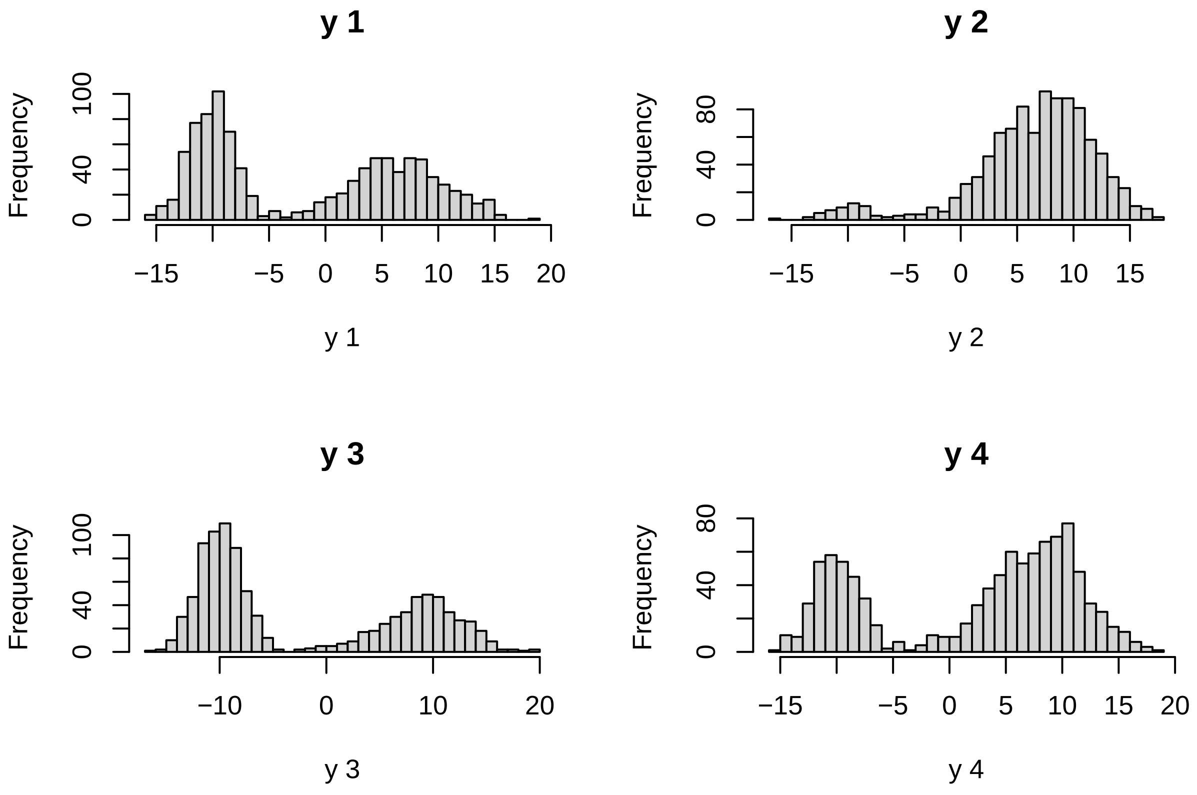
Extra Credit

Yuki Joyama (yj2803)

```r
# import data
data <- read.table("hw4_data_prob3.txt", header = TRUE)

# check histograms
par(mfrow = c(2, 2))
for (i in 1:4) {
  hist(data[[i]], main = paste("y", i), xlab = paste("y", i), breaks = 30)
}
```



```r
par(mfrow = c(1, 1))

# data cleaning for stan
stan_data <- list(
  N = nrow(data),
  D = ncol(data),
  K = 3,
  y = as.matrix(data)
)

# compile the stan model
stan_model <- stan_model("mixture_model.stan")
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 15.0.0 (clang-1500.1.0.2.5)’
## using SDK: ‘MacOSX14.2.sdk’
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Framew
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/RcppEigen/
## In file included from /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/RcppEigen/include/Eigen/src/Core
## #include <cmath>
##          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```
# fit the model
fit <- sampling(
  stan_model,
  data = stan_data,
  iter = 1000,
  chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001745 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 17.45 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 1: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 1: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 1: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 1: Iteration: 400 / 1000 [ 40%]  (Warmup)
```

```
## Chain 1: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 1: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 1: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 1: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 1: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 1: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 1: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 145.602 seconds (Warm-up)
## Chain 1:                248.547 seconds (Sampling)
## Chain 1:                394.149 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001135 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 11.35 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 2: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 2: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 2: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 2: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 2: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 2: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 