BDA - Assignment 7

Anonymous

Contents

```
# To install aaltobda, see the General information in the assignment.
remotes::install_github("avehtari/BDA_course_Aalto", subdir = "rpackage", upgrade = "never")
```

```
## Skipping install of 'aaltobda' from a github remote, the SHA1 (38f34d35) has not changed since last i
##   Use `force = TRUE` to force installation
```

```
library(aaltobda)
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.3, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file
```

1 Linear model: Drowing data with Stan

```
data("drowning")
```

a) Find the three mistakes

Below I marked the three errors within the Stan script.

```
writeLines(readLines("listing1.stan"))
```

```
## Warning in readLines("listing1.stan"): incomplete final line found on
## 'listing1.stan'
```

```
## // The corrected
## data {
##     int<lower=0> N;     // number of observations
##     vector[N] x;        // observations per year
##     vector[N] y;        // observation number of drowned
##     real xpred;         // prediction year --- ERROR 3: See below
## }
## parameters {
##     real alpha;
##     real beta;
```

```
##     real<lower=0> sigma;    // ERROR 1: sigma must be > 0
## }
## transformed parameters {
##     vector[N] mu = alpha + beta*x;
## }
## model {
##     // Priors
##     beta ~ normal(0, 25);
##
##     // Likelihood function
##     y ~ normal(mu, sigma);   // ERROR 2: line not ending in ";"
## }
## generated quantities {
##    // ERROR 3: mu was not using xpred
##    real ypred = normal_rng(alpha + beta*xpred, sigma);
## }
```

b)

I chose $\beta \sim N(0, \sigma_{beta} = 25)$ since I show below that $Pr(-69 < \beta < 69)$ is alittle above .99.

```
integrate(function(beta) {dnorm (beta, 0, 25)}, -69, 69)
```

```
## 0.9942199 with absolute error < 1.1e-07
```

c)

Below I show how I added the priors, before the likelihood function. Notice that in my Stan script, I already have the priors added.

```
// Priors beta ~ normal(0, 25); alpha ~ normal(1980+143, 28)
```

d)

Because I have no information about the constant (e.g., what the value of drownings was when Jesus was borned), then I rather just keep it as uniform.

Extra:

I show if I obtain similar figures to those in the assignment.

```
# Creating the data
data <- list(N = length(drowning$year),
             x = drowning$year,
             y = drowning$drownings,
             xpred = 2020)
fit <- stan(file = 'listing1.stan', data = data, verbose = FALSE)
```

```
## Warning in readLines(file, warn = TRUE): incomplete
## final line found on 'C:\Users\noe.nava\OneDrive -
## USDA\Drive\courses\vehtari_bayesian_data_analysis\BDA_course_Aalto\navaAssignments\assignment
## 7\listing1.stan'
```

```
##
## SAMPLING FOR MODEL 'listing1' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
```

```
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 1.39 seconds (Warm-up)
## Chain 1:                2.535 seconds (Sampling)
## Chain 1:                3.925 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'listing1' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 1.97 seconds (Warm-up)
## Chain 2:                3.085 seconds (Sampling)
## Chain 2:                5.055 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'listing1' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 1.772 seconds (Warm-up)
## Chain 3:                2.156 seconds (Sampling)
## Chain 3:                3.928 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'listing1' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 1.621 seconds (Warm-up)
## Chain 4:                2.294 seconds (Sampling)
## Chain 4:                3.915 seconds (Total)
## Chain 4:
```

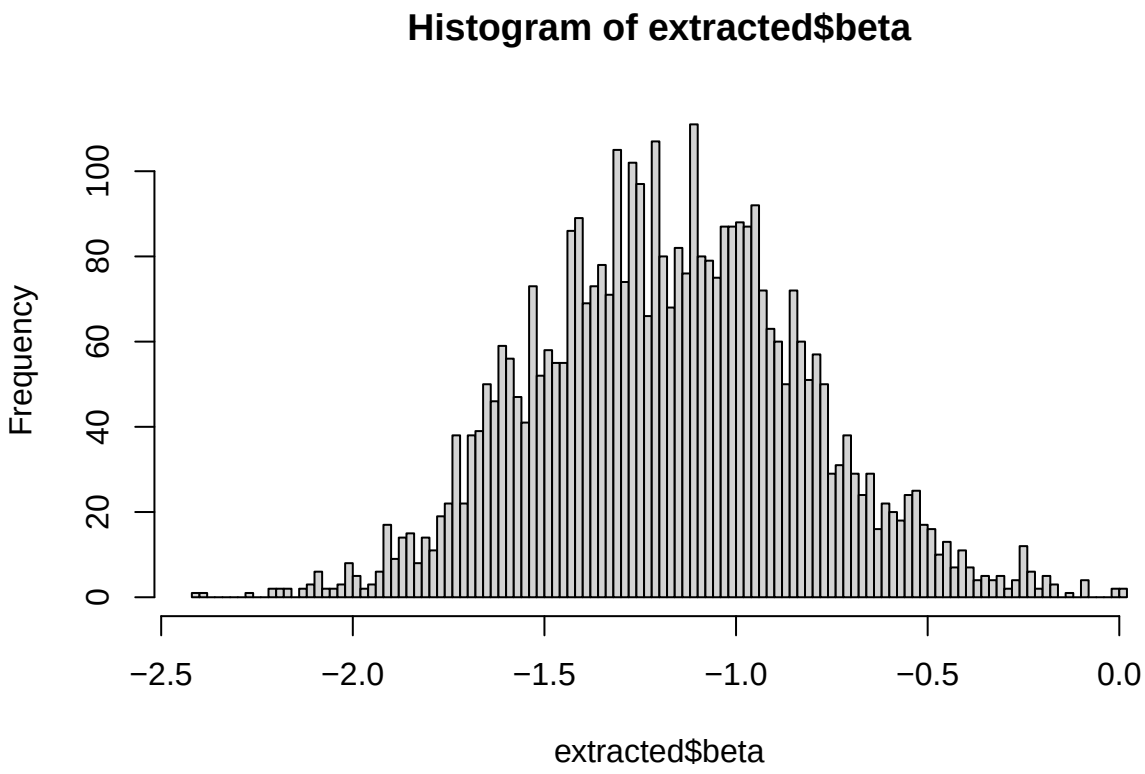
```
## Warning: There were 914 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
extracted <- extract(fit)
```

