Applied Bayesian modeling - Exam 2, fall 2022

Álvaro J. Castro Rivadeneira - 32381790

December 17, 2022

General Information

General instructions and grading information have been omitted.

Info about the data and outcome of interest

In this exam, we examine the same outcome of interest as in Exam 1 but consider a different data set and questions/models.

We consider data y_i for $i = 1, .., n$, where y_i refers to a health score calculated for an individual i. The health score can be any value, more negative health scores indicate poorer health while more positive health scores indicate better health. In addition to an individual i's health score y_i, the available data sets also includes individuals' age a_i (with ages ranging from 15 to 65) and their county of residence, denoted by index $j[i]$.

The data set is saved in `dat_exam2_fall2022.csv`, where `y` refers to the health score, `county` to county, and `age` to age.

Question 1 (10 points)

Consider the following Bayesian model, referred to in the remainder as model 1:

$$y_i|\alpha_{j[i]}, \beta, \sigma_y \overset{i.i.d.}{\sim} N(\alpha_{j[i]} + \beta(a_i - 30), \sigma_y^2),$$

$$\alpha_j|\mu_\alpha, \sigma_\alpha \overset{i.i.d.}{\sim} N(\mu_\alpha, \sigma_\alpha^2),$$

with brm-default priors on model parameters $\beta, \sigma_y, \mu_\alpha, \sigma_\alpha$.

Fit model 1 to the data set and check MCMC-related diagnostics including Rhat and effective sample sizes. If these diagnostics suggest issues, check for coding errors and/or change MCMC-related parameters such that the resulting fit can be used for inference.

To hand in:
- Code to do model fitting and printed summary of model fit for the model that you want to use for inference.
- Report the lowest values of Rhat and the lowest effective sample sizes among the parameters $\beta, \sigma_y, \mu_\alpha, \sigma_\alpha$, and discuss briefly whether these values indicate issues or not.

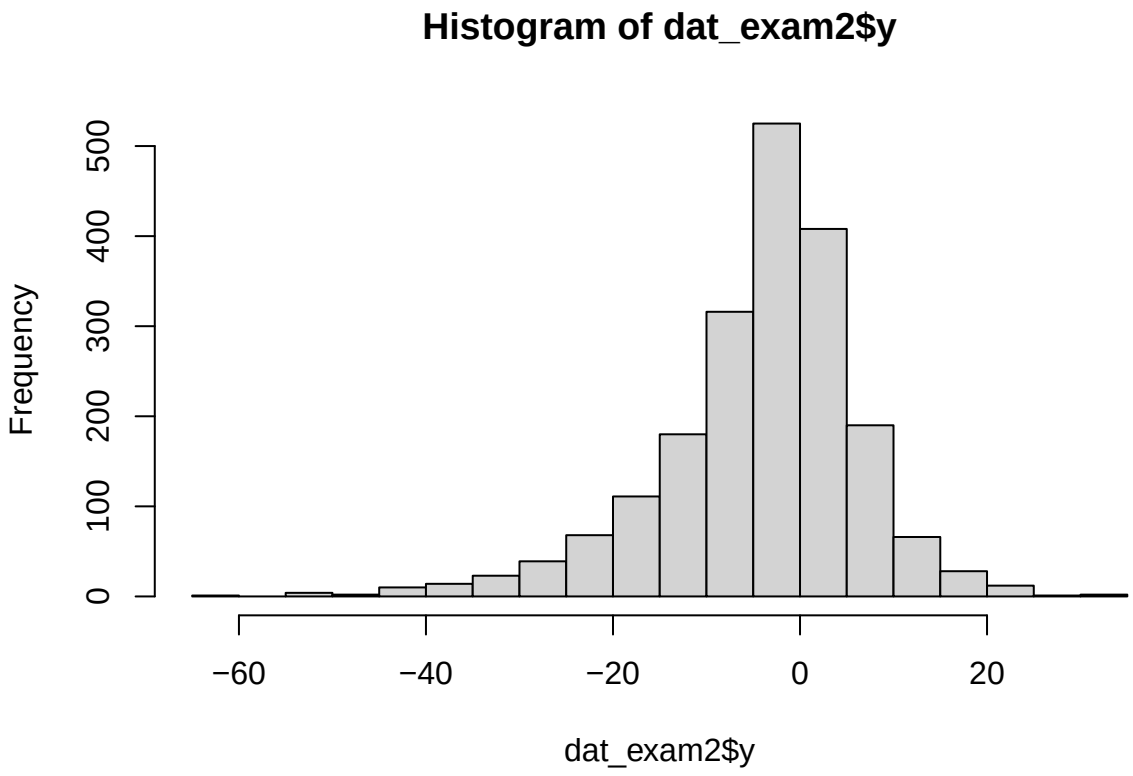
Answer

First, import data and conduct some exploratory data analysis (EDA):

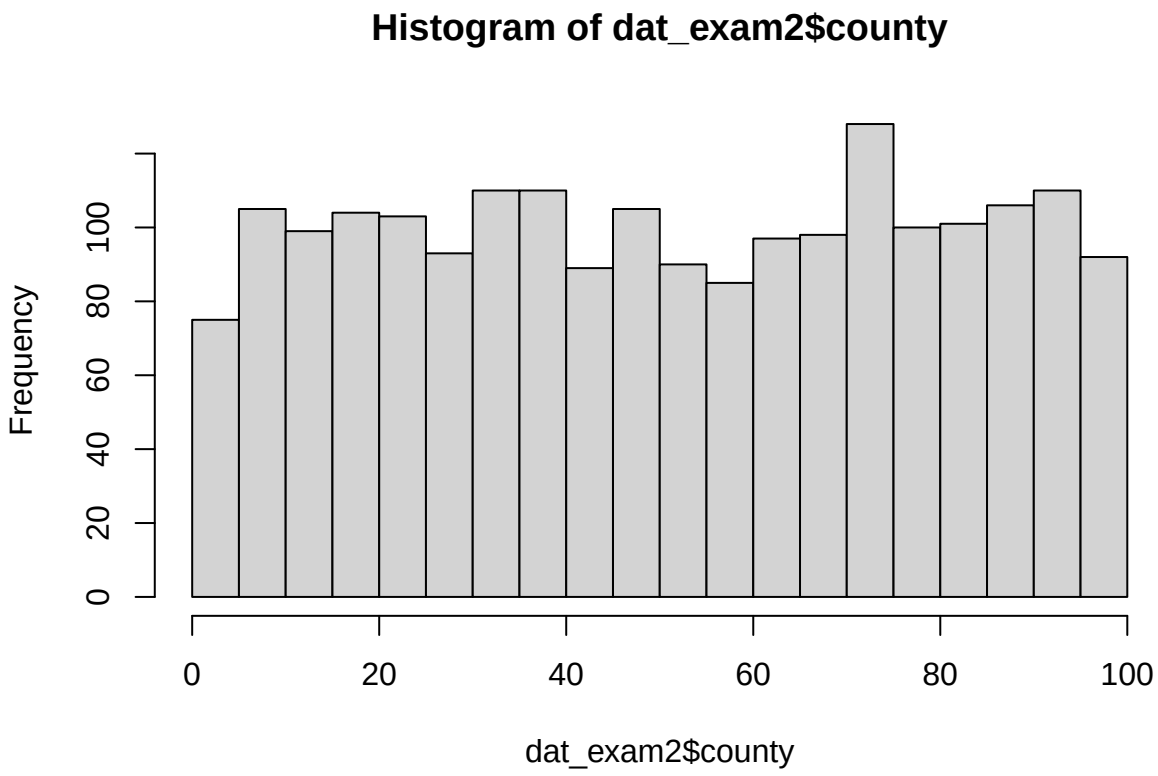
```
dat_exam2 <- read_csv("dat_exam2_fall2022.csv")
summary(dat_exam2)
```

```
##        y                county            age        
##  Min.   :-63.144   Min.   :  1.00   Min.   :15.00  
##  1st Qu.: -9.090   1st Qu.: 26.00   1st Qu.:27.00  
##  Median : -2.502   Median : 51.00   Median :39.00  
##  Mean   : -4.206   Mean   : 51.19   Mean   :39.55  
##  3rd Qu.:  2.174   3rd Qu.: 76.00   3rd Qu.:52.00  
##  Max.   : 31.921   Max.   :100.00   Max.   :64.00
```

