BDA Project

Arthur Aspelin, Jannica Savander, Christian Segercrantz

11/2021

Contents

Introduction

Description of the data

Description of the models

Priors

Stan code

Convergence diagnostics

Posterior predictive checks

Model comparison with LOO-CV

Predictive performance assessment (if applicable)

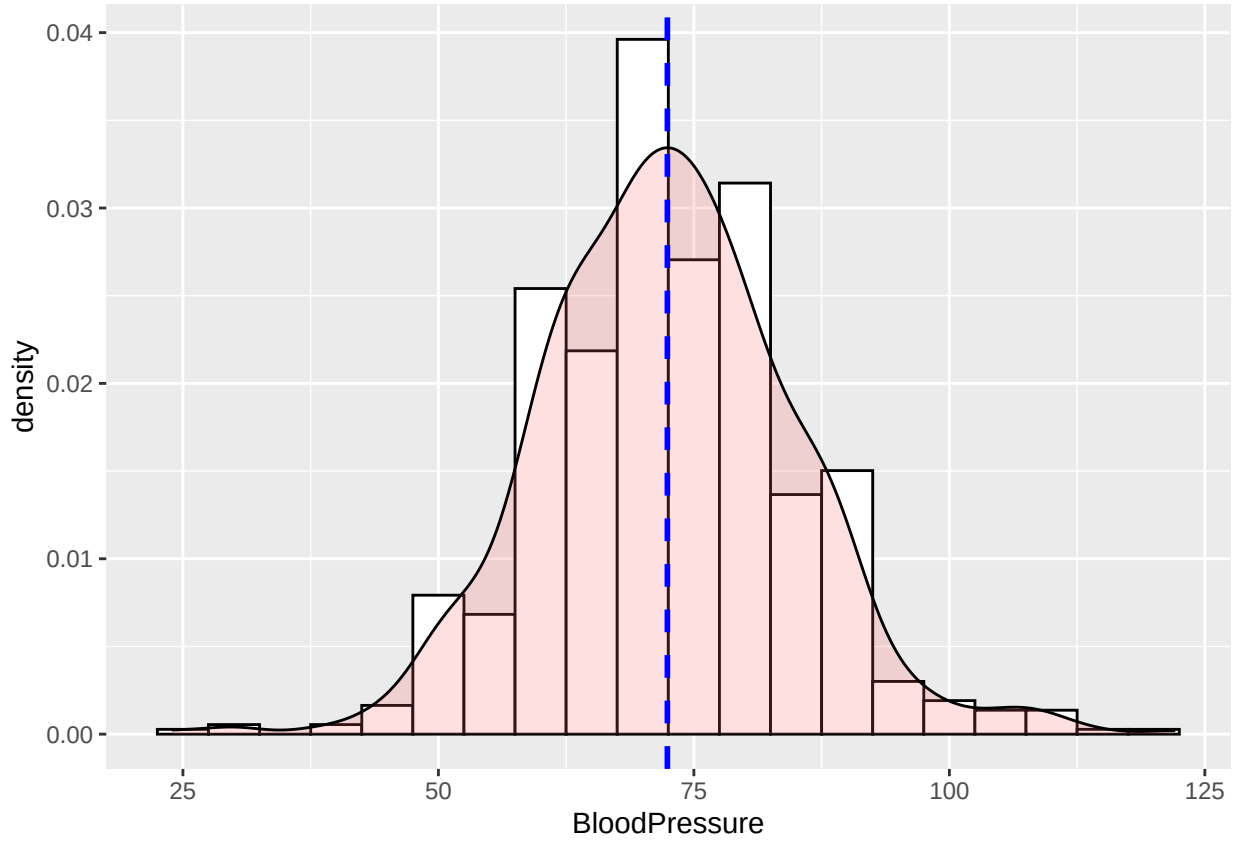
Sensitivity analysis

Discussion

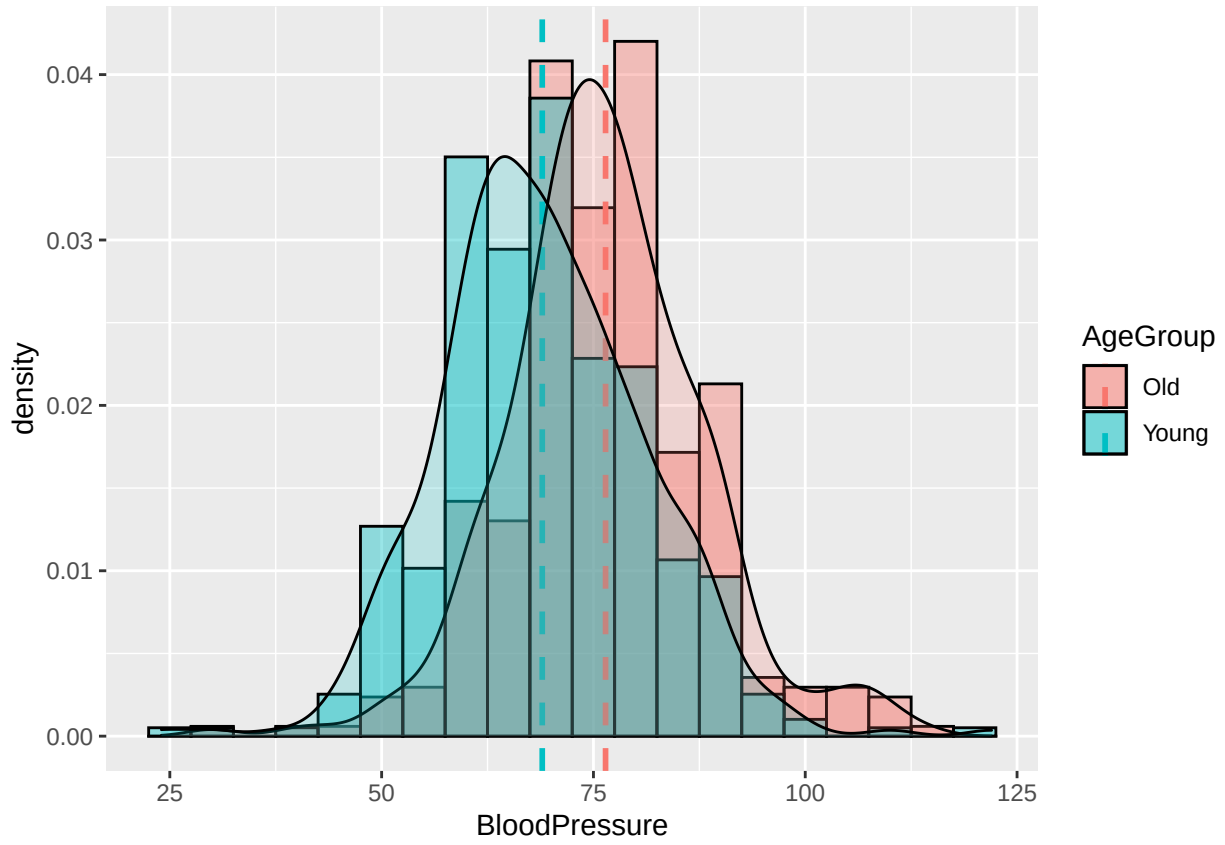
Conclusion

Self-reflection

```
data <- data %>%
  filter(BloodPressure > 0) %>%
  mutate(AgeGroup = case_when(
  Age <= 30      ~ "Young",
  Age > 30       ~ "Old")
  ) %>% head(-1)
#data
```




```
means <- data %>%
  group_by(AgeGroup) %>%
  summarise(mean = mean(BloodPressure), n = n())

ggplot(data, aes(x=BloodPressure, fill=AgeGroup)) +
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", alpha = 0.4) +
  geom_vline(data = means, aes(xintercept=mean, color = AgeGroup), linetype="dashed", size=1) +
  geom_density(alpha=.2)
```




```
ggplot(data, aes(x=Age, y=BloodPressure, color=AgeGroup)) + geom_point()
```


```
data {
  int<lower=0> N;                  //Amount of data points
  vector[N] y;                     //
  real mean_mu_prior;              //
  real<lower=0> mean_sigma_prior;  //
  real<lower=0> var_prior;         //
}

parameters {
  real mu;
  real<lower=0> sigma;
}

model {
  //prior
  mu ~ normal(mean_mu_prior, mean_sigma_prior);
  sigma ~ inv_chi_square(var_prior);
 //likelihoods
  y ~ normal(mu, sigma);
}

generated quantities {
  real ypred;
  vector[N] log_lik;
  ypred = normal_rng(mu, sigma);
  for (n in 1:(N)){
```

```
    log_lik[n] = normal_lpdf(y[n] | mu, sigma);
  }
}
```

