Week 5: Bayesian linear regression and introduction to Stan

Qiaoyu (Terence) Liang

Introduction

Today we will be starting off using Stan, looking at the kid's test score data set (available in resources for the Gelman Hill textbook).

```
library(tidyverse)
library(rstan)
library(tidybayes)
library(here)
```

The data look like this:

```
kidiq <- read_rds("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
## # ... with 424 more rows
```

As well as the kid's test scores, we have a binary variable indicating whether or not the mother completed high school, the mother's IQ and age.

Descriptives

Question 1

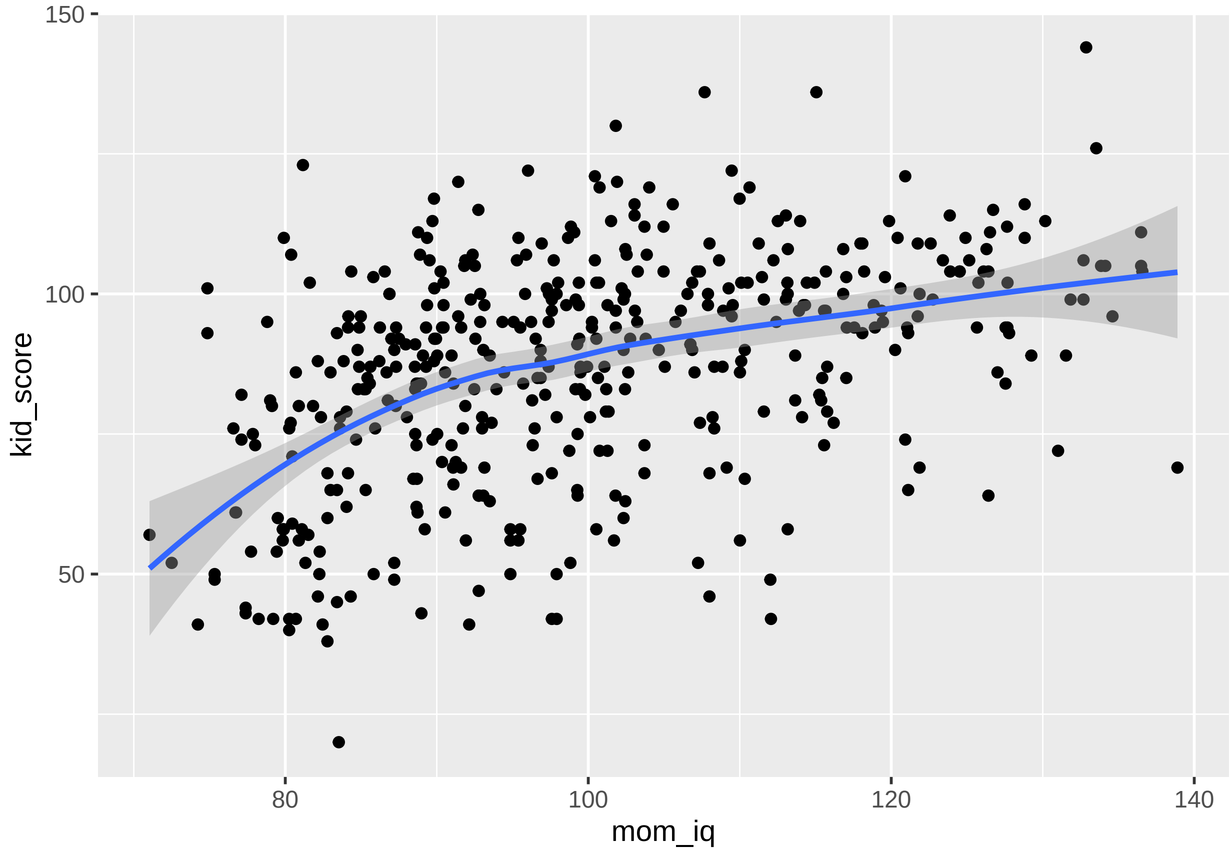
Use plots or tables to show three interesting observations about the data. Remember:

- Explain what your graph/ tables show
- Choose a graph type that's appropriate to the data type

Graph I:

This graph shows that there seems to be a slightly positive relationship between kid's test scores and mom's IQ since we can see kid's test scores increases as mom's IQ increases.