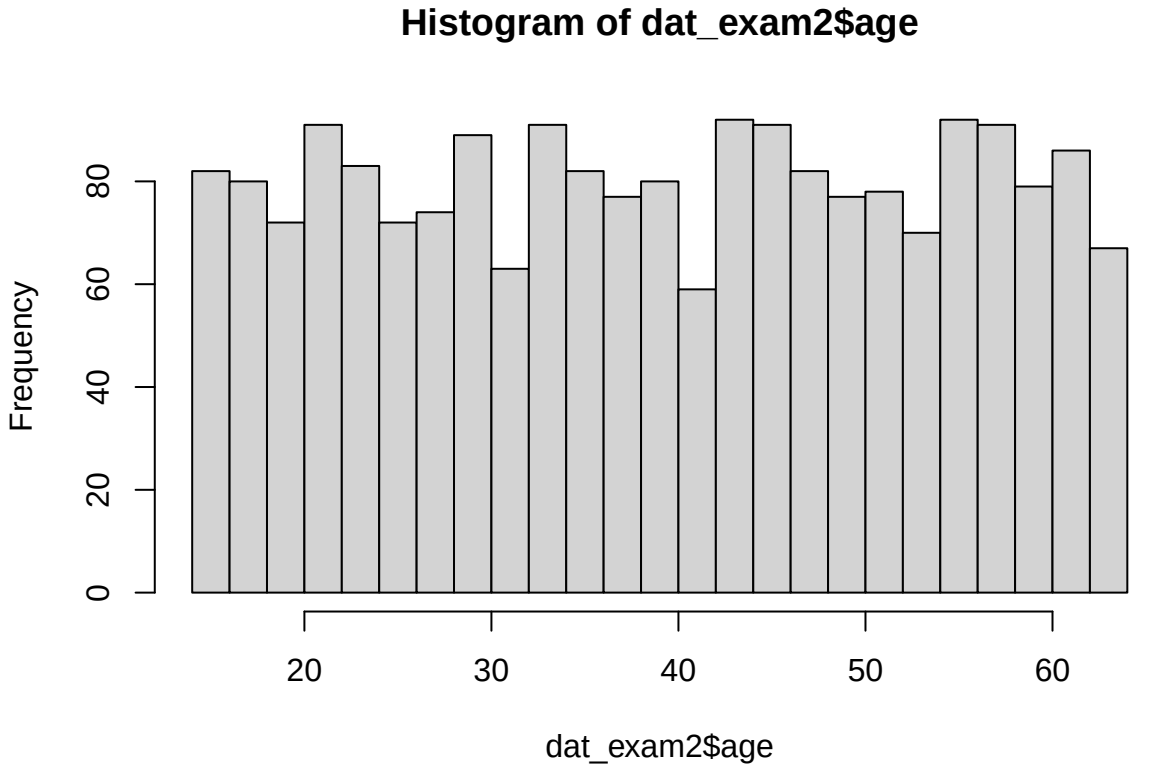
```
 #        y                county            age        
 # Min.   :-63.144   Min.   :  1.00   Min.   :15.00  
 # 1st Qu.: -9.090   1st Qu.: 26.00   1st Qu.:27.00  
 # Median : -2.502   Median : 51.00   Median :39.00  
 # Mean   : -4.206   Mean   : 51.19   Mean   :39.55  
 # 3rd Qu.:  2.174   3rd Qu.: 76.00   3rd Qu.:52.00  
 # Max.   : 31.921   Max.   :100.00   Max.   :64.00

# Visualize distributions
hist(dat_exam2$y, breaks=20)
```



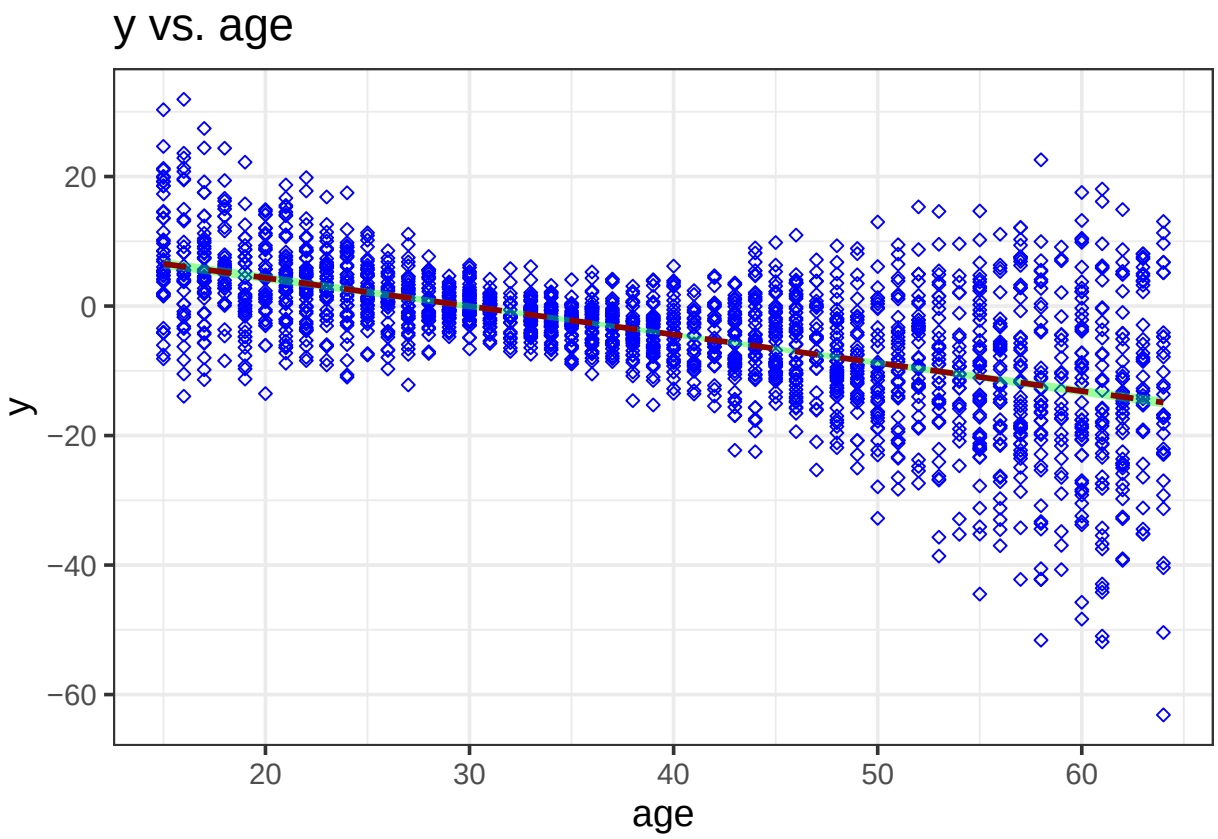

```
# Left skewed normal distribution
hist(dat_exam2$county, breaks=20)
```




```
# Uniform distribution
hist(dat_exam2$age, breaks=20)
```



```
# Uniform distribution
ggplot(dat_exam2, aes(x=age, y=y)) +
  geom_point(color="blue", shape=23) +
  geom_smooth(method=lm, linetype="dashed", color="darkred", fill="green") +
  scale_color_brewer(palette = "Set1") +
  theme_bw(base_size = 14) +
  ggtitle("y vs. age")
```




```
# Younger people have better age, as expected

# Create age-30 (semi-centered) variable
dat_exam2 <- dat_exam2 %>%
  mutate(ageless30 = age-30)
summary(dat_exam2$ageless30)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -15.000  -3.000   9.000   9.549  22.000  34.000
```

```
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# -15.000  -3.000   9.000   9.549  22.000  34.000
# I like to run a traditional glm to compare results:
exam2.q1.1 <- glm(y ~ 1 + ageless30, data = dat_exam2, family = "gaussian")
summary(exam2.q1.1)
```

```
##
```

```
## Call:
## glm(formula = y ~ 1 + ageless30, family = "gaussian", data = dat_exam2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -48.248   -4.111   -0.178    4.154   34.857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03086    0.23443  -0.132    0.895
## ageless30   -0.43722    0.01354 -32.291   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 76.48747)
##
##     Null deviance: 232576  on 1999  degrees of freedom
## Residual deviance: 152822  on 1998  degrees of freedom
## AIC: 14354
##
## Number of Fisher Scoring iterations: 2
```

```
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.03086    0.23443  -0.132    0.895
# ageless30   -0.43722    0.01354 -32.291   <2e-16 ***
exam2.q1.2 <- lmer(y ~ (1 | county) + ageless30, data = dat_exam2)
summary(exam2.q1.2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ (1 | county) + ageless30
##    Data: dat_exam2
##
## REML criterion at convergence: 14104
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0651 -0.4460 -0.0183  0.4641  3.8957
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  county   (Intercept) 14.98    3.871
##  Residual             61.82    7.862
## Number of obs: 2000, groups:  county, 100
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -0.06396    0.44278  -0.144
## ageless30   -0.43643    0.01239 -35.211
##
## Correlation of Fixed Effects:
##           (Intr)
## ageless30 -0.269
```

```
# Random effects:
#  Groups   Name        Variance Std.Dev.
#  county   (Intercept) 14.98    3.871
#  Residual             61.82    7.862
# Number of obs: 2000, groups:  county, 100
#
# Fixed effects:
#             Estimate Std. Error t value
# (Intercept) -0.06396    0.44278  -0.144
# ageless30   -0.43643    0.01239 -35.211
```