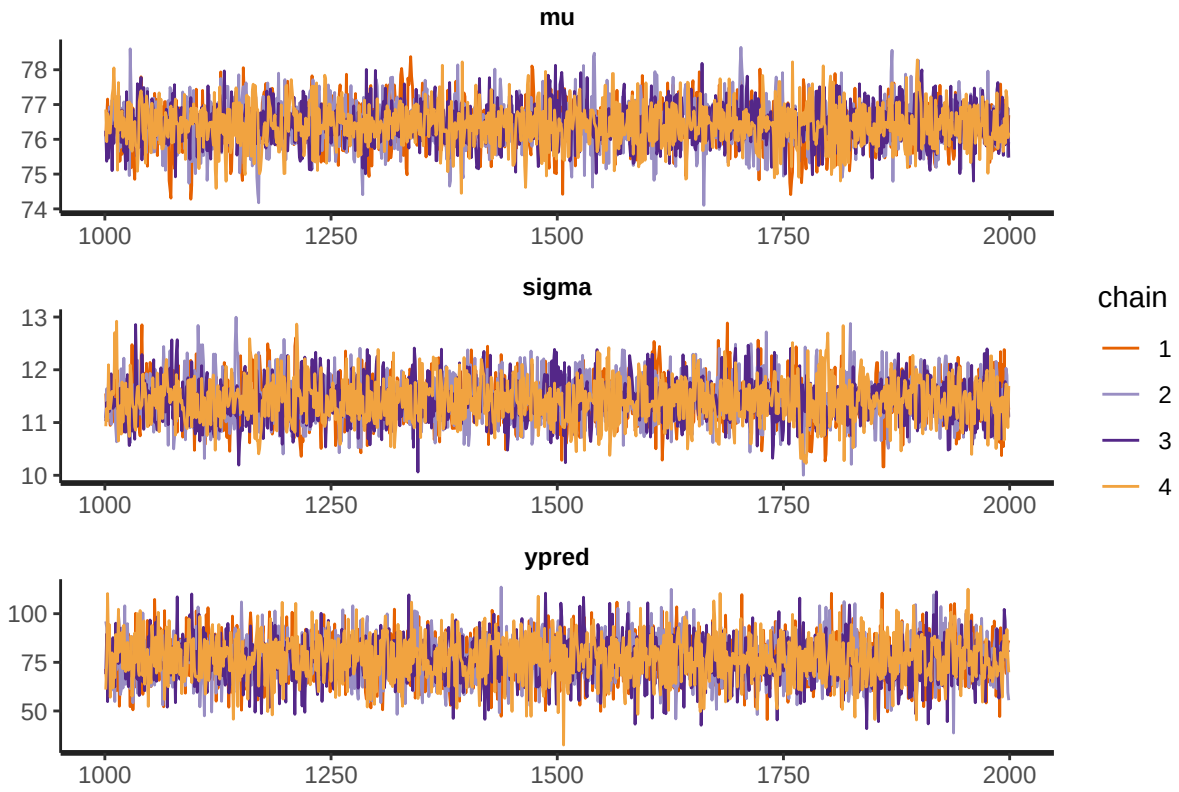
```
data_old <- data %>%
  filter(AgeGroup == "Old")

mean_mu_prior_old = mean(data_old$BloodPressure)
mean_sigma_prior_old = 10
var_prior_old = 20
data_nonhiera_old <- list(
  y = data_old$BloodPressure,
  N = length(data_old$BloodPressure),
  mean_mu_prior = mean_mu_prior_old,
  mean_sigma_prior = mean_sigma_prior_old,
  var_prior = var_prior_old
)
```

```
fit_nonhiera_old = sampling(nonhieramodel,
  data = data_nonhiera_old,            # named list of data
  chains = 4,             # number of Markov chains
  warmup = 1000,          # number of warmup iterations per chain
  iter = 2000,            # total number of iterations per chain
  cores = 4,              # number of cores (could use one per chain)
  refresh = 0             # no progress shown
  )
```

```
head(monitor(fit_nonhiera_old, print = FALSE),3)
```

```
##       mean se_mean     sd 2.5%  25%  50%  75% 97.5% n_eff Rhat valid   Q5  Q50
## mu    76.4 0.01073  0.632 75.1 76.0 76.4 76.8  77.7  3465    1     1 75.4 76.4
## sigma 11.4 0.00731  0.428 10.6 11.1 11.4 11.7  12.3  3423    1     1 10.7 11.4
## ypred 76.5 0.18095 11.478 53.9 68.7 76.6 84.4  99.5  3989    1     1 57.7 76.6
##        Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu    77.5    0.0385  0.01708  0.01493   0.0116     0.0388 0.00758     3496
## sigma 12.2    0.0286  0.00708  0.00832   0.0117     0.0184 0.00517     3430
## ypred 94.9    0.5637  0.25014  0.33790   0.2180     0.5502 0.12796     4032
##       Tail_ESS
## mu        2562
## sigma     2455
## ypred     4122
```

```
traceplot(fit_nonhiera_old, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```




