Trying to predict the patient outcome based on baseline measurables at diagnosis

Bayesian Data Analysis D - 2023

Introduction

Randomized controlled trial (RCT) is a type of scientific experiment often used to test the effectiveness of a medical treatment or intervention. Firstly, in an RCT, patients are categorized randomly into an “experimental” and a “control” group. The experimental group gets the new treatment, while the control group is given a placebo. RCT has become a standard technique used in the drug-discovery pipeline.

Our dataset is from the first recorded RCT that was carried out in 1948 to assess the effectiveness of streptomycin on patients who have pulmonary tuberculosis [1]. The situation in this trial was unique as there were very limited medical supplies in the UK after World War II. This led the study to give treatment to only half of the patients. Secondly, an interesting observation is there is no ethics committee approval, nor any consent, which is consistent with practices. Lastly, the study compared the survival rates to the final results. This leaves the analysis to a minimum. In this report, we will try to continue the analysis through Bayesian data analysis methods. However, the data is quite primitive in comparison to today's standards. Nevertheless, we will try to analyze and use the data for predictions. This may be valuable for medical experts to understand how to allocate better resources based on some simple rules.

Firstly, we describe the dataset and perform explanatory data analysis to understand the correlations in the data. We implement a linear and non-linear logistic regression model to carry out classification of the data based on the patient improvement rate six months after diagnosis. The prior is selected according to expert knowledge and sensitivity analysis is carried out to assess the influence of the prior choice on the posterior sampling. Convergence and HMC analysis was carried out to asses the sampling.

Description of Data and Analysis Problem

The dataset is derived from a prospective, randomized, placebo-controlled clinical trial titled "Streptomycin Treatment of Pulmonary Tuberculosis." This trial was published on October 30, 1948, in the British Medical Journal (BMJ), pages 769-782. It is considered one of the first modern randomized clinical trials ever published. The dataset has been reconstructed from the original article and particularly from tables 1-4. The dataset contains 107 patients. Patients that take part in the study are characterised by baseline measurables that are taken at the time of diagnosis. The baseline measurables include temperature, ESR, cavitation, and strep resistance of bacteria. The patients were monitored size months after first diagnosis. The patient outcome was assessed using X-ray monitoring of the lungs. Based on the X-ray data, the patients were put into six categories defined by the variable radium. For the current data analysis study, we choose baseline temperature and ESR values as descriptors of the state of the patient at the time of diagnosis. These baseline variables, were used to predict the outcome of the patient six months after diagnosis. The outcome is quantified using a binomial variable that classifies patients into two groups: improved and not improved. The data is divided into two groups based on whether they received Streptomycin or not. Table 1. contains the detailed description of each variable in the dataset.

Table 1. Data description

patient_id	participant ID (character)
arm	Study Arm (Streptomycin, Control) (factor)
dose_strep_g	Dose of Streptomycin(our medicine) in Grams (numeric)
gender	Gender (M = Male, F = Female) (factor)
baseline_condition	This variable categorises patients based on their overall health condition at the beginning of the trial. (1_Good, 2_Fair, 3_Poor) (factor)
baseline_temp	Oral Temperature at Baseline (Degrees F) (factor)
baseline_esr	Erythrocyte Sedimentation Rate at Baseline. It is an indicator of

	inflammation or infection (millimetres per hour) (factor) Male <50 years old: ≤15 mm/hr. Female <50 years old: ≤ 20 mm/hr. Male >50 years old: ≤20 mm/hr.[6]
baseline_cavitation	Cavitation(formation of hollow spaces or gaps) of the Lungs on chest X-ray at baseline. (yes, no) (factor)
strep_resistance	Resistance to Streptomycin at 6 months (1_sens_0-8, 2_mod_8-99, 3_resist_100+) (factor). It is not a characteristic of a healthy person but rather a property of the bacteria that can cause tuberculosis. Thus, no normal range can be assigned.
radiologic_6m	Radiologic outcome at 6 months (1_Death, 2_Considerable Deterioration, 3_Moderate Deterioration, 4_No Change, 5_Moderate Improvement, 6_Considerable Improvement) (factor)
radnum	Numeric Rating of Chest X-ray at month 6 (1-6) (numeric)
improved	Dichotomous Outcome of Improved (TRUE, FALSE) (logical)

Research questions

The aim of this work is to predict the probability for improvement of patients, based on their baseline condition at diagnosis. The Bayesian analysis is carried out separately for the group that received Streptomycin and the control group (no Streptomycin).

The research question is to determine the individual probability that each patient will survive based on the baseline descriptors measured at diagnosis.

The analysis aims to address three key research questions related to the effectiveness of streptomycin treatment for tuberculosis. It involves investigating the impact of the

drug on different patient baselines, estimating the number of deaths that could have been prevented, and determining the optimal distribution of the treatment.

Exploratory data analysis

In the process of preparing the streptomycin treatment dataset for analysis, several essential preprocessing and feature engineering steps were undertaken. The initial dataset included information on patients, their baseline conditions, and the effectiveness of streptomycin treatment. To enhance the dataset's suitability for analysis, various columns underwent transformations.

Temperature data in the 'baseline_temp' column was initially represented categorically. To facilitate numerical analysis, temperature ranges were mapped to specific numeric values using a predefined dictionary. Similarly, ESR (Erythrocyte Sedimentation Rate) data in the 'baseline_esr' column, originally expressed in ranges, was transformed into specific numeric values.

Boolean variables were addressed to ensure compatibility with analysis tools. The 'improved' column, initially represented as boolean values, was converted to integers (0 and 1) to align with binary coding standards. The 'arm' column, indicating treatment groups ('Control' and 'Streptomycin'), was likewise mapped to integers (0 and 1).

Additionally, categorical data in the 'baseline_cavitation' column, denoting the presence or absence of cavitation, was converted to numeric values (1 and 2) for analytical consistency. The resulting dataset, containing a subset of relevant columns, was exported to a new CSV file named 'strep_tb_scaled.csv'. These preprocessing and feature engineering steps collectively ensure that the dataset is now better suited for modeling, providing a foundation for gaining insights into the streptomycin treatment outcomes. Below is the correlation matrix for our features:

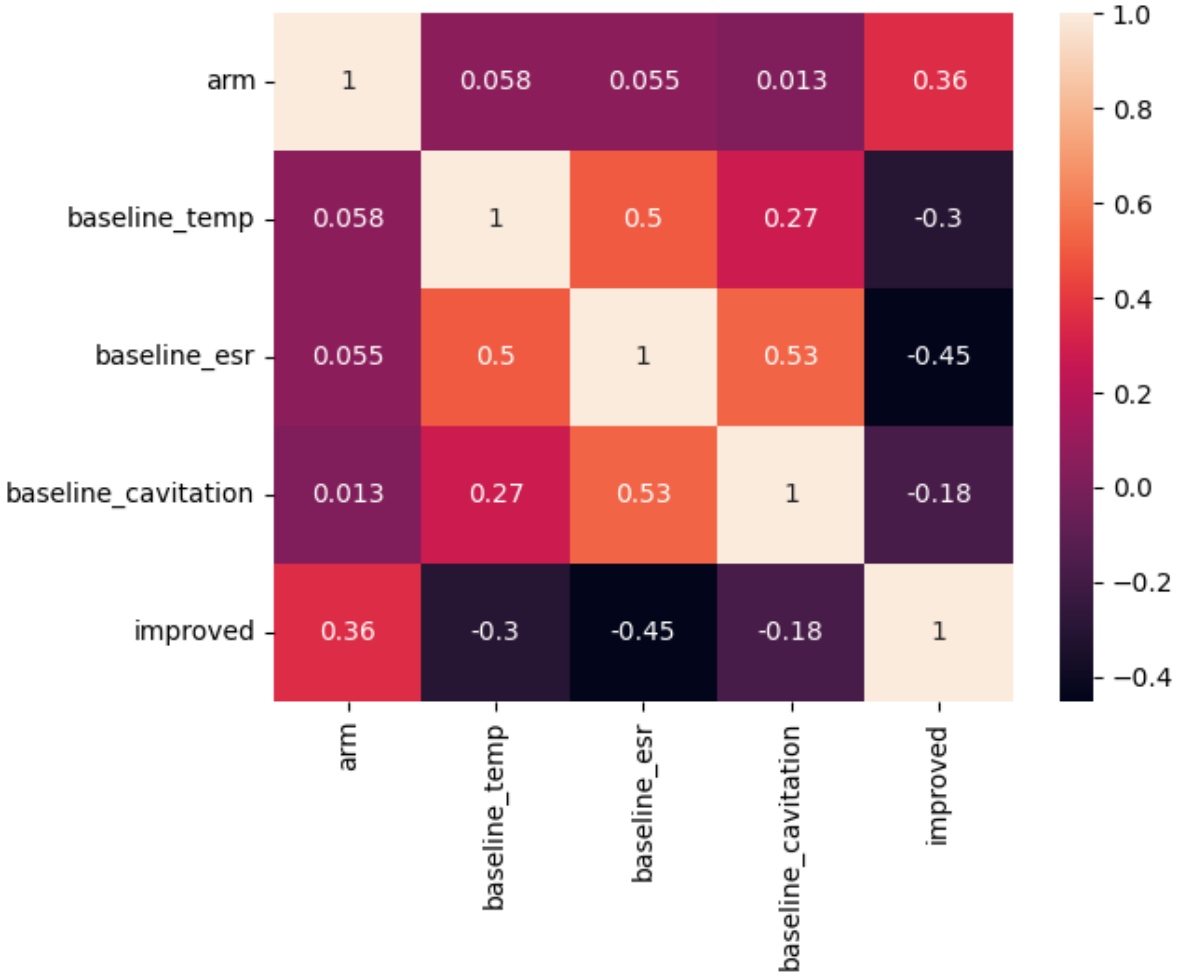


Figure 1. Correlation matrix of he features of the dataset

The correlation coefficient of -0.3 indicates a moderate negative correlation between temperature and improvement. The correlation coefficient of -0.45 indicates a stronger negative correlation between ESR and improvement. The correlation coefficient of -0.18 indicates a relatively weak negative correlation between cavitation and improvement. In conclusion there are some linear relationships between these variables and therefore the final features chosen are baseline temperature and baseline esr values. This is also intuitive as the health condition will have direct consequences on the success rate of the treatment.

These correlations imply that higher values of 'temp,' 'esr,' and 'cavitation' are associated with lower chances of improvement.

Model descriptions

Two models are constructed and compared in their ability to predict patient outcome based on baseline variables. **Model 1** is a linear logistic regression model. The log-odds ratio is

$$lo(y|x_1, x_2) = \alpha + \beta_1 X_1 + \beta_2 X_2$$

where y is a binary label that describes good and bad patient outcomes, X_1 is the temperature and X_2 is the ESR of a patient. α, β_1 and β_2 are fitting parameters of the model. In this model, it is assumed that the relationship between the predictor and the log-odds ratio of success is linear.

Transforming the log odds ratio into probability likelihood

$$P(y|x_1, x_2) = \frac{1}{1+e\text{^}-\{\alpha+ \beta_1 X_1+ \beta_2 X_2\}}$$

it becomes the logistic function.

Model 2 is a non-linear logistic regression model. The log odds ratio is

$$lo(y|x_1, x_2) = -\alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

where the variables are the same as in model 1, however there is an additional term β_3 that describes the correlation between x_1 and x_2.

Transforming the log odds ratio into probability likelihood

$$P(y|x_1, x_2) = \frac{1}{1+e\text{^}-\{-\alpha+ \beta_1 X_1+ \beta_2 X_2 +\beta_3 x_1 x_2\}}$$

it becomes the logistic function.

The posterior is a sample from a Bernoulli distribution for both models.

$$y \sim \text{Bernoulli\_logit}(P(y|x_1, x_2))$$

The reason for choosing this model is as we tried more complicated (seperate distributions and hierarchical) models but these did not converge. This was good design choice as the dataset was challenging to fit with convergence.

Prior selection

Prior selection is an important step in Bayesian modelling. For logistic regression models, noninformative priors are a common choice, see Refs. [2-4]. However, weakly informative priors are also common [4]. In our case it would have been optimal to use

weakly informative priors as domain knowledge requires a medical doctor. However by the assignment requirements we chose with the more informative priors.

In our modelling, we start with the use of weakly informative priors for parameters β_1 (T) and β_2 (ESR). Later, we carry out sensitivity analysis for the prior parameters. Based on medical data [5], the temperature range of healthy and unhealthy adults is between 93 to 106 Fahrenheit [7], while the ESR range is between 5 to 50 mm/hr [8]. This ranges where gathered by taking the average range for a healthy human temperature and ESR scores. This will not be perfect as our dataset contains only people in suboptimal health condition.

