BDA Project

Arthur Aspelin, Jannica Savander, Christian Segercrantz

11/2021

Contents

1. Introduction

The motivation for this project is to estimate the parameters for blood pressure data with the help of Bayesian methods. High blood pressure corresponds with different diseases, such as diabetes and heart diseases. This means that it is essential to predict distribution of blood pressure and its parameters in an accurate way.

Solving the problem, firstly, we want to estimate what type of distribution can describe blood pressure. Then with different Bayesian models estimate the parameters for the distribution. We will also investigate how the parameters differ when dividing the data into different age groups. Higher blood pressure could correspond with higher age.

Main modeling idea is to test with different Bayesian models how to get accurate estimates of the parameters that could describe blood pressure.

Description of the data

We used blood pressure data combined with age data from the Diabetes Dataset from Kaggle. According to the data description, the dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases, and one data point corresponds to a female patient of Pima Indian heritage. All patients are at least 21 years old.

The dataset has more columns than we used, for example number of pregnancies, BMI and diabetes classification. We only used the columns BloodPressure, describing the diastolic blood pressure, and Age, describing the age of the patient in years. The diastolic blood pressure is the pressure the heart applies on the walls of the arteries between the beats (Mayo Clinic Staff (2021)). The unit for the diastolic blood pressure is mmHg, and a normal value is usually below 80. Higher values might indicate hypertension, which increases with age in Western countries (Gurven et al. (2012)).