Now, to fit model1 to the data set:

```
model1 <- brm(y ~ (1 | county) + ageless30,
              file = "output/exam2q1",
              data = dat_exam2,
              chains = 4, iter = 2000, warmup = 1000,
              cores = getOption("mc.cores", 4),
              thin = 1, seed = 1234)
```

Now the results:

```
summary(model1)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ (1 | county) + ageless30
##    Data: dat_exam2 (Number of observations: 2000)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~county (Number of levels: 100)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     3.93      0.34     3.33     4.67 1.00     1383     1861
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.06      0.44    -0.93     0.82 1.00     1443     2246
## ageless30    -0.44      0.01    -0.46    -0.41 1.00     9852     2991
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     7.87      0.13     7.63     8.12 1.00     8566     2870
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

The point estimates are given below, as well as the 95% CIs:

```
posterior_summary(model1, probs = c(0.025, 0.5, 0.975))[1:4,]
```

```
##                           Estimate Est.Error       Q2.5         Q50      Q97.5
## b_Intercept          -0.05892198 0.4448414 -0.9275710 -0.05314737  0.8154342
## b_ageless30          -0.43632344 0.0123171 -0.4605944 -0.43642931 -0.4123524
## sd_county__Intercept  3.92582892 0.3420396  3.3303149  3.90362579  4.6655376
## sigma                 7.86582971 0.1256206  7.6264709  7.86540452  8.1189144
```

```
posterior_interval(model1, prob = 0.95,
                   variable = c("b_ageless30",
                                "sigma",
                                "b_Intercept",
                                "sd_county__Intercept"))
```

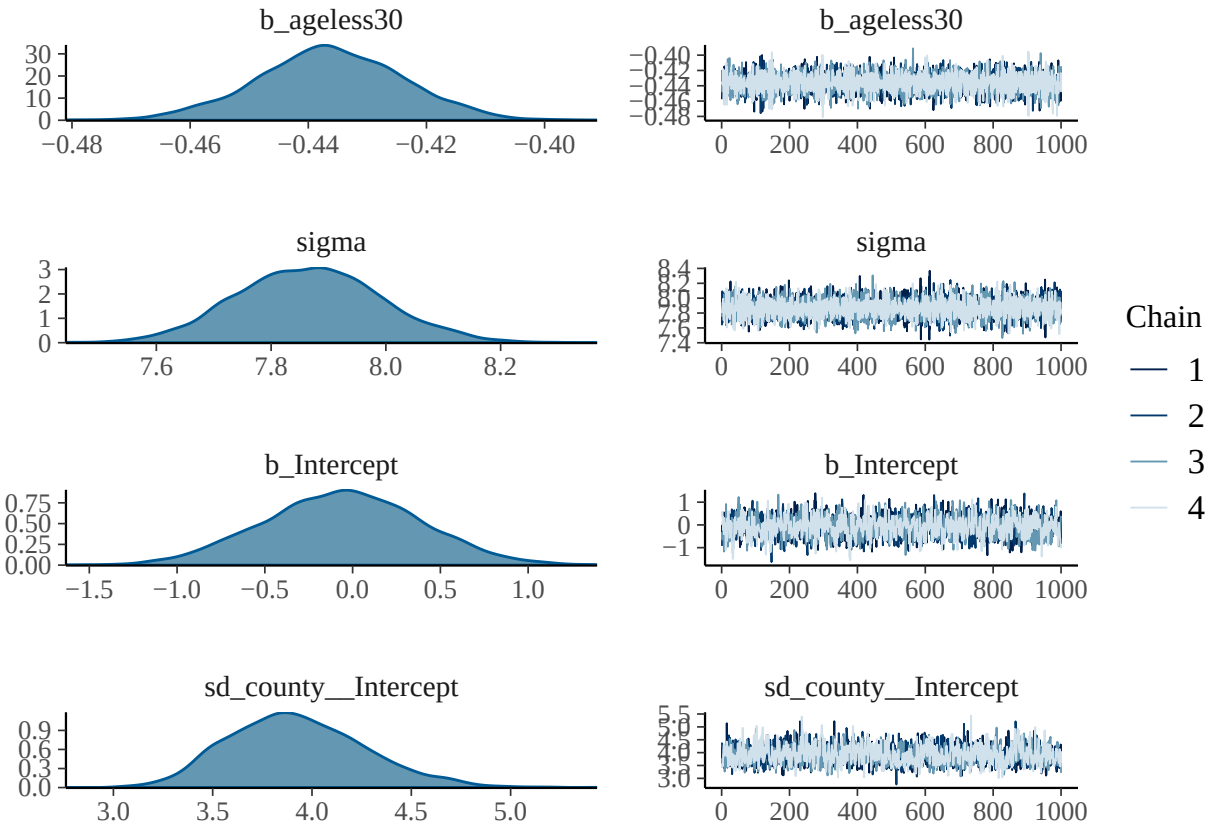
```
##                             2.5%      97.5%
## b_ageless30          -0.4605944 -0.4123524
## sigma                 7.6264709  8.1189144
## b_Intercept          -0.9275710  0.8154342
## sd_county__Intercept  3.3303149  4.6655376
```

```
results <- c("Posterior mean", "Low 95% CI", "High 95% CI", "Rhat", "Lowest ESS")
beta <- c(-0.44, -0.46, -0.41, "1.00", 2991)
sigma <- c(7.87, 7.63, 8.12, "1.00", 2870)
mu_alpha <- c(-0.06, -0.93, 0.82, "1.00", 1443)
sd_intercept <- c(3.93, 3.33, 4.67, "1.00", 1383)

res.model1 <- as.data.frame(cbind(results, beta, sigma, mu_alpha, sd_intercept))
knitr::kable(res.model1, col.names = c("Parameters",
                                       "$\\beta$; beta",
                                       "$\\sigma_y$; sigma",
                                       "$\\mu_\\alpha$; Intercept",
                                       "$\\sigma_\\alpha$; sd(Intercept)"))
```

Parameters	β; beta	σ_y; sigma	μ_α; Intercept	σ_α; sd(Intercept)
Posterior mean	-0.44	7.87	-0.06	3.93
Low 95% CI	-0.46	7.63	-0.93	3.33
High 95% CI	-0.41	8.12	0.82	4.67
Rhat	1.00	1.00	1.00	1.00
Lowest ESS	2991	2870	1443	1383

Now some MCMC diagnostics for μ

```
plot(model1, variable = c("b_ageless30", "sigma", "b_Intercept", "sd_county__Intercept"))
```



In our plots we see that the data are ~normally distributed and that the chains converge and mix well for all parameters. Additionally, as was shown previously, all Rhat values are 1.00 and all the lowest effective sample sizes are well over 1,000 which is adequate for our sample (we hope for ESS values greater than 400, given the 4 chains). Thus, we do not see any issues regarding the diagnostics for our model parameters.

Question 2 (5 points)

Continuing with model fit 1, provide a point estimate, 50% credible interval (NOT 95%), and interpretation of the estimates, for each of the following parameters: $\sigma_y, \sigma_\alpha, \mu_\alpha, \beta, \alpha_2$. Provide a context-specific interpretation of the parameters, do NOT use the terms intercept or slope in your interpretation.

Answer

To get the point estimate and 50% CI for the requested parameters I can use the following:

```
(exam2q2 <- posterior_summary(model1, probs = c(0.25, 0.5, 0.75))[c(1:4,6),])
```

