Assignment7

Contents

Assignment 7

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

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

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

```
## rstan (Version 2.21.5, 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)
```

```
data("drowning")
data("factory")
set.seed(5)
```

Exercise 1: Linear model, drowning data with Stan

a)

Error 1: there is a ";" missing in line 16

Error 2: sigma should have a lower bound lower=0 instead of upper=0

Error 3: function `normal_rng(reals mu, reals sigma)` has `real` value parameters, so the correct implementation should be using `alpha+beta*xpred` instead of `mu`

We have the corrected Stan code:

```
data {
  int<lower=0> N; // number of data points
  vector[N] x; // observation year
  vector[N] y; // observation number of drowned
  real xpred; // prediction year
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
transformed parameters {
  vector[N] mu = alpha + beta*x;
```

```
}
model {
  y ~ normal(mu, sigma);
}
generated quantities {
  real ypred = normal_rng(alpha + beta*xpred, sigma);
}
```

b)

Set σ_b with β follows $\mathcal{N}(0, \sigma_b)$ so that: $Pr(-69 < \beta < 69) = Pr(\frac{-69}{\sigma_b} < \frac{\beta - 0}{\sigma_b} < \frac{69}{\sigma_b}) = Pr(\frac{-69}{\sigma_b} < z < \frac{69}{\sigma_b}) = 0.99$

$\Leftrightarrow Pr(z \leq \frac{-69}{\sigma_b}) = \frac{0.01}{2}$

$\Leftrightarrow \frac{-69}{\sigma_b} = -2.58 \Leftrightarrow \sigma_b = \frac{-69}{-2.58} = 26.744$

c)

Added line `real sigma_b;` in the `data` block

Added line `beta ~ normal(0, sigma_b)` in the `model` block

d)

The calculated prior is $\mathcal{N}(0, 26.744)$.

We have the approximate historical mean yearly number of drownings of 138 $\Rightarrow \mu = \alpha + \beta x = 138$.

From this, having α follows $\mathcal{N}(138, \sigma_a)$, we have: $P(138 - 69 \times 1980 < \alpha < 138 + 69 \times 1980)$

$= P(-69 \times 1980 < \alpha - 138 < 69 \times 1980)$

$= P(\frac{-69 \times 1980}{\sigma_a} < \frac{\alpha - 138}{\sigma_a} < \frac{69 \times 1980}{\sigma_a})$

$= P(\frac{-69 \times 1980}{\sigma_a} < z < \frac{69 \times 1980}{\sigma_a}) = 0.99$

$\Leftrightarrow Pr(z \leq \frac{-69 \times 1980}{\sigma_a}) = \frac{0.01}{2} \Leftrightarrow \frac{-69 \times 1980}{\sigma_a} = -2.58$

