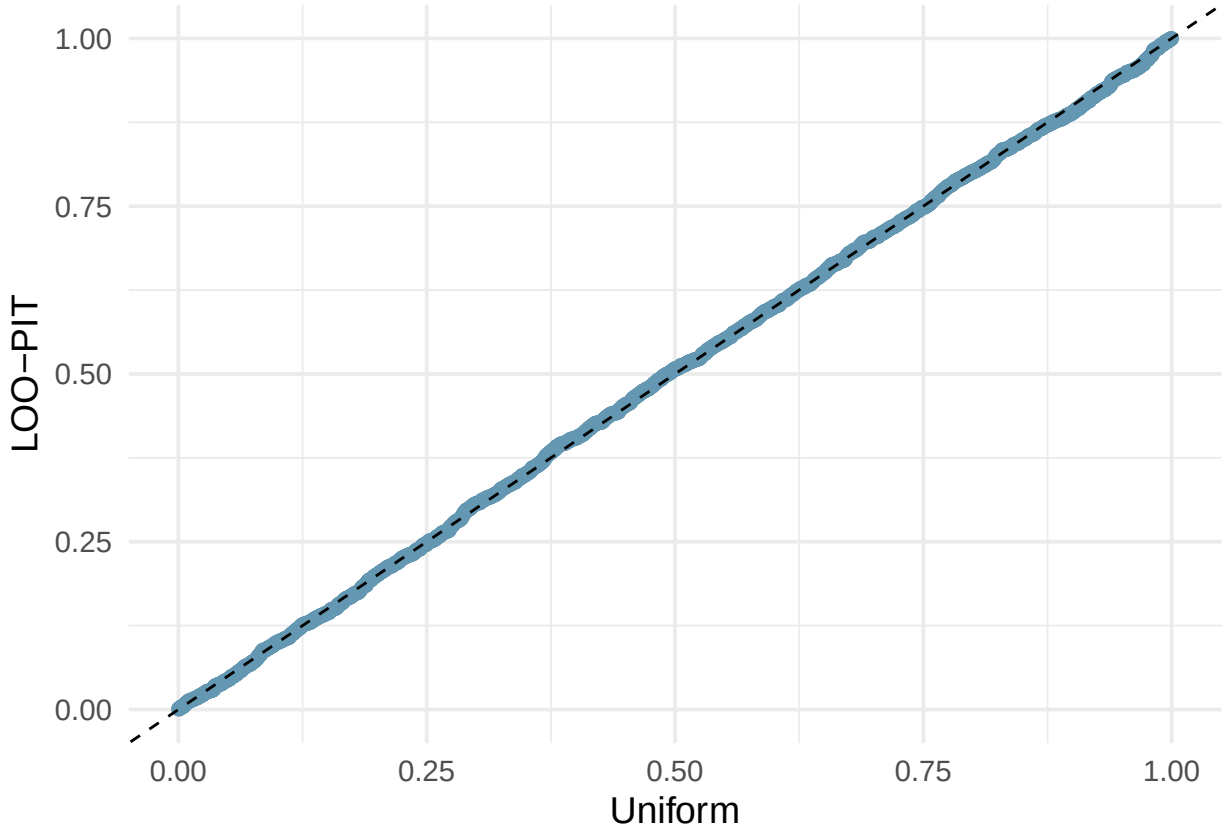
3.3.5 LOO-PIT check

```
## Recomputing 'loo' for model '.x1'
```




```
## Warning: Found 1 observations with a pareto_k > 0.7 in model '.x1'. We
## recommend to set 'moment_match = TRUE' in order to perform moment matching for
## problematic observations.
```

model	elpd$_diff$	se$_diff$
thiomon$_data_model_hier_1$	0	0
thiomon$_data_model_separate$	-0.15	1.3
thiomon$_data_model_hier_2$	-0.2	0.33
thiomon$_data_model_pooled$	-122	15



Try to see if a better hierarchical model fits. The 'sd' prior required a bit of tweaking. Keeping it wider results in 1 divergent transaction which we try to avoid by making it narrow but not quite narrow that we get a prior-data conflict. With this balancing, there are no problems we spot.

There is no issue any more of prior sensitivity.

