Lab exercise 4

Chenxi Liu 1010615050

2024-02-08

Question 4

Add in mother's IQ as a covariate and rerun the model. Please mean center the covariate before putting it into the model. Interpret the coefficient on the (centered) mum's IQ.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(rstan)
```

```
## Loading required package: StanHeaders
##
## rstan version 2.32.5 (Stan version 2.32.2)
##
## 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)
## For within-chain threading using ‘reduce_sum()‘ or ‘map_rect()‘ Stan functions,
## change ‘threads_per_chain‘ option:
## rstan_options(threads_per_chain = 1)
##
##
## Attaching package: 'rstan'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(tidybayes)
library(here)
```

```
## here() starts at /Users/dawn/Desktop/uoft/sta2201/HW
```

```r
kidiq <- readRDS("/Users/dawn/Desktop/uoft/sta2201/HW/kidiq.RDS")
kidiq
```

```
## # A tibble: 434 x 4
##    kid_score mom_hs mom_iq mom_age
##        <int>  <dbl>  <dbl>   <int>
##  1        65      1  121.       27
##  2        98      1   89.4      25
##  3        85      1  115.       27
##  4        83      1   99.4      25
##  5       115      1   92.7      27
##  6        98      0  108.       18
##  7        69      1  139.       20
##  8       106      1  125.       23
##  9       102      1   81.6      24
## 10        95      1   95.1      19
## # i 424 more rows
```

```r
y <- kidiq$kid_score
mu0 <- 80
sigma0 <- 10

X <- cbind(kidiq$mom_hs, kidiq$mom_iq - mean(kidiq$mom_iq))
K <- 2

data <- list(y = y, N = length(y),
             X =X, K = K)
fit <- stan(file = "/Users/dawn/Desktop/uoft/sta2201/HW/kids3.stan",
            data = data,
            iter = 1000)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## '/Users/dawn/Desktop/uoft/sta2201/HW/kids3.stan'
```

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

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: 'Apple clang version 14.0.3 (clang-1403.0.22.14.1)'
## using SDK: 'MacOSX13.3.sdk'
## clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Framew
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Core
## namespace Eigen {
```

```
## ^
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/src/Cor
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/StanHeade
## In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/
## /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/Core:96
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.37 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 1: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 1: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 1: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 1: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 1: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 1: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 1: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 1: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 1: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 1: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 1: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.063 seconds (Warm-up)
## Chain 1:                0.045 seconds (Sampling)
## Chain 1:                0.108 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 9e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 2: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 2: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 2: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 2: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 2: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 2: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 2: Iteration: 600 / 1000 [ 60%]  (Sampling)
```

```
## Chain 2: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 2: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 2: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 2: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.08 seconds (Warm-up)
## Chain 2:                0.048 seconds (Sampling)
## Chain 2:                0.128 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 9e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 3: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 3: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 3: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 3: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 3: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 3: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 3: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 3: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 3: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 3: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 3: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.085 seconds (Warm-up)
## Chain 3:                0.048 seconds (Sampling)
## Chain 3:                0.133 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 8e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.08 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 4: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 4: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 4: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 4: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 4: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 4: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 4: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 4: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 4: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 4: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 4: Iteration: 1000 / 1000 [100%]  (Sampling)
```

```
## Chain 4:
## Chain 4:  Elapsed Time: 0.07 seconds (Warm-up)
## Chain 4:                0.049 seconds (Sampling)
## Chain 4:                0.119 seconds (Total)
## Chain 4:
```

```
fit
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##             mean se_mean   sd     2.5%      25%      50%      75%    97.5%
## alpha      82.24    0.05 1.82    78.59    81.04    82.30    83.45    85.72
## beta[1]     5.77    0.06 2.04     1.88     4.40     5.74     7.13     9.90
## beta[2]     0.56    0.00 0.06     0.44     0.52     0.56     0.61     0.68
## sigma      18.09    0.02 0.61    16.90    17.69    18.07    18.48    19.32
## lp__    -1474.39    0.05 1.40 -1477.87 -1475.09 -1474.06 -1473.35 -1472.65
##         n_eff Rhat
## alpha    1147    1
## beta[1]  1151    1
## beta[2]  1354    1
## sigma    1517    1
## lp__      917    1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb  8 13:09:09 2024.
## 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).
```

beta[2]: The coefficient for the centered mom_iq variable. The mean value is 0.57, with a standard error of 0.00, and a standard deviation of 0.06. The 95% credible interval for the coefficient is between 0.45 and 0.69. The mean of coefficient of mom's iq is positive which shows a positive association with kids iq, this means that for a one-unit increase in the mum's IQ from its mean value, the kid's score is expected to increase by 0.57 units, holding all other variables constant.