$\Leftrightarrow \sigma_a = 52953$

Therefore, we have the final Stan code for question 1:

```
data {
  int<lower=0> N; // number of data points
  vector[N] x; // observation year
  vector[N] y; // observation number of drowned
  real xpred; // prediction year
  real sigma_b; // added line
  real sigma_a; // added line
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
```

```
}
transformed parameters {
  vector[N] mu = alpha + beta*x;
}
model {
  y ~ normal(mu, sigma);
  beta ~ normal(0, sigma_b); // added line
  alpha ~ normal(138, sigma_a); // added line
}
generated quantities {
  real ypred = normal_rng(alpha + beta*xpred, sigma);
}
```

```r
data = list(N = length(drowning$year), x = drowning$year,
            y = drowning$drownings, xpred = 2020,
            sigma_a = 52953, sigma_b = 26.744)
data_ = stan(file = "~/notebooks/bda2022/linear.stan", data=data, refresh=0)
```

```
## Trying to compile a simple C file
```

```
## Warning: There were 779 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
```

```r
print(data_)
```

```
## Inference for Stan model: linear.
## 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
## alpha   2451.64   22.24 721.78 1054.22 1967.90 2437.54 2928.91 3900.48  1054
## beta      -1.16    0.01   0.36   -1.88   -1.40   -1.15   -0.92   -0.46  1053
## sigma     26.29    0.09   3.15   21.17   24.04   25.90   28.11   33.28  1265
## mu[1]    157.01    0.22   8.05  141.35  151.76  156.73  162.31  173.37  1369
## mu[2]    155.85    0.21   7.75  140.66  150.82  155.58  160.95  171.61  1400
## mu[3]    154.69    0.20   7.45  140.09  149.83  154.46  159.60  169.82  1436
## mu[4]    153.53    0.19   7.15  139.65  148.87  153.31  158.26  168.01  1478
## mu[5]    152.38    0.18   6.86  138.98  147.91  152.14  156.94  166.25  1527
## mu[6]    151.22    0.17   6.58  138.35  146.92  151.01  155.61  164.55  1585
## mu[7]    150.06    0.16   6.31  137.69  145.95  149.87  154.30  162.92  1654
## mu[8]    148.90    0.15   6.04  137.05  145.00  148.74  152.90  161.25  1736
## mu[9]    147.74    0.14   5.79  136.40  143.94  147.61  151.55  159.53  1835
## mu[10]   146.58    0.13   5.55  135.67  142.93  146.47  150.24  157.79  1955
## mu[11]   145.42    0.12   5.32  134.99  141.92  145.28  148.94  156.11  2101
## mu[12]   144.26    0.11   5.11  134.16  140.88  144.11  147.63  154.56  2279
## mu[13]   143.10    0.10   4.92  133.45  139.89  142.95  146.36  153.00  2492
## mu[14]   141.95    0.09   4.74  132.73  138.86  141.87  145.08  151.39  2734
## mu[15]   140.79    0.08   4.59  131.91  137.85  140.67  143.77  149.93  2998
## mu[16]   139.63    0.08   4.46  130.91  136.75  139.52  142.55  148.62  3252
## mu[17]   138.47    0.07   4.35  129.85  135.69  138.39  141.31  147.18  3501
## mu[18]   137.31    0.07   4.28  128.94  134.54  137.22  140.04  145.75  3714
## mu[19]   136.15    0.07   4.23  127.91  133.44  136.09  138.85  144.61  3849
## mu[20]   134.99    0.07   4.22  126.79  132.25  134.94  137.68  143.36  3873
## mu[21]   133.83    0.07   4.23  125.69  131.04  133.85  136.56  142.30  3780
```

```
## mu[22]  132.67    0.07   4.28  124.48  129.77  132.66  135.44  141.21  3591
## mu[23]  131.52    0.08   4.35  123.11  128.53  131.55  134.34  140.17  3345
## mu[24]  130.36    0.08   4.46  121.79  127.36  130.40  133.34  139.25  3049
## mu[25]  129.20    0.09   4.59  120.38  126.12  129.24  132.34  138.26  2750
## mu[26]  128.04    0.10   4.74  118.89  124.86  128.05  131.29  137.39  2447
## mu[27]  126.88    0.10   4.92  117.28  123.58  126.92  130.27  136.55  2199
## mu[28]  125.72    0.11   5.11  115.57  122.28  125.73  129.26  135.68  2033
## mu[29]  124.56    0.12   5.32  113.92  120.96  124.61  128.25  134.89  1887
## mu[30]  123.40    0.13   5.55  112.24  119.65  123.47  127.25  134.17  1767
## mu[31]  122.24    0.14   5.79  110.51  118.35  122.34  126.23  133.42  1669
## mu[32]  121.08    0.15   6.05  108.82  117.05  121.18  125.27  132.62  1589
## mu[33]  119.93    0.16   6.31  107.09  115.70  120.05  124.31  131.98  1522
## mu[34]  118.77    0.17   6.58  105.42  114.42  118.92  123.30  131.40  1466
## mu[35]  117.61    0.18   6.87  103.67  113.09  117.75  122.32  130.73  1420
## mu[36]  116.45    0.19   7.15  101.88  111.81  116.60  121.35  130.13  1380
## mu[37]  115.29    0.20   7.45  100.15  110.51  115.46  120.36  129.57  1345
## mu[38]  114.13    0.21   7.75   98.37  109.20  114.33  119.39  128.97  1315
## mu[39]  112.97    0.22   8.05   96.56  107.81  113.20  118.42  128.34  1289
## mu[40]  111.81    0.24   8.36   94.78  106.44  112.06  117.49  127.79  1267
## ypred   110.31    0.45  27.65   56.53   91.79  110.40  128.82  165.14  3799
## lp__   -146.69    0.04   1.27 -150.15 -147.27 -146.37 -145.77 -145.26   993
##        Rhat
## alpha     1
## beta      1
## sigma     1
## mu[1]     1
## mu[2]     1
## mu[3]     1
## mu[4]     1
## mu[5]     1
## mu[6]     1
## mu[7]     1
## mu[8]     1
## mu[9]     1
## mu[10]    1
## mu[11]    1
## mu[12]    1
## mu[13]    1
## mu[14]    1
## mu[15]    1
## mu[16]    1
## mu[17]    1
## mu[18]    1
## mu[19]    1
## mu[20]    1
## mu[21]    1
## mu[22]    1
## mu[23]    1
## mu[24]    1
## mu[25]    1
## mu[26]    1
## mu[27]    1
## mu[28]    1
## mu[29]    1
```

```
## mu[30]     1
## mu[31]     1
## mu[32]     1
## mu[33]     1
## mu[34]     1
## mu[35]     1
## mu[36]     1
## mu[37]     1
## mu[38]     1
## mu[39]     1
## mu[40]     1
## ypred      1
## lp__       1
##
## Samples were drawn using NUTS(diag_e) at Sun Nov  6 18:34:10 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).
```