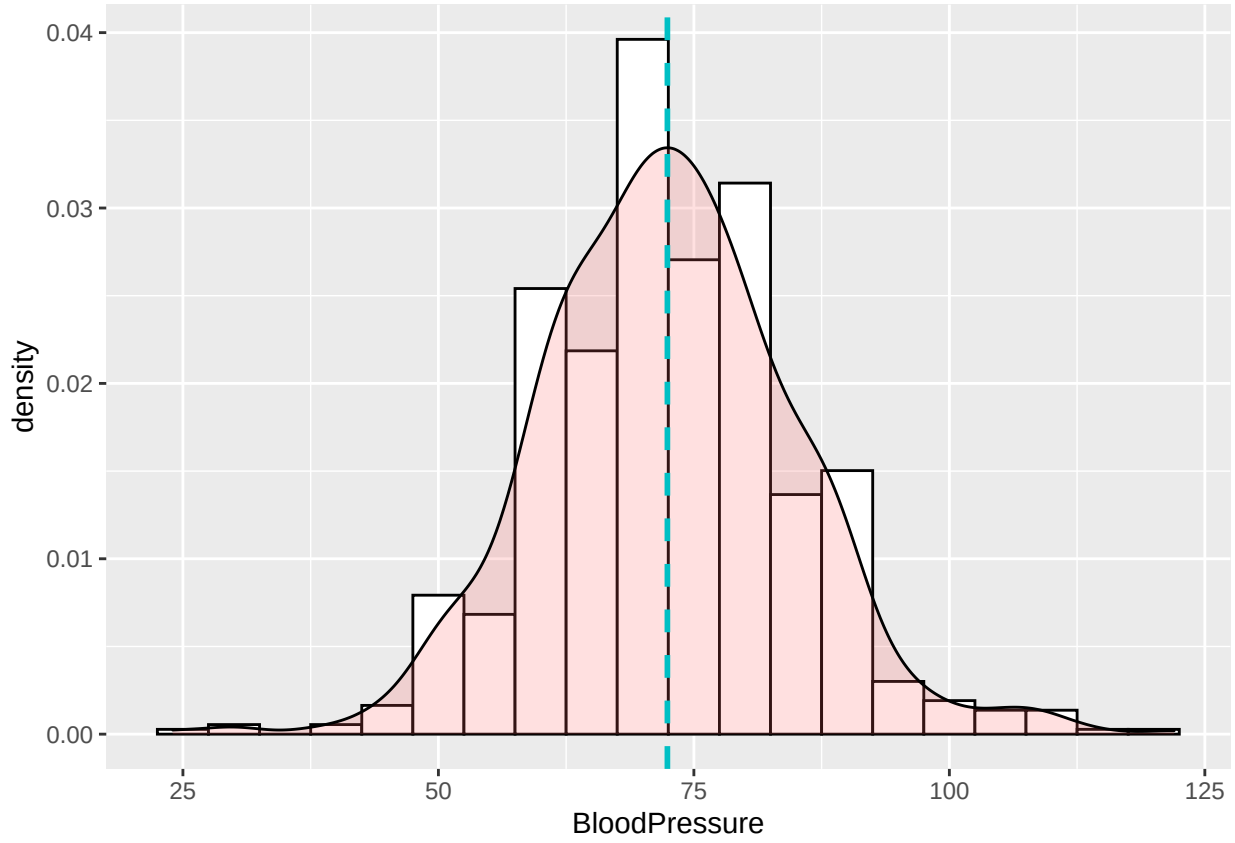
Description of the models

Priors

Convergence diagnostics

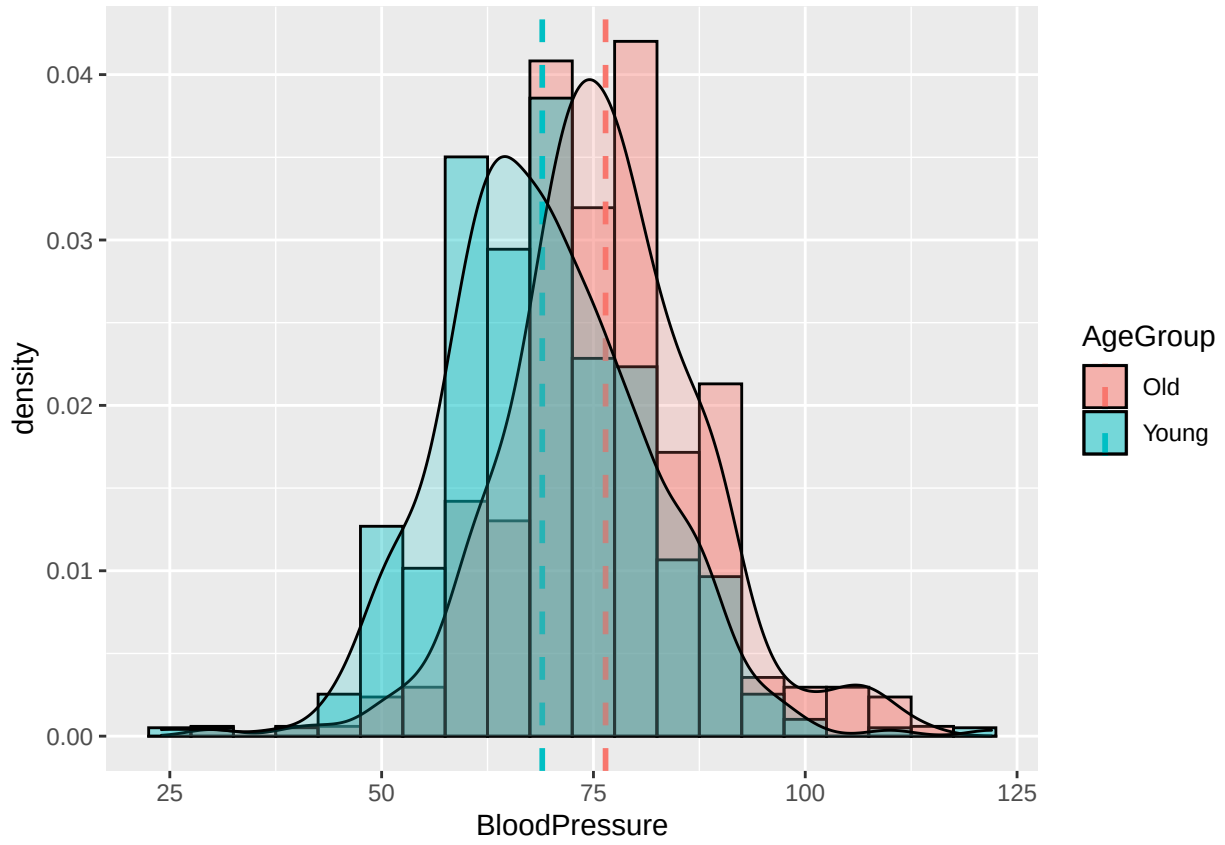
Predictive performance assessment (if applicable)

```
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()
```

Stan code

```
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
  )
```

```
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
  )
```

```
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