```
##                          Estimate Est.Error        Q25         Q50        Q75
## b_Intercept           -0.05892198 0.4448414 -0.3564491 -0.05314737  0.2432771
## b_ageless30           -0.43632344 0.0123171 -0.4444441 -0.43642931 -0.4280074
## sd_county__Intercept   3.92582892 0.3420396  3.6812529  3.90362579  4.1469168
## sigma                  7.86582971 0.1256206  7.7786107  7.86540452  7.9498668
## r_county[2,Intercept]  0.13309337 1.7339773 -1.0113714  0.18364941  1.2469871
```

```
(res.model1.2 <- t(exam2q2))
```

```
##           b_Intercept b_ageless30 sd_county__Intercept     sigma
## Estimate  -0.05892198  -0.4363234            3.9258289 7.8658297
## Est.Error  0.44484144   0.0123171            0.3420396 0.1256206
## Q25       -0.35644905  -0.4444441            3.6812529 7.7786107
## Q50       -0.05314737  -0.4364293            3.9036258 7.8654045
## Q75        0.24327712  -0.4280074            4.1469168 7.9498668
##           r_county[2,Intercept]
## Estimate              0.1330934
## Est.Error             1.7339773
## Q25                  -1.0113714
## Q50                   0.1836494
## Q75                   1.2469871
```

```
# eta_2 = alpha - mu_alpha (random effects)
# To get the alpha_2 = eta_2 + mu_alpha, I will use code from class:
eta <- as_tibble(posterior_summary(model1, probs = c(0.25, 0.5, 0.75))[c(5:104),], rownames = "county")
eta$county <- c(1:100)
alphas <- coef(model1, summary = T, probs = c(0.25, 0.5, 0.75))$county %>%
  as_tibble(rownames = "county") %>%
  rename(Estimate = Estimate.Intercept)
alphas[2,]
```

```
## # A tibble: 1 x 11
##   county Estim~1 Est.E~2 Q25.I~3 Q50.I~4 Q75.I~5 Estim~6 Est.E~7 Q25.a~8 Q50.a~9
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 2       0.0742    1.71   -1.05   0.103    1.20  -0.436  0.0123  -0.444  -0.436
## # ... with 1 more variable: Q75.ageless30 <dbl>, and abbreviated variable names
## #   1: Estimate, 2: Est.Error.Intercept, 3: Q25.Intercept, 4: Q50.Intercept,
## #   5: Q75.Intercept, 6: Estimate.ageless30, 7: Est.Error.ageless30,
## #   8: Q25.ageless30, 9: Q50.ageless30
```

```
(alpha_2 <- unlist(alphas[2,c(2:6)]))
```

```
##            Estimate Est.Error.Intercept       Q25.Intercept       Q50.Intercept
##          0.07417139          1.70906834         -1.05402802          0.10330741
##       Q75.Intercept
##          1.19550928
```

```
# Update the estimate for alpha_2:
res.model1.2[,5] <- alpha_2
colnames(res.model1.2)[5] <- "alpha_2"
res.model1.2
```

```
##           b_Intercept b_ageless30 sd_county__Intercept     sigma     alpha_2
## Estimate  -0.05892198  -0.4363234            3.9258289 7.8658297  0.07417139
## Est.Error  0.44484144   0.0123171            0.3420396 0.1256206  1.70906834
## Q25       -0.35644905  -0.4444441            3.6812529 7.7786107 -1.05402802
## Q50       -0.05314737  -0.4364293            3.9036258 7.8654045  0.10330741
## Q75        0.24327712  -0.4280074            4.1469168 7.9498668  1.19550928
```

```
res.model1.2 <- round(res.model1.2, 2)
rownames(res.model1.2) <- c("Point estimate (mean)", "Standard Error",
                        "Low 50% CI (Q25)", "Median (Q50)", "High 50% CI (Q75)")
(res.model1.2 <- as.data.frame(res.model1.2) %>% select(4,3,1,2,5))
```

```
##                        sigma sd_county__Intercept b_Intercept b_ageless30
## Point estimate (mean)   7.87                3.93       -0.06       -0.44
## Standard Error          0.13                0.34        0.44        0.01
## Low 50% CI (Q25)        7.78                3.68       -0.36       -0.44
## Median (Q50)            7.87                3.90       -0.05       -0.44
## High 50% CI (Q75)       7.95                4.15        0.24       -0.43
##                        alpha_2
## Point estimate (mean)     0.07
## Standard Error            1.71
## Low 50% CI (Q25)         -1.05
## Median (Q50)              0.10
## High 50% CI (Q75)         1.20
```

```
knitr::kable(res.model1.2, col.names = c("$\\sigma_y$; sigma",
                                         "$\\sigma_\\alpha$; sd(Intercept)",
                                         "$\\mu_\\alpha$; Intercept",
                                         "$\\beta$; beta",
                                         "$\\alpha_2$; alpha_2"))
```

	σ_y; sigma	σ_α; sd(Intercept)	μ_α; Intercept	β; beta	α_2; alpha_2
Point estimate (mean)	7.87	3.93	-0.06	-0.44	0.07
Standard Error	0.13	0.34	0.44	0.01	1.71
Low 50% CI (Q25)	7.78	3.68	-0.36	-0.44	-1.05
Median (Q50)	7.87	3.90	-0.05	-0.44	0.10
High 50% CI (Q75)	7.95	4.15	0.24	-0.43	1.20

Interpretation: The results listed in the table are in the order they were requested in the question, but it makes more sense to start with μ_α, which refers to the estimated average health score among mean county health scores. In other words, it is an estimated mean among county means, which is -0.06. The 50% credible interval (50% CI) indicates that given the current information, we believe it is 50% probable that the true parameter lies between -0.36 and 0.24. β refers to the effect of age on average health scores. In our results, this means that approximately for every additional year of age, a person's health score is expected to decline by an average of 0.44 units with a 50% CI between -0.44 and -0.43. σ_y refers to the standard deviation of health scores among individuals within counties - how much these results vary in each individual county. In our results, this is estimated to be 7.87, with a 50% CI between 7.78 and 7.95. σ_α refers to the standard deviation of mean health scores across counties, or how much the health scores differ on average between counties, which in our results is estimated to be 3.93, with a 50% CI between 3.68 and 4.15. Finally, as requested, an estimate was found for the second county, whose average health score is estimated at 0.07, with a 50% credible interval between -1.05 and 1.20. So, people in the second county are on average healthier than people in other counties.

Question 3 (5 points)

Continuing with model fit 1, obtain the posterior predictive probability that a yet-to-be-sampled individual with age $a = 20$ in a yet-to-be-sampled county has a health outcome greater than 10.

In your answer, in addition to producing and reporting the outcome of interest, also introduce notation and give an expression for the probability using the samples of model parameters, or, if needed, using samples obtained in additional sampling steps. If using additional sampling steps, explain with additional equations how those samples are obtained.

Answer

We assume that the hierarchical sampling distribution holds true, and we want to predict the probability that a yet-to-be-sampled individual k with age $a = 20$ in a yet-to-be-sampled county $h = j[k]$ has a health score greater than 10. To do that, we can obtain samples from the posterior predictive distribution, denoted by $\tilde{y}_k^{(s)} \sim p(\tilde{y}_k|\boldsymbol{y}, a_k = 20)$.

This can be sampled with the following steps:
(1) Sample $(\mu_\alpha^{(s)}, \sigma_\alpha^{(s)}, \sigma_y^{(s)}, \beta^{(s)}) \sim p(\mu_\alpha, \sigma_\alpha, \sigma_y, \beta|\boldsymbol{y})$,
(2) Sample $\tilde{\alpha}_h^{(s)} \sim p(\tilde{\alpha}_h|\mu_\alpha^{(s)}, \sigma_\alpha^{2(s)})$,
(3) Sample $\tilde{y}_k^{(s)} \sim p(\tilde{y}_k|\tilde{\alpha}_h^{(s)}, \beta^{(s)}, a_k = 20, \sigma_y^{2(s)})$.

I already have samples for $(\mu_\alpha^{(s)}, \sigma_\alpha^{(s)}, \sigma_y^{(s)}, \beta^{(s)})$ from fitting the model, so, I will obtain random samples for $\tilde{\alpha}_h^{(s)}$ using random draws from: $\tilde{\alpha}_h^{(s)}|(\mu_\alpha^{(s)}, \sigma_\alpha^{2(s)}) \sim N(\mu_\alpha^{(s)}, \sigma_\alpha^{2(s)})$.

Once I have samples of $\tilde{\alpha}_h^{(s)}$ from the previous step, I can obtain samples from $p(\tilde{y}_k|\boldsymbol{y}, \alpha_h, a_k = 20)$ with $j[k] = h$ by sampling $\tilde{y}_k^{(s)} \sim p(\tilde{y}_k|\tilde{\alpha}_h^{(s)}, \beta^{(s)}, a = 20, \sigma_y^{2(s)})$, knowing that $\tilde{y}_k|(\tilde{\alpha}_h^{(s)}, \beta^{(s)}, a = 20, \sigma_y^{2(s)}) \sim N(\alpha_h^{(s)} - 10\beta, \sigma_y^{2(s)})$. For ease of notation, I will set $\theta_k^{(s)} = \alpha_h^{(s)} - 10\beta$.

This will give me random draws of $\tilde{y}_k$, the individual's health score, with which I can create a histogram of my posterior predictive density. Moreover, with results from these samples $\tilde{y}_k^{(s)}$, I can estimate the probability that the individual k has a health score greater than 10, by recognizing that $P(\tilde{y}_k > 10|\boldsymbol{y}) \approx 1/S \sum \boldsymbol{I}(\tilde{y}_k^{(s)} > 10)$.