```
data_young <- data %>%
  filter(AgeGroup == "Young")

mean_mu_prior_old = mean(data_young$BloodPressure)
mean_sigma_prior_old = 10
var_prior_old = 20
data_nonhiera_young <- list(
  y = data_young$BloodPressure,
  N = length(data_young$BloodPressure),
  mean_mu_prior = mean_mu_prior_old,
  mean_sigma_prior = mean_sigma_prior_old,
  var_prior = var_prior_old
)
```

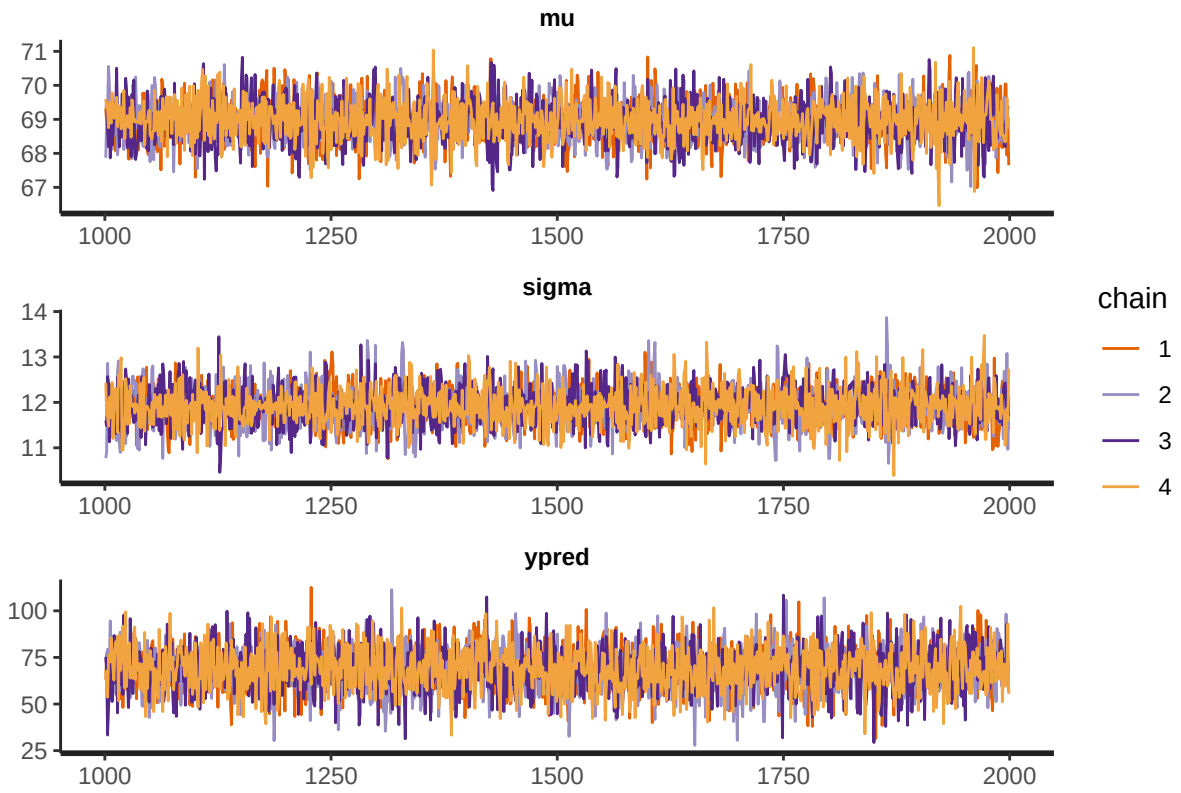
```
fit_nonhiera_young = sampling(nonhieramodel,
  data = data_nonhiera_young,            # named list of data
  chains = 4,             # number of Markov chains
  warmup = 1000,          # number of warmup iterations per chain
  iter = 2000,            # total number of iterations per chain
  cores = 4,              # number of cores (could use one per chain)
  refresh = 0             # no progress shown
  )
```

```
head(monitor(fit_nonhiera_young, print = FALSE),3)
```

```
##       mean se_mean     sd 2.5%  25%  50%  75% 97.5% n_eff Rhat valid   Q5  Q50
## mu    69.0 0.00978  0.610 67.7 68.5 69.0 69.4  70.2  3886    1     1 68.0 69.0
```

```
## sigma 11.9 0.00637  0.418 11.1 11.6 11.9 12.2  12.8  4291    1     1 11.3 11.9
## ypred 68.9 0.18902 11.685 45.7 61.2 69.3 76.6  91.6  3807    1     1 49.2 69.3
##         Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu    70.0    0.0313   0.0153  0.01169   0.0131     0.0297 0.00692     3909
## sigma 12.6    0.0172   0.0114  0.00541   0.0110     0.0252 0.00452     4338
## ypred 87.8    0.5562   0.2684  0.17039   0.2668     0.4778 0.13367     3822
##       Tail_ESS
## mu        2945
## sigma     2870
## ypred     3849
```

```
traceplot(fit_nonhiera_young, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```




```
data {
  int<lower=0> N;                  //Amount of data points
  vector[N] y;                     //
  real mean_mu_prior;              //
  real<lower=0> mean_sigma_prior;  //
  real<lower=0> var_prior;         //
}

parameters {
  real mu;
  real<lower=0> sigma;
  real mu_hypo;
  real<lower=0> tau;
}
```

```
model {
  //hyperpriors
  mu_hypo ~ normal(mean_mu_prior, mean_sigma_prior);
  tau ~ inv_chi_square(var_prior);
  //prior
  mu ~ normal(mu_hypo, tau);
  sigma ~ inv_chi_square(var_prior);
 //likelihoods
  y ~ normal(mu, sigma);
}

generated quantities {
  real ypred;
  vector[N] log_lik;
  ypred = normal_rng(mu, sigma);
  for (n in 1:(N)){
    log_lik[n] = normal_lpdf(y[n] | mu, sigma);
  }
}
```

```
mean_mu_prior = mean(data$BloodPressure)
mean_sigma_prior = 10
var_prior = 20
data_hiera_old <- list(
  y = data_old$BloodPressure,
  N = length(data_old$BloodPressure),
  mean_mu_prior = mean_mu_prior,
  mean_sigma_prior = mean_sigma_prior_old,
  var_prior = var_prior
)
data_hiera_young <- list(
  y = data_young$BloodPressure,
  N = length(data_young$BloodPressure),
  mean_mu_prior = mean_mu_prior,
  mean_sigma_prior = mean_sigma_prior,
  var_prior = var_prior
)
```

```
fit_hiera_old = sampling(hieramodel,
  data = data_hiera_old,            # named list of data
  chains = 4,             # number of Markov chains
  warmup = 1000,          # number of warmup iterations per chain
  iter = 2000,            # total number of iterations per chain
  cores = 4,              # number of cores (could use one per chain)
  refresh = 0             # no progress shown
  )
```

