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)
```

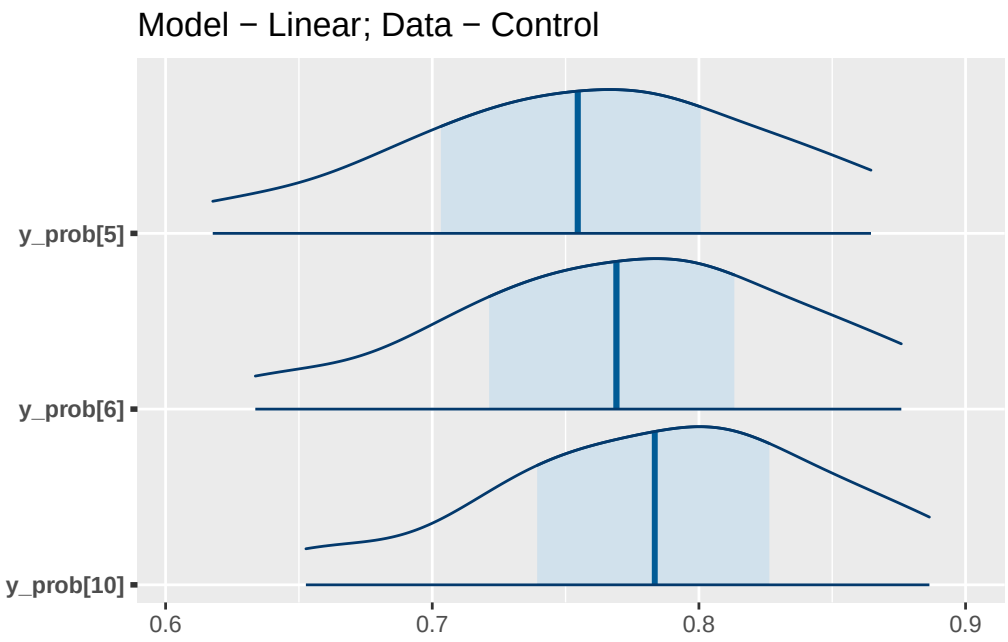
```
[1] 0.6785714
```

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

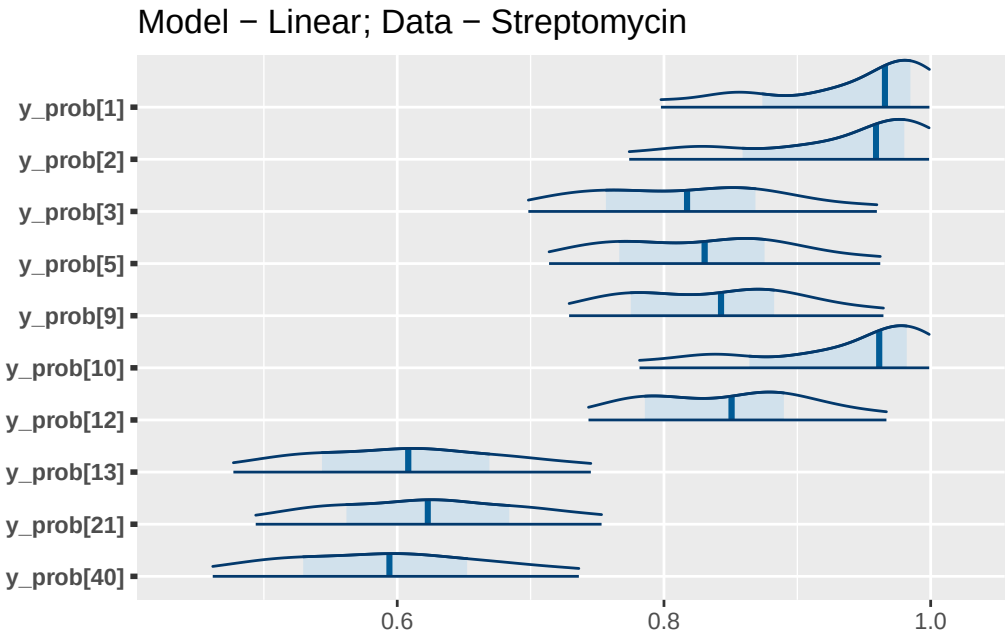
```
Warning in cbind(probs, data_labels): number of rows of result is not a
multiple of vector length (arg 2)
```

```
[1] 0.8461538
```

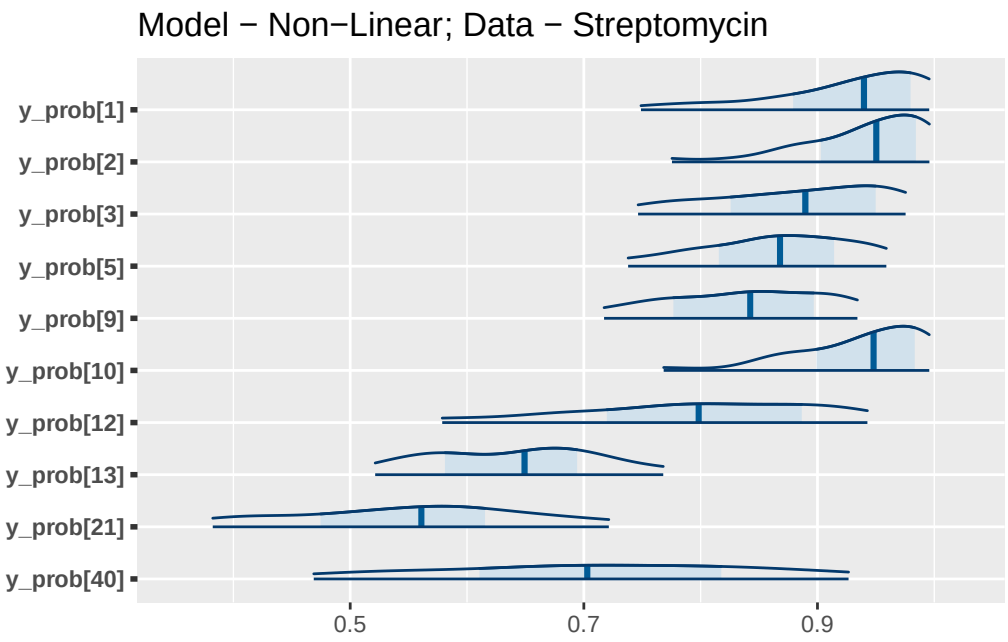
```
unique_arg_values <- function(fit, if_linear, is_medicin) {
  means <- fit$summary()[['mean']]
  unique_means <- unique(means)
  indices_of_unique_means <- match(unique_means, means)
if(if_linear){
  if(is_medicin){
    return(indices_of_unique_means[5:length(indices_of_unique_means)])
  }else{
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