```
# Step 1
set.seed(1234)
samp <- as_draws_df(model1)
dim(samp)
```

```
## [1] 4000  109
```

```
names(samp)[1:5]
```

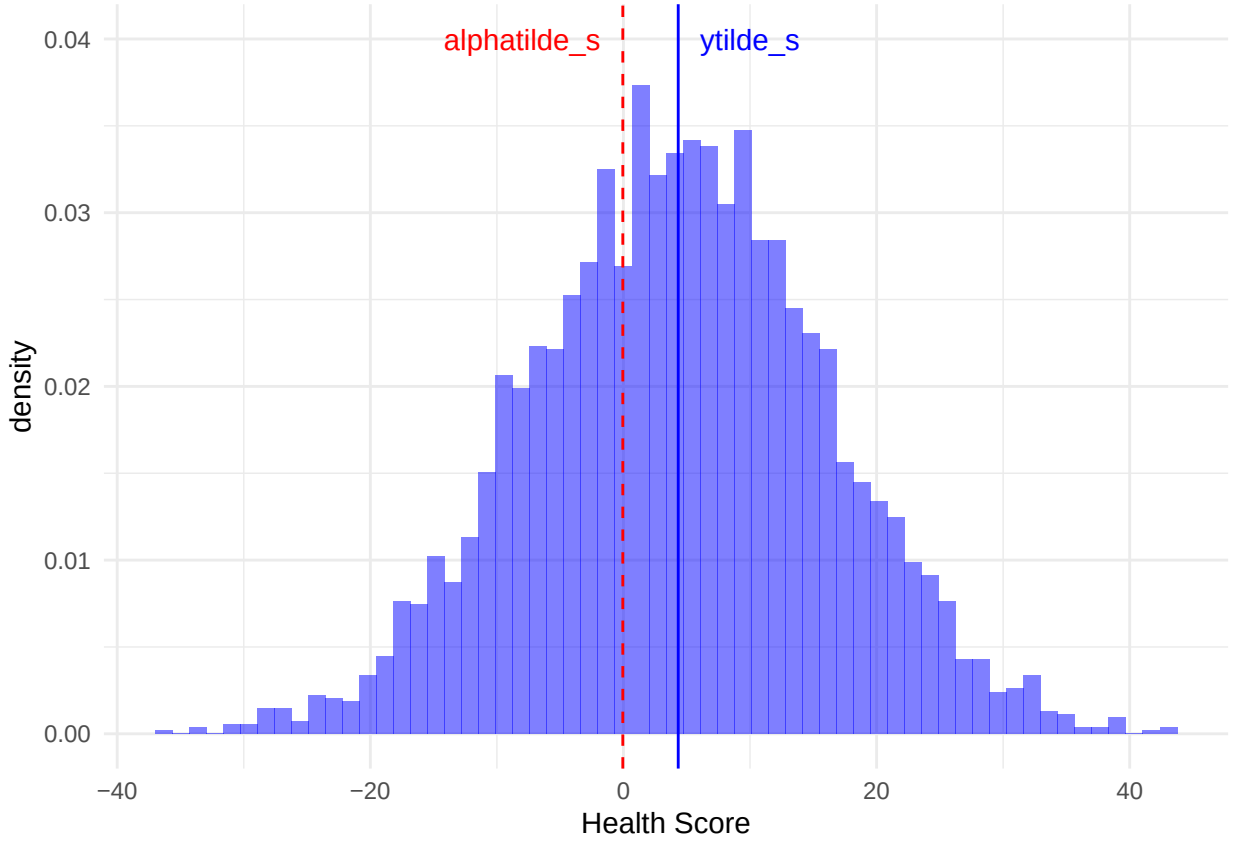
```
## [1] "b_Intercept"           "b_ageless30"           "sd_county__Intercept"
## [4] "sigma"                 "r_county[1,Intercept]"
```

```
mualpha_s <- samp$b_Intercept
beta_s <- samp$b_ageless30
sigmaalpha_s <- samp$sd_county__Intercept
sigmay_s <- samp$sigma
S <- length(sigmay_s)
```

```r
# Obtain a normally distributed random sample of alphas using the sampled posterior parameters
alphatilde_s <- rnorm(S, mualpha_s, sigmaalpha_s)
# Step 2
set.seed(1234)
alphanew_s <- rnorm(S, mualpha_s, sigmaalpha_s)
theta_s <- alphanew_s - (10 * beta_s)
# Step 3
set.seed(1234)
ytilde_s <- rnorm(S, theta_s, sigmay_s)
# Obtain point estimates and 95% CI (using tidybayes):
point_interval(ytilde_s, .point = mean)
```

```
##          y      ymin     ymax .width .point .interval
## 1 4.326585 -18.35611 27.08927   0.95   mean        qi
```

```r
# We can see that the point estimate is much higher than the overall mean, which is expected, given age
# Step 4
p <- as_tibble(ytilde_s) %>%
  ggplot(aes(ytilde_s, after_stat(density), fill = "blue")) +
  geom_histogram(alpha = .5, fill = "blue", bins = 60, size = 1.5) +
  theme_minimal() +
  xlab("Health Score") +
  geom_vline(xintercept = mean(ytilde_s), col = "blue") +
  geom_vline(xintercept = mean(alphatilde_s), col = "red", linetype = "dashed")
p + annotate("text", x = 10, y = 0.04, label = "ytilde_s", color = "blue") +
  annotate("text", x = -8, y = 0.04, label = "alphatilde_s", color = "red")
```



Finally, as indicated above, to get the predicted probability that the health score will be greater than 10, we can use:

```
mean(ytilde_s > 10)
```

```
## [1] 0.3125
```

Thus, the probability is 0.31.

Question 4 (10 points)

I was running out of time, so I stopped writing down the questions...

```
model2 <- brm(y ~ (1 + ageless30 | county) + ageless30,
              file = "output/exam2q4",
              data = dat_exam2,
              chains = 4, iter = 2000, warmup = 1000,
              cores = getOption("mc.cores", 4),
              thin = 1, seed = 1234)
```

Now the results:

```
summary(model2)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ (1 + ageless30 | county) + ageless30
##    Data: dat_exam2 (Number of observations: 2000)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~county (Number of levels: 100)
##                           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                 2.29      0.18     1.96     2.67 1.00      957
## sd(ageless30)                 0.53      0.04     0.46     0.61 1.00      619
## cor(Intercept,ageless30)     -0.62      0.07    -0.73    -0.48 1.00      529
##                           Tail_ESS
## sd(Intercept)                 1534
## sd(ageless30)                 1229
## cor(Intercept,ageless30)       999
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.15      0.23    -0.61     0.31 1.01      739     1105
## ageless30    -0.43      0.05    -0.54    -0.32 1.01      492      763
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     1.95      0.03     1.89     2.01 1.00     6681     3168
##
```

```
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

All Rhat values are close to 1, with the highest values being at 1.01, and most at 1.00 which is excellent. Further, the smallest ESS are greater than 500, which is adequate, as we expect them to be greater than 400.