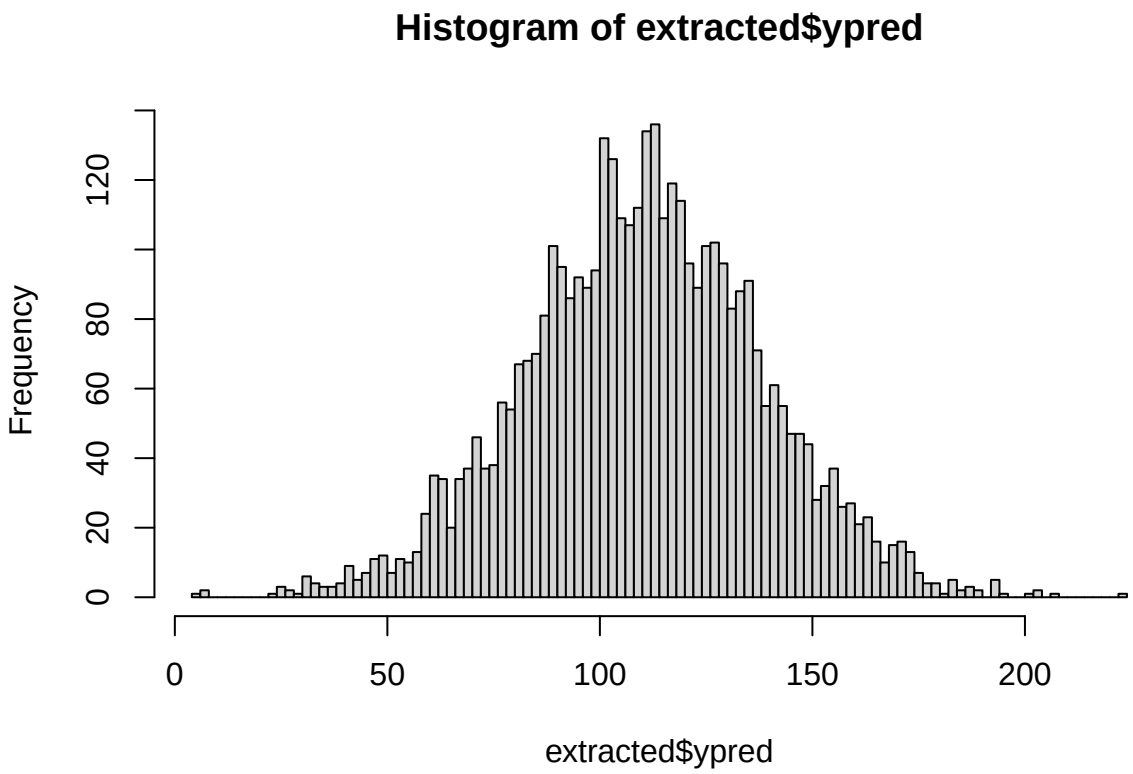
Below I show the histogram for beta:

```
hist(extracted$beta, breaks = 100)
```



Below I show the histogram for the value in 2020:

```
hist(extracted$ypred, breaks = 100)
```



2 Hierarchical model: factory data with Stan

```
data("factory")
ypool <- matrix(as.matrix(factory), ncol = 1)
ypool <- as.vector(ypool) # previous dimensions were 30x1, but Stan expects 30 ... do not know the diff
stan_data <- list(
  y = factory,        # Data in 5x6 dimension
  N = nrow(factory),  # Number of observations per machine
  J = ncol(factory),  # Number of machines
  ypool = ypool
)
```

Why strange results?

Notice that the strange results come from the fact that we are predicting negative factory measurements. This seem to happen when we use weakly informative priors.

```
test <- stan(file = 'test.stan', data = stan_data)
```

```
##
## SAMPLING FOR MODEL 'test' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
```

```
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.06 seconds (Warm-up)
## Chain 1:                0.056 seconds (Sampling)
## Chain 1:                0.116 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'test' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.061 seconds (Warm-up)
## Chain 2:                0.055 seconds (Sampling)
## Chain 2:                0.116 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'test' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.069 seconds (Warm-up)
## Chain 3:                0.055 seconds (Sampling)
## Chain 3:                0.124 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'test' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.07 seconds (Warm-up)
## Chain 4:                0.053 seconds (Sampling)
## Chain 4:                0.123 seconds (Total)
## Chain 4:
```

```
print(test)
```

```
## Inference for Stan model: test.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff
## mu[1]       0.12    0.01  1.00   -1.89   -0.54    0.12    0.77    2.10 10206
## mu[2]       0.08    0.01  1.02   -1.91   -0.63    0.08    0.79    2.07  7961
## mu[3]       0.10    0.01  1.00   -1.83   -0.59    0.11    0.77    2.05  8612
## mu[4]       0.09    0.01  1.01   -1.86   -0.62    0.09    0.78    2.00  9841
## mu[5]       0.09    0.01  0.98   -1.85   -0.57    0.11    0.74    2.03 11445
## mu[6]       0.11    0.01  0.97   -1.78   -0.56    0.11    0.75    2.03  7642
## sigma[1]   60.03    0.22 15.81   38.44   49.05   57.25   67.56   99.00  5378
## sigma[2]   82.34    0.33 21.04   52.57   67.38   78.74   92.79  133.72  4098
```

```
## sigma[3]   67.90    0.24 17.04   43.60   55.82   65.12   76.12  108.65  5251
## sigma[4]   85.58    0.30 21.54   55.73   70.88   81.93   95.77  137.50  5161
## sigma[5]   69.32    0.26 17.38   44.58   57.23   66.23   77.98  112.31  4633
## sigma[6]   67.03    0.23 16.66   43.21   55.05   64.22   75.91  107.73  5482
## ypred       0.89    0.97 60.66 -121.24  -37.46    0.17   39.17  120.18  3916
## lp__     -287.39    0.06  2.47 -293.32 -288.79 -287.06 -285.60 -283.55  1646
##          Rhat
## mu[1]       1
## mu[2]       1
## mu[3]       1
## mu[4]       1
## mu[5]       1
## mu[6]       1
## sigma[1]    1
## sigma[2]    1
## sigma[3]    1
## sigma[4]    1
## sigma[5]    1
## sigma[6]    1
## ypred       1
## lp__        1
##
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:43:55 2022.
## 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).
```

```
testExtracted <- extract(test)
```

a) Description of models:

- Separate model

Our machine observations are assumed to come from

$y_{ij} \sim N(\mu_j, \sigma_j)$

where the prior distribution of the parameters are

$\mu_j \sim N(0, 1)$

$\sigma_j \sim N(0, 1)$

Notice here that what is happening is that the modelling discerns between observations as they come from different machines.

- Pooled model

Our machine observations are assumed to come from a common *pooled* distribution as defined by

$y_i \sim N(\mu, \sigma)$

where in addition, the prior distribution of the parameters are

$\mu \sim N(0, 1)$

$\sigma \sim N(0, 1)$

Notice here that what is happening is that the modelling does not discern between observations as they come from different machines; they are pooled together.