```

```
  )
```

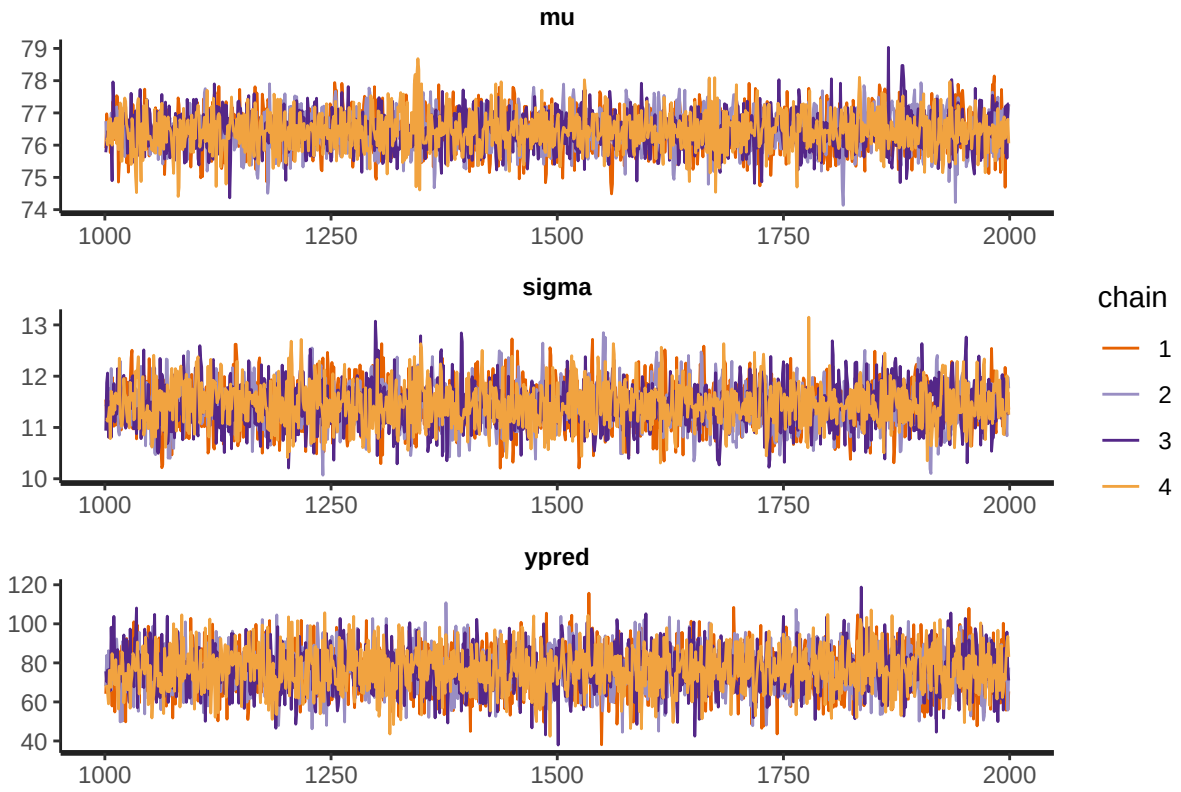
```
fit_hiera_young = sampling(hieramodel,
  data = data_hiera_young,            # named list of data
  chains = 4,             # number of Markov chains
  warmup = 1000,          # number of warmup iterations per chain
  iter = 2000,            # total number of iterations per chain
  cores = 4,   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

Posterior predictive checks