Furthermore, since the data is binary, we calculate the expected values for the parameters: $\beta_1 = 1/(106 - 93) = 0.07$ and $\beta_2 = 1/(50 - 5) = 0.22$. Based on Refs. [2-4], we choose the standard deviation for both β_1 and β_2 to be 5, such that the priors become weakly informative. For α, we choose a noninformative normal prior with expected value 0 and standard deviation 10.

$$\beta_1 \sim Normal(0.07, 5)$$

$$\beta_2 \sim Normal(0.22, 5)$$

$$\alpha \sim Normal(0, 10)$$

R and Stan code parameters

```r
matrix_control <- cbind(data_control$baseline_temp, data_control$baseline_esr)

data_list_control <- list(
  N = length(data_control$baseline_temp),
  X = matrix_control,
  y = data_control$improved,
  beta1_prior_mean =  beta_1_mean,
  beta2_prior_mean = beta_1_sd,
  beta1_prior_sd = beta_2_mean,
  beta2_prior_sd= beta_2_sd
  )
```

```r
## Linear model, dataset control
```{r}

# Sampling from the posterior distribution happens here:
fit_linear_control <- model_linear$sample(data = data_list_control,
                                  refresh=0,
                                  max_treedepth = 20,
                                  iter_sampling = 4000,
                                  show_messages=FALSE,
                                  show_exceptions=FALSE)
print(fit_linear_control)


```
```

Description: df [10 × 10]

variable <chr>	mean <S3: AsIs>	median <S3: AsIs>	sd <S3: AsIs>
lp__	-13.63	-13.34	1.29
alpha	370.25	296.99	299.99
beta1	0.08	0.08	0.02
beta2	-10.61	-8.56	8.45

```r
## Non-Linear model, dataset control
```{r}

# Sampling from the posterior distribution happens here:
fit_nonlinear_control <- model_non_linear$sample(data = data_list_control, refresh=0,
                                  max_treedepth = 20,
                                  show_messages=FALSE,
                                  show_exceptions=FALSE)
print(fit_nonlinear_control )
fit_nonlinear_control.
```
```

Description: df [10 × 10]

variable <chr>	mean <S3: AsIs>	median <S3: AsIs>	sd <S3: AsIs>
lp__	-14.08	-13.72	1.58
alpha	411.78	316.96	353.49
beta1	0.08	0.08	0.02
beta2	-10.30	-7.46	10.08
beta3	-0.01	-0.01	0.02
y_prob[1]	1.00	1.00	0.00
y_prob[2]	1.00	1.00	0.00
y_prob[3]	0.82	0.86	0.15

For the details in code see appendix.

Stan code

```
//  Linear Logistic regression model
data {
  int<lower=0> N; // Number of observations
  array[N] int<lower=0, upper=1> y; // Outcome variable (1 for improved, 0 for not improved)
  matrix[N, 2] X; // Predictor matrix (including baseline temp and ESR)
  real beta1_prior_mean;
  real beta2_prior_mean;
  real beta1_prior_sd;
  real beta2_prior_sd;
}

parameters {
  real alpha; // Intercept
  real beta1; // Tempeture
  real beta2; // ESR
}

model {
  // Priors
  beta1 ~ normal(beta1_prior_mean, beta1_prior_sd); // Own prior for beta[1] temp
  beta2 ~ normal(beta2_prior_mean, beta2_prior_sd);  // Own prior for beta[2] esr

  // Logistic regression
  y ~ bernoulli_logit(alpha + X * to_vector({beta1, beta2}));
}

generated quantities {
  vector[N] y_prob = inv_logit(alpha + X * to_vector({beta1, beta2}));
}
```

```
// Non-Linear Logistic regression model
data {
  int<lower=0> N; // Number of observations
  array[N] int<lower=0, upper=1> y; // Outcome variable (1 for improved, 0 for not improved)
  matrix[N, 2] X; // Predictor matrix (including baseline temp and ESR)
  real beta1_prior_mean;
  real beta2_prior_mean;
  real beta1_prior_sd;
  real beta2_prior_sd;
}

parameters {
  real alpha; // Intercept
  real beta1; // Regression coefficient for baseline temp
  real beta2; // Regression coefficient for ESR
  real beta3; // Regression coefficient for the interaction term
}

model {
  // Priors
  beta1 ~ normal(beta1_prior_mean, beta1_prior_sd); // Own prior for beta[1] temp
  beta2 ~ normal(beta2_prior_mean, beta2_prior_sd);  // Own prior for beta[2] esr

  // Logistic regression with interaction
  y ~ bernoulli_logit(alpha + X * to_vector({beta1, beta2}) + beta3 * X[:, 1] .* X[:, 2]);
}

generated quantities {
  vector[N] y_prob = inv_logit(alpha + X * to_vector({beta1, beta2}) + beta3 * X[:, 1] .* X[:, 2]);
}

```

Convergence analysis of R hat, ESS and HMC

R_hat, also known as the Gelman-Rubin statistic , is a diagnostic tool used in the context of Markov Chain Monte Carlo (MCMC) simulations, particularly for assessing the convergence of multiple chains. It helps determine whether multiple chains have mixed well and whether the MCMC sampler has adequately explored the target distribution. Common practice is to consider chains converged if R_hat < 1.05.

Effective Sample Size (ESS) is a measure used in Markov Chain Monte Carlo (MCMC) simulations to estimate the number of independent samples equivalent to the generated samples obtained from the MCMC procedure. Generated and data samples are paired by high correlation. It provides an indication of how effectively the MCMC algorithm explores the parameter space. A higher ESS means that the chain is providing more information about the posterior distribution, suggesting better convergence and mixing. A satisfactory score for ESS in the industry is 400. ESS is usually divided into two ESS tail and ESS bulk. ESS tail refers to the effective sample size for the extreme tails of the distribution. It evaluates the efficiency of sampling in the tails of the posterior distribution. ESS bulk refers to the effective sample size for the bulk of the distribution. It's a measure of how many independent draws the MCMC chains would equivalently provide for the central part of the posterior distribution. To summarise ESS bulk measures the effectiveness of the mean and median values when tail measures the effectiveness of the 95th and 5th quantile.

For HMC we used No-U-Turn Sampler (NUTS), which automatically tunes the trajectory length (treedepth) of the sampler. If a high number of iterations are hitting the maximum treedepth, it suggests that the sampler is turning prematurely and may be exploring the posterior distribution inefficiently.

Table 2. Convergence analysis of linear model

Linear Logistic regression			
Dataset Control (Threedepth = 20)		**Dataset Streptomycin (Threedepth = 10)**	
Rhat Alpha = 1.0 Beta1 = 1.0 Beta2 = 1.0	**ESS (bulk, tail)** Alpha = (1520, 2450) Beta1 = (3532, 3960) Beta2 = (1520, 2424)	**Rhat** Alpha = 1.0 Beta1 = 1.0 Beta2 = 1.0	**ESS (bulk, tail)** Alpha = (4433,5547) Beta1 = (5129, 5051 Beta2 = (5421, 5090)

Table 3. Convergence analysis of nonlinear model

Linear Logistic regression			
Dataset Control (Threedepth = 20)		**Dataset Streptomycin (Threedepth = 20)**	
Rhat Alpha = 1.0 Beta1 = 1.0 Beta2 = 1.0	**ESS (bulk, tail)** Alpha = (396, 805) Beta1 = (1741,1828) Beta2 = (386, 773)	**Rhat** Alpha = 1.0 Beta1 = 1.0 Beta2 = 1.0	**ESS (bulk, tail)** Alpha= (1262, 1185) Beta1= (1305, 1162) Beta2= (1307, 1657)

As earlier concluded our dataset is divided into two equally size sets. All variables converged to the R_hat value of 1. This was achieved by using 4 chains which all iterated 5 times to find the R_hat.