Exercise 2: Hierarchical model, factory data with Stan

a)

```
data = list(y = factory, N = nrow(factory), M = ncol(factory))
data
```

```
## $y
##   V1  V2  V3  V4  V5  V6
## 1 83 117 101 105  79  57
## 2 92 109  93 119  97  92
## 3 92 114  92 116 103 104
## 4 46 104  86 102  79  77
## 5 67  87  67 116  92 100
##
## $N
## [1] 5
##
## $M
## [1] 6
```

Separate model: $y_{ij} \sim \mathcal{N}(\mu, \sigma)$

$\mu_j \sim \mathcal{N}(0, 10)$

$\sigma_j \sim Gamma(1, 1)$

Pooled model: $y_{ij} \sim \mathcal{N}(\mu, \sigma)$

$\mu_j \sim \mathcal{N}(0, 10)$

$\sigma_j \sim Gamma(1, 1)$

Hierarchical model: $y_{ij} \sim \mathcal{N}(\mu, \sigma)$

$\mu_0 \sim \mathcal{N}(0, 10)$

$\sigma_0 \sim Gamma(1, 1)$

$\mu_j \sim \mathcal{N}(\mu_0, \sigma_0)$

$\sigma_j \sim Gamma(1, 1)$

b)

Separate model: We choose the weakly informative prior Normal(0, 10) for μ and Gamma(1, 1) for σ.

```
data {
  int<lower=0> N;
  int<lower=0> M;
  vector[M] y[N];
}
parameters {
  vector[M] mu;
  vector<lower=0>[M] sigma;
}
model {
  for (i in 1:M) {
    mu[i] ~ normal(0, 10);
    sigma[i] ~ gamma(1, 1);
  }
  for (i in 1:M) {
    y[, i] ~ normal(mu[i], sigma[i]);
  }
}
generated quantities {
  real ypred = normal_rng(mu[6], sigma[6]);
}
```

```r
separate = stan(file = "~/notebooks/bda2022/separate.stan", data = data, refresh = 0)
```

```
## Trying to compile a simple C file
```

```
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## clang -flto=thin -I"/usr/share/R/include" -DNDEBUG   -I"/usr/local/lib/R/site-library/Rcpp/include/"
## In file included from <built-in>:1:
## In file included from /usr/local/lib/R/site-library/StanHeaders/include/stan/math/prim/mat/fun/Eigen
## In file included from /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Dense:1:
## In file included from /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Core:88:
## /usr/local/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:1: error: unknown ty
## namespace Eigen {
## ^
## /usr/local/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:16: error: expected
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /usr/local/lib/R/site-library/StanHeaders/include/stan/math/prim/mat/fun/Eigen
## In file included from /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Dense:1:
## /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Core:96:10: fatal error: 'complex' file not fou
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [/usr/lib/R/etc/Makeconf:168: foo.o] Error 1
```

```r
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]       50.42    0.17  9.07   30.92   44.42   50.91   56.96   66.39  2808
## mu[2]       49.24    0.23 12.22   25.95   40.91   48.78   56.88   74.67  2934
## mu[3]       58.25    0.21 11.82   34.26   50.16   58.88   66.97   79.03  3038
## mu[4]       47.96    0.21 12.07   25.39   39.71   47.75   55.88   72.93  3287
## mu[5]       60.45    0.23 12.64   34.55   51.65   60.83   70.06   82.16  3040
## mu[6]       51.18    0.18 10.52   29.81   44.25   51.39   58.60   70.33  3308
## sigma[1]    16.01    0.07  3.43   10.07   13.55   15.72   18.19   23.74  2748
## sigma[2]    24.55    0.08  4.54   15.35   21.60   24.51   27.51   33.64  3121
## sigma[3]    16.08    0.08  4.42    8.58   12.75   15.86   19.04   25.20  3184
## sigma[4]    26.28    0.08  4.57   17.55   23.18   26.20   29.29   35.48  3313
## sigma[5]    15.84    0.09  4.79    7.57   12.31   15.65   19.01   25.73  2854
## sigma[6]    18.68    0.07  3.90   11.86   15.97   18.45   21.21   26.80  3428
## ypred       50.70    0.35 21.83    3.38   37.01   52.57   65.95   89.74  3890
## lp__      -347.03    0.06  2.47 -352.66 -348.50 -346.69 -345.20 -343.22  1741
##           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 Sun Nov  6 18:34:34 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).
```

Pooled model: We choose the weakly informative prior Normal(0, 10) for μ and Gamma(1, 1) for σ.

```stan
data {
  int<lower=0> N;
  int<lower=0> J;
  vector[J] y[N];
}
parameters {
  real mu;
  real sigma;
}
model {
```

```
  mu ~ normal(0, 10);
  sigma ~ gamma(1, 1);
  for (i in 1:J) {
    y[, i] ~ normal(mu, sigma);
  }
}
generated quantities {
  real ypred = normal_rng(mu, sigma);
}
```

```r
pooled = stan(file = "~/notebooks/bda2022/pooled.stan", data = data, refresh = 0)
```

```
## Trying to compile a simple C file
```

```r
print(pooled)
```