```
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  0.0102  0.615 75.3 76.0 76.4 76.9  77.6  3598    1     1 75.5 76.4
## sigma 11.4  0.0074  0.443 10.6 11.1 11.4 11.7  12.3  3565    1     1 10.7 11.4
## ypred 76.2  0.1766 11.464 54.2 68.4 76.0 84.1  98.6  4203    1     1 57.6 76.0
##        Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu    77.4    0.0339  0.01909  0.01143   0.0164     0.0294 0.00724     3605
## sigma 12.2    0.0269  0.00997  0.00806   0.0112     0.0211 0.00524     3578
## ypred 95.3    0.5850  0.26792  0.22403   0.2657     0.4429 0.12498     4216
##       Tail_ESS
## mu        2311
## sigma     2818
## ypred     4104
```

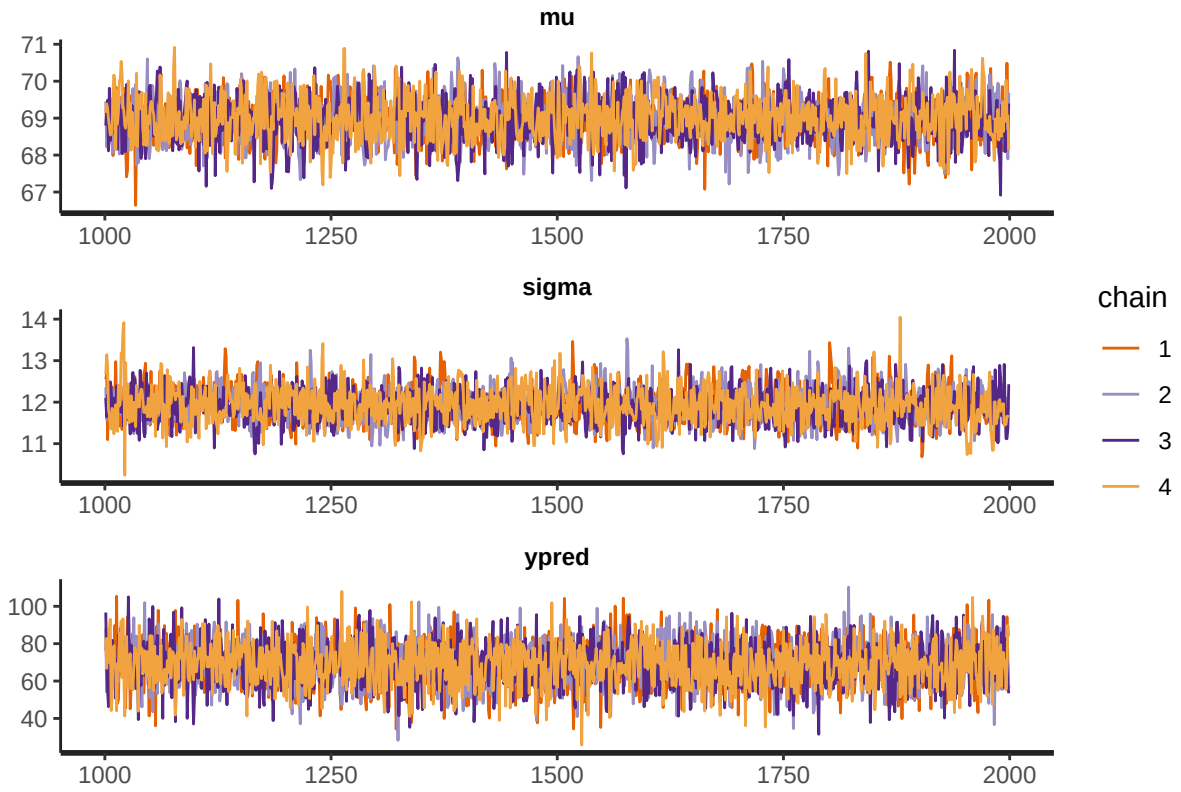
```
traceplot(fit_nonhiera_old, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```




```
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  0.0109  0.600 67.8 68.6 69.0 69.4  70.2  3007    1     1 68.0 69.0
## sigma 11.9  0.0076  0.425 11.1 11.6 11.9 12.2  12.8  3127    1     1 11.2 11.9
## ypred 69.0  0.1861 11.810 45.7 61.0 69.0 77.2  92.0  4023    1     1 49.8 69.0
##        Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu    70.0    0.0292   0.0140  0.01378  0.01258     0.0257 0.00771     3029
## sigma 12.6    0.0181   0.0101  0.00906  0.00912     0.0362 0.00539     3158
## ypred 88.4    0.3674   0.2277  0.25689  0.20768     0.3358 0.13157     4028
##       Tail_ESS
## mu        2601
## sigma     2677
## ypred     4001
```

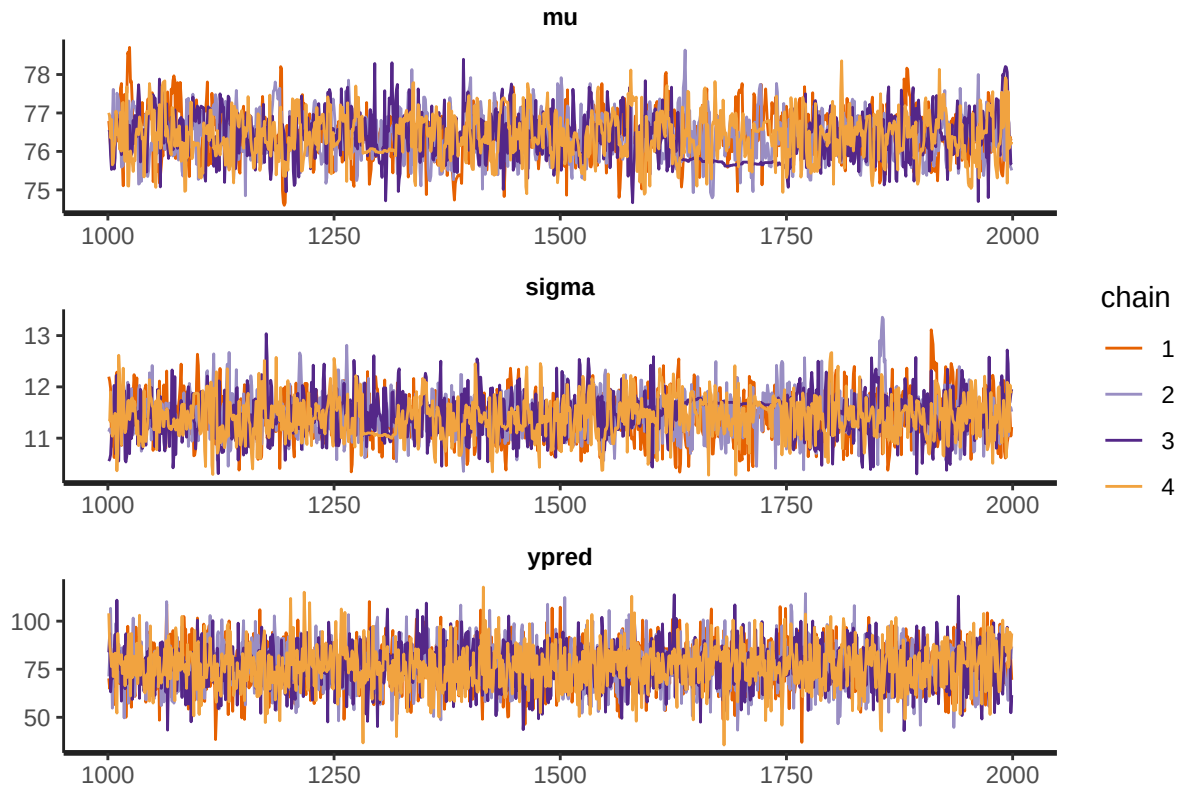
```
traceplot(fit_nonhiera_young, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```