- Hierarchical model

Our machine observations are assumed to come from

$y_{ij} \sim N(\mu_j, \sigma)$

where in addition, the prior distribution of the parameters are

$\mu_j \sim N(0, \tau)$

where τ $N(0, 1)$ is our *hyper-parameters*. The distribution of the shape parameter is drawn as:

$\sigma \sim N(0, 1)$

Notice that here, we are restricting our parameters of location mu_j to be drawn from a common distribution whose parameter of location is, in turn, drawn from another normal distribution, a weekly *hyper-prior*: $\tau \sim N(0, 10)$

b) Stan scripts for each model

- Pooled model:

```
pooled <- stan(file = 'pooled.stan', data = stan_data, verbose = FALSE)
```

```
##
## SAMPLING FOR MODEL 'pooled' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.014 seconds (Warm-up)
## Chain 1:                0.011 seconds (Sampling)
## Chain 1:                0.025 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'pooled' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.015 seconds (Warm-up)
## Chain 2:                0.011 seconds (Sampling)
## Chain 2:                0.026 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'pooled' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.018 seconds (Warm-up)
## Chain 3:                0.013 seconds (Sampling)
## Chain 3:                0.031 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'pooled' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.01 seconds (Warm-up)
## Chain 4:                0.021 seconds (Sampling)
## Chain 4:                0.031 seconds (Total)
## Chain 4:
```

```
pooledExtracted <- extract(pooled)
writeLines(readLines("pooled.stan"))
```

```
## data {
##     int<lower=0> N; // Numnber of observations per machine
##     int<lower=0> J; // Number of machines
##     vector[J] y[N]; // This seems to create a matrix
##     vector[J*N] ypool; // pooled ys
## }
##
## parameters {
##     real mu;
##     real<lower=0> sigma;
## }
##
## model {
##     // Priors
##     mu ~ normal(0,1);
##     sigma ~ normal(0,10);
##
##     // Likelihood
##     ypool ~ normal(mu, sigma);
## }
##
## generated quantities {
##    real ypred6;
## // for the sixth machine
##    ypred6 = normal_rng(mu, sigma);
## }
```

- separate model:

```
separate <- stan(file = 'separate.stan', data = stan_data, verbose = FALSE)
```

```
##
## SAMPLING FOR MODEL 'separate' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.073 seconds (Warm-up)
## Chain 1:                0.056 seconds (Sampling)
## Chain 1:                0.129 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'separate' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.092 seconds (Warm-up)
## Chain 2:                0.077 seconds (Sampling)
## Chain 2:                0.169 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'separate' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.087 seconds (Warm-up)
## Chain 3:                0.054 seconds (Sampling)
## Chain 3:                0.141 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'separate' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.069 seconds (Warm-up)
## Chain 4:                0.053 seconds (Sampling)
## Chain 4:                0.122 seconds (Total)
## Chain 4:
```

```
separateExtracted <- extract(separate)
writeLines(readLines("separate.stan"))
```

```
## data {
##     int<lower=0> N; // Numnber of observations per machine
##     int<lower=0> J; // Number of machines
##     vector[J] y[N]; // This seems to create a matrix
##     vector[J*N] ypool; // pooled ys
## }
##
## parameters {
##     vector[J] mu;
##     vector<lower=0>[J] sigma;
## }
##
## model {
##     // Priors
##     for(j in 1:J) {
##         mu[j] ~ normal(0,1);
```

```
##         sigma[j] ~ normal(0,10);
##     }
##
##     // Likelihood
##     for(j in 1:J) {
##         y[,j] ~ normal(mu[j], sigma[j]);
##     }
## }
##
## generated quantities {
##    real ypred6;
## // for the sixth machine
##    ypred6 = normal_rng(mu[6], sigma[6]);
## }
```

- Hierarchical model

```
hierarchical <- stan(file = 'hierarchical.stan', data = stan_data, verbose = FALSE)
```

```
##
## SAMPLING FOR MODEL 'hierarchical' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.14 seconds (Warm-up)
## Chain 1:                0.098 seconds (Sampling)
## Chain 1:                0.238 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'hierarchical' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.105 seconds (Warm-up)
## Chain 2:                0.077 seconds (Sampling)
## Chain 2:                0.182 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'hierarchical' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.268 seconds (Warm-up)
## Chain 3:                0.098 seconds (Sampling)
## Chain 3:                0.366 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'hierarchical' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.116 seconds (Warm-up)
## Chain 4:                0.097 seconds (Sampling)
## Chain 4:                0.213 seconds (Total)
## Chain 4:
```

```
## Warning: There were 4 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

```
hierarchicalExtracted <- extract(hierarchical)
writeLines(readLines("hierarchical.stan"))
```

```
## data {
##     int<lower=0> N; // Numnber of observations per machine
##     int<lower=0> J; // Number of machines
##     vector[J] y[N]; // This seems to create a matrix
##     vector[J*N] ypool; // pooled ys
## }
##
## parameters {
##     vector[J+1] mu;      // Notice dimensions J+1 in order to get info on the seventh mu and get prec
##     real<lower=0> sigma;
## // Hyper-parameter
##     real<lower=0> tau;
## }
##
## model {
## // Hyper-priors
##     tau ~ normal(0,1);
##
## // Priors
##     for(j in 1:(J+1)) {
##         mu[j] ~ normal(0,tau);
##     }
##     sigma ~ normal(0,10);
##
## // Likelihood
##     for(j in 1:J) {
##         y[,j] ~ normal(mu[j], sigma);
##     }
## }
```