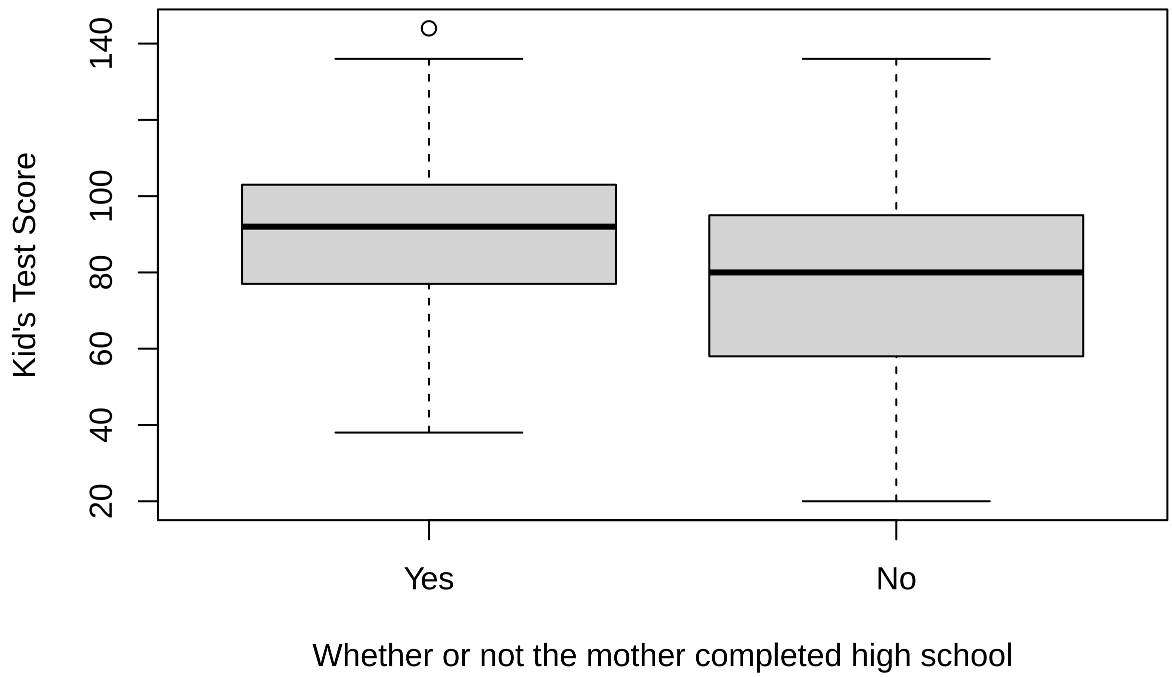
```
ggplot(data=kidiq, aes(x=mom_iq,y=kid_score)) +
  geom_point() +
  geom_smooth()
```



Graph II:

This boxplot shows that the kid's score for the kids whose moms completed high school tends to be higher than those kids whose mothers did not complete high school.