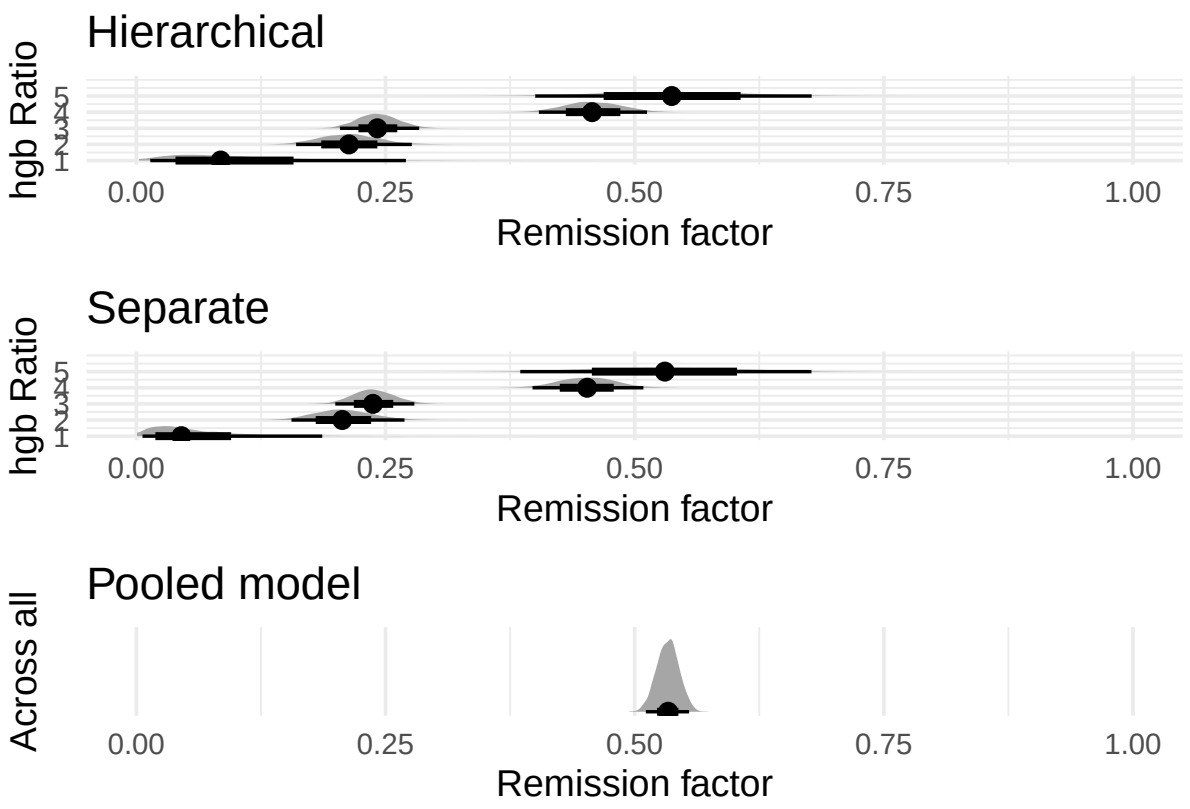
3.4 Pooled, Separate, and Hierarchical Model Comparison

3.4.1 LOO comparing all the 4 models

```
## Warning: Found 1 observations with a pareto_k > 0.7 in model
## 'thiomon_data_model_hier_2'. We recommend to set 'moment_match = TRUE' in order
## to perform moment matching for problematic observations.
```

The best-performing model is the hierarchical, closely followed by the separate with hardly much difference in EPLD (less than 0.5), and finally the pooled (with an EPLD score 18 points below both of the hierarchical and separate models).

3.4.2 Model posterior distributions



The reasons are mostly clear - the separate and hierarchical models perform almost the same but the pooled model does not factor in any covariates and hence only relies on the remission bits as information and hence it has poor predictive power. The close performance of the separate and hierarchical models is because the patients segregated by the 'hgbrat' do share some information mutually in the hierarchical model but this is not much which means that the predicted remissions are not widely varying.

3.4.3 Bayes and LOO R2 checks

The pooled model has negligible values as they don't have any covariates factored in. The separate and hierarchical models do have 14% and 15% explanatory power of the variance of the data. Again this shows the separate and hierarchical models do not differ much here.

The LOO-R2 and Bayes-R2 values both match for all the models indicating that there is no likely over-fitting or under-fitting of the data.

3.5. Non-Bayesian model

Besides the implementation of the Bayesian model, we would like to introduce the machine learning aspect to the model as well for comparison between the Bayesian approach and the simple machine learning approach, while finding and evaluating the correlation between model parameters for saving computational resources and suggesting the observation that we can try in Bayesian model as well. General Linear model (GLM) will be described with the following parameters based on the abbreviation of dataset description /cite something here. In general, the model will use all of the parameters of the dataset, without any modification, for comparison only.

```
model_glm <- glm(remission ~ 0 + days_of_life + wbc + hgb + hct + plt + rbc + mcv + mch + mchc + rdw + r
```

4. Evaluation results

After building the model, due to the objective of the model being classification, we can use the classification evaluation metrics aspect for the evaluation of the model result. Two suggestions that we have implemented: using a confusion matrix with evaluation metrics mathematical calculation, and using ROC curves with AUC values for different model comparisons, which can be elaborated on in session 4.1. and 4.2.