```
##
## generated quantities {
##    real ypred6;
##    real ypred7;
## // for the sixth machine
##    ypred6 = normal_rng(mu[6], sigma);
## // for the seventh machine
##     ypred7 = normal_rng(mu[7], sigma);
## }
```

c)

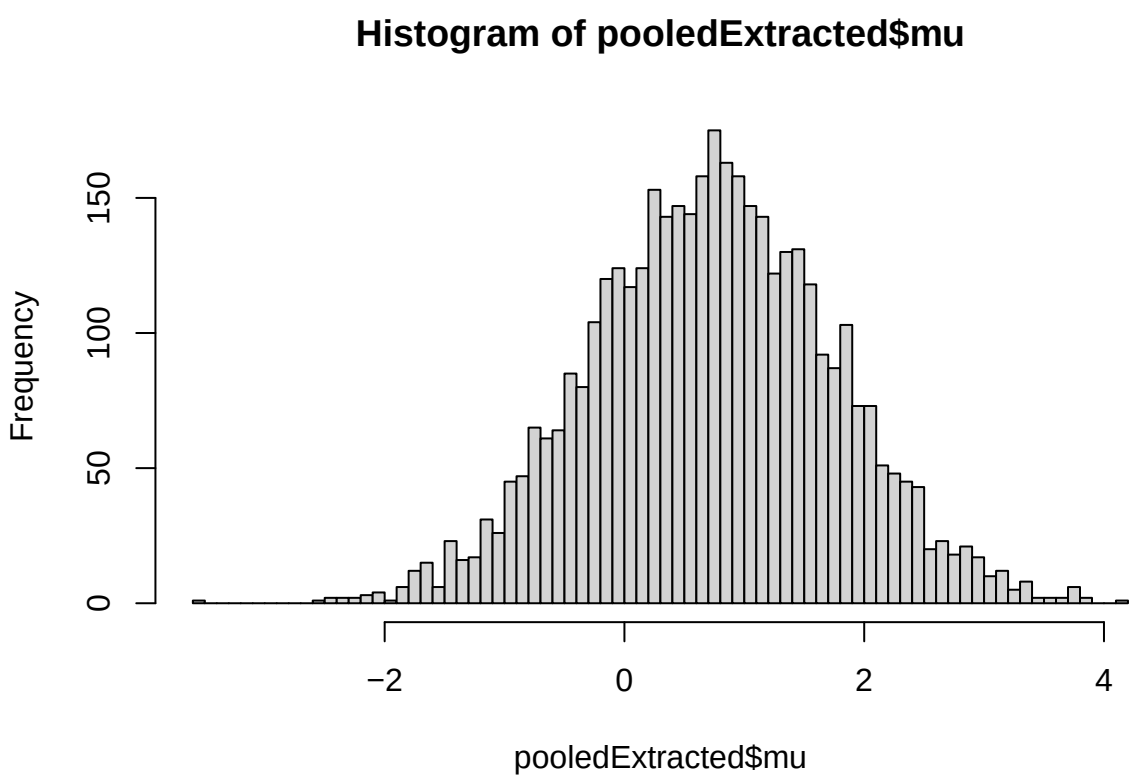
the posterior distribution of the mean of the quality measurements of the sixth machine — and — the predictive distribution for another quality measurement of the sixth machine — and — the posterior distribution of the mean of the quality measurements of the seventh machine

- Pooled: Below I show both statistics on the location and dispersion parameter of the pooled model, which represents the sixth and seventh machines (since they are all pooled together). I also show histograms.

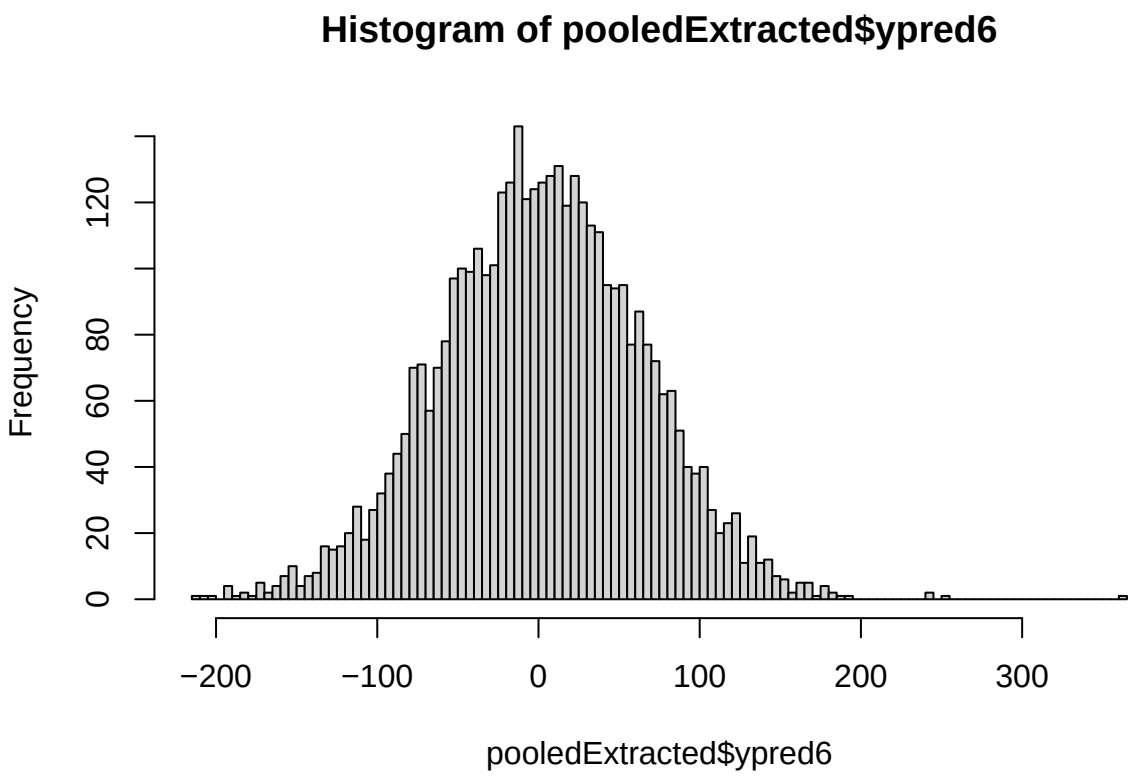
```
print(pooled)
```

```
## Inference for Stan model: pooled.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff Rhat
## mu        0.72    0.02  1.01   -1.25    0.04    0.74    1.41    2.74  3857    1
## sigma    62.67    0.07  4.34   54.95   59.70   62.52   65.33   72.32  3496    1
## ypred6    2.06    1.03 62.47 -120.81  -40.16    2.24   44.24  123.45  3707    1
## lp__   -174.57    0.02  1.03 -177.35 -174.92 -174.27 -173.86 -173.59  1740    1
##
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:44:22 2022.
## 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).
```