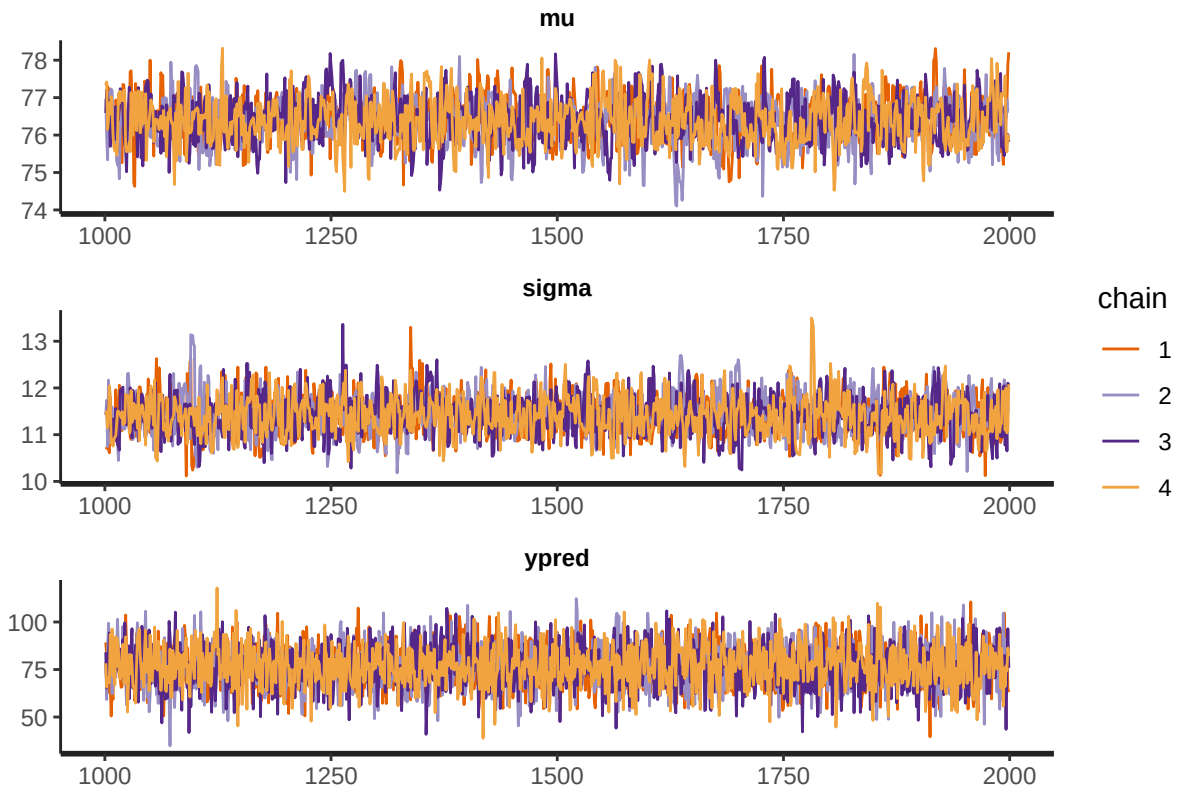
```
fit_hiera_young = sampling(hieramodel,
  data = data_hiera_young,            # named list of data
  chains = 4,             # number of Markov chains
  warmup = 1000,          # number of warmup iterations per chain
  iter = 2000,            # total number of iterations per chain
  cores = 4,              # number of cores (could use one per chain)
  refresh = 0             # no progress shown
```

```
  )
```

```
head(monitor(fit_hiera_old, print = FALSE),3)
```

```
##          mean se_mean    sd 2.5%  25%  50%  75% 97.5% n_eff Rhat valid   Q5  Q50
## mu       76.4 0.01526 0.617 75.2 76.0 76.4 76.8  77.6  1626    1     1 75.4 76.4
## sigma    11.4 0.00922 0.431 10.6 11.1 11.4 11.7  12.3  2172    1     1 10.7 11.4
## mu_hypo  76.4 0.01528 0.618 75.2 76.0 76.4 76.8  77.6  1628    1     1 75.4 76.4
##           Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu       77.4    0.0268   0.0190   0.0186   0.0209     0.0336 0.01079     1643
## sigma    12.1    0.0239   0.0104   0.0102   0.0125     0.0331 0.00654     2208
## mu_hypo  77.4    0.0402   0.0203   0.0204   0.0190     0.0490 0.01081     1645
##          Tail_ESS
## mu           1691
## sigma        1866
## mu_hypo      1675
```

```
traceplot(fit_hiera_old, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```



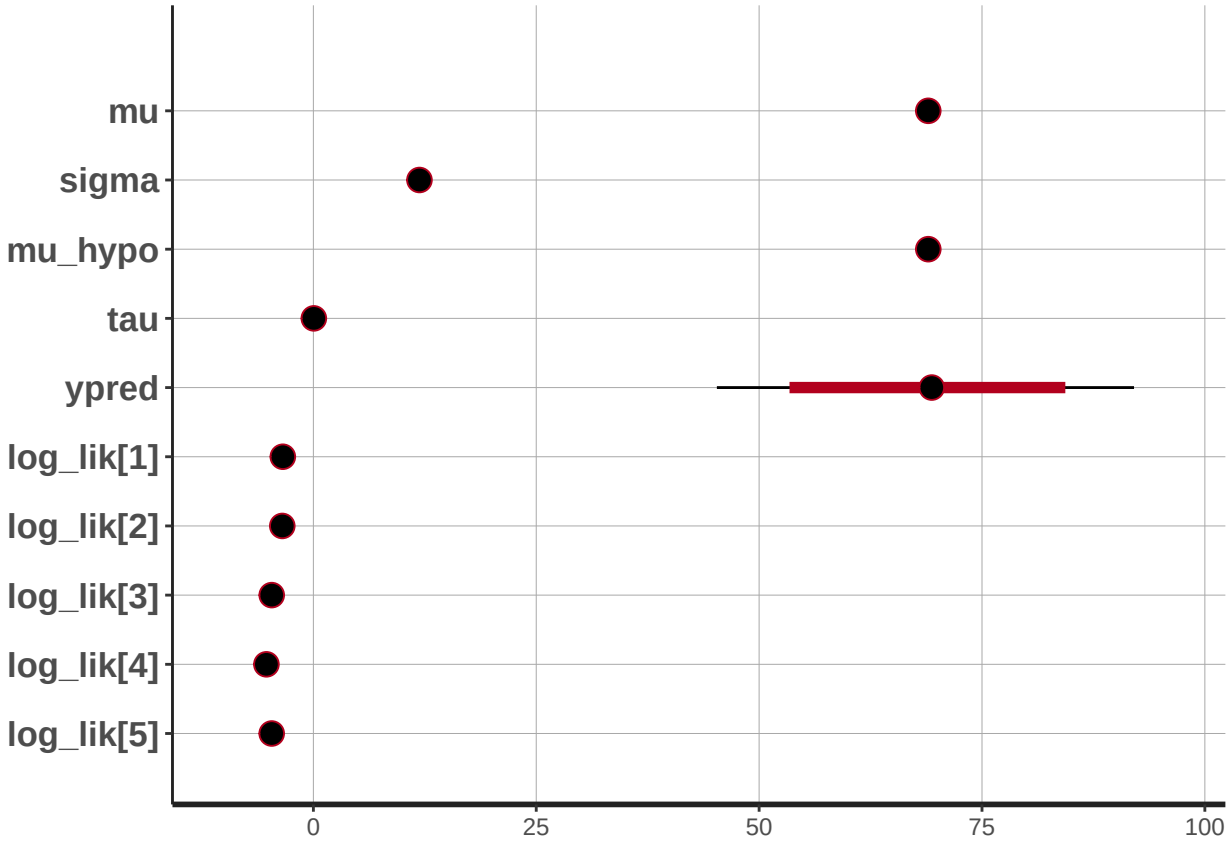