2: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 2: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 2: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 2: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 2: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 71.991 seconds (Warm-up)
## Chain 2:                43.594 seconds (Sampling)
## Chain 2:                115.585 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001161 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 11.61 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
```

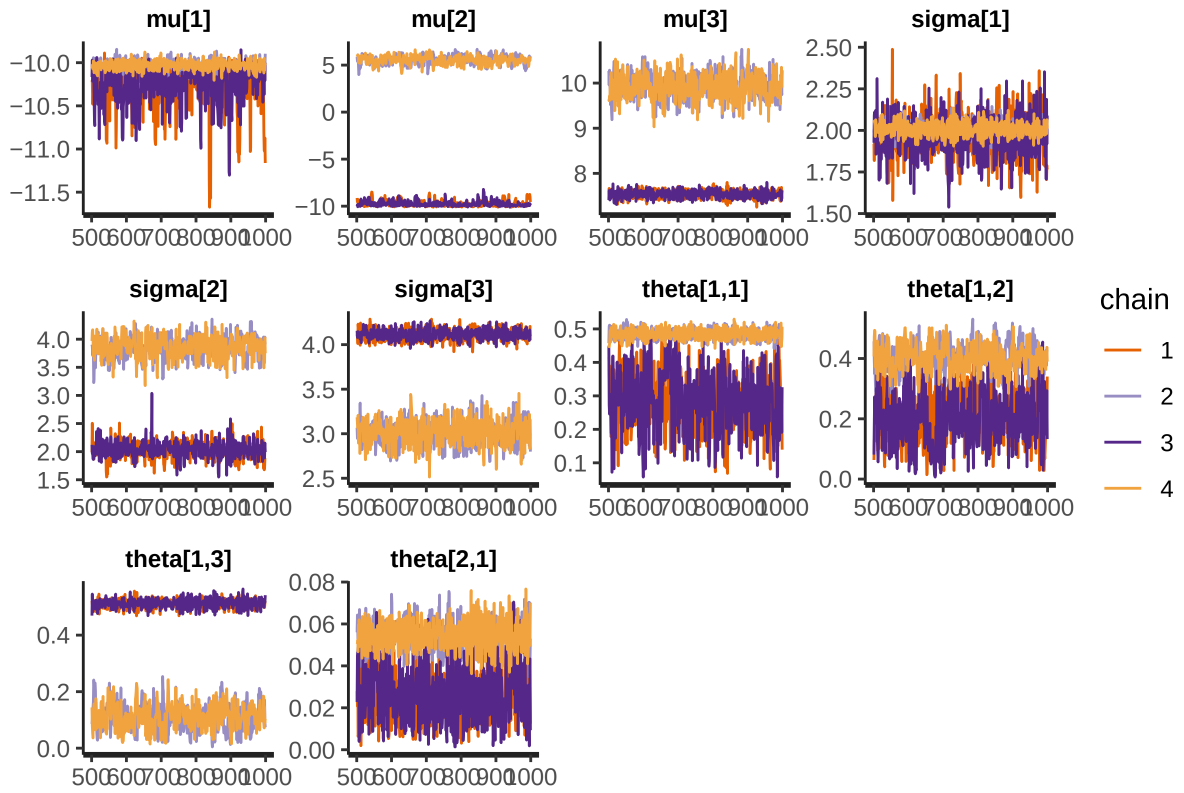
```
## Chain 3: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 3: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 3: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 3: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 3: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 3: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 3: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 3: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 3: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 3: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 3: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 3: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 134.107 seconds (Warm-up)
## Chain 3:                125.603 seconds (Sampling)
## Chain 3:                259.71 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0.001161 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 11.61 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 4: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 4: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 4: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 4: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 4: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 4: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 4: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 4: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 4: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 4: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 4: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 50.035 seconds (Warm-up)
## Chain 4:                34.737 seconds (Sampling)
## Chain 4:                84.772 seconds (Total)
## Chain 4: 
```

```r
# summary of posterior distribution of parameters:
print(fit)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=1000; warmup=500; thin=1; 