4.1. Confusion matrix and Evaluation metrics results

Confusion matrix is a common method for binary classification tasks, where the prediction will be built in the augmented matrix, describing the actual status of the model prediction. Here are the detailed results for each model:

4.1.1. Pooled Model

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls > cases
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 168 210
##          1  98  81
##
##                Accuracy : 0.447
##                  95% CI : (0.4052, 0.4894)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 0.9998
##
##                   Kappa : -0.0885
##
##  Mcnemar's Test P-Value : 2.535e-10
##
##             Sensitivity : 0.6316
##             Specificity : 0.2784
##          Pos Pred Value : 0.4444
##          Neg Pred Value : 0.4525
##              Prevalence : 0.4776
##          Detection Rate : 0.3016
##    Detection Prevalence : 0.6786
##       Balanced Accuracy : 0.4550
##
##        'Positive' Class : 0
##
```

The accuracy of the pooled model is close to random guessing, as indicated by the balanced accuracy of approximately 50%

4.1.2. Separate model

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 207 163
##          1  59 128
##
##                Accuracy : 0.6014
##                  95% CI : (0.5594, 0.6424)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 0.0001056
##
##                   Kappa : 0.2145
##
##  Mcnemar's Test P-Value : 4.748e-12
##
##             Sensitivity : 0.7782
##             Specificity : 0.4399
##          Pos Pred Value : 0.5595
##          Neg Pred Value : 0.6845
##              Prevalence : 0.4776
##          Detection Rate : 0.3716
##    Detection Prevalence : 0.6643
##       Balanced Accuracy : 0.6090
##
##        'Positive' Class : 0
##
```

The separate model performs better than the pooled model. The model has a higher sensitivity, making it better at identifying true positives.

4.1.3. Hierarchical model with global concept

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 192 150
##          1  74 141
##
##                Accuracy : 0.5978
```

```
##                  95% CI : (0.5558, 0.6389)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 0.0002048
##
##                   Kappa : 0.2038
##
##  Mcnemar's Test P-Value : 5.411e-07
##
##             Sensitivity : 0.7218
##             Specificity : 0.4845
##          Pos Pred Value : 0.5614
##          Neg Pred Value : 0.6558
##              Prevalence : 0.4776
##          Detection Rate : 0.3447
##    Detection Prevalence : 0.6140
##       Balanced Accuracy : 0.6032
##
##        'Positive' Class : 0
##
```

This hierarchical model has an accuracy of nearly 60% (which may be changed when knitting the docume). The sensitivity is higher than the specificity, indicating that the model is better at identifying true positives.

4.1.4. Hierarchical model without global concept

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 214 190
##          1  52 101
##
##                Accuracy : 0.5655
##                  95% CI : (0.5232, 0.6071)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 0.02295
##
##                   Kappa : 0.1483
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.8045
##             Specificity : 0.3471
##          Pos Pred Value : 0.5297
##          Neg Pred Value : 0.6601
##              Prevalence : 0.4776
##          Detection Rate : 0.3842
##    Detection Prevalence : 0.7253
##       Balanced Accuracy : 0.5758
```

```
##
##        'Positive' Class : 0
##
```

Hierarchical model 2 performs approximately with hierarchical model 1. The sensitivity is higher, indicating better identification of true positives.

4.1.5. GLM Model

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

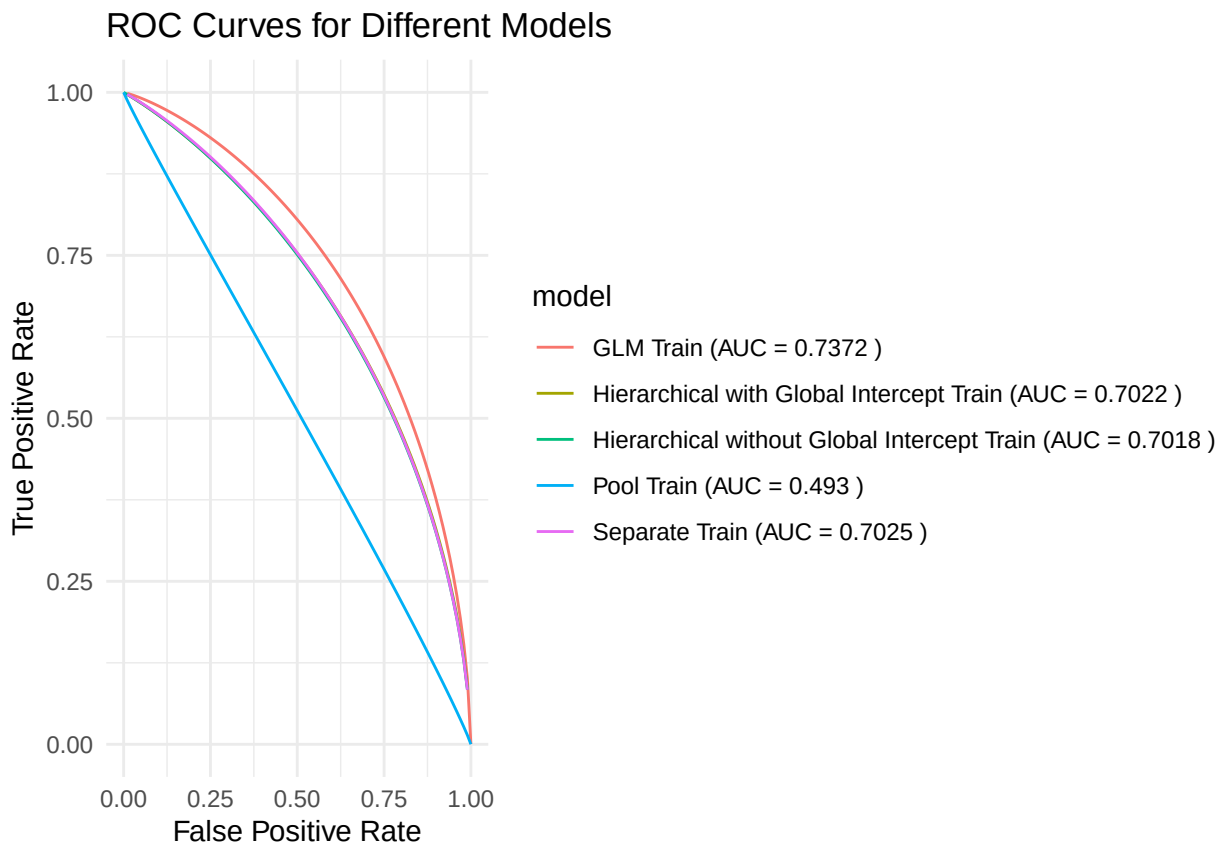
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 174  91
##          1  92 200
##
##                Accuracy : 0.6715
##                  95% CI : (0.6307, 0.7104)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 7.077e-13
##
##                   Kappa : 0.3415
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.6541
##             Specificity : 0.6873
##          Pos Pred Value : 0.6566
##          Neg Pred Value : 0.6849
##              Prevalence : 0.4776
##          Detection Rate : 0.3124
##    Detection Prevalence : 0.4758
##       Balanced Accuracy : 0.6707
##
##        'Positive' Class : 0
##
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