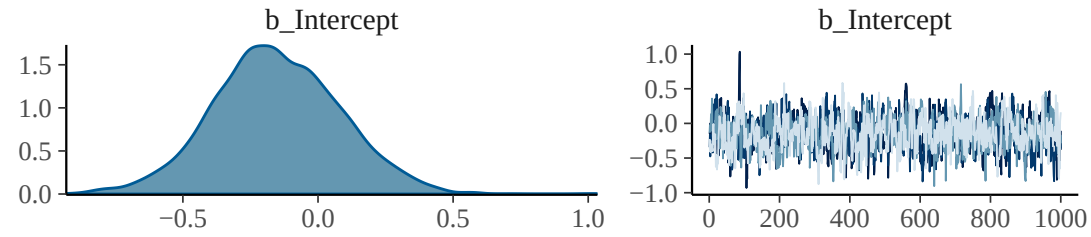
The point estimates are given below, as well as the 95% CIs:

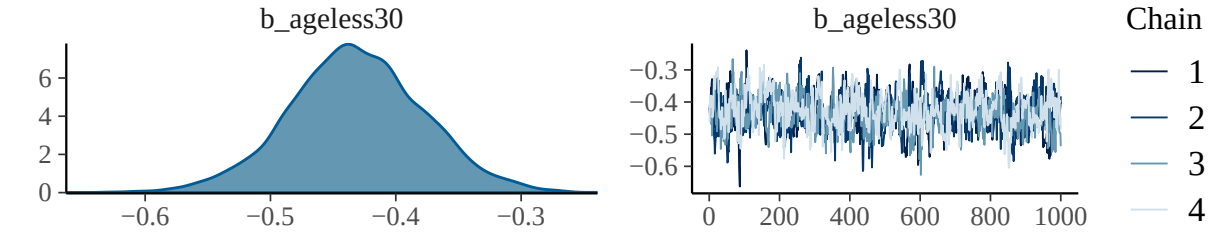
```
posterior_summary(model2, probs = c(0.025, 0.5, 0.975))[1:6,]
```

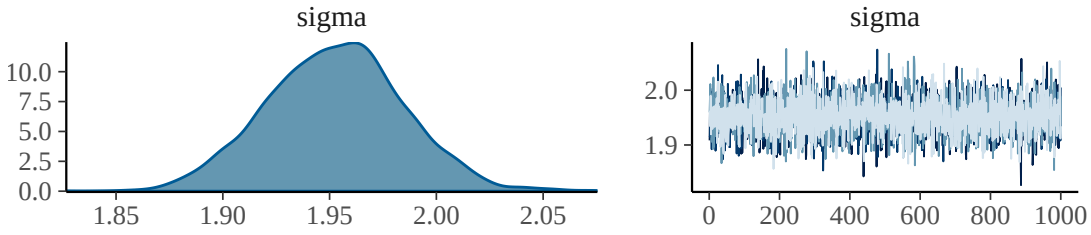
```
##                                   Estimate  Est.Error       Q2.5        Q50
## b_Intercept                     -0.1529238 0.23378814 -0.6083606 -0.1607241
## b_ageless30                     -0.4316096 0.05366921 -0.5386426 -0.4325322
## sd_county__Intercept             2.2894467 0.18120296  1.9598026  2.2777749
## sd_county__ageless30             0.5294144 0.03907310  0.4598159  0.5280241
## cor_county__Intercept__ageless30 -0.6175124 0.06580988 -0.7326014 -0.6230692
## sigma                            1.9525687 0.03199244  1.8908771  1.9528654
##                                      Q97.5
## b_Intercept                      0.3125194
## b_ageless30                     -0.3242481
## sd_county__Intercept             2.6704991
## sd_county__ageless30             0.6104845
## cor_county__Intercept__ageless30 -0.4801420
## sigma                            2.0142636
```

Now some MCMC diagnostics:

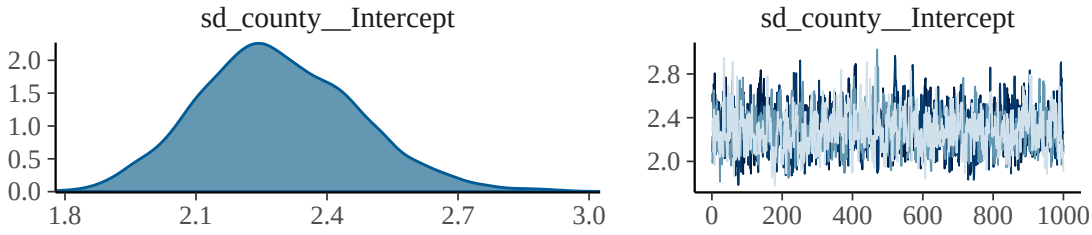
```
plot(model2, variable = c("b_Intercept", "b_ageless30", "sigma"))
```

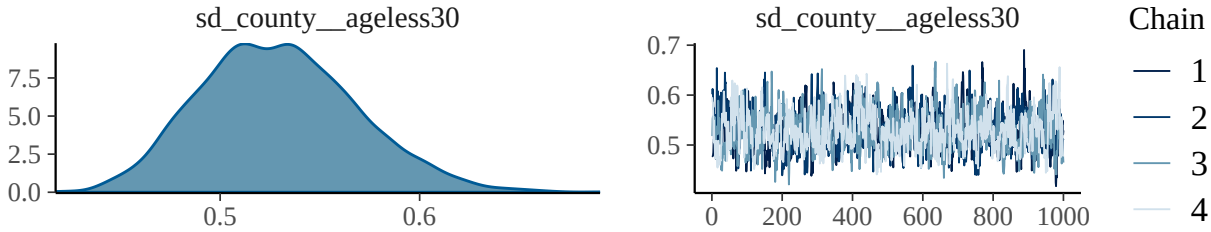


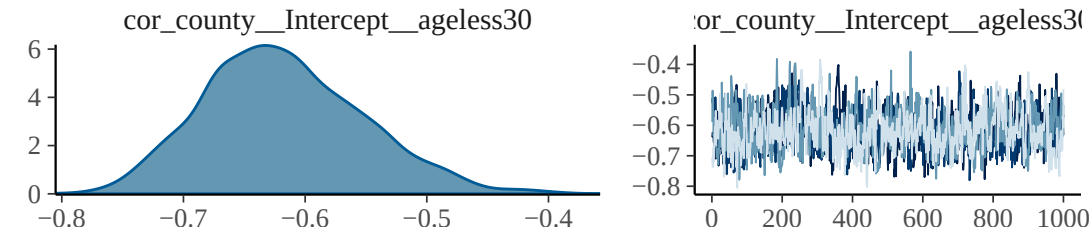





```
plot(model2, variable = c("sd_county__Intercept", "sd_county__ageless30", "cor_county__Intercept__ageles
```