```
head(monitor(fit_hiera_young, print = FALSE),3)
```

```
##          mean se_mean    sd 2.5%  25%  50%  75% 97.5% n_eff Rhat valid   Q5  Q50
## mu       69.0 0.01717 0.612 67.8 68.6 69.0 69.4  70.2  1201    1     1 68.0 69.0
## sigma    11.9 0.00943 0.424 11.1 11.6 11.9 12.2  12.8  1965    1     1 11.2 11.9
## mu_hypo  69.0 0.01741 0.614 67.8 68.6 69.0 69.4  70.2  1184    1     1 68.0 69.0
##           Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu       70.0     0.035   0.0199   0.0163   0.0227     0.0349 0.01215     1266
```

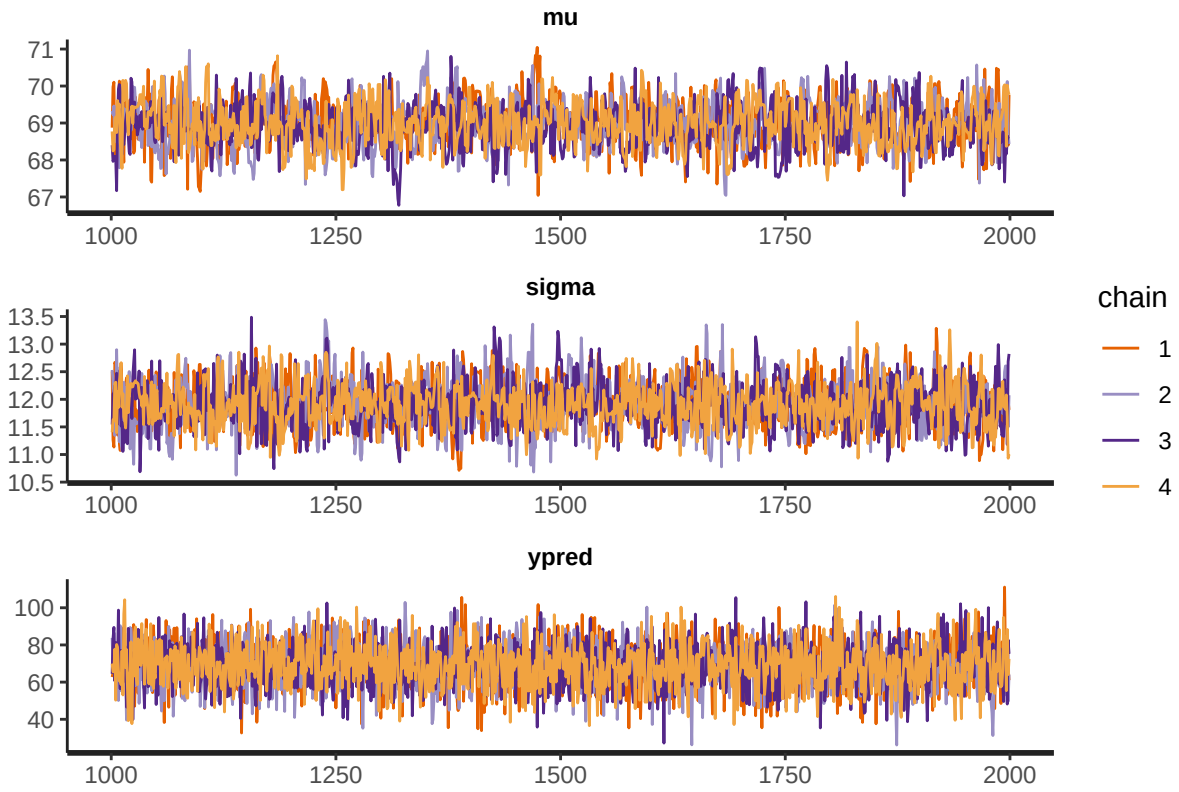
```
## sigma   12.6     0.017   0.0147   0.0114   0.0131     0.0327 0.00668     2033
## mu_hypo 70.0     0.034   0.0219   0.0191   0.0241     0.0429 0.01232     1239
##         Tail_ESS
## mu          1510
## sigma       1899
## mu_hypo     1624
```

```
plot(fit_hiera_young)
```

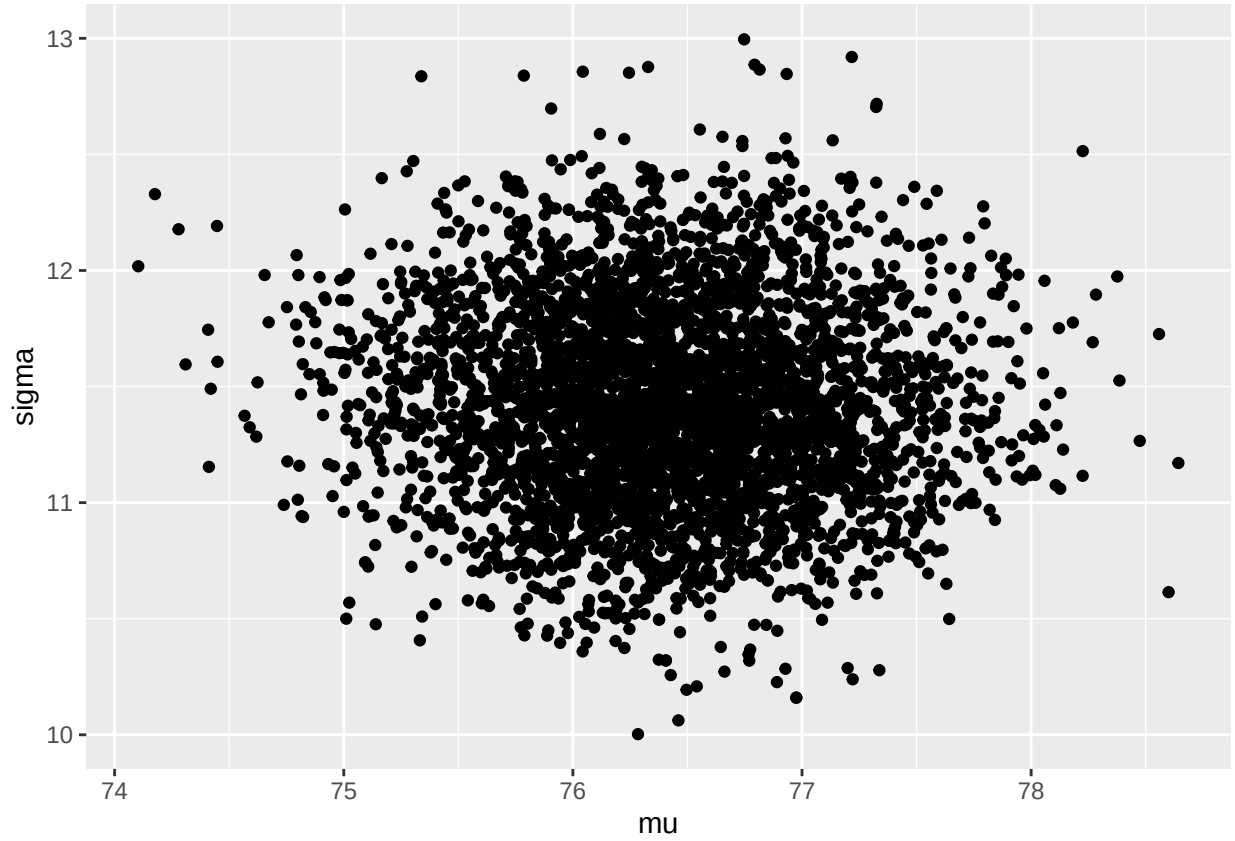
```
## 'pars' not specified. Showing first 10 parameters by default.
## ci_level: 0.8 (80% intervals)
## outer_level: 0.95 (95% intervals)
```