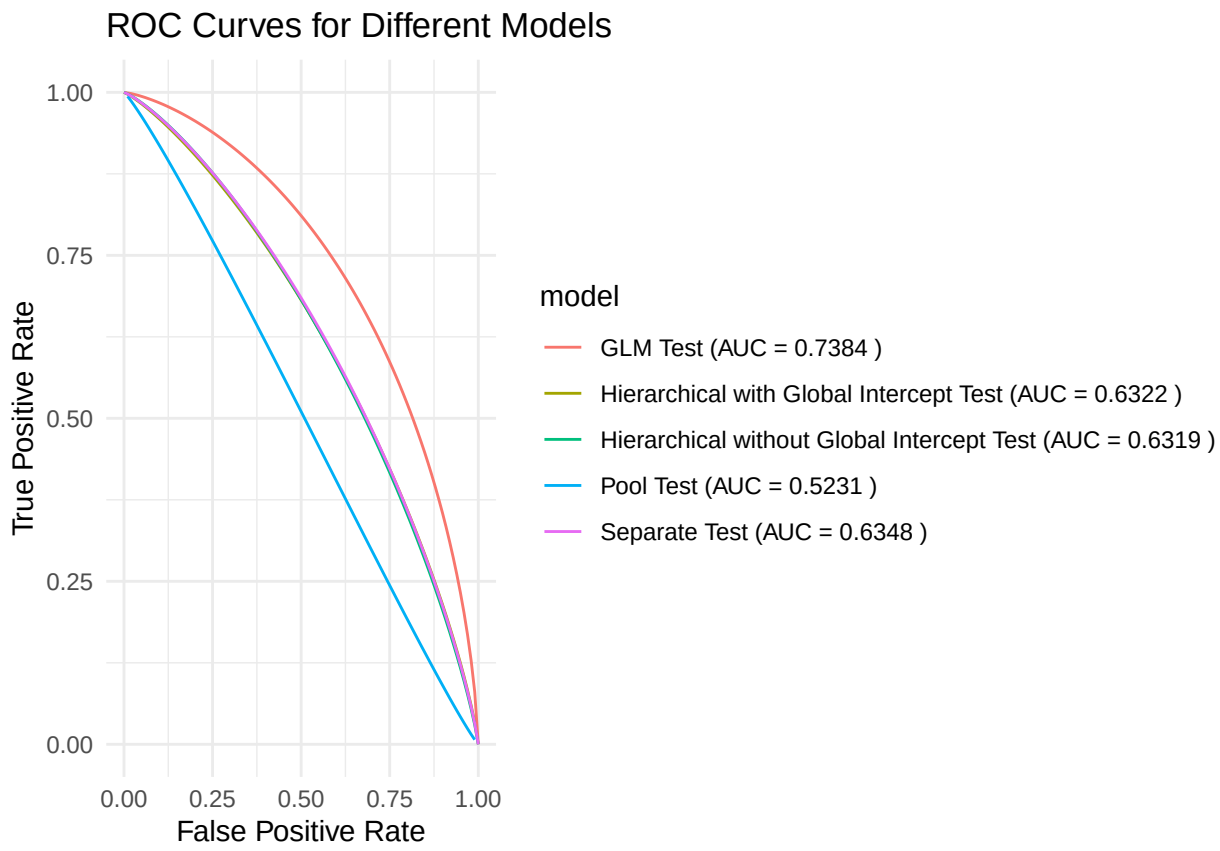
The GLM model outperforms all other models with an accuracy of over 65%. The model has a good balance between sensitivity and specificity, making it effective at identifying both true positives and true negatives.