```
hist(pooledExtracted$mu, breaks = 100)
```




```
hist(pooledExtracted$ypred6, breaks = 100)
```



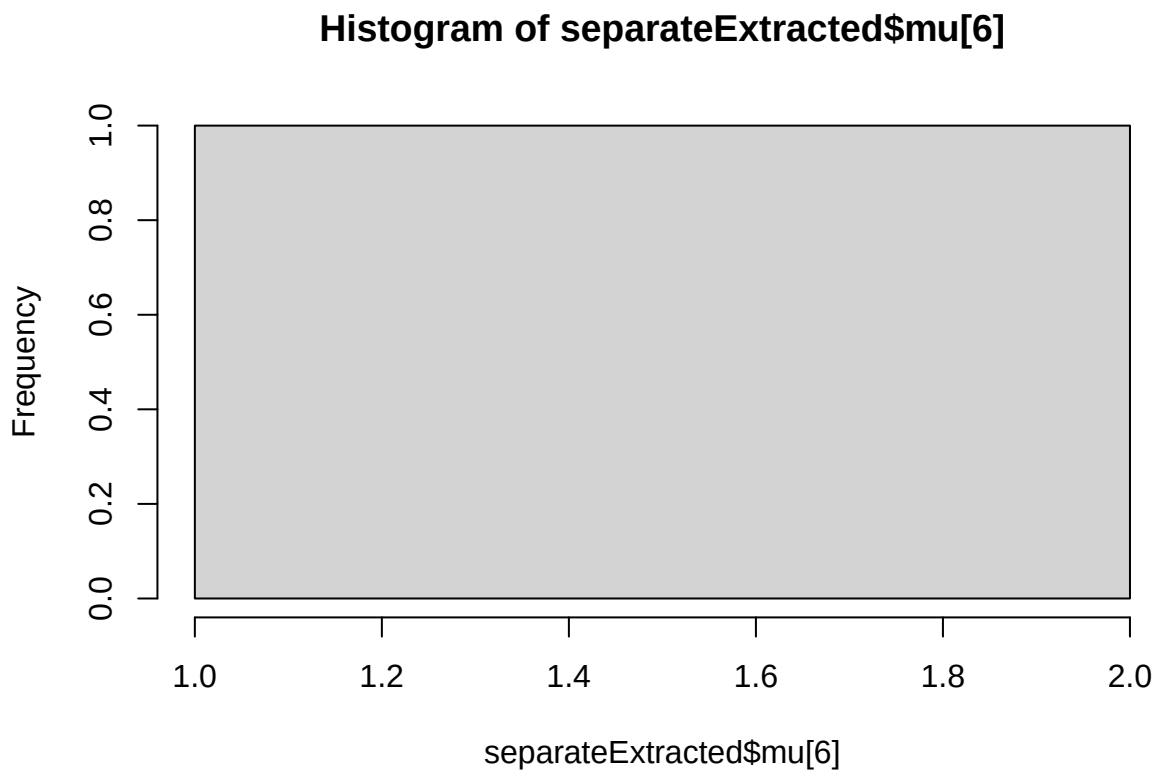
- Separate: Below I show both statistics on the location and dispersion parameter of the separate model. Because I could not simply draw another mu, I am unable to say something about the seventh machine. I show histograms for mu_6 and notice that since this is not a distribution, then I only have one value. The predictive value for the sixth machine is also shown with negative values.

```
print(separate)
```

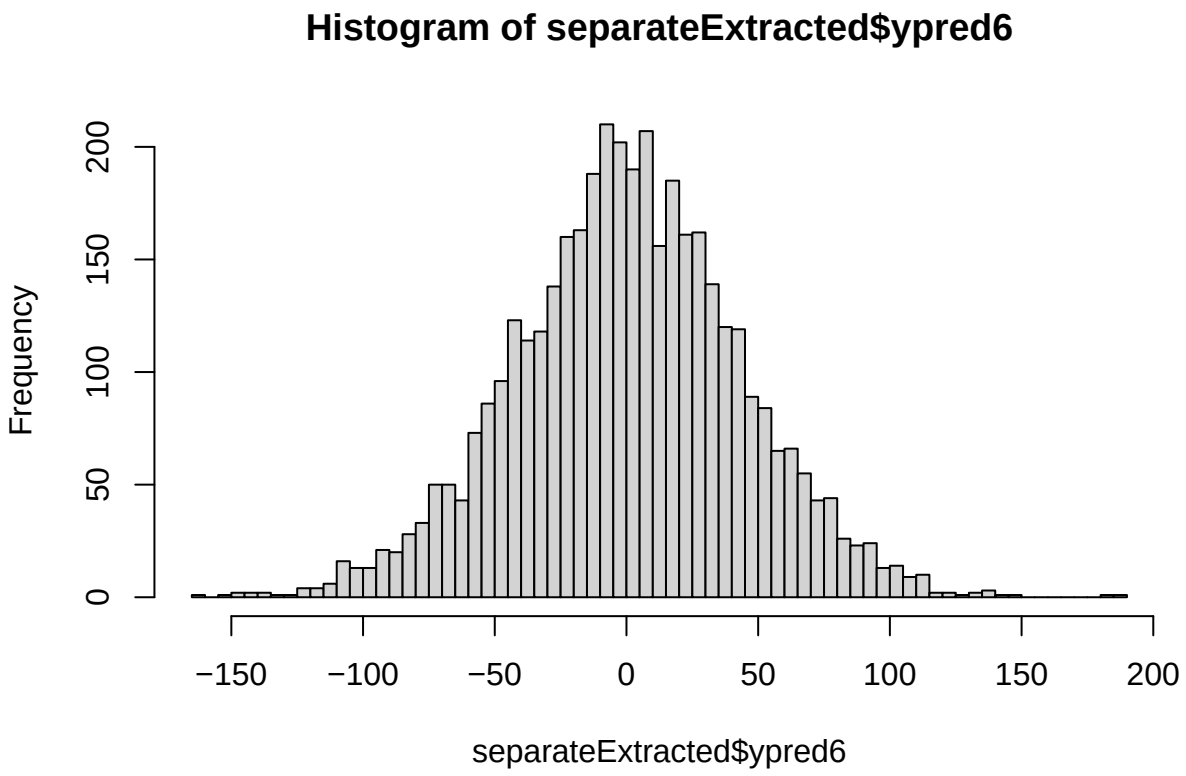
```
## Inference for Stan model: separate.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff
## mu[1]       0.24    0.01  1.00   -1.72   -0.47    0.25    0.93    2.14  7787
## mu[2]       0.24    0.01  1.00   -1.74   -0.43    0.26    0.91    2.24  8218
## mu[3]       0.24    0.01  1.00   -1.70   -0.45    0.24    0.91    2.20  9392
## mu[4]       0.26    0.01  1.01   -1.70   -0.42    0.26    0.95    2.22 10444
## mu[5]       0.24    0.01  1.01   -1.76   -0.46    0.25    0.93    2.19  8262
## mu[6]       0.25    0.01  1.00   -1.68   -0.45    0.26    0.93    2.21  9248
## sigma[1]   39.72    0.06  4.81   31.12   36.39   39.50   42.70   49.93  7220
## sigma[2]   47.09    0.05  4.64   38.64   43.83   46.84   50.24   56.78  7922
## sigma[3]   42.59    0.05  4.72   34.26   39.19   42.35   45.74   52.53 10501
## sigma[4]   48.33    0.05  4.83   39.55   44.89   48.08   51.40   58.39  8563
## sigma[5]   43.02    0.05  4.77   34.41   39.66   42.78   46.02   53.19  8343
## sigma[6]   42.28    0.05  4.67   34.22   38.98   41.95   45.38   52.35  8425
## ypred6      0.56    0.74 43.34  -86.82  -27.31    0.48   28.52   86.67  3410
## lp__     -222.87    0.06  2.33 -228.30 -224.20 -222.58 -221.21 -219.19  1604
##          Rhat
```