## post-warmup draws per chain=500, total post-warmup draws=2000.
## 
##                 mean se_mean    sd      2.5%       25%       50%       75%
## mu[1]         -10.15    0.08  0.21    -10.73    -10.20    -10.08    -10.02
## mu[2]          -2.08    5.41  7.66    -10.03     -9.81     -2.11      5.59
## mu[3]           8.73    0.85  1.21      7.40      7.54      8.42      9.92
## sigma[1]        1.99    0.01  0.09      1.76      1.95      2.00      2.04
## sigma[2]        2.94    0.65  0.93      1.79      2.03      3.11      3.88
## sigma[3]        3.56    0.39  0.56      2.78      3.02      3.69      4.11
## theta[1,1]      0.39    0.07  0.12      0.13      0.29      0.45      0.49
## theta[1,2]      0.30    0.07  0.12      0.06      0.20      0.34      0.41
## theta[1,3]      0.31    0.14  0.20      0.04      0.11      0.36      0.51
## theta[2,1]      0.04    0.01  0.02      0.01      0.03      0.05      0.06
## theta[2,2]      0.30    0.19  0.27      0.01      0.03      0.19      0.57
## theta[2,3]      0.66    0.20  0.29      0.26      0.38      0.78      0.94
## theta[3,1]      0.43    0.11  0.17      0.09      0.29      0.53      0.58
## theta[3,2]      0.21    0.07  0.12      0.07      0.11      0.15      0.29
## theta[3,3]      0.36    0.04  0.06      0.26      0.31      0.38      0.42
## theta[4,1]      0.24    0.05  0.09      0.06      0.16      0.28      0.31
## theta[4,2]      0.25    0.07  0.11      0.05      0.15      0.26      0.34
## theta[4,3]      0.52    0.12  0.17      0.25      0.35      0.57      0.68
## lp__       -12505.45   23.52 33.38 -12543.90 -12538.52 -12509.65 -12471.88
##                97.5% n_eff  