```
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  0.0198 0.610 75.3 76.0 76.4 76.8  77.6   918 1.00     1 75.4 76.4
## sigma    11.4  0.0094 0.426 10.6 11.1 11.4 11.7  12.3  2060 1.01     1 10.7 11.4
## mu_hypo  76.4  0.0198 0.610 75.3 76.0 76.4 76.8  77.6   919 1.01     1 75.5 76.4
##           Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu       77.4    0.0368   0.0346   0.0284  0.01878     0.0316 0.01398     1009
## sigma    12.1    0.0205   0.0164   0.0145  0.00952     0.0402 0.00668     2102
## mu_hypo  77.4    0.0344   0.0286   0.0272  0.02140     0.0293 0.01399      992
##          Tail_ESS
## mu           1443
## sigma        2121
## mu_hypo      1395
```

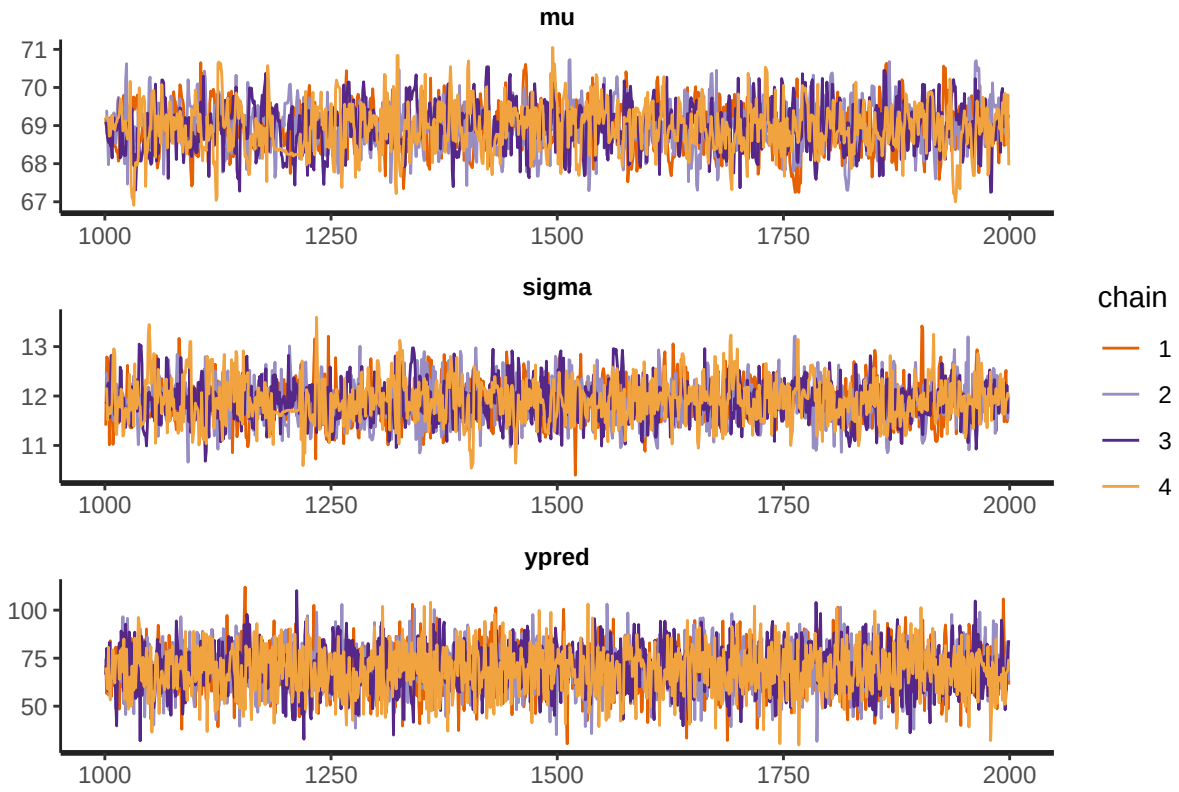
```
traceplot(fit_hiera_old, inc_warmup = FALSE, nrow = 3, pars=c("mu", "sigma", "ypred"))
```



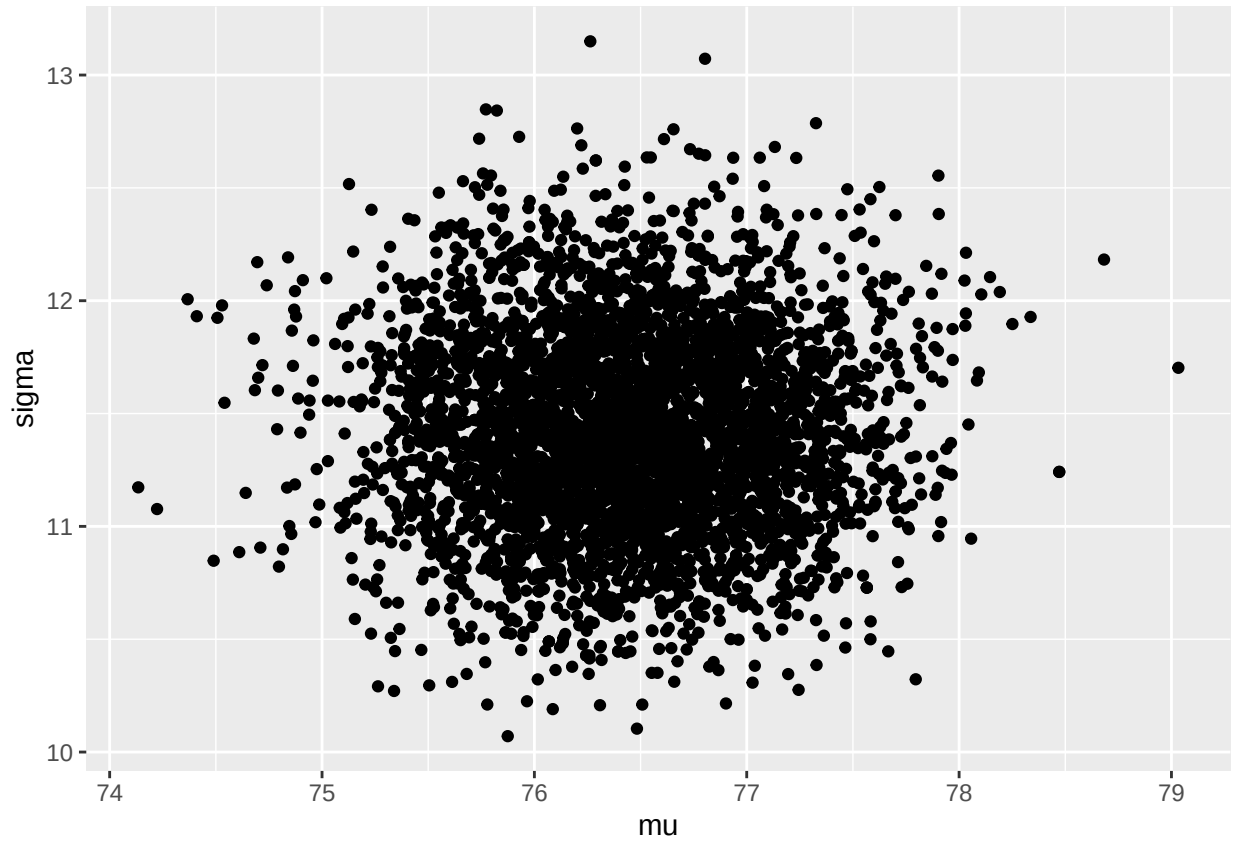