```
## mu[1]       1
## mu[2]       1
## mu[3]       1
## mu[4]       1
## mu[5]       1
## mu[6]       1
## sigma[1]    1
## sigma[2]    1
## sigma[3]    1
## sigma[4]    1
## sigma[5]    1
## sigma[6]    1
## ypred6      1
## lp__        1
##
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:44:53 2022.
## 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).
```

```
hist(separateExtracted$mu[6], breaks = 100)
```




```
hist(separateExtracted$ypred6, breaks = 100)
```



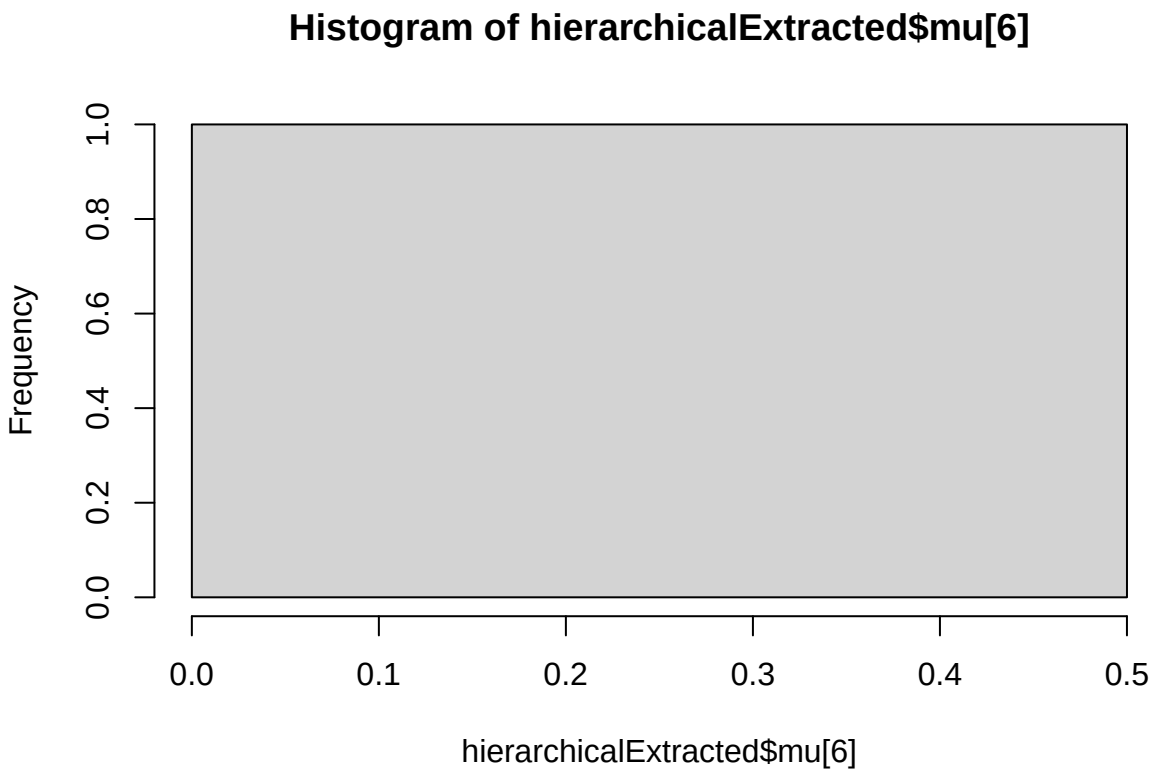
- Hierarchical: Below I show both statistics on the location and dispersion parameter of the hierarchical model. Because I could simply draw another mu, I am able to say something about the seventh machine. I show histograms for mu_6 and notice that since this is not a distribution, then I only have one value. The predictive value for the sixth and seventh machine is also shown with negative values.