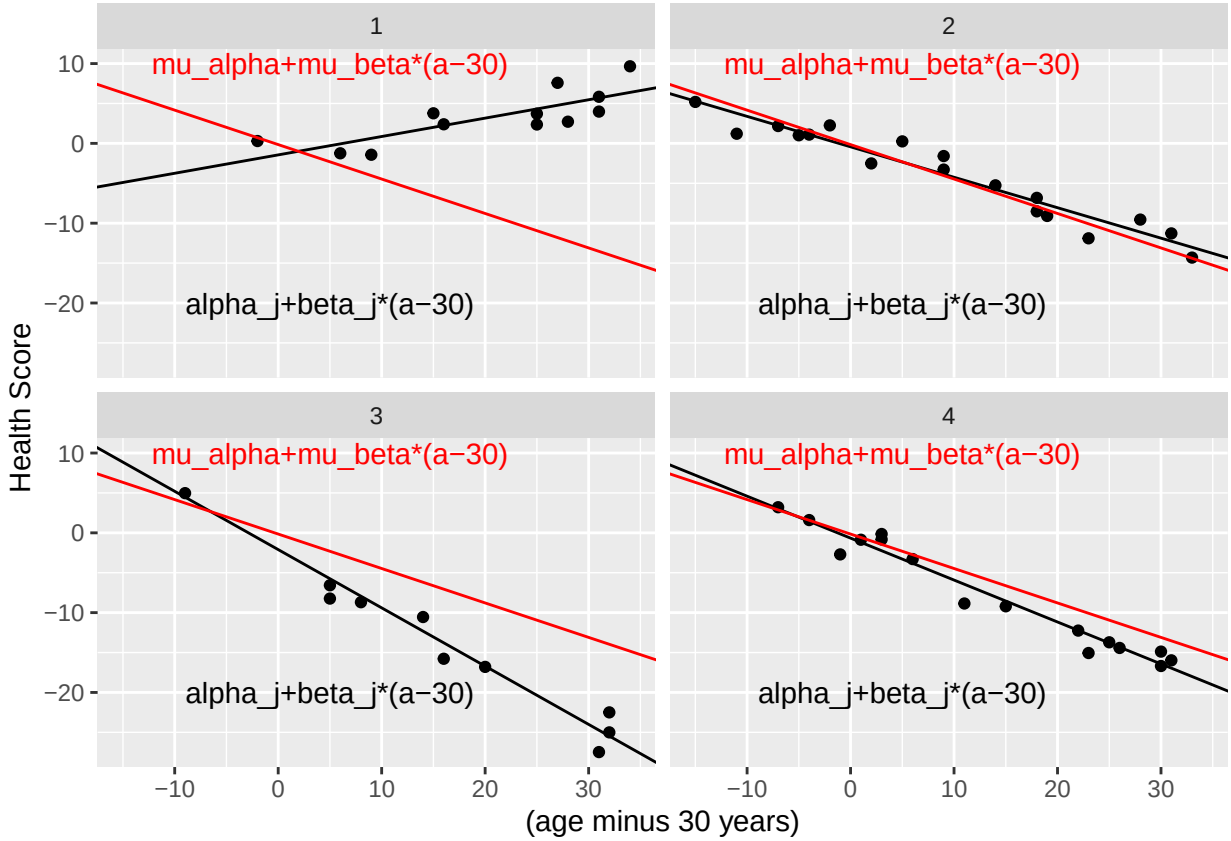




In all our plots we see that the data are ~normally distributed and that the chains converge and mix well for all parameters. Additionally, as was shown previously, all Rhat values are between 1.00 and 1.01 and all the lowest effective sample sizes are over 500 which is adequate for our sample (we hope for ESS values greater than 400, given the 4 chains). Thus, we do not see any issues regarding the diagnostics for our model parameters.

Now, for a visualization of the relation between age and health scores for the first 4 counties:

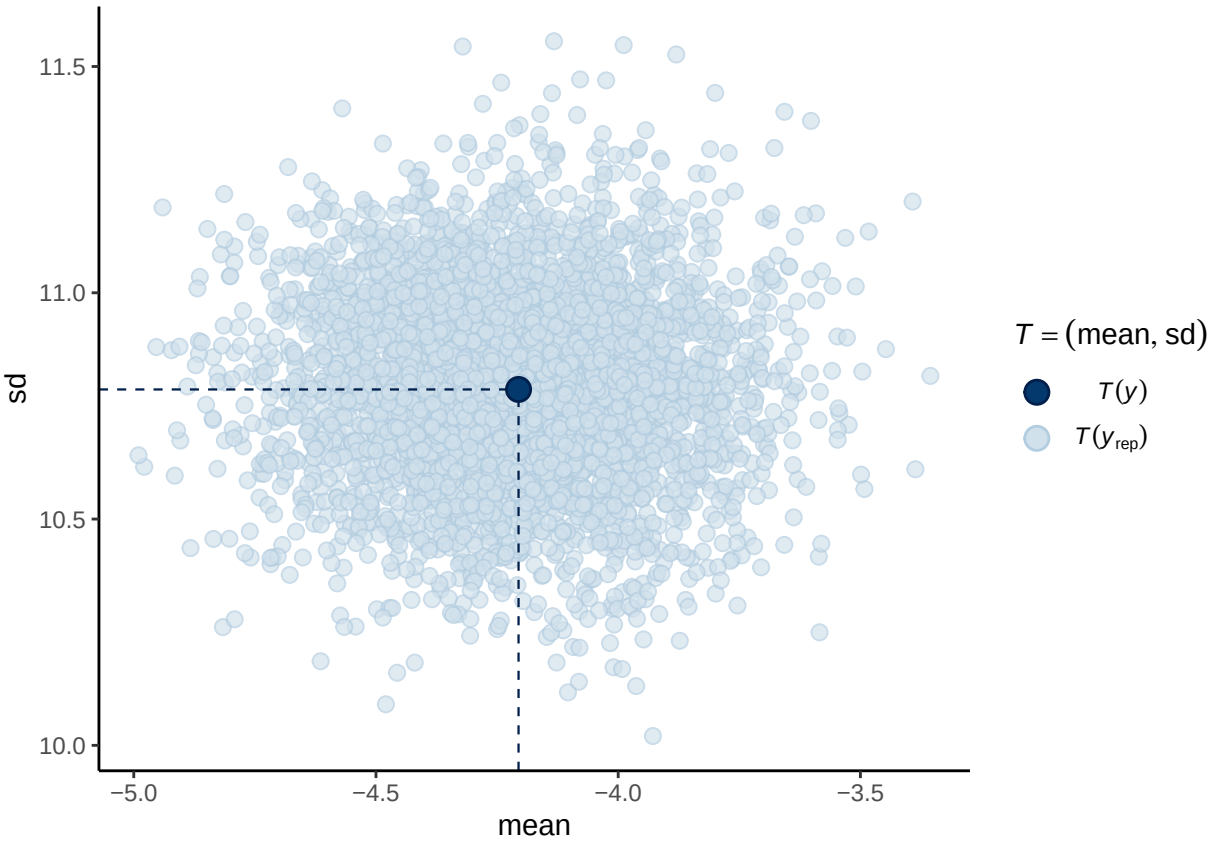
```
coefs <- coef(model2)$county[, 'Estimate', c("Intercept", "ageless30")]
coefs_tibble <- as_tibble(rownames =  "county", coefs) %>%
  rename(slope = ageless30) %>%
  mutate(county = as.numeric(county))
q <- dat_exam2 %>% full_join(coefs_tibble, by = "county") %>%
  filter(county %in% coefs_tibble$county[1:4]) %>%
  ggplot(aes(x = ageless30, y = y)) +
  geom_point() +
  geom_abline(aes(intercept = Intercept, slope = slope)) +
  geom_abline(aes(intercept = fixef(model2)[, "Estimate"][1],
                  slope = fixef(model2)[, "Estimate"][2]), col = "red") +
  xlab("(age minus 30 years)") +
  ylab("Health Score") +
  facet_wrap( ~ county)
q + annotate("text", x = 5, y = -20, label = "alpha_j+beta_j*(a-30)", color = "black") +
  annotate("text", x = 5, y = 10, label = "mu_alpha+mu_beta*(a-30)", color = "red")
```



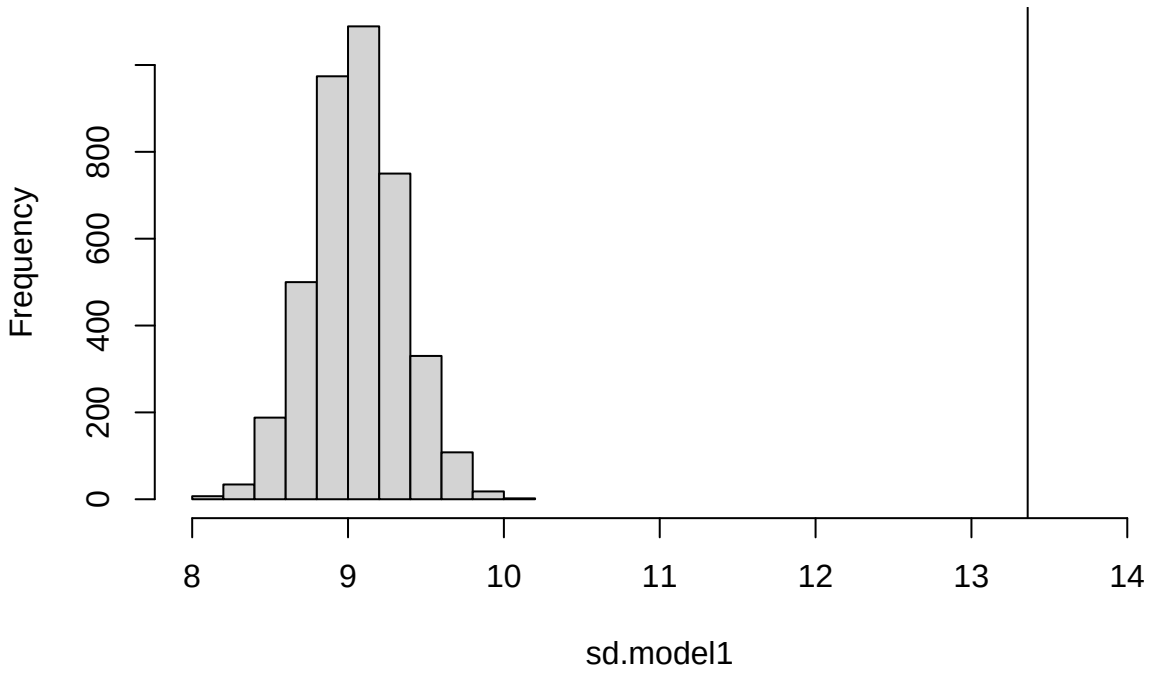
Question 5 (5 points)