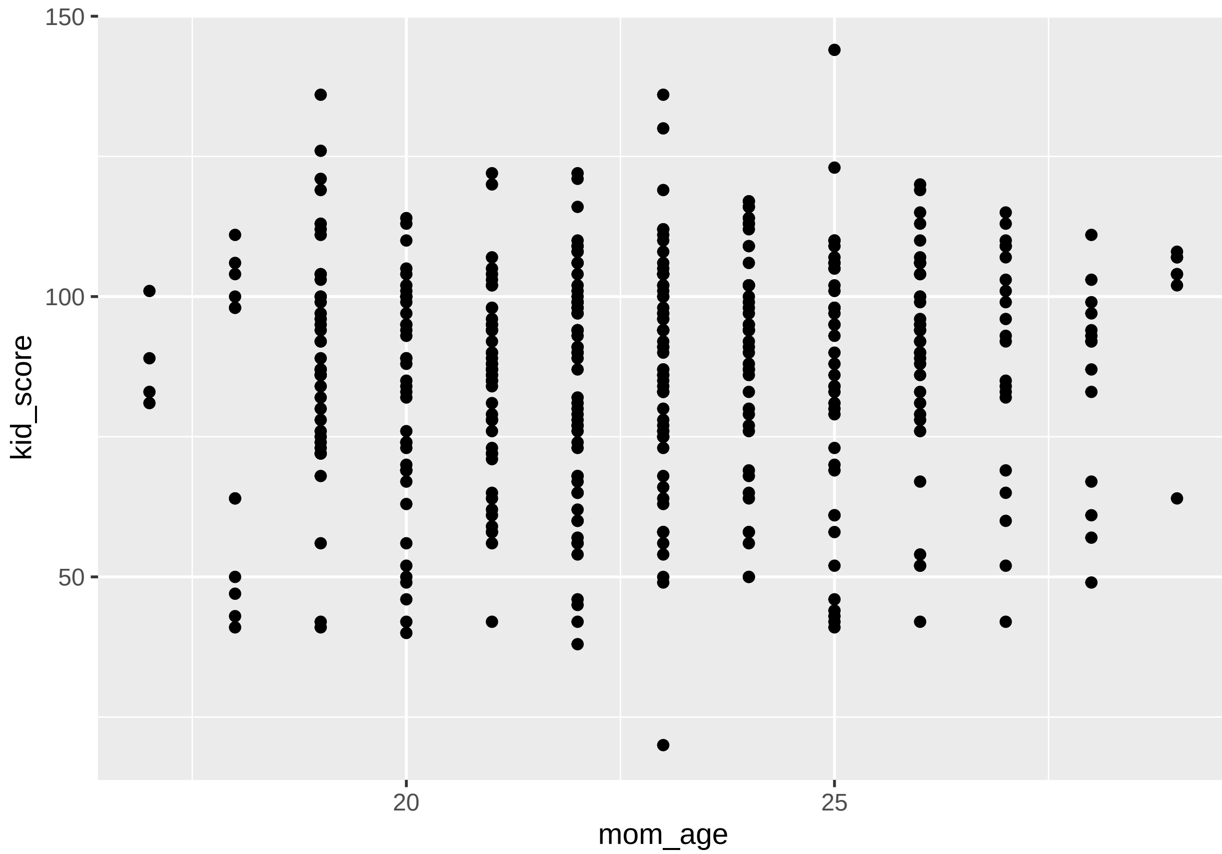
```
boxplot(kidiq[kidiq$mom_hs == 1,]$kid_score,
        kidiq[kidiq$mom_hs == 0,]$kid_score,
        names = c("Yes", "No"),
        xlab = "Whether or not the mother completed high school",
        ylab = "Kid's Test Score")
```



Graph III:

Based on this scatter plot, it seems hard to find a clear pattern between the kid's test score and the mom's age.

```
ggplot(data=kidiq)+
  geom_point(aes(x=mom_age, y=kid_score))
```



Estimating mean, no covariates

In class we were trying to estimate the mean and standard deviation of the kid's test scores. The `kids2.stan` file contains a Stan model to do this. If you look at it, you will notice the first `data` chunk lists some inputs that we have to define: the outcome variable `y`, number of observations `N`, and the mean and standard deviation of the prior on `mu`. Let's define all these values in a `data` list.

```
y <- kidiq$kid_score
mu0 <- 80
sigma0 <- 10
# named list to input for stan function
data <- list(y = y,
             N = length(y),
             mu0 = mu0,
             sigma0 = sigma0)
```

Now we can run the model:

```
set.seed(2201)
fit <- stan(file = "kids2.stan",
            data = data,
            chains = 3,
            iter = 500)
```