```
print(hierarchical)
```

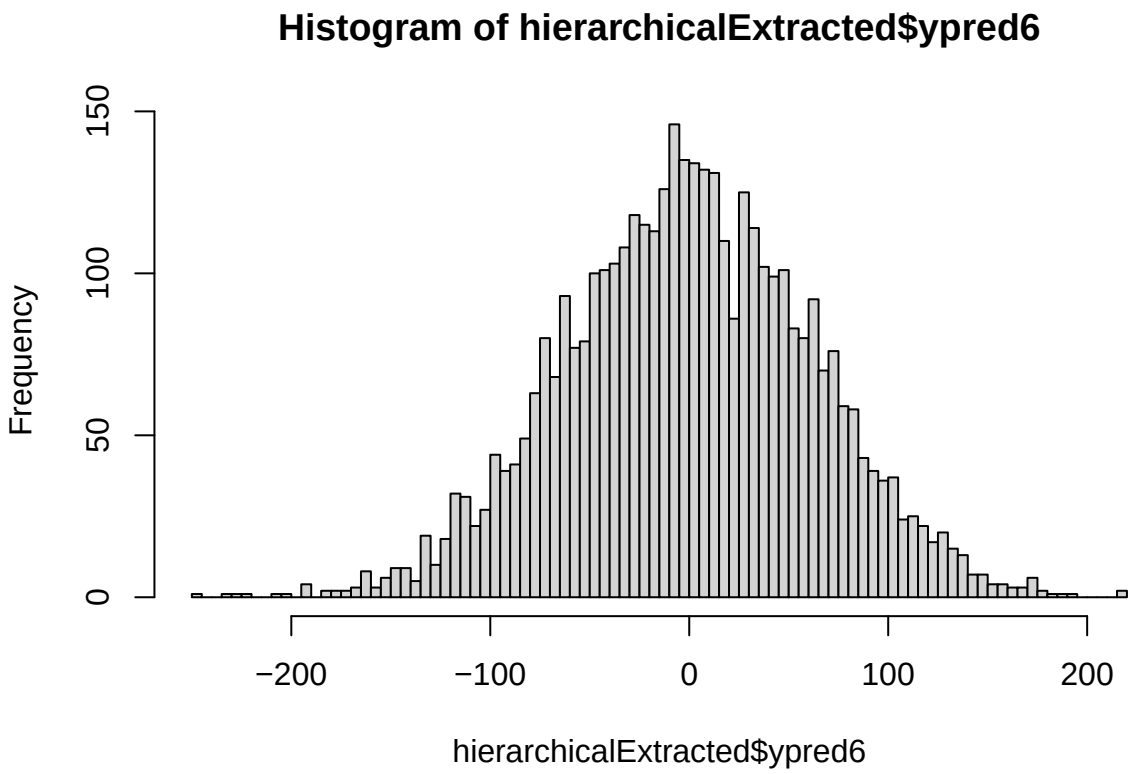
```
## Inference for Stan model: hierarchical.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff Rhat
## mu[1]     0.10    0.02  1.13   -2.17   -0.41    0.05    0.55    2.65  4040 1.00
## mu[2]     0.17    0.02  1.12   -2.05   -0.37    0.08    0.60    2.90  2912 1.00
## mu[3]     0.11    0.02  1.15   -2.18   -0.39    0.04    0.55    2.81  3995 1.00
## mu[4]     0.16    0.02  1.15   -2.10   -0.37    0.06    0.63    2.82  4343 1.00
## mu[5]     0.13    0.02  1.16   -2.23   -0.38    0.07    0.59    2.84  3958 1.00
## mu[6]     0.15    0.02  1.19   -2.24   -0.40    0.07    0.63    2.95  4519 1.00
## mu[7]     0.02    0.02  1.11   -2.42   -0.46    0.02    0.50    2.46  4183 1.00
## sigma    62.98    0.11  4.32   54.84   59.98   62.88   65.83   71.70  1492 1.00
## tau       0.95    0.03  0.61    0.18    0.46    0.82    1.32    2.40   337 1.01
## ypred6   -0.45    0.99 62.51 -120.96  -42.71   -0.64   42.43  121.45  3949 1.00
## ypred7    0.51    0.99 64.16 -127.12  -43.15    0.71   44.98  124.14  4211 1.00
## lp__   -176.86    0.32  5.22 -186.92 -180.58 -177.02 -173.04 -167.33   260 1.02
##
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:45:26 2022.
## 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).
```

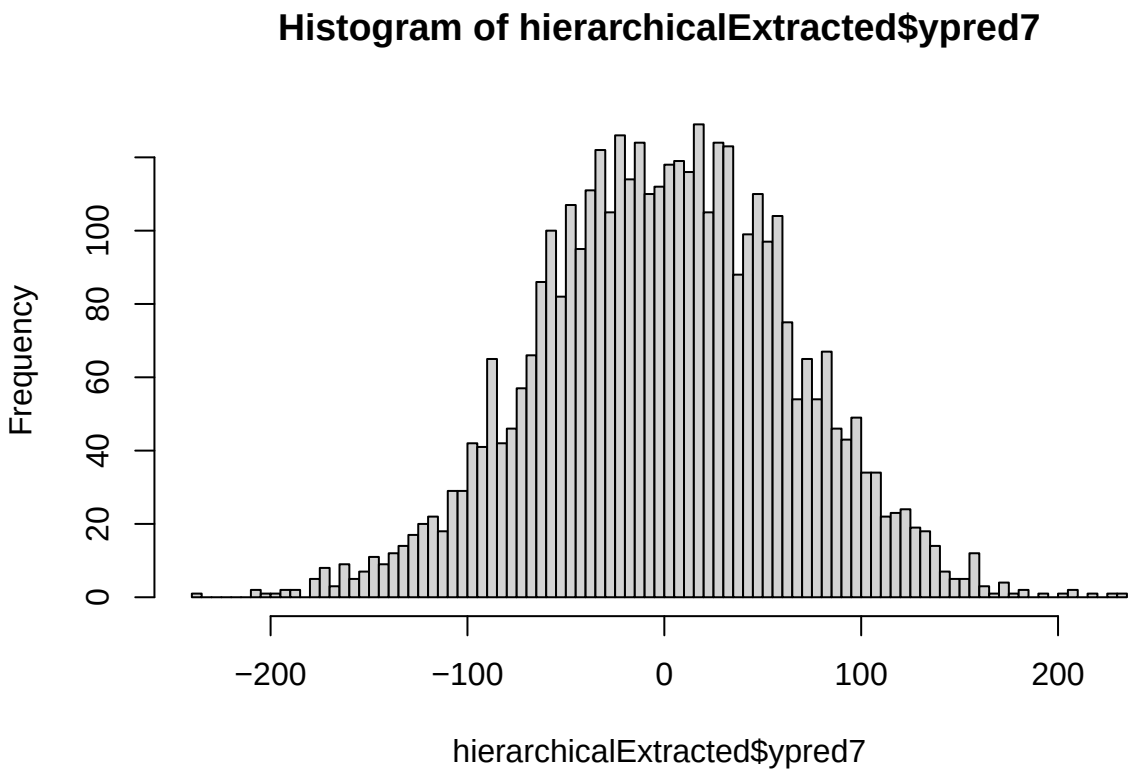
```
hist(hierarchicalExtracted$mu[6], breaks = 100)
```




```
hist(hierarchicalExtracted$ypred6, breaks = 100)
```




```
hist(hierarchicalExtracted$ypred7, breaks = 100)
```



d)

First, we create the function to report the mean with the 90% true intervals, but also notice that we do not have negative predictions, implying that prios matter a lot!:

```
estIntervals <- function(sims) {

  est <- mean(sims)
  low <- quantile(sims, .05)
  upp <- quantile(sims, .95)

  value <- list(
    est = est,
    low = low,
    upp = upp
  )

  return(value)
}
```

- Pooled model:

```
pooledv2 <- stan(file = 'pooledv2.stan', data = stan_data, verbose = FALSE)
```

```
##
## SAMPLING FOR MODEL 'pooledv2' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
```

```
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.02 seconds (Warm-up)
## Chain 1:                0.012 seconds (Sampling)
## Chain 1:                0.032 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'pooledv2' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.022 seconds (Warm-up)
## Chain 2:                0.017 seconds (Sampling)
## Chain 2:                0.039 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'pooledv2' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
```

```
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.019 seconds (Warm-up)
## Chain 3:                0.008 seconds (Sampling)
## Chain 3:                0.027 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'pooledv2' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.029 seconds (Warm-up)
## Chain 4:                0.013 seconds (Sampling)
## Chain 4:                0.042 seconds (Total)
## Chain 4:
```

```
print(pooledv2)
```

```
## Inference for Stan model: pooledv2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff Rhat
## mu      85.54    0.08  3.26   78.33   83.57   85.71   87.78   91.40  1749    1
## sigma   16.09    0.04  1.81   13.05   14.80   15.92   17.14   20.22  1839    1
## ypred   85.20    0.26 16.39   52.82   74.34   85.03   96.15  117.21  3940    1
## lp__  -155.52    0.02  1.02 -158.35 -155.91 -155.21 -154.79 -154.55  1755    1
##
```