In our analysis of the linear model, 2000 Iterations was not enough for the linear model to find good ESS tail. Therefore chain iterations was increased to 4000 to find the acceptable threshold value of 400+.

In comparison for the Non linear model 2000 iterations was enough, However themaximum treedepth of 10 was not enough as 19.0%(for control) and 78%(for Streptomycin) of transitions hit the maximum. No-U-Turn Sampler (NUTS) is a limit to prevent the algorithm from running too long and using excessive computational resources. NUTS constraint of 10 was to restrictive. To solve this the the treedepth was dubbed to 20. Lastly, after some further analysis we noticed the same happened to Linear dataset with linear regression, we solved it the same way as the earlier.

The output was satisfactory for all models. Detailed code can be found in appendix.

```
Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
```

Predictive performance analysis

The accuracy score was used to assess the performance of the three models.

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ number\ of\ predictions}$$

Table 4 shows the prediction accuracy for the two groups and models tested. The Linear model had higher accuracy for both patient groups i.e. control and received Streptomycin. The prediction accuracy is significantly better for the control group. The reason for this is that most patients in the control group did not improve, therefore misclassification becomes less frequent.

Table 4. Accuracy in predicting patient outcome for different models

	Linear model	Nonlinear model
Control	0.90	0.85
Recieved Streptomycin	0.69	0.68

Discussion of issues and potential improvements

The Bayesian data analysis results show that patients that got Streptomycin incorporated in their treatment plan had a higher probability of survival. The main caveat of the modelling is the small dataset that only contains 107 patient data.

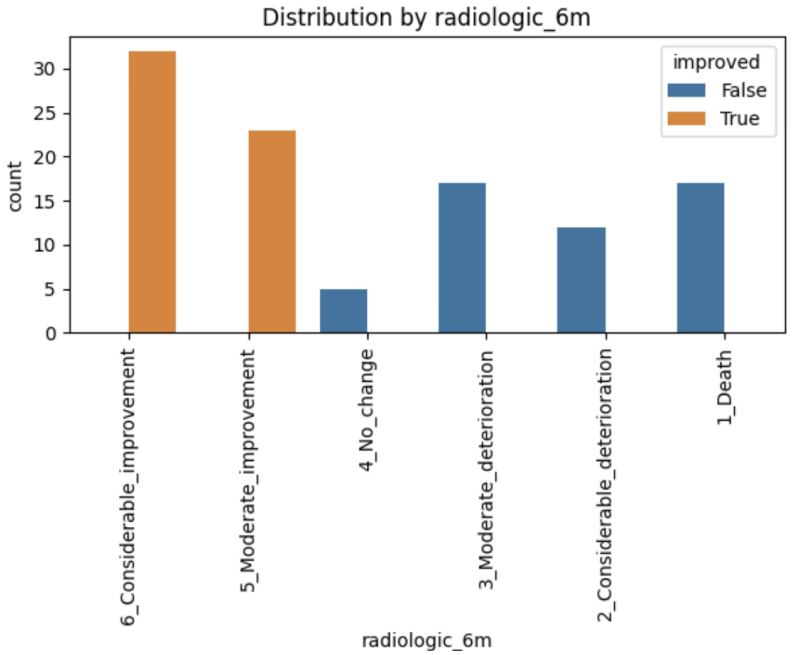


Figure 2. Classification of patients at six months after diagnosis

Additionally, the number of baseline descriptors to describe the condition of the patient at diagnosis is very low. For the current work, only two descriptors per patient were used. By increasing the size of the dataset and the number of descriptors one could get better prediction accuracy and more statistically significant results.

Using a larger dataset, it would be possible to construct a model for multiclass classification. Figure 2 shows the patients in the dataset classified into six categories using X-ray imaging six months after diagnosis. These categories give insight into how well the patient recovered with and without treatment.

Furthermore, a larger dataset would enable examining the effectiveness of Streptomycin across different patient baseline conditions. This study would be useful in understanding which patients should be administered the drug for best outcome and maximise the number of saved lives.

In the results, there will be presented distributions and some of these are point distributions which because of an

Sensitivity analysis to prior distributions

Tabel 5. Initially chosen prior values based on expert knowledge.

beta_1_mean	1/(106−93)
beta_1_sd	20
beta_2_mean	1/(50−5)
beta_2_sd	20

In an effort to assess the sensitivity of our Bayesian model to prior specifications, we systematically modified the standard deviation (beta_1_sdbeta_1_sd) for the linear and non-linear models in the Control and Treatment groups. Specifically, we adjusted beta_1_sdbeta_1_sd values to observe their impact on beta_2_meanbeta_2_mean convergence. The intent was to explore how changes in the spread of the prior for beta_1beta_1 influence the posterior mean of beta_2beta_2.

Table 6. Chosen prior mean and standard deviation for sensitivity analysis.

beta_1_sd/ beta_2_mean	Linear, Control	Linear, Treatment	Non-linear, Control	Non-linear, Treatment
20	-10,72	-0,09	-9,37	0,78
40	-7,61	-0,09	-5,82	0,77

Despite the attempt to assess the convergence of beta_2beta_2 under various conditions, challenges were encountered. The proposed approach of plugging in values led to negative values, resulting in code crashes. Consequently, the examination was limited to the Non-linear Treatment model, where beta_1_sd=40beta_1_sd=40 and beta_2_mean=0.77beta_2_mean=0.77. However, this model exhibited instability, indicating potential issues with the chosen priors.

A subsequent exploration involved iteratively adjusting priors to stabilize the model. This led to variations in beta_2_meanbeta_2_mean from 0.24 to 2.5 and back to 0.84. While the model demonstrated convergence, the observed instability raises concerns about the soundness of the results and underscores the complex relationship between prior specifications and model behaviour. Further investigation and refinement of the model or prior choices may be necessary to enhance stability and reliability.

Results

Posterior predictive checking is a Bayesian method to assess the fit of a statistical model by comparing the observed data to data simulated from the posterior predictive distribution. The idea is that if the model is a good fit, then simulated data generated under the model should look similar to the observed data.

The four graphs you provided appear to be visualizations of posterior predictive checks, showing the range of predictions made by the models for various observations. They plot the 50% and 90% credible intervals for predicted probabilities of each observation, with the density indicating where the mass of the posterior distribution lies.

In the absence of overlaid observed data points, the interpretation is somewhat limited as we there is not the data available to plot the reason for this is quite simple as the dataset doent allow this as all of the features are categoricall between 3 or 4 values for each column. You would normally look to see if the observed data fall within the plotted intervals, particularly within the inner 50% interval, as this would suggest that the model is capturing the central tendency of the data well. If the observed data frequently falls outside the 90% interval, this would suggest that the model is missing something fundamental about the data.