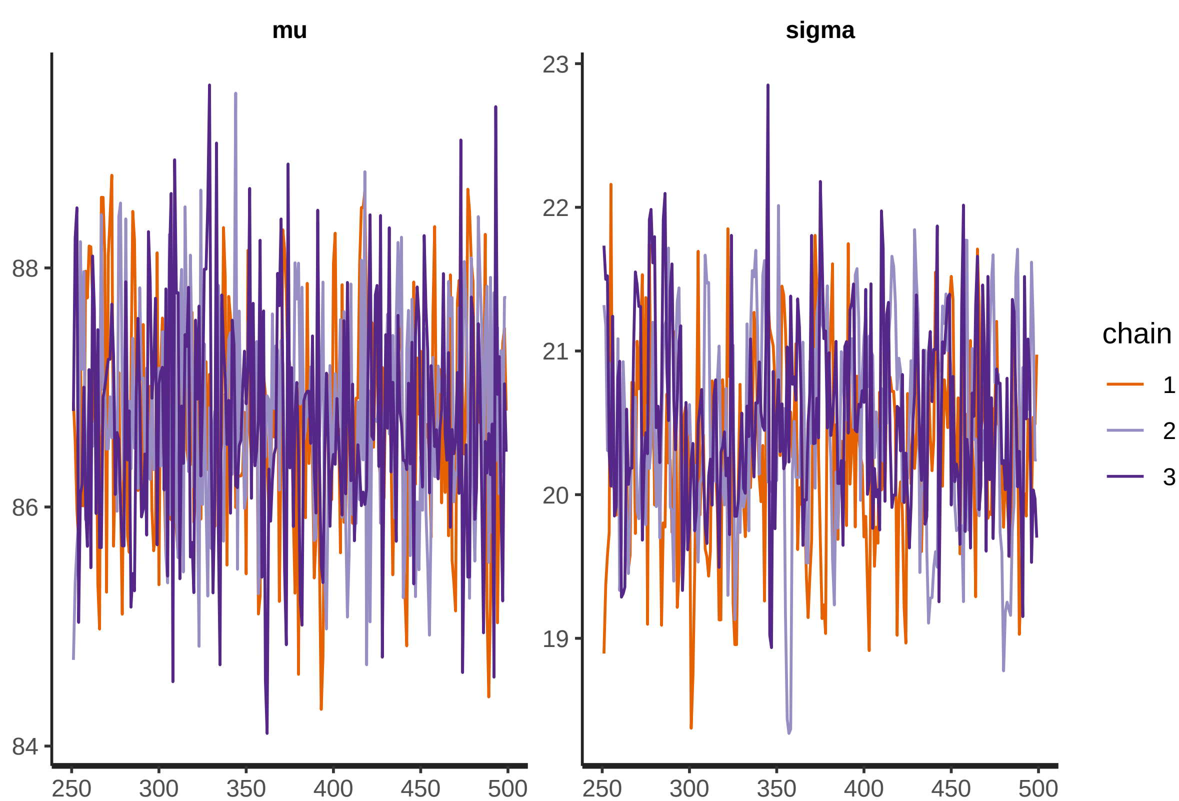
Look at the summary

```
fit
```

```
## Inference for Stan model: anon_model.
## 3 chains, each with iter=500; warmup=250; thin=1;
## post-warmup draws per chain=250, total post-warmup draws=750.
##
##            mean se_mean   sd     2.5%      25%      50%      75%    97.5% n_eff
## mu        86.73    0.04 0.92    84.98    86.08    86.70    87.36    88.51   687
## sigma     20.43    0.04 0.69    19.12    19.96    20.40    20.91    21.75   332
## lp__   -1525.72    0.04 0.91 -1527.90 -1526.17 -1525.46 -1525.05 -1524.78   431
##        Rhat
## mu     1.00
## sigma 1.01
## lp__   1.00
##
## Samples were drawn using NUTS(diag_e) at Mon Feb 13 17:55:31 2023.
## 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).
```