I'll check how extreme the variability in health scores at ages 50 or above is compared to the outcomes from replicated data sets.

```
dat_exam2over50 <- dat_exam2 %>% filter(age>=50)
dat_sd <- sd(dat_exam2over50$y)
# SD = 13.36087
dat_var <- dat_sd^2
# Var = 178.5129

# Model 1
# First I just want to compare the overall variability
pp_check(model1, type = "stat_2d", x = "sigma") +
  theme_classic()
```



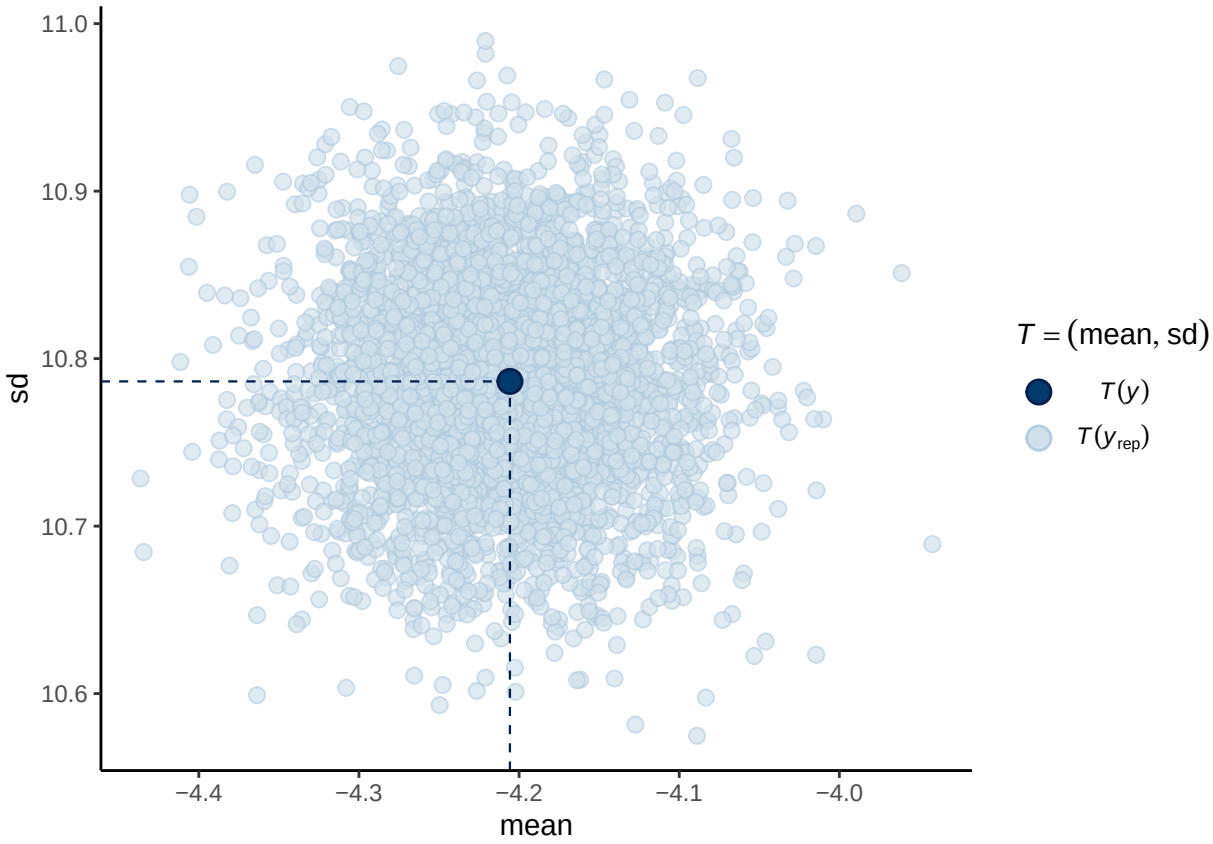

```
# Now, I will look at the variability of health scores in the replicated datasets
set.seed(1234)
samp1 <- posterior_predict(model1, newdata = dat_exam2over50)
sd.model1 <- apply(samp1,1,sd)
hist(sd.model1, xlim = c(8,14))
abline(v = dat_sd)
```



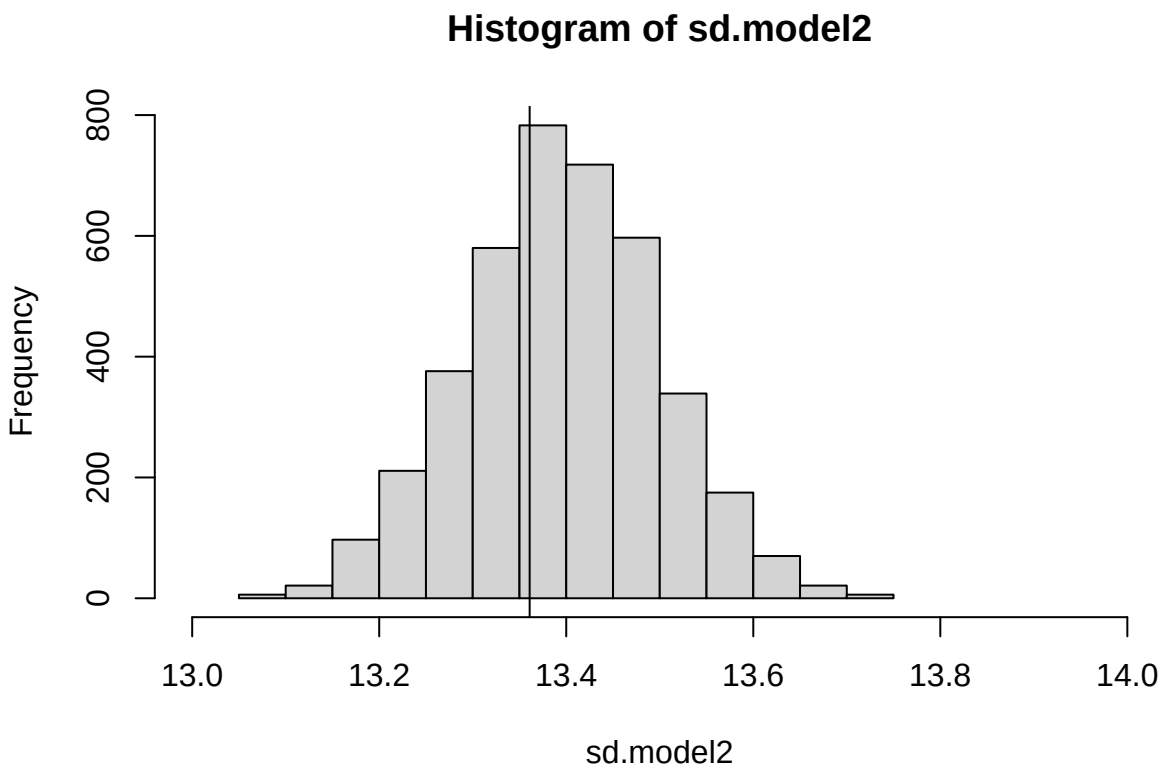

```
# So, our model does not capture the variability in participants over 50 well at all.
# I can capture the probability that our statistic is as variable as the true variability
# I use P(var_rep > dat_sd)
mean(sd.model1 > 13.36087)
```

```
## [1] 0
```

```
# P= 0. So in our replicated datasets, it's impossible to get variability that high.
# This suggests our model is not capturing well the variability in participants over 50.

# Model 2
# First I just want to compare the overall variability
pp_check(model2, type = "stat_2d", x = "sigma") +
  theme_classic()
```




```
# Now, I will look at the variability of health scores in the replicated datasets
set.seed(1234)
samp2 <- posterior_predict(model2, newdata = dat_exam2over50)
sd.model2 <- apply(samp2,1,sd)
hist(sd.model2, xlim = c(13,14))
abline(v = dat_sd)
```




```
# So, our model 2 is MUCH better and accurately captures the variability.
# In the histogram it is around the middle, showing it is frequent in our replicated data.
# I use P(var_rep > dat_sd)
mean(sd.model2 > 13.36087)
```

```
## [1] 0.637
```

```
# P=0.6255
# Thus, I am very happy with how model 2 captured the variability in participants > 50.
```