4.2. ROC curves and AUC values

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



According to the evaluation result, we could see that almost all the models have an AUC difference between training and testing data maximum of 7%, which can be an acceptable value. Moreover, the Bayesian approach cannot be better than the GLM approach. One of the proposals for this is because of having less occupied parameters in the model. We can base on that to improve our model as well.

4.3. Discussion.

According to the evaluation results, the GLM model outperforms the Bayesian models in terms of accuracy and AUC. Moreover, as expected from the Bayesian analysis, the difference between Separate and Hierarchical here is not clear. The Bayesian models show moderate performance but do not surpass the GLM model.

5. Discussion of Issues and Potential Improvements

The main issue observed is the moderate performance of the Bayesian models compared to the GLM model. To improve the accuracy of the Bayesian models, we can consider increasing the number of variables, introducing different methods for approaching the prior, or normalizing the parameters into different categories and medical terms. This can help improve the model's accuracy and predictive power.

5.1. Improving the number of training sample

An example of applying 4500 variables into a separate model.

```
invisible({capture.output({
ref_thiomon_formulae_separate <- bf(remission ~ 0 + hgbrat + lymph_percent, family = "bernoulli")

thiomon_priors_default_priors_separate <- get_prior(ref_thiomon_formulae_separate, data = train_data)
```

```
(thiomon_set_priors_informative_separate <- c(
  prior(
    normal(-2,5),
    class = "b"
  ),

  prior(
    normal(0,1),
    class = "b",
    coef = "lymph_percent"
  )
))

thiomon_data_model_separate_improvement <- brm(
    formula = ref_thiomon_formulae_separate,
    prior = thiomon_set_priors_informative_separate,
    data = train_data,
    iter = 2000,
    warmup = 1000,
    chains = 4,                            # Number of chains
    control = list(
      adapt_delta = 0.9,                  # Increase acceptance rate target if needed
      max_treedepth = 20                  # Increase the tree depth if needed
    ),
    threads = threading(2), cores = 8, backend = "cmdstanr"
)
})})
```

All of the Bayesian check methods have been produced internally and did not provide any problems compared to the previous model. Therefore, we could come to an evaluation session

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 216 188
##          1  50 103
##
##                Accuracy : 0.5727
##                  95% CI : (0.5304, 0.6142)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 0.009715
##
##                   Kappa : 0.1624
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8120
```

```
##             Specificity : 0.3540
##          Pos Pred Value : 0.5347
##          Neg Pred Value : 0.6732
##              Prevalence : 0.4776
##          Detection Rate : 0.3878
##    Detection Prevalence : 0.7253
##       Balanced Accuracy : 0.5830
##
##        'Positive' Class : 0
##
```

Based on observation, the actual accuracy of the model here is larger, but not significant. However, the generalization of the model is better due to the decrease in the sensitivity and increase and specificity. This is a good sign for the model as well.

Moreover, a comparison between using less and more data in ROC/AUC values can be presented below.

5.2. Increasing the number of parameters in the original model.

The model here can be computed with more parameters, with the purpose of producing a better result. However, using more parameters can also lead to an increase in the model complexity, which may not be appropriate for the task's project. Therefore, an example of changing the separate model formula with adding 4 more parameters hct, wbc, mpv, mchc into the model and we can observe the result.

```
invisible({capture.output({
ref_thiomon_formulae_separate <- bf(remission ~ 0 + hgbrat + lymph_percent + hct + wbc + mpv + mchc, fa

thiomon_priors_default_priors_separate <- get_prior(ref_thiomon_formulae_separate, data = train_data)

(thiomon_set_priors_informative_separate <- c(
  prior(
    normal(19.5,20),
    class = "b",
    coef = "lymph_percent"
  ),

  prior(
    normal(7.5,10),
    class = "b",
    coef = "wbc"
  ),

  prior(
    normal(34,5),
    class = "b",
    coef = "mchc"
  ),

  prior(
    normal(37.1,10),
    class = "b",
    coef = "hct"
  ),

  prior(
```

```
    normal(8.2,5),
    class = "b",
    coef = "mpv"
  )
))