```
head(monitor(fit_hiera_young, print = FALSE),3)
```

```
##          mean se_mean    sd 2.5%  25%  50%  75% 97.5% n_eff Rhat valid   Q5  Q50
## mu       69.0  0.0159 0.605 67.8 68.5 69.0 69.4  70.1  1453    1     1 68.0 69.0
## sigma    11.9  0.0089 0.417 11.1 11.6 11.9 12.2  12.8  2205    1     1 11.2 11.9
## mu_hypo  69.0  0.0159 0.607 67.8 68.5 69.0 69.4  70.2  1464    1     1 68.0 69.0
##           Q95 MCSE_Q2.5 MCSE_Q25 MCSE_Q50 MCSE_Q75 MCSE_Q97.5 MCSE_SD Bulk_ESS
## mu       69.9    0.0486  0.01840  0.01774   0.0202     0.0471 0.01122     1455
## sigma    12.6    0.0173  0.00938  0.00975   0.0133     0.0287 0.00631     2206
## mu_hypo  69.9    0.0424  0.01877  0.01611   0.0206     0.0442 0.01122     1463
##          Tail_ESS
## mu           1465
## sigma        2097
## mu_hypo      1609
```

```
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()
```



Model comparison with LOO-CV

```
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         2.8  0.5
## looic      2618.3 34.9
## ------
## 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      3082.5 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         2.8  0.5
## looic      2618.3 34.9
## ------
## 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      3082.5 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:"
```

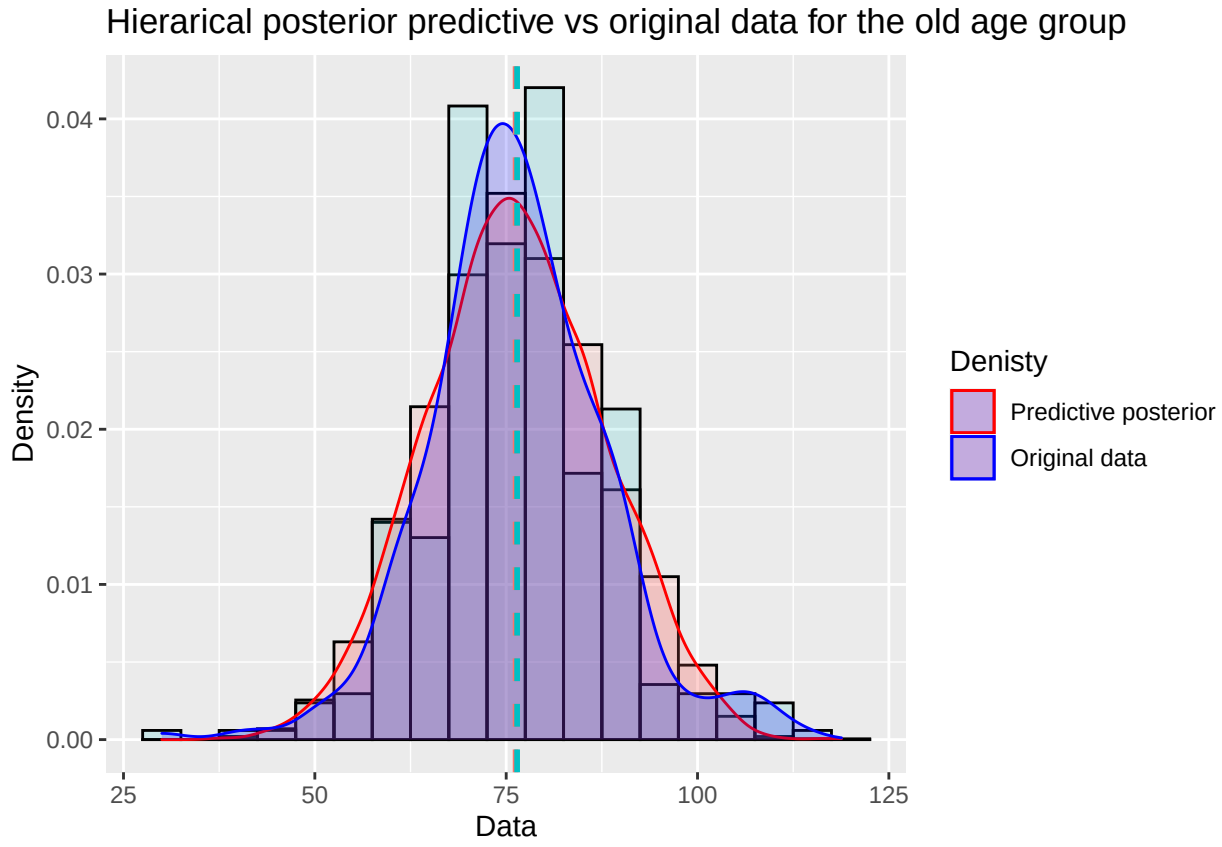
```
loo_compare(loo_nonhiera_young, loo_hiera_young)
```

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

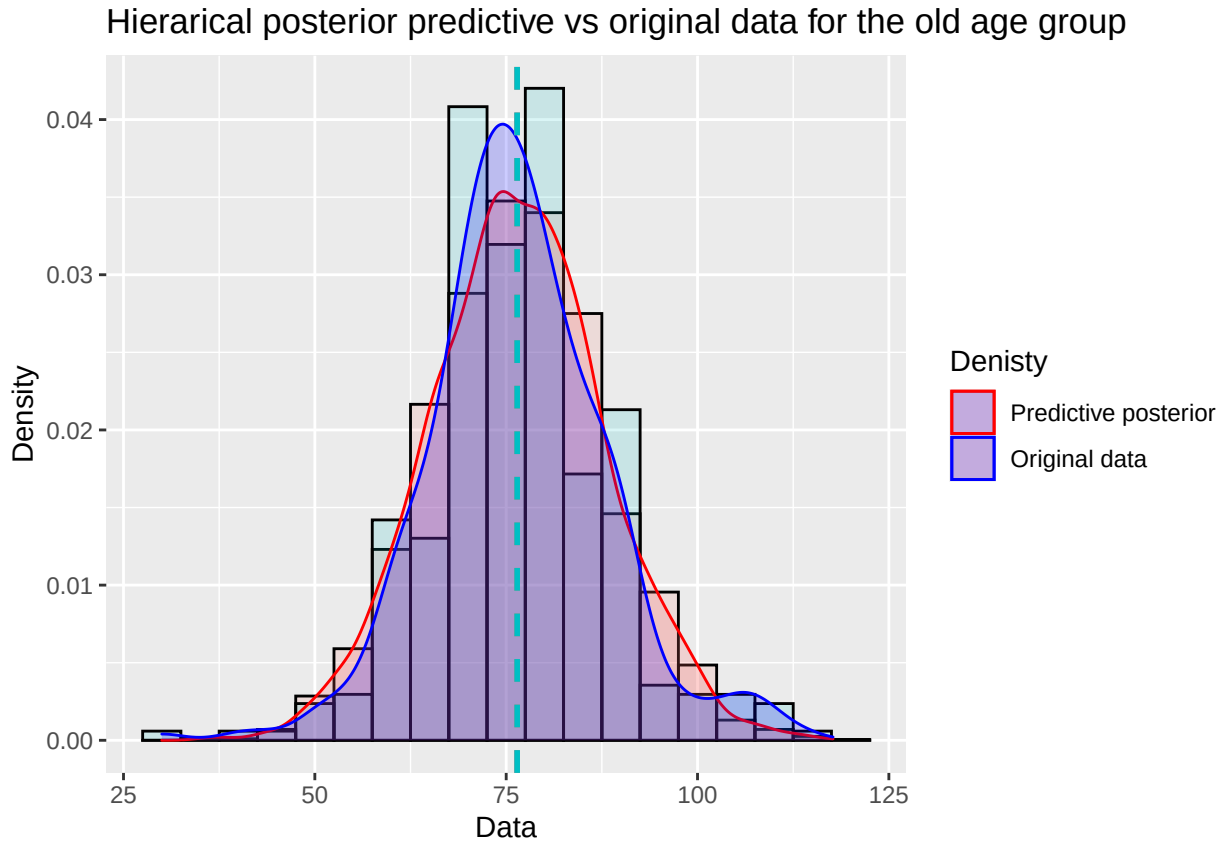
```
## model2 0.0       0.0
```