In comparing the models of a linear and non-linear logistic regression models, to classify patient improvement six months post-diagnosis. The linear model showed higher accuracy across control and treatment groups with the Predictive performance analysis.

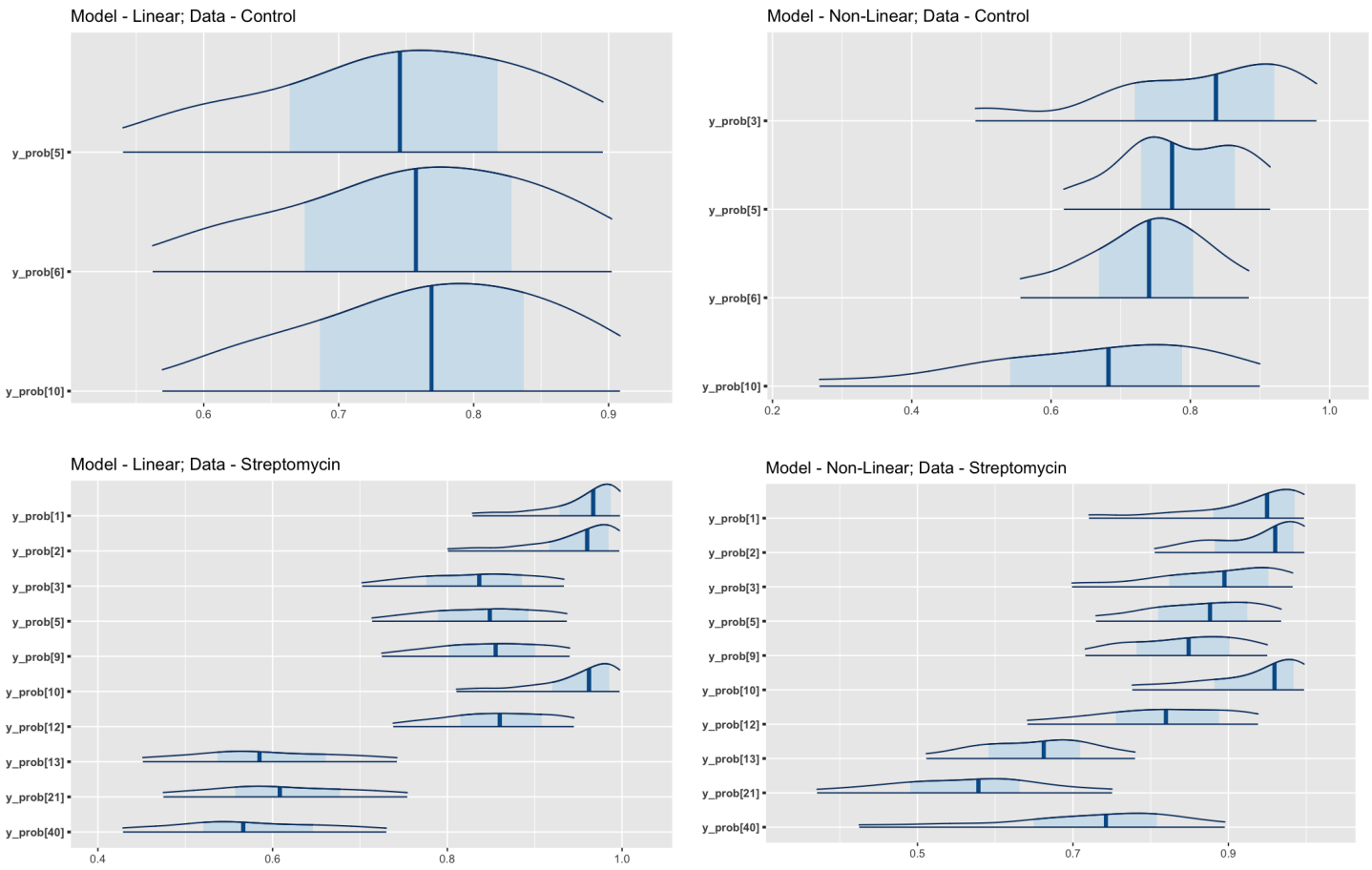


Figure 3. Posterior predictive checking of individual patients for improvement six months after diagnosis.

Conclusions

Based on the results and the sensitivity analysis, we can conclude that the sampling is highly unstable and the dataset is not appropriate for Bayesian data analysis. However, the methods used are transferable to more extensive datasets with more patients and descriptors. In addition, we were able to infer from the modelling that the use of Streptomycin significantly increases the probability for patients to improve. Altough the results were disappointing Bayesian data analysis shows its power as one could still find some patterns without data.

Self-reflection

The actual project process has helped us to understand the Bayesian process in more detail. The hands-on experience with Bayesian modeling techniques has been invaluable. The group dynamic played a crucial role in the project's success. Collaborating with team members with diverse perspectives and skill sets fostered an environment conducive to learning. Engaging in discussions about model choices, interpreting results, and troubleshooting challenges allowed for a collective exchange of knowledge.

Initially, we explored the possibility of normalizing feature values. The normalization aimed to bring all features to a similar scale, potentially improving the performance of the analysis. However, after careful consideration and observation of the results, we made a decision to abandon the normalization approach. Encountering challenges like this during the project, whether in specifying models, addressing convergence issues, or interpreting results, provided valuable learning opportunities.

Lastly, the golden rule of predicting power is to have high quality data that is still relevant.

References:

[1] Streptomycin treatment of pulmonary tuberculosis. Br Med J. 2(4582):769-82 (1948)

[2] Yang , R. and B. Erger , J. O. Estimation of a covariance matrix using reference prior. Ann. Statist. 22 1195–1211 (1994)

[3] Kall , R. E. and Wasserman , L. The selection of prior distributions by formal rules. J. Amer. Statist. Assoc. 91 1343–1370 (1996)

[4] Gelman, A., Jakulin, A., Grazai, M., and Su, Y.S. A weakly informative default prior distribution for logistic and other regression models. Ann. Appl. Stat. 2 1360-1383 (2008)

[5] Walter, H.K., Hall, W.D., and Hurst, J.W. Clinical Methods: The History, Physical and Laboratory Examinations. 3rd ed. Boston:Butterworths. (1990)

[6] Tishkowski K, Gupta V. Erythrocyte Sedimentation Rate. [Updated 2023 Apr 23]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK557485/

[7] Blum FC, Biros MH. Fever in the adult patient. In: Walls RM, ed. *Rosen's Emergency Medicine: Concepts and Clinical Practice*. 10th ed. Philadelphia, PA: Elsevier; 2023:chap 8.