thiomon_data_model_separate_improvement_1 <- brm(
    formula = ref_thiomon_formulae_separate,
    prior = thiomon_set_priors_informative_separate,
    data = train_data,
    iter = 2000,
    warmup = 1000,
    chains = 4,                          # Number of chains
    control = list(
      adapt_delta = 0.9,                # Increase acceptance rate target if needed
      max_treedepth = 20                # Increase the tree depth if needed
    ),
    threads = threading(2), cores = 8, backend = "cmdstanr"
)
})})
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 208 161
##          1  58 130
##
##                Accuracy : 0.6068
##                  95% CI : (0.5649, 0.6476)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 3.706e-05
##
##                   Kappa : 0.2249
##
##  Mcnemar's Test P-Value : 5.481e-12
##
##             Sensitivity : 0.7820
##             Specificity : 0.4467
##          Pos Pred Value : 0.5637
##          Neg Pred Value : 0.6915
##              Prevalence : 0.4776
##          Detection Rate : 0.3734
##    Detection Prevalence : 0.6625
##       Balanced Accuracy : 0.6143
##
##        'Positive' Class : 0
##
```

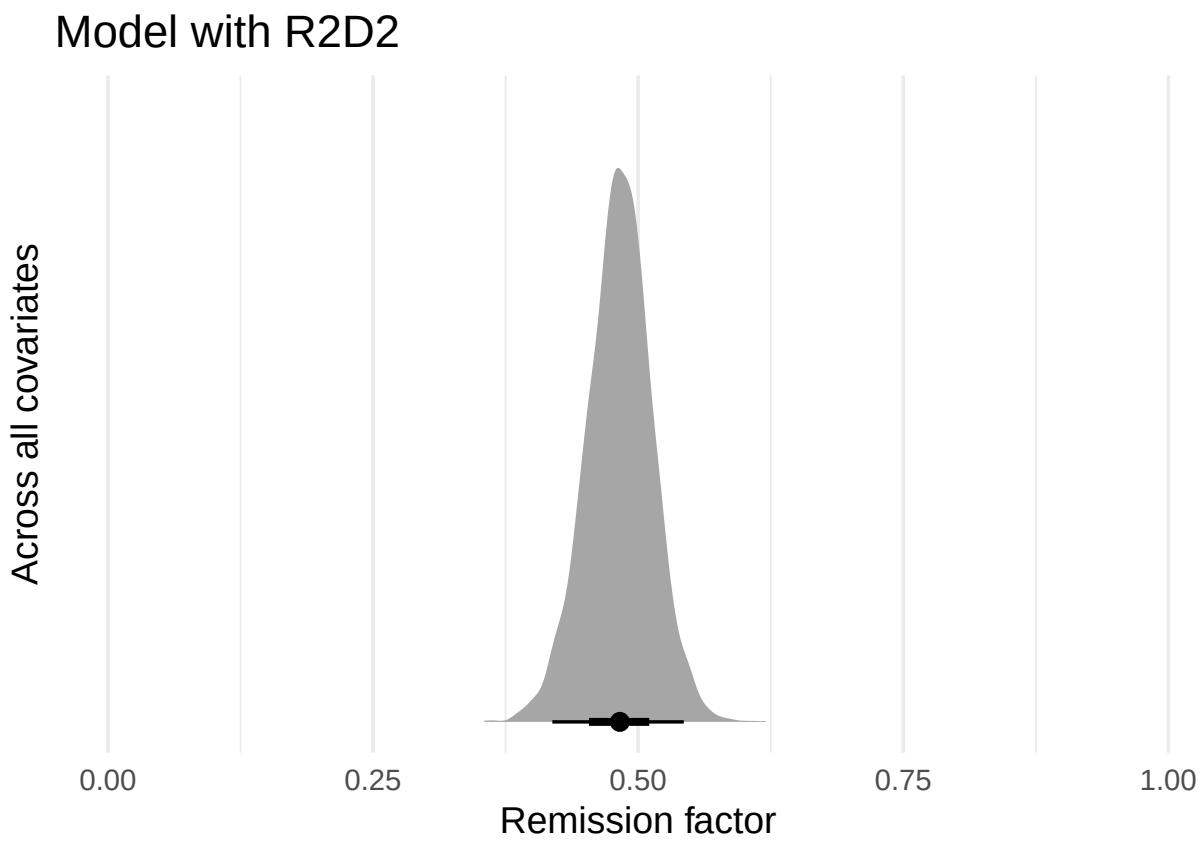
5.3. R2D2 method for the priors and using the full set of covariates for the modeling

Moreover, besides these trials that have been implemented, there can be room for improvement. An example is normalizing the parameters into medical terms for building a model with efficient parameters only. Furthermore, we have not tried to use with spline approach in the model, or with multi-parameters in hierarchical models. They are worth considering when we approach the problems and can be the necessary steps for improvement.

One improvement that we consider is to use the R2D2 method for the priors and use the full set of covariates for the modeling.

```
invisible({capture.output({
formula_r2d2 <- bf(remission ~ 0 + wbc + hgb + hct + plt + rbc + mcv + mch + mchc + rdw + mpv + neut_pe