Rhat
## mu[1]          -9.94     7  1.22
## mu[2]           6.27     2 21.33
## mu[3]          10.37     2  6.40
## sigma[1]        2.16   108  1.04
## sigma[2]        4.15     2  5.86
## sigma[3]        4.22     2  5.53
## theta[1,1]      0.51     3  1.96
## theta[1,2]      0.48     3  1.84
## theta[1,3]      0.54     2  6.42
## theta[2,1]      0.07     3  1.80
## theta[2,2]      0.68     2  5.56
## theta[2,3]      0.95     2  5.90
## theta[3,1]      0.61     2  2.25
## theta[3,2]      0.49     3  1.59
## theta[3,3]      0.44     2  2.74
## theta[4,1]      0.34     3  2.02
## theta[4,2]      0.43     3  1.99
## theta[4,3]      0.71     2  4.68
## lp__       -12468.40     2 12.51
## 
```

```
## Samples were drawn using NUTS(diag_e) at Wed Dec 11 19:49:36 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
# traceplots
traceplot(fit)
```



Variational Bayes

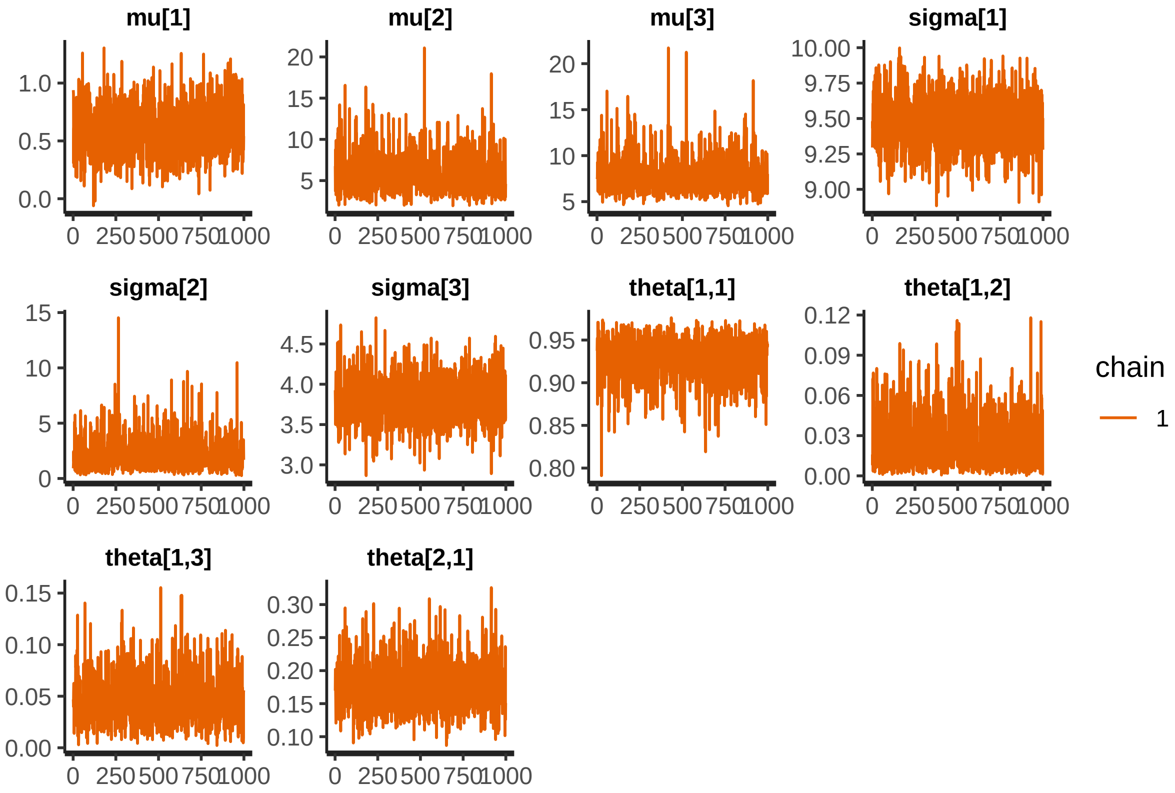
```
# VB works
vb_fit <-
  vb(
    stan_model,
    data = stan_data,
    iter = 1000,
    elbo_samples = 500,
    algorithm = c("fullrank"),
    output_samples = 1000,
    tol_rel_obj = 0.00001
  )
```

```
## Chain 1: ------------------------------------------------------------
## Chain 1: EXPERIMENTAL ALGORITHM:
## Chain 1:   This procedure has not been thoroughly tested and may be unstable
## Chain 1:   or buggy. The interface is subject to change.
## Chain 1: ------------------------------------------------------------
## Chain 1:
## Chain 1:
## Chain 1:
## Chain 1: Gradient evaluation took 0.001454 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 14.54 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Begin eta adaptation.