[8] Tishkowski K, Gupta V. Erythrocyte Sedimentation Rate. [Updated 2023 Apr 23]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK557485/

Appendix: project code

anonymous

```r
# data preperation
data("strep_tb")
strep_tb <- na.omit(strep_tb)
summary(strep_tb)
```

```
 patient_id                  arm      dose_strep_g     dose_PAS_g gender
Length:106         Streptomycin:55   Min.   :0.000   Min.   :0    F:58
Class :character   Control     :51   1st Qu.:0.000   1st Qu.:0    M:48
Mode  :character                     Median :2.000   Median :0
                                     Mean   :1.038   Mean   :0
                                     3rd Qu.:2.000   3rd Qu.:0
                                     Max.   :2.000   Max.   :0
baseline_condition      baseline_temp  baseline_esr baseline_cavitation
1_Good:16          1_98-98.9F  : 7    1_0-10 : 0    no :45
2_Fair:37          2_99-99.9F  :25    2_11-20: 5    yes:61
3_Poor:53          3_100-100.9F:31    3_21-50:36
                   4_100F+     :43    4_51+  :65


     strep_resistance                      radiologic_6m    rad_num
1_sens_0-8   :64      6_Considerable_improvement  :32    Min.   :1.000
2_mod_8-99   : 8      5_Moderate_improvement      :23    1st Qu.:2.000
3_resist_100+:34      4_No_change                 : 5    Median :5.000
                      3_Moderate_deterioration    :17    Mean   :3.953
                      2_Considerable_deterioration:12    3rd Qu.:6.000
                      1_Death                     :17    Max.   :6.000
 improved
Mode :logical
FALSE:51
TRUE :55
```

```r
data <- read.csv(file = "strep_tb_scaled.csv", header = TRUE)
data$X <- NULL
data$n = 1

data_medicin = data[data$arm == 1,]
data_control = data[data$arm == 0,]
```

```r
#priors

beta_1_mean = 1/(106-93) # tempetrue in farenhieth
beta_1_sd = 20
beta_2_mean = 1/(50-5) # ESR mm/hr
beta_2_sd = 20
```

```r
## creating data_list

matrix_control <- cbind(data_control$baseline_temp, data_control$baseline_esr)

data_list_control <- list(
  N = length(data_control$baseline_temp),
  X = matrix_control,
  y = data_control$improved,
  beta1_prior_mean =  beta_1_mean,
  beta2_prior_mean = beta_1_sd,
  beta1_prior_sd = beta_2_mean,
  beta2_prior_sd= beta_2_sd
  )



matrix_medicin <- cbind(data_medicin$baseline_temp, data_medicin$baseline_esr)

data_list_medicin <- list(
  N = length(data_medicin$baseline_temp),
  X = matrix_medicin,
  y = data_medicin$improved,
  beta1_prior_mean =  beta_1_mean,
  beta2_prior_mean = beta_1_sd,
  beta1_prior_sd = beta_2_mean,
  beta2_prior_sd= beta_2_sd
  )
```

```r
## compiling
model_linear<- cmdstan_model(stan_file = "linear_logistic.stan")
model_non_linear<- cmdstan_model(stan_file = "non_linear_logistic.stan")
```

0.1 Linear model, dataset control

```r
# Sampling from the posterior distribution happens here:
fit_linear_control <- model_linear$sample(data = data_list_control,
                                          refresh=0,
                                          max_treedepth = 20,
                                          iter_sampling = 4000,
                                          show_messages=FALSE,
                                          show_exceptions=FALSE)
print(fit_linear_control)
```

```
Warning: NAs introduced by coercion
```

```
 variable   mean median     sd    mad     q5     q95 rhat ess_bulk ess_tail
lp__      -13.58 -13.28   1.29   1.07 -16.04  -12.12 1.00     2346     3978
alpha     370.59 289.89 314.36 272.20  26.43 1000.53 1.00     1230     2051
beta1       0.08   0.08   0.02   0.02   0.04    0.11 1.00     3571     4226
beta2     -10.62  -8.35   8.86   7.66 -28.33   -0.93 1.00     1235     2060
y_prob[1]   1.00   1.00   0.00   0.00   1.00    1.00 1.00     3937       NA
y_prob[2]   1.00   1.00   0.00   0.00   1.00    1.00 1.00     3944       NA
y_prob[3]   0.73   0.74   0.10   0.10   0.55    0.88 1.00    16564    10430
y_prob[4]   0.73   0.74   0.10   0.10   0.55    0.88 1.00    16564    10430
y_prob[5]   0.74   0.75   0.10   0.10   0.57    0.89 1.00    16645    10540
y_prob[6]   0.76   0.76   0.09   0.09   0.59    0.89 1.00    16673    10782

# showing 10 of 55 rows (change via 'max_rows' argument or 'cmdstanr_max_rows' option)
```

0.2 Linear model, dataset medicin

```
# Sampling from the posterior distribution happens here:
fit_linear_medicin <- model_linear$sample(data = data_list_medicin, refresh=0,
                                  iter_sampling = 4000,
                                  max_treedepth = 20,
                                  show_messages=FALSE,
                                  show_exceptions=FALSE)
print(fit_linear_medicin )
```

```
 variable   mean median   sd  mad     q5    q95 rhat ess_bulk ess_tail
lp__      -34.17 -33.85 1.26 1.02 -36.66 -32.81 1.00     4864     6398
alpha      -2.75  -2.88 2.97 2.94  -7.42   2.34 1.00     4822     5479
beta1       0.08   0.08 0.02 0.02   0.04   0.11 1.00     5575     5727
beta2      -0.08  -0.08 0.04 0.04  -0.16  -0.02 1.00     5202     4762
y_prob[1]   0.94   0.97 0.07 0.03   0.80   1.00 1.00     5260     5061
y_prob[2]   0.93   0.96 0.08 0.04   0.77   1.00 1.00     5212     4982
y_prob[3]   0.82   0.83 0.08 0.08   0.67   0.94 1.00     5777     5970
y_prob[4]   0.82   0.83 0.08 0.08   0.67   0.94 1.00     5777     5970
y_prob[5]   0.83   0.84 0.08 0.08   0.69   0.95 1.00     5818     6010
y_prob[6]   0.83   0.84 0.08 0.08   0.69   0.95 1.00     5818     6010

# showing 10 of 59 rows (change via 'max_rows' argument or 'cmdstanr_max_rows' option)
```