```
traceplot(fit_hiera_young, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```




```
extract_nonhiera_old <- data.frame(extract(fit_nonhiera_old))
ggplot(data = extract_nonhiera_old ,aes(x=mu, y=sigma))+geom_point()
```




```
loo_nonhiera_old <- loo(fit_nonhiera_old, pars="log_lik")
loo_nonhiera_old
```

```
##
## Computed from 4000 by 338 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1309.1 17.4
## p_loo         2.7  0.5
## looic      2618.2 34.8
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_nonhiera_young <- loo(fit_nonhiera_young, pars="log_lik")
loo_nonhiera_young
```

```
##
## Computed from 4000 by 394 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1541.3 18.1
## p_loo         2.6  0.6
## looic      3082.6 36.2
## ------
```

```
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_hiera_old <- loo(fit_nonhiera_old, pars="log_lik")
loo_hiera_old
```

```
##
## Computed from 4000 by 338 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1309.1 17.4
## p_loo         2.7  0.5
## looic      2618.2 34.8
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_hiera_young <- loo(fit_nonhiera_young, pars="log_lik")
loo_hiera_young
```

```
##
## Computed from 4000 by 394 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1541.3 18.1
## p_loo         2.6  0.6
## looic      3082.6 36.2
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
print("The model for the old:")
```

```
## [1] "The model for the old:"
```

```
loo_compare(loo_nonhiera_old, loo_hiera_old)
```

```
##        elpd_diff se_diff
## model1 0.0       0.0
## model2 0.0       0.0
```

```
print("The model for the young:")
```

```
## [1] "The model for the young:"
```

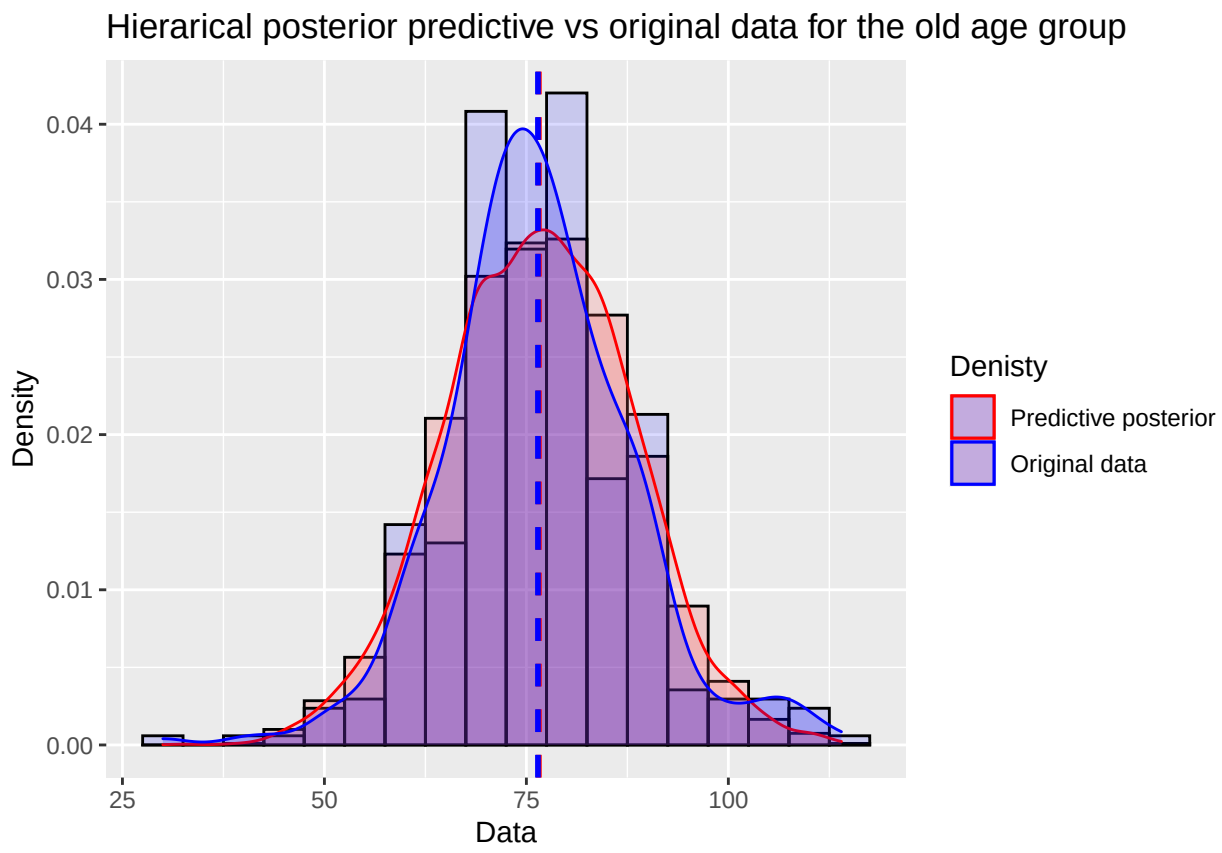
```
loo_compare(loo_nonhiera_young, loo_hiera_young)
```

```
##        elpd_diff se_diff
## model1 0.0       0.0
## model2 0.0       0.0
```