fit_r2d2 <- brm(formula_r2d2, data = train_data,
            normalize = FALSE,
            prior=c(prior(R2D2(mean_R2 = 0.5, prec_R2 = 1, cons_D2 = 1,
                               autoscale = FALSE),class=b)),
            threads = threading(2), cores = 8, backend = "cmdstanr")
})})
```

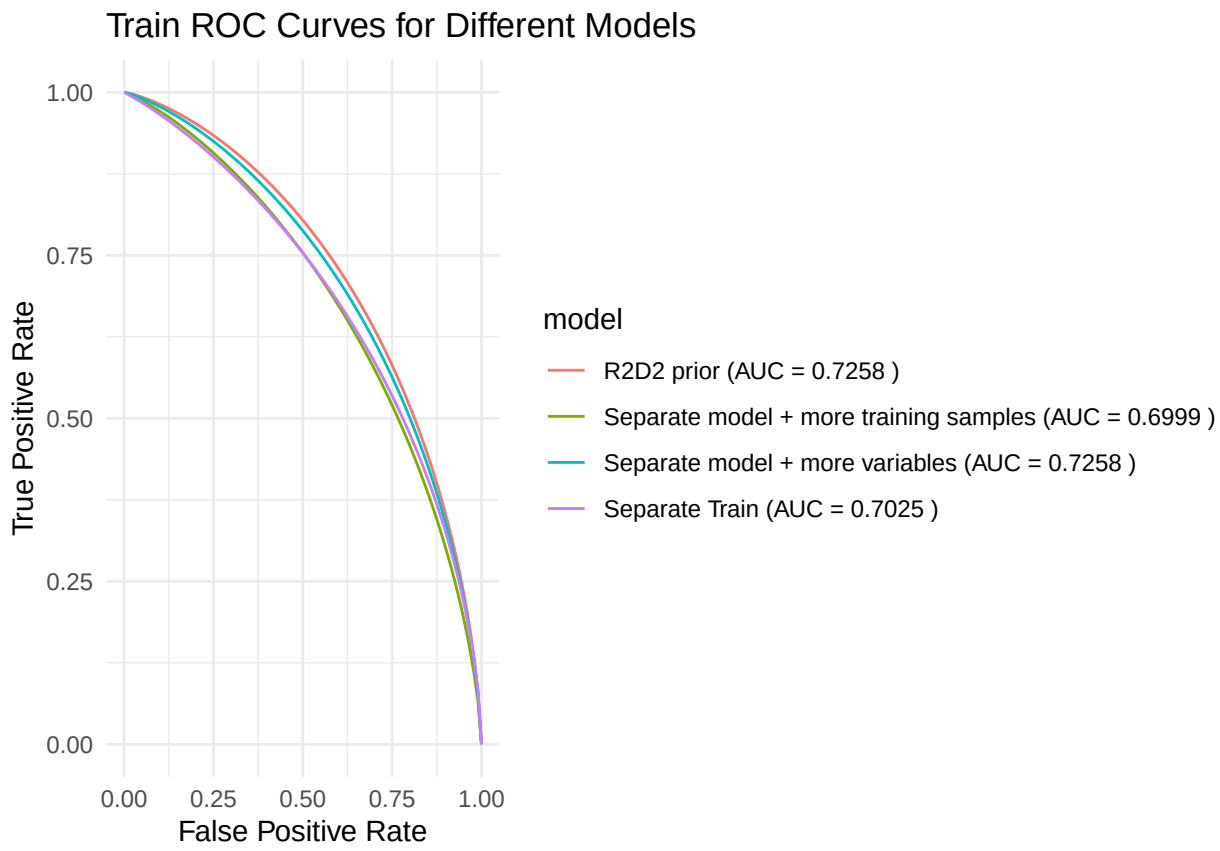
```
p_fit_r2d2 <- fit_r2d2 |> as_draws_df() |> as_draws_rvars() |> spread_rvars(b_wbc, b_hgb, b_hct, b_plt,
  ggplot(aes(xdist=inv_logit_scaled(mean_value), y=0)) +
  stat_halfeye() +
  scale_y_continuous(breaks=NULL) +
  labs(x='Remission factor', y='Across all covariates', title='Model with R2D2')
(p_fit_r2d2) * xlim(c(0.0,1.0))
```



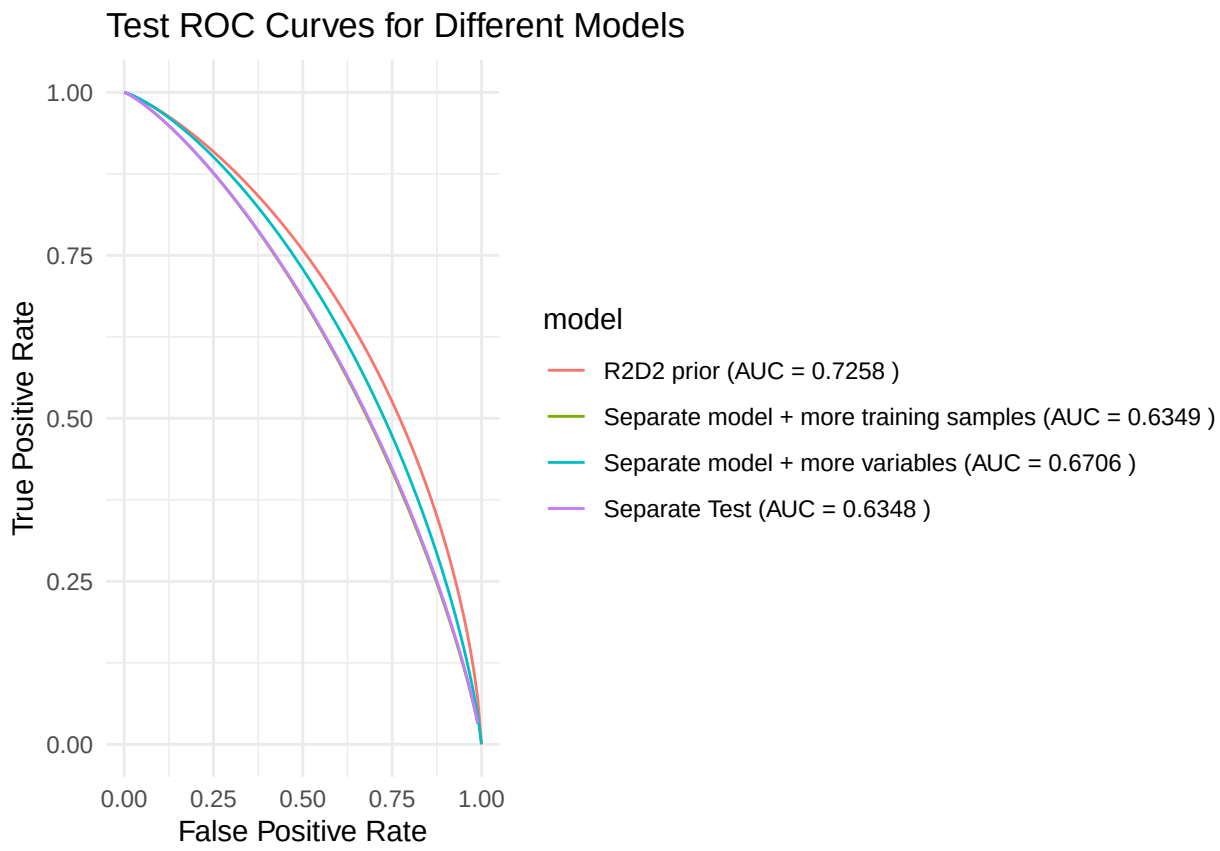
```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 177 109
##          1  89 182
##
##                Accuracy : 0.6445
##                  95% CI : (0.6032, 0.6843)
##     No Information Rate : 0.5224
##     P-Value [Acc > NIR] : 3.853e-09
##
##                   Kappa : 0.2899
##
##  Mcnemar's Test P-Value : 0.1769
##
##             Sensitivity : 0.6654
##             Specificity : 0.6254
##          Pos Pred Value : 0.6189
##          Neg Pred Value : 0.6716
##              Prevalence : 0.4776
##          Detection Rate : 0.3178
##    Detection Prevalence : 0.5135
##       Balanced Accuracy : 0.6454
##
##        'Positive' Class : 0
##
```

5.4. ROC curves/AUC values comparison

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```