## Chain 1: Iteration:   1 / 250 [  0%]  (Adaptation)
## Chain 1: Iteration:  50 / 250 [ 20%]  (Adaptation)
## Chain 1: Iteration: 100 / 250 [ 40%]  (Adaptation)
## Chain 1: Iteration: 150 / 250 [ 60%]  (Adaptation)
## Chain 1: Iteration: 200 / 250 [ 80%]  (Adaptation)
## Chain 1: Iteration: 250 / 250 [100%]  (Adaptation)
## Chain 1: Success! Found best value [eta = 0.1].
## Chain 1:
## Chain 1: Begin stochastic gradient ascent.
## Chain 1:   iter             ELBO   delta_ELBO_mean   delta_ELBO_med   notes
## Chain 1:    100       -14857.805             1.000            1.000
## Chain 1:    200       -14508.785             0.512            1.000
## Chain 1:    300       -14409.065             0.015            0.024
## Chain 1:    400       -14346.607             0.006            0.007
## Chain 1:    500       -14282.541             0.004            0.004
## Chain 1:    600       -14244.869             0.004            0.004
## Chain 1:    700       -14186.891             0.003            0.004
## Chain 1:    800       -14146.482             0.003            0.004
## Chain 1:    900       -14141.265             0.002            0.003
## Chain 1:   1000       -14068.186             0.003            0.005
## Chain 1: Informational Message: The maximum number of iterations is reached! The algorithm may not ha
## Chain 1: This variational approximation is not guaranteed to be meaningful.
## Chain 1:
## Chain 1: Drawing a sample of size 1000 from the approximate posterior...
## Chain 1: COMPLETED.
```

```r
# vb esitmates
print(vb_fit)
```

```
## Inference for Stan model: anon_model.
## 1 chains, each with iter=1000; warmup=0; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##             mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff khat
## mu[1]       0.61     NaN 0.21 0.21 0.46 0.61 0.75  1.02   NaN 9.53
## mu[2]       5.71     NaN 2.37 2.64 4.07 5.17 6.75 11.79   NaN 9.48
## mu[3]       7.66     NaN 1.97 5.26 6.40 7.17 8.37 12.70   NaN 9.54
## sigma[1]    9.47     NaN 0.19 9.10 9.35 9.47 9.60  9.84   NaN 9.58
## sigma[2]    2.14     NaN 1.49 0.47 1.15 1.76 2.63  5.85   NaN 9.48
## sigma[3]    3.81     NaN 0.29 3.29 3.62 3.80 4.00  4.40   NaN 9.58
## theta[1,1] 0.93     NaN 0.03 0.87 0.91 0.93 0.95  0.97   NaN 9.57
## theta[1,2] 0.02     NaN 0.02 0.00 0.01 0.02 0.03  0.07   NaN 9.56
## theta[1,3] 0.05     NaN 0.02 0.01 0.03 0.04 0.06  0.10   NaN 9.56
## theta[2,1] 0.18     NaN 0.04 0.11 0.15 0.17 0.20  0.26   NaN 9.57
## theta[2,2] 0.06     NaN 0.04 0.01 0.03 0.05 0.08  0.16   NaN 9.56
## theta[2,3] 0.77     NaN 0.06 0.62 0.74 0.78 0.81  0.86   NaN 9.54
## theta[3,1] 0.96     NaN 0.02 0.90 0.95 0.96 0.97  0.99   NaN 9.58
## theta[3,2] 0.01     NaN 0.01 0.00 0.00 0.00 0.01  0.03   NaN 9.56
## theta[3,3] 0.03     NaN 0.02 0.01 0.02 0.03 0.05  0.09   NaN 9.56
## theta[4,1] 0.66     NaN 0.05 0.55 0.63 0.66 0.70  0.76   NaN 9.57
## theta[4,2] 0.03     NaN 0.03 0.00 0.01 0.02 0.04  0.13   NaN 9.56
## theta[4,3] 0.31     NaN 0.06 0.20 0.27 0.31 0.34  0.42   NaN 9.57
## lp__       0.00     NaN 0.00 0.00 0.00 0.00 0.00  0.00   NaN 9.56
##
## Approximate samples were drawn using VB(fullrank) at Wed Dec 11 19:49:43 2024.
```

```r
# get estimated and generating values for wanted parameters
pars <- vb_fit %>% names %>% `[`(1:10) %>% sort()
sim_summary <- as.data.frame(summary(vb_fit)[[1]])
estimated_values <- sim_summary[pars, c("mean", "2.5%", "97.5%")]
rstan::traceplot(vb_fit)
```


mu[1]
1.0
0.5
0.0
mu[2]
20
15
10
5
mu[3]
20
15
10
5
sigma[1]
10.00
9.75
9.50
9.25
9.00
sigma[2]
15
10
5
0
sigma[3]
4.5
4.0
3.5
3.0
theta[1,1]
0.95
0.90
0.85
0.80
theta[1,2]
0.12
0.09
0.06
0.03
0.00
theta[1,3]
0.15
0.10
0.05
0.00
theta[2,1]
0.30
0.25
0.20
0.15
0.10
0
250
500
750
1000
chain
1