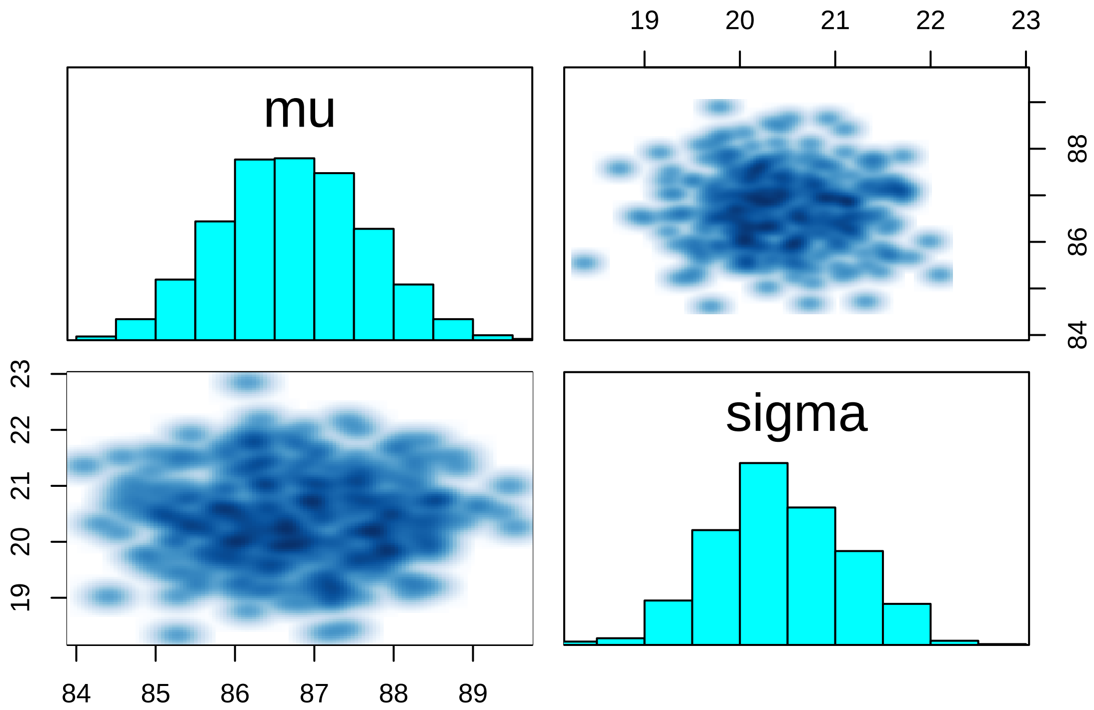
Traceplot

```
traceplot(fit)
```

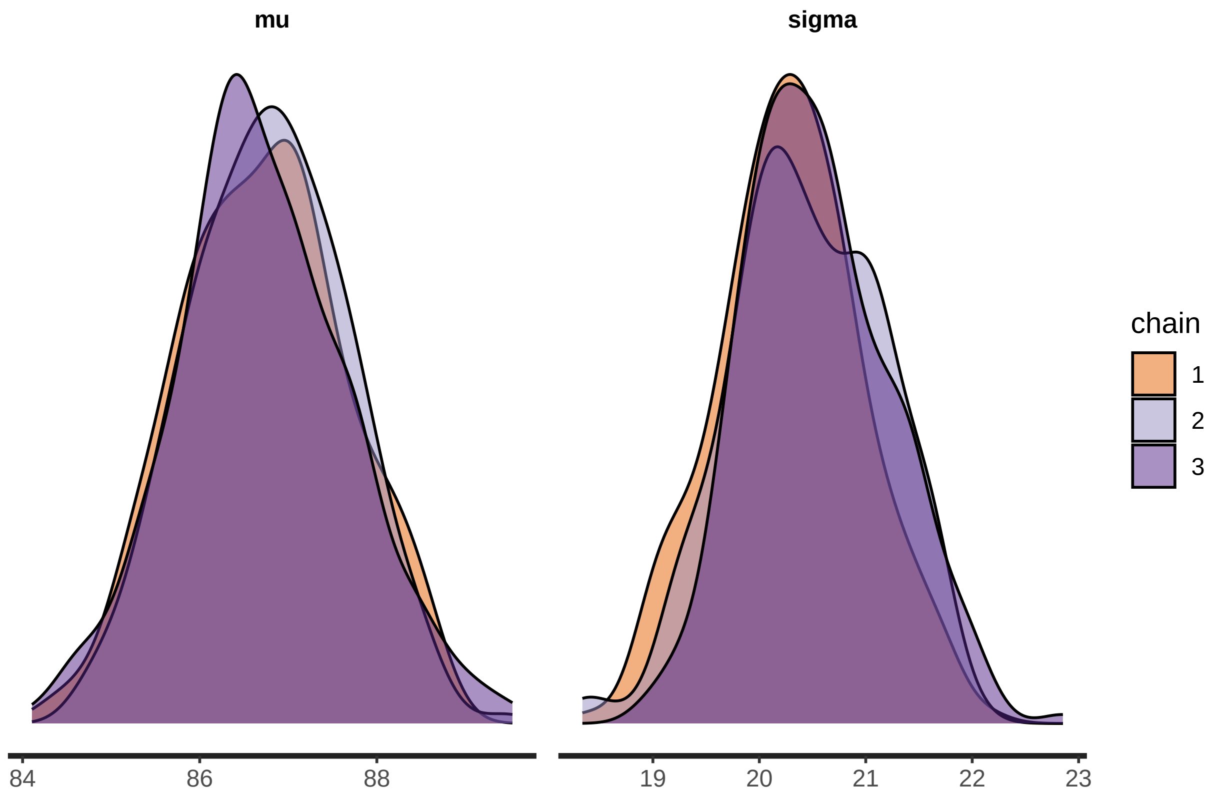


All looks fine.

```r
pairs(fit, pars = c("mu", "sigma"))
```




```r
stan_dens(fit, separate_chains = TRUE)
```