```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



As you can observe, more training samples here do not print any differences, while more variables can lead to better model accuracy, having better ROC curves and AUC values. This can be a potential option for us to analyze and improve in the future.

6. Conclusion

After evaluating various models on the Thiomon dataset, we can conclude that the effects of the variables regarding remission are considerable. The Bayesian approach for this remission classification task is appropriate. However, the Bayesian approach is computationally complex, where increasing the number of parameters will lead to more time consumption during the fitting process, and the model accuracy will be better proportional to the time training. Therefore, we can expect more accuracy improvement when we build a larger model or use different Bayesian methods, for example, choosing different prior, or applying a hybrid approach in an actual research project.

Compared to these models, the hierarchical models, particularly the one with a global coefficient, provided a good balance between capturing global and group-level variations, leading to the best overall performance. The GLM models also showed promising results with fewer computational resources, which can also be a threshold for us to recompute the Bayesian model in a research project. Moreover, GLM also describes to us the potential of using more parameters in the Bayesian approach can lead to better results, which have also been experimented with in the Potential Improvements section.

7. Self-Reflection

Regarding the project, we have learned how to collaborate and help each other in the group. The project has been done with multiple trials and errors, with different parameter fitting. And the final result is the balance trade-off of accuracy and time training. In some cases, some evaluation metrics need knowledge, leading

to our discussion and revision of course content material and providing the opportunity for us to improve ourselves throughout the professor's project report. Even though the Bayesian classification result cannot be used in real research work and does not have a good accuracy compared to the other research publications, the application of Bayesian and how to evaluate the model can help us build our fundamental knowledge and inspire our future research path. Moreover, the process of doing the project together can also enhance our skills in teamwork and communication. We appreciate the opportunity from the course orientation to the final project.

References

1. Axelrad JE, Roy A, Lawlor G, Korelitz B, Lichtiger S. Thiopurines and inflammatory bowel disease: Current evidence and a historical perspective. World Journal of Gastroenterology. 2016;22(46):10103–17.
2. GitHub & BitBucket HTML preview. https://htmlpreview.github.io/?https://github.com/higgi13425/medicaldata/blob/master/man/description_docs/thiomon_desc.html;
3. Vehtari A. Collinear demo with mesquite bushes [Internet]. n.d. Available from: https://users.aalto.fi/~ave/modelselection/mesquite.html