```
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))
```

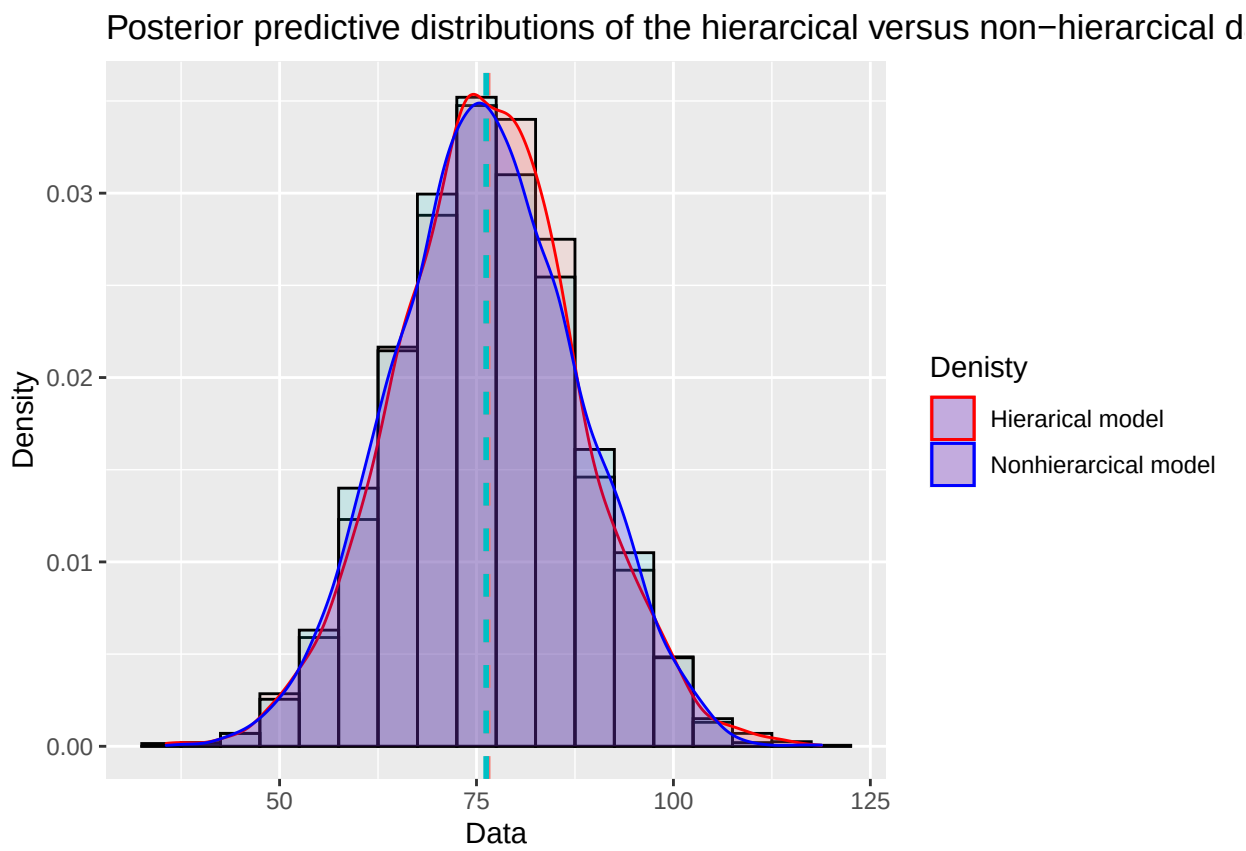
```
ggplot(extract_nonhiera_old, aes(x=ypred)) + ggtitle("Hierarical posterior predictive vs original data f
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", colour="black"
  geom_histogram(data= data_old, aes(x=BloodPressure,y=..density..), binwidth = 5, colour="black",positi
  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="#F8766D", linetype="dashed", size=1) +
  geom_vline(data=data_old, aes(xintercept=mean(BloodPressure), color="Orig"), color="#00BFC4", linetype
  labs(x="Data", y ="Density", colour = "legend") +
  scale_colour_manual(name = 'Denisty', values=c('Sim'='red','Orig'='blue'), labels = c('Predictive post
```



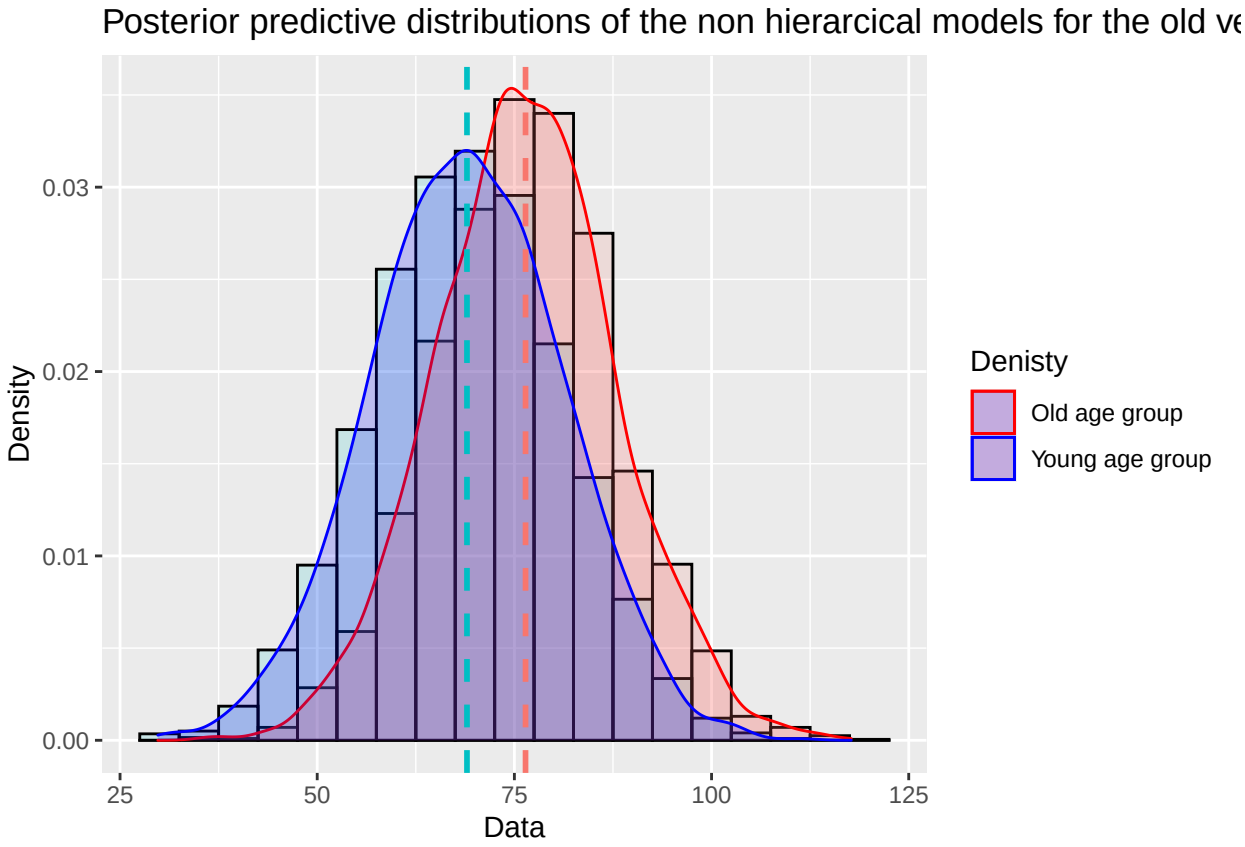