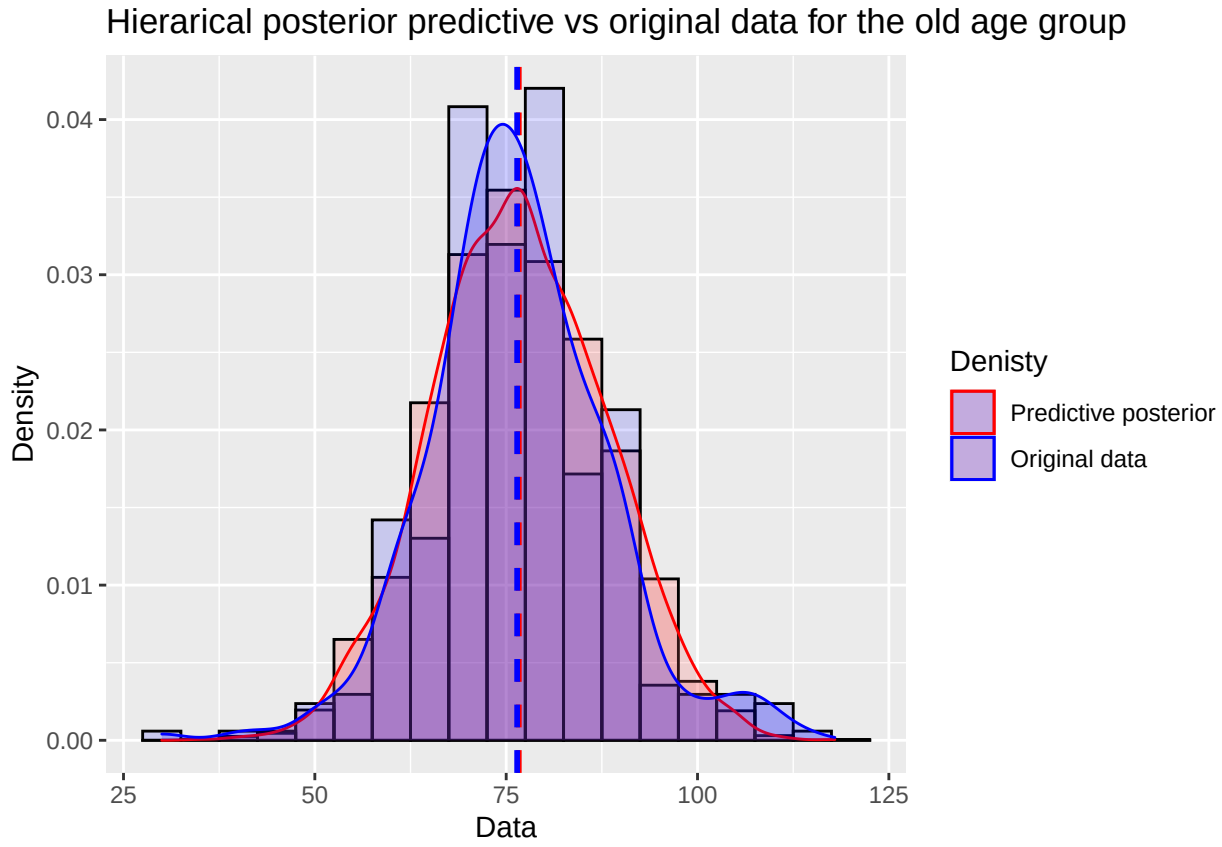
```
extract_hiera_old <- data.frame(extract(fit_hiera_old))
extract_nonhiera_old <- data.frame(extract(fit_nonhiera_old))
```

```
extract_hiera_young <- data.frame(extract(fit_hiera_young))
extract_nonhiera_young <- data.frame(extract(fit_nonhiera_young))
```

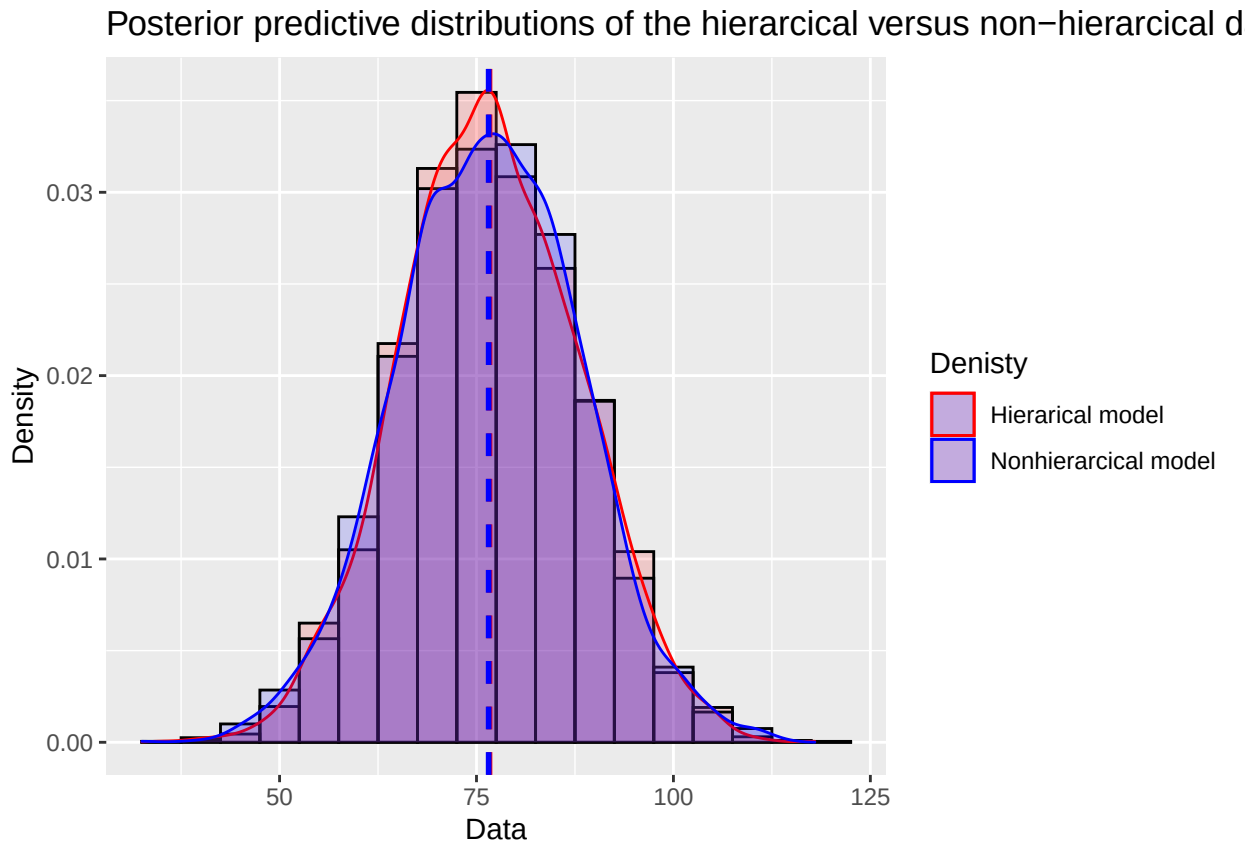
```
ggplot(extract_nonhiera_old, aes(x=ypred)) + ggtitle("Hierarical posterior predictive vs original data f
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", colour="black"
  geom_histogram(data= data_old, aes(x=BloodPressure,y=..density..), binwidth = 5, colour="black",positi
  geom_density(aes(colour="Sim"),alpha=.2, fill="#FF6666") +
  geom_density(data=data_old, aes(x=BloodPressure, colour="Orig"),alpha=.2, fill="#0000FF") +
  geom_vline(aes(xintercept=mean(ypred)), colour="red", linetype="dashed", size=1) +
  geom_vline(data=data_old, aes(xintercept=mean(BloodPressure), color="Orig"), color="blue", linetype="d
  labs(x="Data", y ="Density", colour = "legend") +
  scale_colour_manual(name = 'Denisty', values=c('Sim'='red','Orig'='blue'), labels = c('Predictive post
```