Question 5

Confirm the results from Stan agree with `lm()`

```
lm_model <- lm(y ~ X[,1] + X[,2])
summary(lm_model)
```

```
##
## Call:
## lm(formula = y ~ X[, 1] + X[, 2])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.873 -12.663   2.404  11.356  49.545
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 82.12214    1.94370  42.250  < 2e-16 ***
## X[, 1]       5.95012    2.21181   2.690  0.00742 **
## X[, 2]       0.56391    0.06057   9.309  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.14 on 431 degrees of freedom
## Multiple R-squared:  0.2141, Adjusted R-squared:  0.2105
## F-statistic: 58.72 on 2 and 431 DF,  p-value: < 2.2e-16
```

The coefficients obtained from the Bayesian model using Stan are indeed consistent with the coefficients obtained from the frequentist linear regression model using lm().

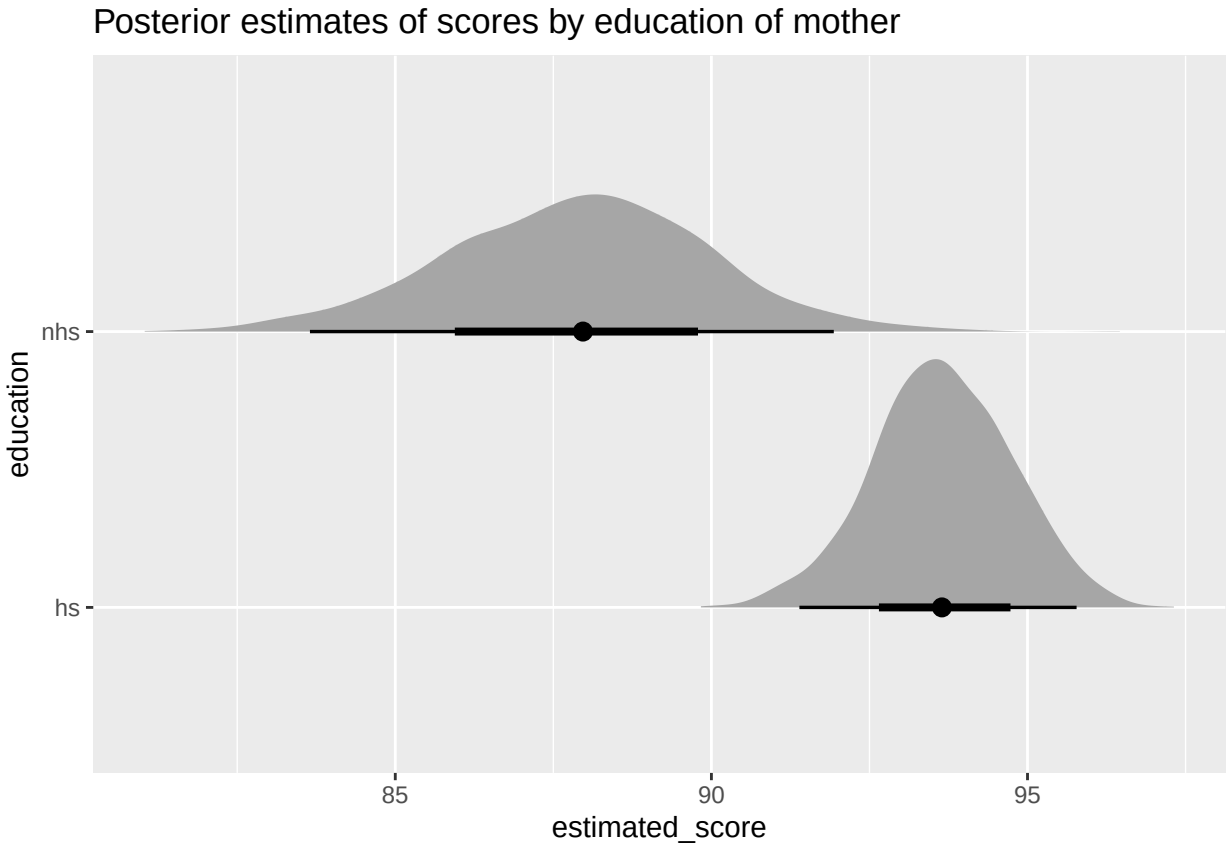
Question 6

Plot the posterior estimates of scores by education of mother for mothers who have an IQ of 110.

```
center_IQ <- 110-mean(kidiq$mom_iq) #which is 10

fit %>%
  gather_draws(alpha, beta[condition]) %>%
  group_by(.draw) %>%
  mutate(.value = ifelse(!is.na(condition)&condition==2, .value*center_IQ,  .value)) %>%