```
ggplot(extract_hiera_old, aes(x=ypred)) + ggtitle("Hierarical posterior predictive vs original data for
  geom_histogram(aes(y=..density..), binwidth = 5, colour="black",position = "identity", colour="black"
  geom_histogram(data= data_old, aes(x=BloodPressure,y=..density..), binwidth = 5, colour="black",positi
  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="#F8766D", linetype="dashed", size=1) +
  geom_vline(data=data_old, aes(xintercept=mean(BloodPressure), color="Orig"), color="#00BFC4", linetype
  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="#F8766D", linetype="dashed", size=1) +
  geom_vline(data=extract_nonhiera_old, aes(xintercept=mean(ypred), color="Orig"), color="#00BFC4", lin
  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 hierarc
  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="#F8766D", linetype="dashed", size=1) +
  geom_vline(data=extract_hiera_young, aes(xintercept=mean(ypred), color="Orig"), color="#00BFC4", linet
  labs(x="Data", y ="Density", colour = "legend") +
  scale_colour_manual(name = 'Denisty', values=c('Sim'='red','Orig'='blue'), labels = c('Old age group'
```



Sensitivity analysis

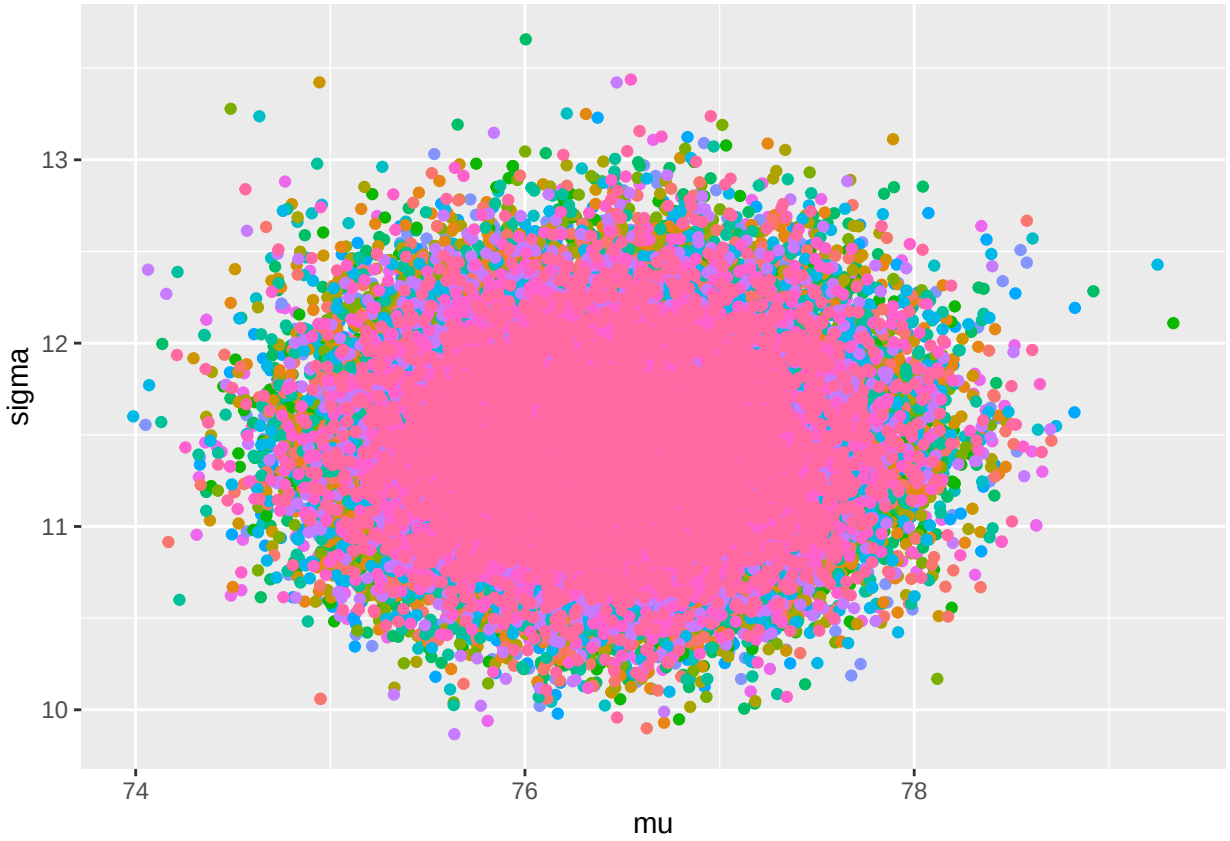
```
mean_mu_prior_sensitivity = c(0, 50, 100, 1000)
mean_sigma_prior_old_sensitivity = c(1, 10, 100, 1000)
var_prior_old_sensitivity = c(1, 10, 100, 1000)
fit_sensitivity = c()
for (i in 1:length(mean_mu_prior_sensitivity)){
  for (j in 1:length(mean_sigma_prior_old_sensitivity)){
  data_sensitivity <- 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
    ))
```

```
  }
}
```

```
gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}
colors = gg_color_hue(16)
```

```
  ggplot() +
  geom_point(data=data.frame(extract(fit_sensitivity[[1]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[2]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[3]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[4]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[5]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[6]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[7]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[8]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[9]],inc_warmup=TRUE)), aes(x=mu, sigma, color=colo
  geom_point(data=data.frame(extract(fit_sensitivity[[10]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  geom_point(data=data.frame(extract(fit_sensitivity[[11]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  geom_point(data=data.frame(extract(fit_sensitivity[[12]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  geom_point(data=data.frame(extract(fit_sensitivity[[13]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  geom_point(data=data.frame(extract(fit_sensitivity[[14]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  geom_point(data=data.frame(extract(fit_sensitivity[[15]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  geom_point(data=data.frame(extract(fit_sensitivity[[16]],inc_warmup=TRUE)), aes(x=mu, sigma, color=col
  theme(legend.position = "None")
```



Discussion

Conclusion

Self-reflection

References

Gurven, Michael, Aaron D. Blackwell, Daniel Eid Rodríguez, Jonathan Stieglitz, and Hillard Kaplan. 2012. "Does Blood Pressure Inevitably Rise with Age?" *Hypertension* 60 (1): 25–33. https://doi.org/10.1161/hypertensionaha.111.189100.

Mayo Clinic Staff. 2021. "Blood Pressure Chart: What Your Reading Means." 2021. https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/in-depth/blood-pressure/art-20050982.