0.3 Non-Linear model, dataset control

```
# Sampling from the posterior distribution happens here:
fit_nonlinear_control <- model_non_linear$sample(data = data_list_control, refresh=0,
                                  max_treedepth = 20,
                                  show_messages=FALSE,
                                  show_exceptions=FALSE)
print(fit_nonlinear_control )
```

```
Warning: NAs introduced by coercion
```

```
 variable   mean median     sd    mad     q5     q95 rhat ess_bulk ess_tail
lp__      -13.96 -13.63   1.60   1.39 -16.88  -12.09 1.00      933     1520
alpha     384.92 300.02 327.14 285.81  31.24 1047.38 1.01      374      655
beta1       0.08   0.08   0.02   0.02   0.04    0.11 1.00     1586     1509
beta2      -9.64  -7.33   9.42   8.12 -28.45    1.28 1.01      369      682
beta3      -0.01  -0.01   0.02   0.02  -0.04    0.02 1.00     1245     1312
y_prob[1]   1.00   1.00   0.00   0.00   1.00    1.00 1.00     2157       NA
y_prob[2]   1.00   1.00   0.00   0.00   1.00    1.00 1.00     2072       NA
y_prob[3]   0.82   0.85   0.14   0.13   0.53    0.98 1.00     1500     1581
y_prob[4]   0.82   0.85   0.14   0.13   0.53    0.98 1.00     1500     1581
y_prob[5]   0.79   0.80   0.10   0.10   0.60    0.93 1.00     2436     2215

# showing 10 of 56 rows (change via 'max_rows' argument or 'cmdstanr_max_rows' option)
```

0.4 Non-Linear model, dataset medicin

```
# Sampling from the posterior distribution happens here:
fit_nonlinear_medicin <- model_non_linear$sample(data = data_list_medicin, refresh=0,
                                    max_treedepth = 20,
                                    show_messages=FALSE,
                                    show_exceptions=FALSE)
print(fit_nonlinear_medicin )
```

```
 variable   mean median   sd  mad     q5    q95 rhat ess_bulk ess_tail
lp__      -34.31 -33.96 1.48 1.24 -37.34 -32.62 1.00     1288     1689
alpha      -3.52  -3.63 3.08 3.05  -8.33   1.63 1.00     1080     1577
beta1       0.08   0.08 0.02 0.02   0.04   0.11 1.00     1247     1445
beta2       0.81   0.79 0.97 0.96  -0.74   2.38 1.00     1167     1353
beta3      -0.01  -0.01 0.01 0.01  -0.02   0.01 1.00     1175     1305
y_prob[1]   0.91   0.95 0.11 0.06   0.67   1.00 1.00     1869     2080
y_prob[2]   0.93   0.96 0.09 0.05   0.75   1.00 1.00     2096     2101
y_prob[3]   0.87   0.90 0.09 0.08   0.70   0.98 1.00     1780     1889
y_prob[4]   0.87   0.90 0.09 0.08   0.70   0.98 1.00     1780     1889
y_prob[5]   0.86   0.87 0.08 0.07   0.72   0.96 1.00     2172     2438

# showing 10 of 60 rows (change via 'max_rows' argument or 'cmdstanr_max_rows' option)
```

0.5 Diagnosis

```
print(fit_linear_medicin$cmdstan_diagnose())
```

```
Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/linear_logistic-2023

Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
```

```
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
$status
[1] 0

$stdout
[1] "Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/linear_logistic

$stderr
[1] ""

$timeout
[1] FALSE
```

```
print(fit_linear_control$cmdstan_diagnose())
```

```
Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/linear_logistic-2023

Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
$status
[1] 0

$stdout
[1] "Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/linear_logistic

$stderr
[1] ""

$timeout
[1] FALSE
```

```
print(fit_nonlinear_medicin$cmdstan_diagnose())
```

```
Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/non_linear_logistic-
```

```
Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
$status
[1] 0

$stdout
[1] "Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/non_linear_logi

$stderr
[1] ""

$timeout
[1] FALSE
```

```
print(fit_nonlinear_control$cmdstan_diagnose())
```

```
Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/non_linear_logistic-

Checking sampler transitions treedepth.
Treedepth satisfactory for all transitions.

Checking sampler transitions for divergences.
No divergent transitions found.

Checking E-BFMI - sampler transitions HMC potential energy.
E-BFMI satisfactory.

Effective sample size satisfactory.

Split R-hat values satisfactory all parameters.

Processing complete, no problems detected.
$status
[1] 0

$stdout
[1] "Processing csv files: /var/folders/lf/v92_5zrn6k3ch09dnnm3cj1r0000gn/T/RtmpRKqbNT/non_linear_logi

$stderr
[1] ""
```

```
$timeout
[1] FALSE
```

```
  print(fit_nonlinear_medicin$summary()[,"mean"])
```

```
# A tibble: 60 x 1
        mean
       <num>
 1 -34.3
 2  -3.52
 3   0.0768
 4   0.807
 5  -0.00875
 6   0.907
 7   0.925
 8   0.875
 9   0.875
10   0.860
# i 50 more rows
```

```
  #generated_values <- extract(fit)
  # two colums X1 is probs_mean
  accuracy_score <- function(data) {
    binary_predictions <- ifelse(data[, 1] > 0.5, 1, 0)
    correct_predictions <- binary_predictions == data[, 2]
    return(sum(correct_predictions) / nrow(data))
  }
  fit_to_accuracy <- function(fit, data_labels){
    probs <- fit$summary()[['mean']]
    probs <- probs[5:length(probs)]

    output <- cbind(probs, data_labels)
    return(accuracy_score(output))
  }

  print(fit_to_accuracy(fit=fit_linear_medicin,  data_medicin$improved ))
```

```
[1] 0.6909091
```

```
  print(fit_to_accuracy(fit=fit_linear_control,  data_control$improved ))
```

```
[1] 0.9019608
```

```
  print(fit_to_accuracy(fit=fit_nonlinear_medicin,  data_medicin$improved ))
```

```
Warning in cbind(probs, data_labels): number of rows of result is not a
multiple of vector length (arg 2)
```

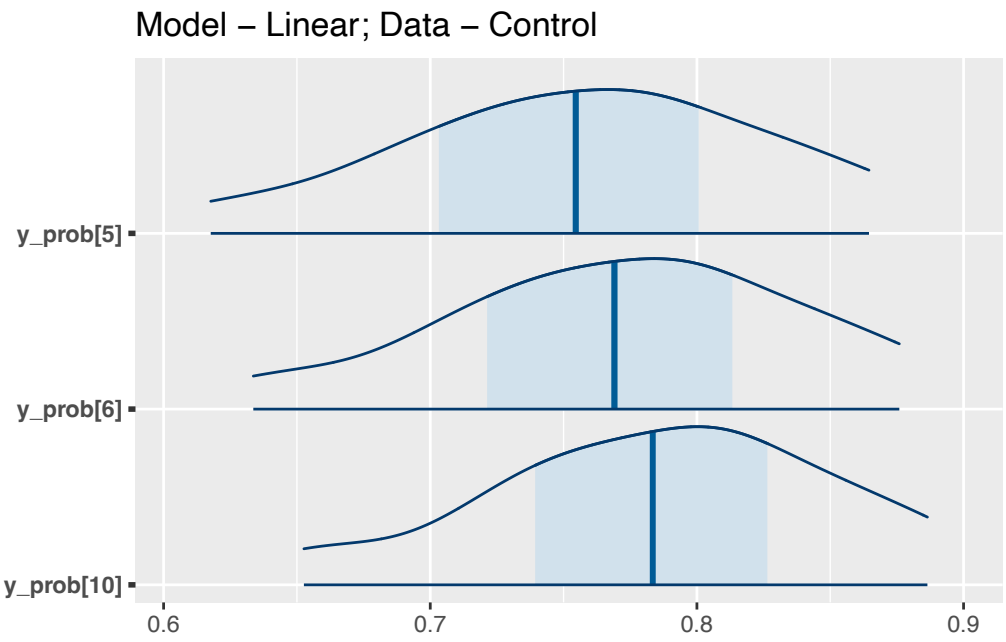
```
[1] 0.6785714
```

```
print(fit_to_accuracy(fit=fit_nonlinear_control,  data_control$improved ))
```