```
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:45:56 2022.
## 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).
```

```
pooledv2Extracted <- extract(pooledv2)

estIntervals(pooledv2Extracted$mu)
```

```
## $est
## [1] 85.54127
##
## $low
##       5%
## 79.84953
##
## $upp
##      95%
## 90.57616
```

- separate model:

```
separatev2 <- stan(file = 'separatev2.stan', data = stan_data, verbose = FALSE)
```

```
## Warning in readLines(file, warn = TRUE): incomplete
## final line found on 'C:\Users\noe.nava\OneDrive -
## USDA\Drive\courses\vehtari_bayesian_data_analysis\BDA_course_Aalto\navaAssignments\assignment
## 7\separatev2.stan'
```

```
##
## SAMPLING FOR MODEL 'separatev2' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.313 seconds (Warm-up)
## Chain 1:                0.112 seconds (Sampling)
## Chain 1:                0.425 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'separatev2' NOW (CHAIN 2).
```

```
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.276 seconds (Warm-up)
## Chain 2:                0.138 seconds (Sampling)
## Chain 2:                0.414 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'separatev2' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.299 seconds (Warm-up)
## Chain 3:                0.131 seconds (Sampling)
## Chain 3:                0.43 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'separatev2' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
```

```
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.266 seconds (Warm-up)
## Chain 4:                0.121 seconds (Sampling)
## Chain 4:                0.387 seconds (Total)
## Chain 4:
```

```
print(separatev2)
```

```
## Inference for Stan model: separatev2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff
## mu[1]      50.53    0.15  8.99   31.75   44.44   51.25   56.99   66.44  3450
## mu[2]      48.95    0.23 12.22   25.84   40.46   48.64   57.13   73.85  2859
## mu[3]      58.79    0.23 11.90   34.00   50.64   59.53   67.72   79.05  2715
## mu[4]      47.66    0.22 12.25   23.65   39.52   47.39   55.96   72.11  3087
## mu[5]      61.08    0.23 12.44   35.96   52.70   61.61   70.38   82.35  2932
## mu[6]      51.31    0.19 10.58   29.64   44.20   51.94   58.96   70.69  3224
## sigma[1]   15.96    0.06  3.33   10.37   13.52   15.76   18.04   23.08  3370
## sigma[2]   24.67    0.09  4.50   16.07   21.72   24.69   27.72   33.41  2624
## sigma[3]   15.96    0.08  4.48    8.49   12.62   15.59   18.87   25.73  2815
## sigma[4]   26.46    0.08  4.53   17.76   23.31   26.40   29.47   35.60  3151
## sigma[5]   15.67    0.09  4.76    7.46   12.18   15.37   18.85   25.45  2672
## sigma[6]   18.59    0.07  3.88   11.71   15.78   18.35   21.18   26.86  3508
## ypred      50.77    0.30 18.73    9.79   39.18   51.90   63.72   84.55  3966
## lp__     -347.01    0.06  2.37 -352.41 -348.44 -346.70 -345.27 -343.34  1782
##          Rhat
## mu[1]       1
## mu[2]       1
## mu[3]       1
## mu[4]       1
## mu[5]       1
## mu[6]       1
## sigma[1]    1
## sigma[2]    1
## sigma[3]    1
## sigma[4]    1
## sigma[5]    1
## sigma[6]    1
## ypred       1
```

```
## lp__        1
##
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:46:26 2022.
## 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).
```

```
separatev2Extracted <- extract(separatev2)

estIntervals(separatev2Extracted$mu[1])
```

```
## $est
## [1] 41.94509
##
## $low
##       5%
## 41.94509
##
## $upp
##      95%
## 41.94509
```

- Hierarchical model

```
hierarchicalv2 <- stan(file = 'hierarchicalv2.stan', data = stan_data, verbose = FALSE)
```

```
##
## SAMPLING FOR MODEL 'hierarchicalv2' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.33 seconds (Warm-up)
## Chain 1:                0.139 seconds (Sampling)
## Chain 1:                0.469 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'hierarchicalv2' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
```

```
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.196 seconds (Warm-up)
## Chain 2:                0.103 seconds (Sampling)
## Chain 2:                0.299 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'hierarchicalv2' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.222 seconds (Warm-up)
## Chain 3:                0.109 seconds (Sampling)
## Chain 3:                0.331 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'hierarchicalv2' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.193 seconds (Warm-up)
## Chain 4:                0.682 seconds (Sampling)
## Chain 4:                0.875 seconds (Total)
## Chain 4:
```

```
## Warning: There were 498 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

```
print(hierarchicalv2)
```

```
## Inference for Stan model: hierarchicalv2.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean    sd    2.5%     25%     50%     75%   97.5% n_eff
## mu[1]      77.97    0.33  6.28   64.15   74.39   78.16   82.74   88.43   365
## mu[2]      88.59    0.82  9.48   73.91   81.68   86.97   95.65  107.75   134
## mu[3]      82.22    0.38  5.73   71.16   78.14   82.44   85.92   93.75   228
## mu[4]      90.44    0.96 11.04   74.17   81.75   87.83   99.33  112.81   133
## mu[5]      82.96    0.41  5.78   72.25   78.96   82.89   86.74   94.60   199
## mu[6]      81.57    0.34  5.47   70.62   77.87   81.88   85.16   92.01   262
## sigma      18.32    0.29  3.69   12.29   15.59   17.90   21.05   26.38   165
## tauMu      75.20    0.61 10.16   48.95   71.09   77.45   82.24   87.71   278
## tauSigma    6.77    0.51  6.36    0.69    1.43    4.05   11.25   21.46   156
## ypred      77.71    0.39 19.83   39.34   64.84   77.50   90.61  118.18  2631
## lp__     -150.03    0.51  5.10 -158.26 -153.90 -151.29 -145.74 -141.05    99
##          Rhat
## mu[1]    1.01
## mu[2]    1.02
## mu[3]    1.02
## mu[4]    1.02
## mu[5]    1.02
## mu[6]    1.02
## sigma    1.02
```

```
## tauMu    1.01
## tauSigma 1.01
## ypred    1.00
## lp__     1.02
##
## Samples were drawn using NUTS(diag_e) at Wed Mar 30 15:46:56 2022.
## 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).
```

```
hierarchicalv2Extracted <- extract(hierarchicalv2)

estIntervals(hierarchicalv2Extracted$mu[1])
```

```
## $est
## [1] 71.53446
##
## $low
##       5%
## 71.53446
##
## $upp
##      95%
## 71.53446
```