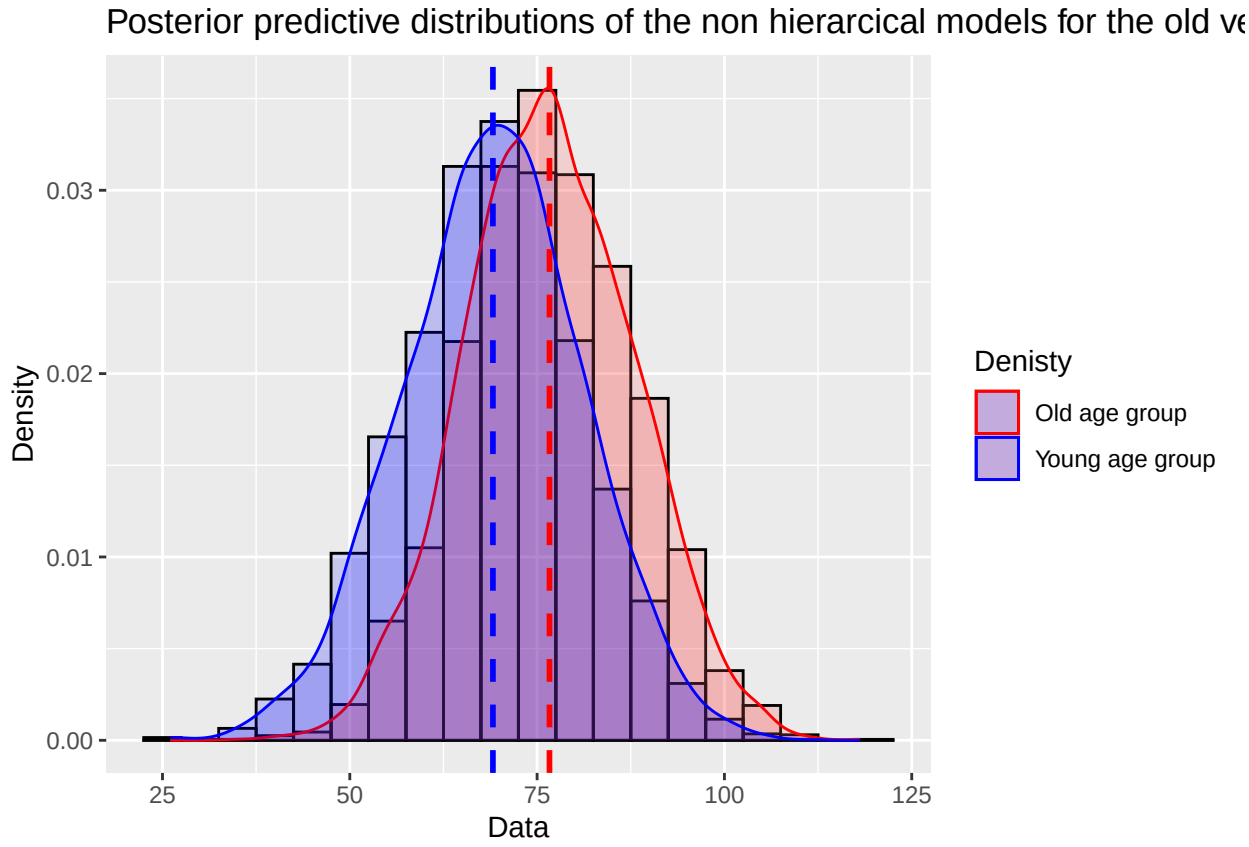

```
ggplot(extract_hiera_old, aes(x=ypred)) + ggtitle("Hierarical posterior predictive vs original data for
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", colour="black"
  geom_histogram(data= data_old, aes(x=BloodPressure,y=..density..), binwidth = 5, colour="black",positi
  geom_density(aes(colour="Sim"),alpha=.2, fill="#FF6666") +
  geom_density(data=data_old, aes(x=BloodPressure, colour="Orig"),alpha=.2, fill="#0000FF") +
  geom_vline(aes(xintercept=mean(ypred)), colour="red", linetype="dashed", size=1) +
  geom_vline(data=data_old, aes(xintercept=mean(BloodPressure), color="Orig"), color="blue", linetype="d
  labs(x="Data", y ="Density", colour = "legend") +
  scale_colour_manual(name = 'Denisty', values=c('Sim'='red','Orig'='blue'), labels = c('Predictive post
```




```
ggplot(extract_hiera_old, aes(x=ypred)) + ggtitle("Posterior predictive distributions of the hierarcical
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", colour="black"
  geom_histogram(data= extract_nonhiera_old, aes(x=ypred,y=..density..), binwidth = 5, colour="black",p
  geom_density(aes(colour="Sim"),alpha=.2, fill="#FF6666") +
  geom_density(data=extract_nonhiera_old, aes(x=ypred, colour="Orig"),alpha=.2, fill="#0000FF") +
  geom_vline(aes(xintercept=mean(ypred)), colour="red", linetype="dashed", size=1) +
  geom_vline(data=extract_nonhiera_old, aes(xintercept=mean(ypred), color="Orig"), color="blue", linety
  labs(x="Data", y ="Density", colour = "legend") +
  scale_colour_manual(name = 'Denisty', values=c('Sim'='red','Orig'='blue'), labels = c('Hierarical mod
```




```
ggplot(extract_hiera_old, aes(x=ypred)) + ggtitle("Posterior predictive distributions of the non hierar
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", colour="black"
  geom_histogram(data= extract_hiera_young, aes(x=ypred,y=..density..), binwidth = 5, colour="black",pos
  geom_density(aes(colour="Sim"),alpha=.2, fill="#FF6666") +
  geom_density(data=extract_hiera_young, aes(x=ypred, colour="Orig"),alpha=.2, fill="#0000FF") +
  geom_vline(aes(xintercept=mean(ypred)), colour="red", linetype="dashed", size=1) +
  geom_vline(data=extract_hiera_young, aes(xintercept=mean(ypred), color="Orig"), color="blue", linetyp
  labs(x="Data", y ="Density", colour = "legend") +
  scale_colour_manual(name = 'Denisty', values=c('Sim'='red','Orig'='blue'), labels = c('Old age group'
```




```
mean_mu_prior_sensitivity = c(0, 50, 100, 1000)
mean_sigma_prior_old_sensitivity = c(1, 10, 100, 1000)
var_prior_old_sensitivity = c(1, 10, 100, 1000)
fit_sensitivity = c()
for (i in 1:length(mean_mu_prior_sensitivity)){
  for (j in 1:length(mean_sigma_prior_old_sensitivity)){
  data_sensitivity <- list(
    y = data_old$BloodPressure,
    N = length(data_old$BloodPressure),
    mean_mu_prior = mean_mu_prior_sensitivity[i],
    mean_sigma_prior = mean_sigma_prior_old_sensitivity[j],
    var_prior = var_prior_old_sensitivity[j]
  )

  fit_sensitivity = c(fit_sensitivity,sampling(nonhieramodel,
    data = data_nonhiera_old,            # named list of data
    chains = 4,             # number of Markov chains
    warmup = 1000,          # number of warmup iterations per chain
    iter = 2000,            # total number of iterations per chain
    cores = 4,              # number of cores (could use one per chain)
    refresh = 0             # no progress shown
    ))
  }
}
```

```
ggplot() +
geom_line(data=data.frame(extract(fit_sensitivity[[1]],inc_warmup=TRUE)), aes(x=mu, sigma)) +
geom_line(data=data.frame(extract(fit_sensitivity[[2]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[3]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[4]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[5]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[6]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[7]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[8]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[9]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[10]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[11]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[12]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[13]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[14]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[15]],inc_warmup=TRUE)), aes(x=mu, sigma))+
geom_line(data=data.frame(extract(fit_sensitivity[[16]],inc_warmup=TRUE)), aes(x=mu, sigma))
```