  summarise(nhs = sum(.value[is.na(condition)|condition==2]),
            hs = sum(.value)) %>%
  pivot_longer(nhs:hs, names_to = "education", values_to = "estimated_score") %>%
  ggplot(aes(y = education, x = estimated_score)) +
  stat_halfeyeh() +
  ggtitle("Posterior estimates of scores by education of mother")
```

```
## Warning: 'stat_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```



Question 7

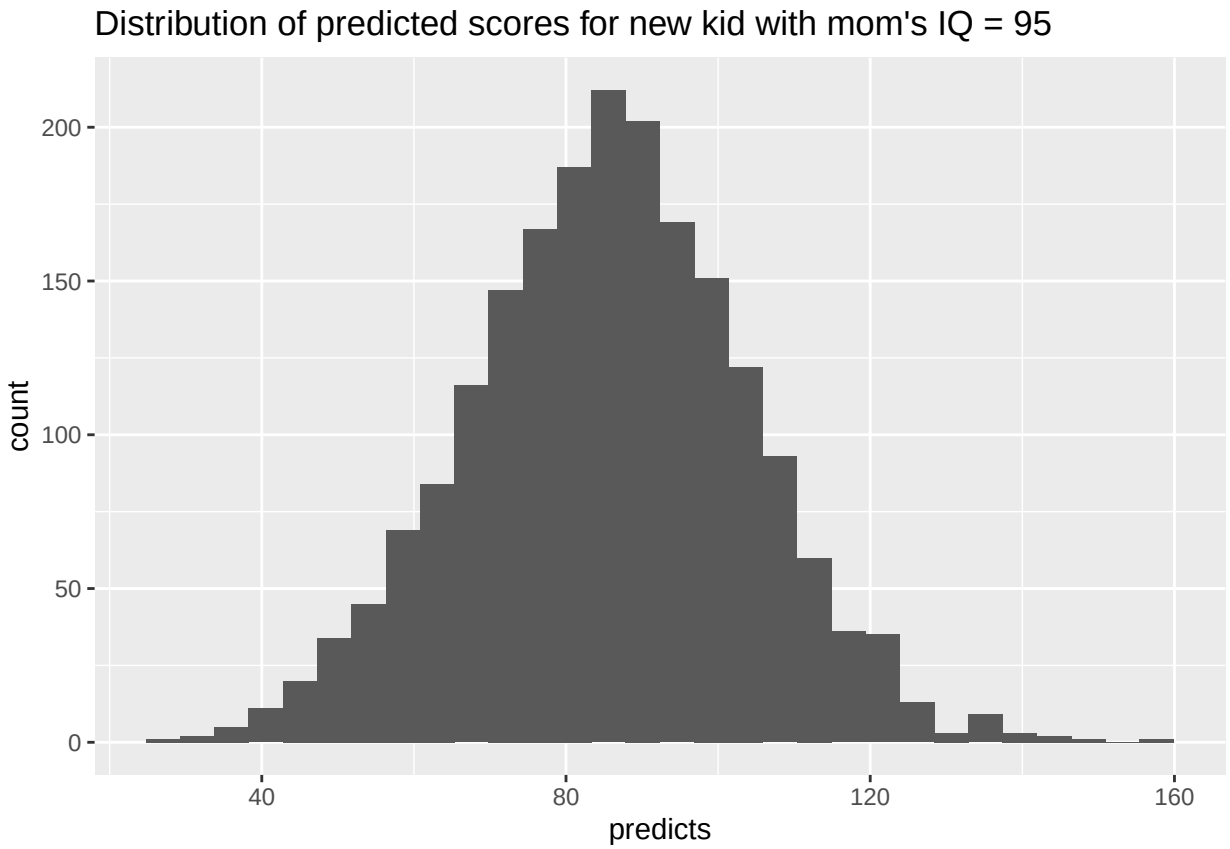
Generate and plot (as a histogram) samples from the posterior predictive distribution for a new kid with a mother who graduated high school and has an IQ of 95.

```
center_IQ <- 95-mean(kidiq$mom_iq)

samples <- extract(fit)
mu <- samples[["alpha"]] + samples[["beta"]][,1] + samples[["beta"]][,2]*center_IQ
sigmas <- samples[["sigma"]]

predicts <- tibble(predicts = rnorm(length(sigmas), mean = mu, sd = sigmas))
ggplot(predicts,aes(predicts)) + geom_histogram() + ggtitle("Distribution of predicted scores for new ki
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Distribution of predicted scores for new kid with mom's IQ = 95
count
200
150
100
50
0
40
80
120
160
predicts