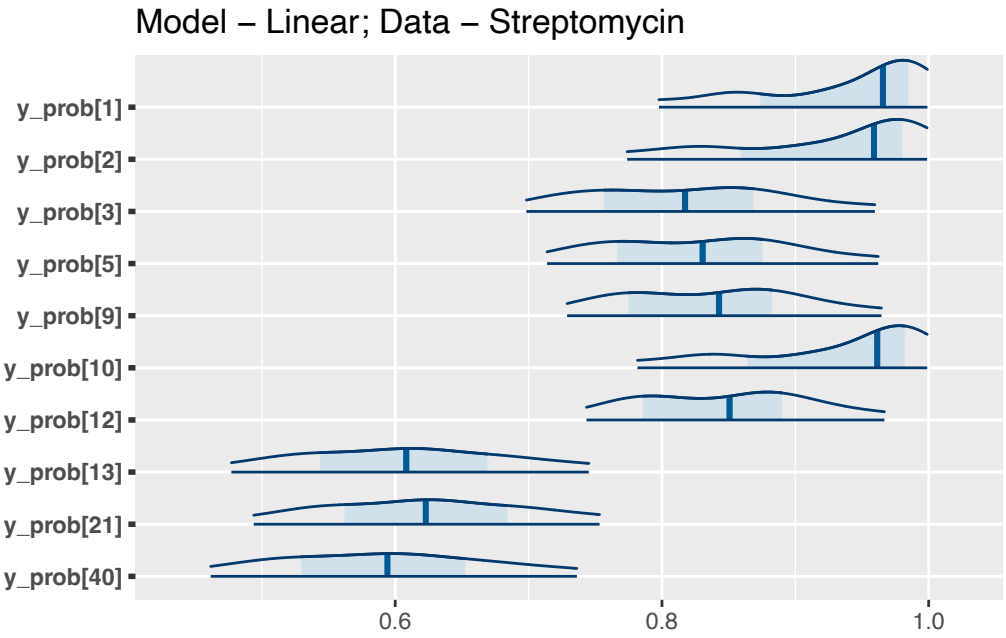
```
Warning in cbind(probs, data_labels): number of rows of result is not a
multiple of vector length (arg 2)
```

```
[1] 0.8461538
```

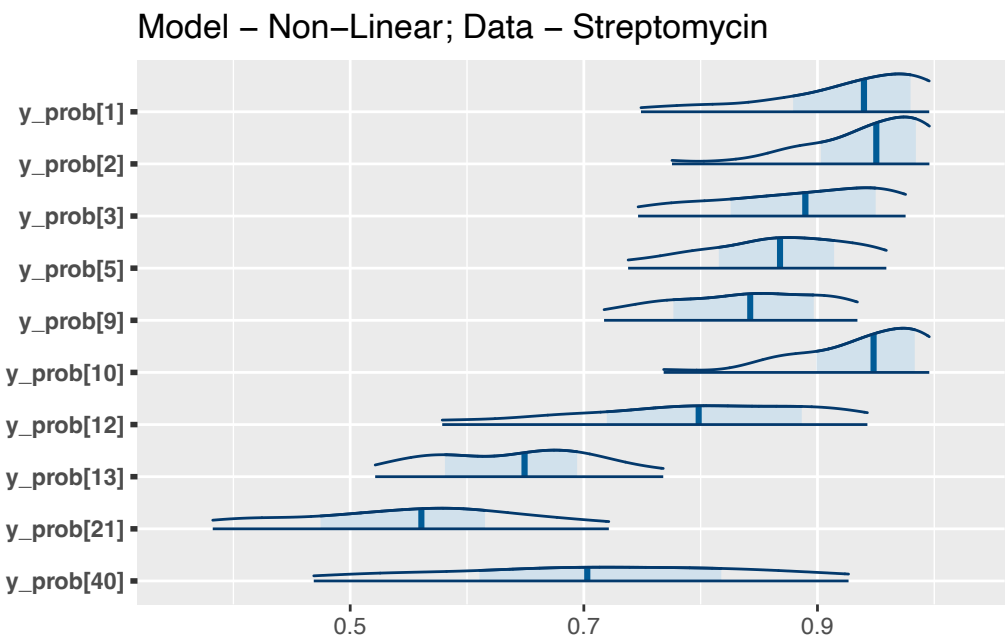
```
unique_arg_values <- function(fit, if_linear, is_medicin) {
  means <- fit$summary()[['mean']]
  unique_means <- unique(means)
  indices_of_unique_means <- match(unique_means, means)
if(if_linear){
  if(is_medicin){
    return(indices_of_unique_means[5:length(indices_of_unique_means)])
  }else{
    return(indices_of_unique_means[11:length(indices_of_unique_means)-3])
  }
}else{
    if(is_medicin){
        return(indices_of_unique_means[6:length(indices_of_unique_means)])
    }else{
        return(indices_of_unique_means[11:length(indices_of_unique_means)-3])
    }
  }
}


plot_mcmc <- function(fit, title_name , if_linear, is_medicin) {
  unique_indexes <- unique_arg_values(fit , if_linear, is_medicin)
  posterior_samples <- fit$draws()
  posterior_len <- length(posterior_samples[,1,1])
  y_prob_mean_vector <-posterior_samples[(posterior_len-100):posterior_len, 4, unique_indexes]

  plot <- bayesplot::mcmc_areas(y_prob_mean_vector, prob = 0.5, prob_outer = 0.90)

  plot_with_title <- plot + ggtitle(title_name)

  print(plot_with_title)
}

plot_mcmc(fit_linear_control, "Model - Linear; Data - Control", 1, 0)
```




```
plot_mcmc(fit_linear_medicin, "Model - Linear; Data - Streptomycin", 1, 1)
```




```
plot_mcmc(fit_nonlinear_medicin, "Model - Non-Linear; Data - Streptomycin", 0, 1)
```




```
plot_mcmc(fit_nonlinear_control, "Model - Non-Linear; Data - Control", 0, 0)
```