Hierarchical model: We choose the Normal(0, 10) and Gamma(1, 1) as hyper-priors for μ_0 and σ_0, respectively, and the Normal(μ_0, σ_0) for prior μ_j.

```
data {
  int<lower=0> N;
  int<lower=0> M;
  vector[M] y[N];
}
parameters {
  vector[M] mu;
  real mu_0;
  real sigma_0;
  real sigma;
}
model {
  mu_0 ~ normal(0, 10);
  sigma_0 ~ gamma(1, 1);
  mu ~ normal(mu_0, sigma_0);
  sigma ~ gamma(1, 1);
  for (i in 1:M) {
    y[, i] ~ normal(mu[i], sigma);
  }
}
generated quantities {
  real ypred = normal_rng(mu[6], sigma);
  real mu_7 = normal_rng(mu_0, sigma_0);
}
```

```r
hierarchical = stan(file = "~/notebooks/bda2022/hierarchical.stan", data = data, refresh = 0)
```

```
## Trying to compile a simple C file

## Warning: There were 61 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 quant
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

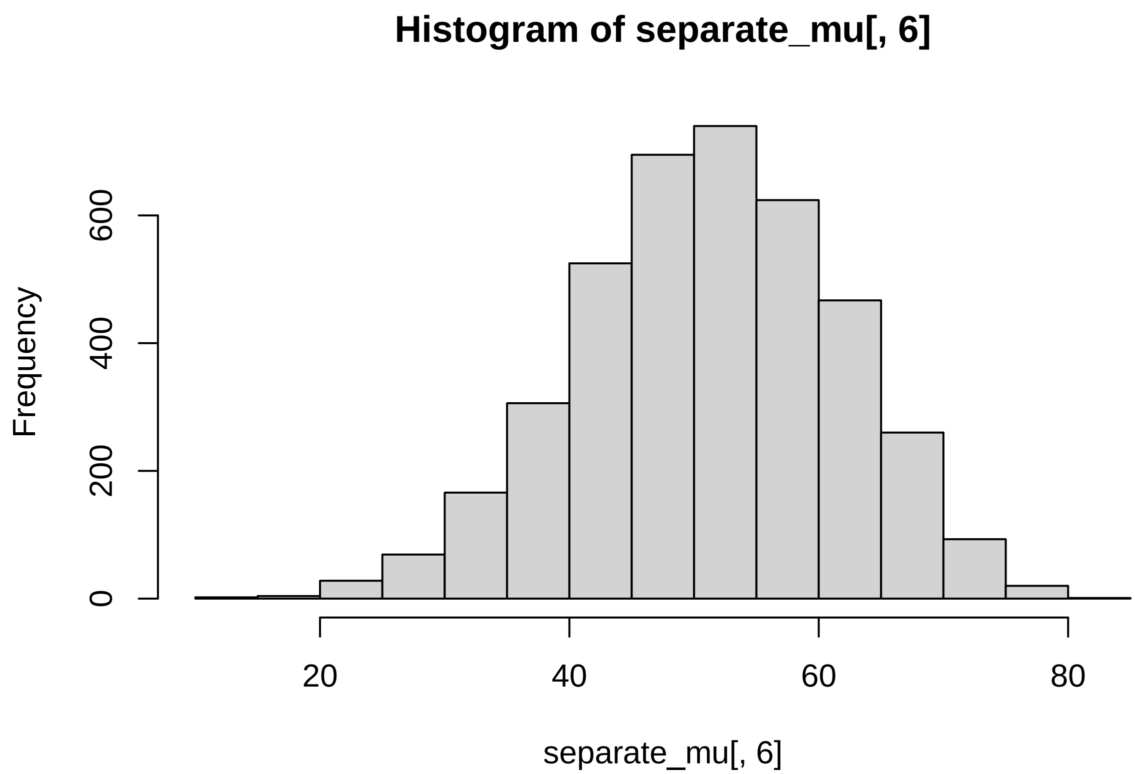
```
print(hierarchical)
```

c)

Separate model:

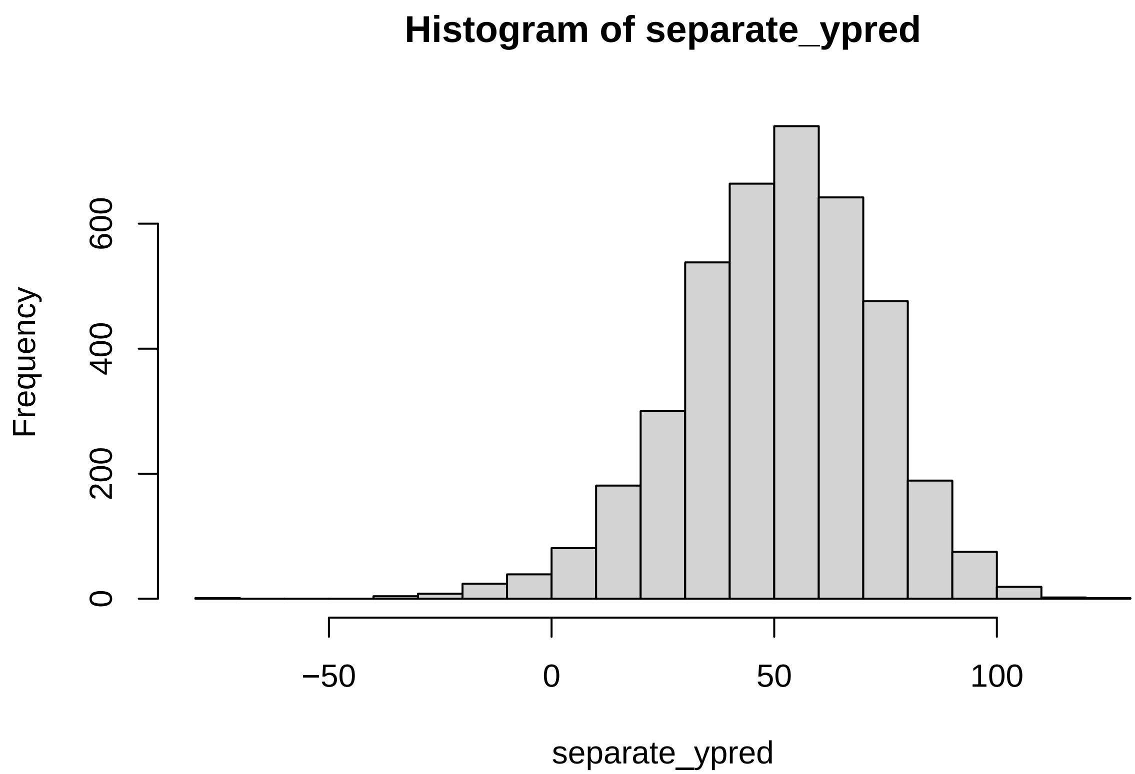
```
separate_extract = extract(separate, permuted= TRUE)
separate_ypred = separate_extract$ypred
separate_mu = separate_extract$mu
hist(separate_mu[,6], breaks=15)
```

i) The posterior distribution of the mean of the quality measurements of the sixth machine:




```
hist(separate_ypred, breaks=15)
```

ii) The predictive distribution for another quality measurement of the sixth machine:

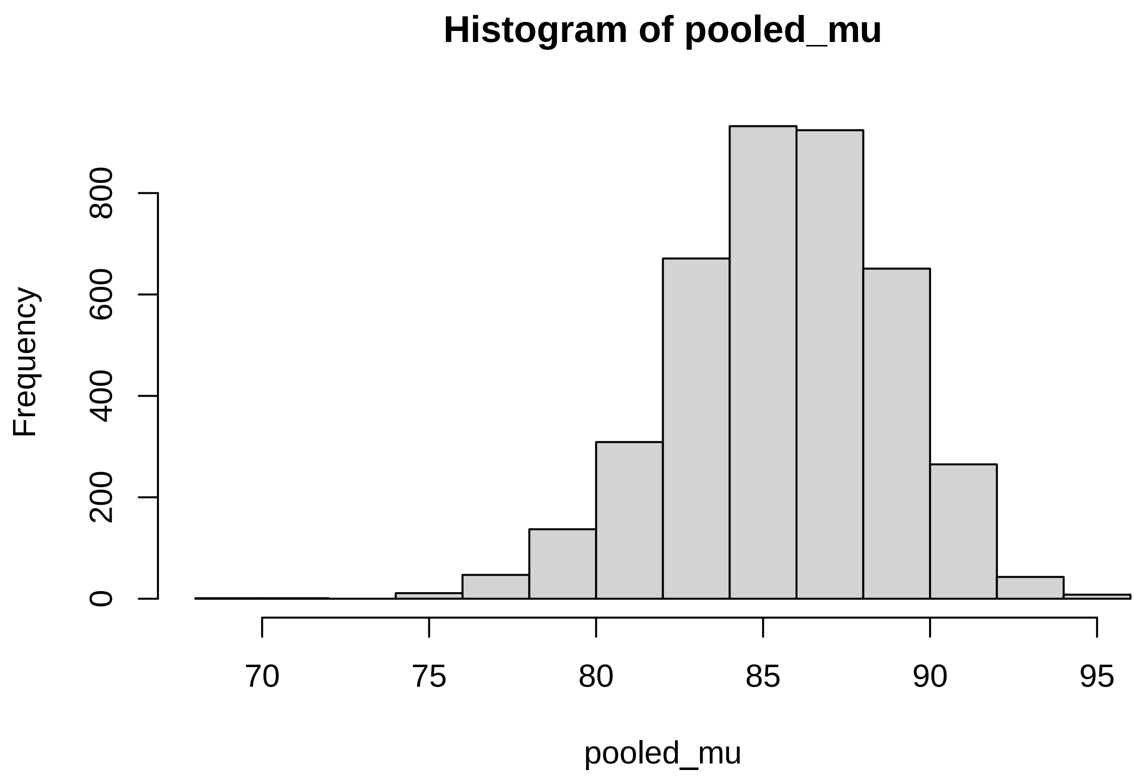


iii) The posterior distribution of the mean of the quality measurements of the seventh machine: In the separate model, each machine has its own model so we cannot plot the posterior distribution of the mean of the quality measurements of the seventh machine.

Pooled model:

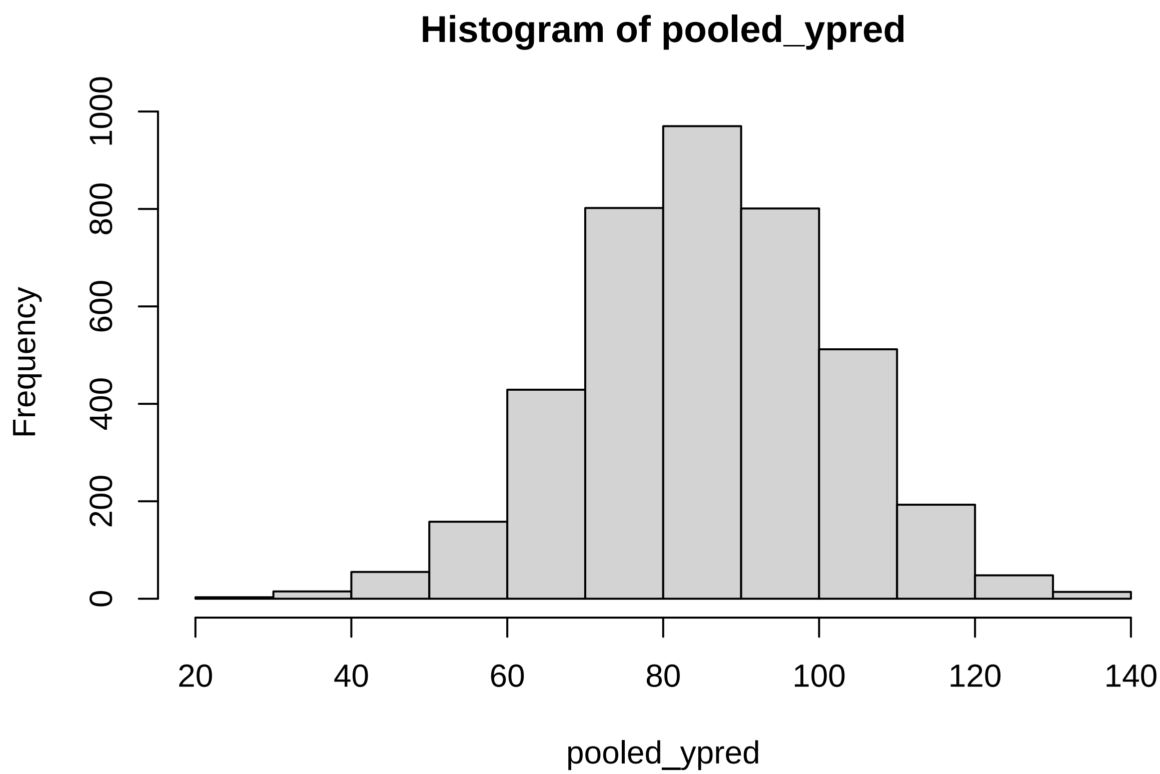
```
pooled_extract = extract(pooled,permuted= TRUE)
pooled_ypred = pooled_extract$ypred
pooled_mu = pooled_extract$mu
hist(pooled_mu,breaks=15)
```

i) The posterior distribution of the mean of the quality measurements of the sixth machine:




```
hist(pooled_ypred, breaks=15)
```

ii) The predictive distribution for another quality measurement of the sixth machine:

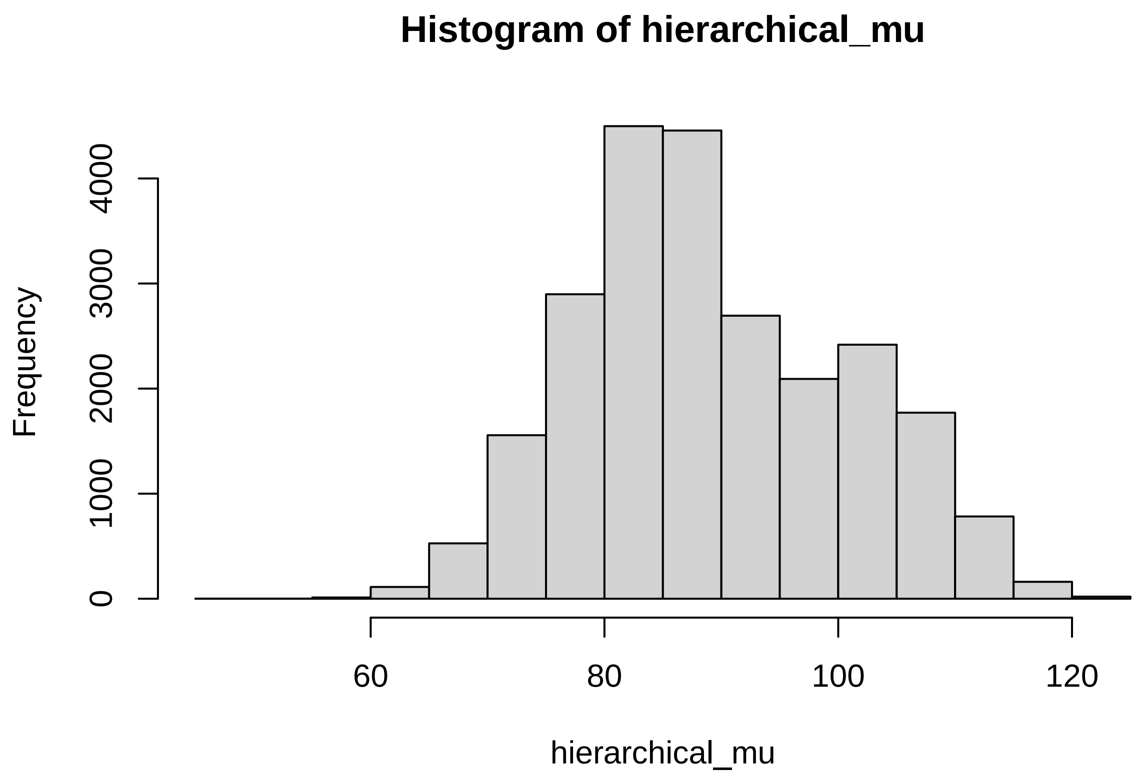


iii) The posterior distribution of the mean of the quality measurements of the seventh machine:
In the pooled model, since all measurements are combined, there is no distinction between machines. Therefore, the posterior distribution of the mean of the quality measurements of the seventh machine is the same as that of any other machine (as plotted in part (i)).

Hierarchical model:

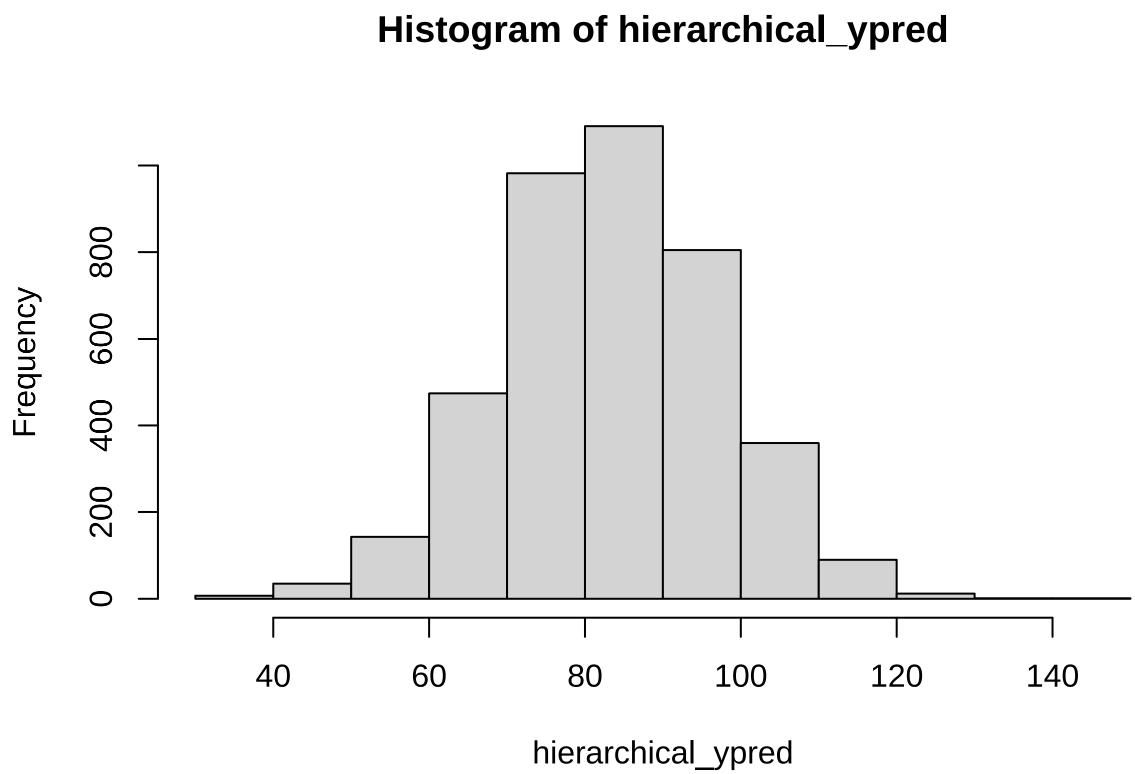
```
hierarchical_extract = extract(hierarchical,permuted= TRUE)
hierarchical_ypred = hierarchical_extract$ypred
hierarchical_mu = hierarchical_extract$mu
hist(hierarchical_mu,breaks=15)
```

i) The posterior distribution of the mean of the quality measurements of the sixth machine:



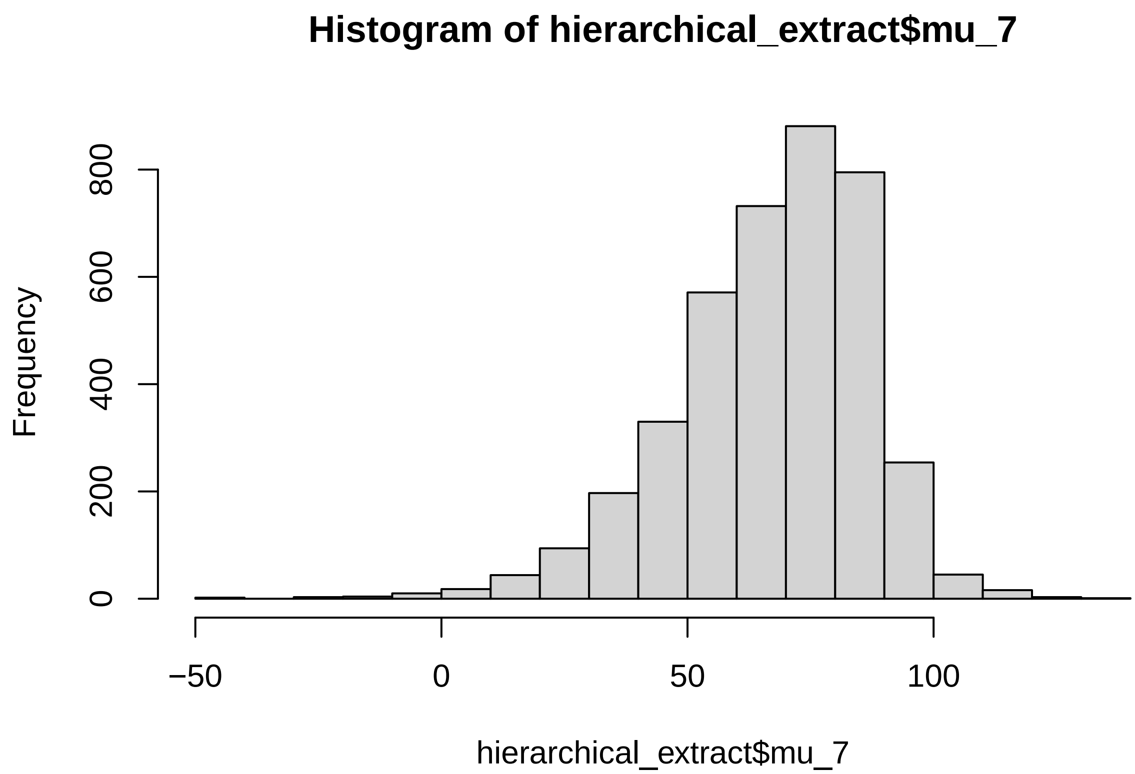

```
hist(hierarchical_ypred, breaks=15)
```

ii) The predictive distribution for another quality measurement of the sixth machine:




```
hist(hierarchical_extract$mu_7, breaks=15)
```

iii) The posterior distribution of the mean of the quality measurements of the seventh machine:



d)

Separate model: The posterior expectation for μ_1 with a 90% credible interval:

```
quantile(separate_extract$mu[,1], probs=c(0.05, 0.95))
```

```
##       5%      95%
## 34.60757 64.26979
```

Therefore, the 90% interval posterior expectation is approximately (34.56, 64.58).

Pooled model: The posterior expectation for μ_1 with a 90% credible interval:

```
quantile(pooled_mu, probs=c(0.05, 0.95))
```

```
##       5%      95%
## 80.02949 90.44701
```

Therefore, the 90% interval posterior expectation is approximately (79.98, 90.50).

Separate model: The posterior expectation for μ_1 with a 90% credible interval:

```
quantile(hierarchical_extract$mu[,1], probs=c(0.05, 0.95))
```

```
##       5%      95%
## 66.48400 84.63321
```

Therefore, the 90% interval posterior expectation is approximately (66.95, 85.53).