Understanding output

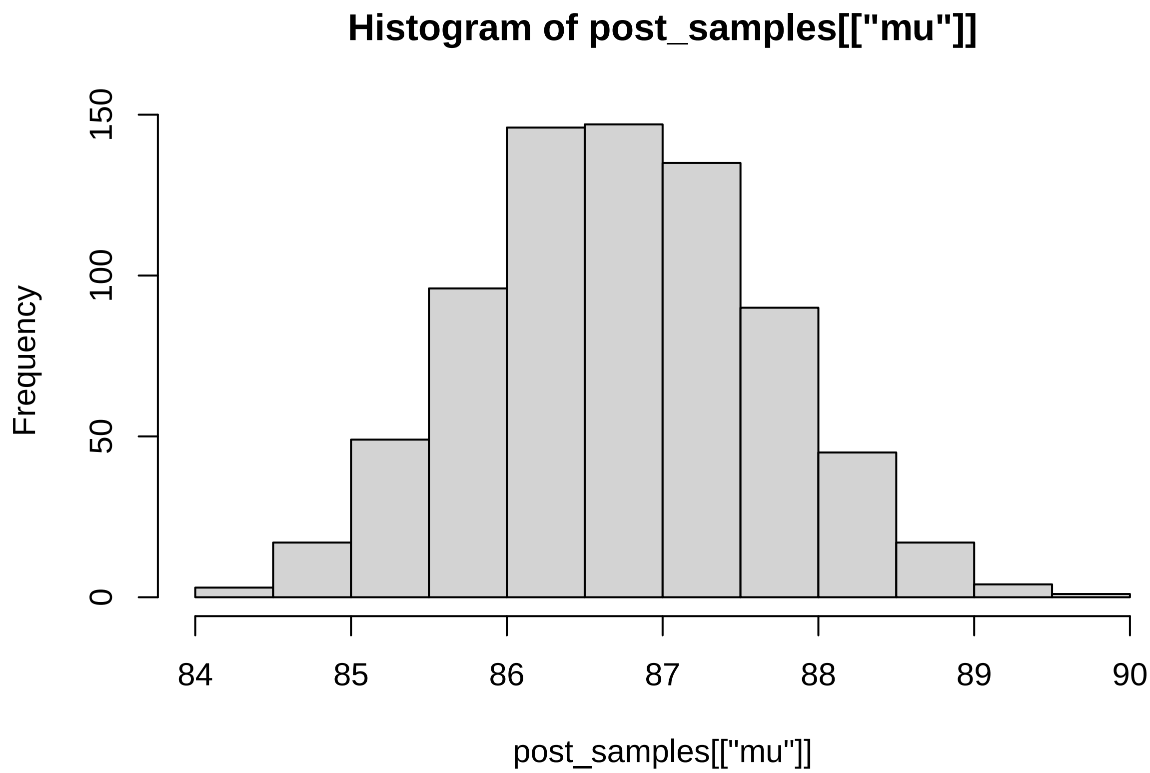
What does the model actually give us? A number of samples from the posteriors. To see this, we can use `extract` to get the samples.

```
post_samples <- extract(fit)
head(post_samples[["mu"]])
```

```
## [1] 87.59275 86.88078 86.07369 86.04646 88.05493 87.42127
```

This is a list, and in this case, each element of the list has 4000 samples. E.g. quickly plot a histogram of mu

```
hist(post_samples[["mu"]])
```




```
median(post_samples[["mu"]])
```

```
## [1] 86.70497
```

```
# 95% bayesian credible interval
quantile(post_samples[["mu"]], 0.025)
```

```
##     2.5%
## 84.97897
```

```
quantile(post_samples[["mu"]], 0.975)
```

```
##   97.5%
## 88.5071
```

Plot estimates

There are a bunch of packages, built-in functions that let you plot the estimates from the model, and I encourage you to explore these options (particularly in `bayesplot`, which we will most likely be using later on). I like using the `tidybayes` package, which allows us to easily get the posterior samples in a tidy format (e.g. using gather draws to get in long format). Once we have that, it's easy to just pipe and do ggplots as usual.

Get the posterior samples for mu and sigma in long format:

```r
dsamples <- fit  |>
  gather_draws(mu, sigma) # gather = long format
dsamples
```

```
## # A tibble: 1,500 x 5
## # Groups:   .variable [2]
##    .chain .iteration .draw .variable .value
##     <int>      <int> <int> <chr>      <dbl>
##  1      1          1     1 mu          86.9
##  2      1          2     2 mu          86.6
##  3      1          3     3 mu          86.0
##  4      1          4     4 mu          85.7
##  5      1          5     5 mu          87.4
##  6      1          6     6 mu          86.0
##  7      1          7     7 mu          86.0
##  8      1          8     8 mu          88.0
##  9      1          9     9 mu          87.7
## 10      1         10    10 mu          88.2
## # ... with 1,490 more rows
```

```r
# wide format
fit  |>  spread_draws(mu, sigma)
```

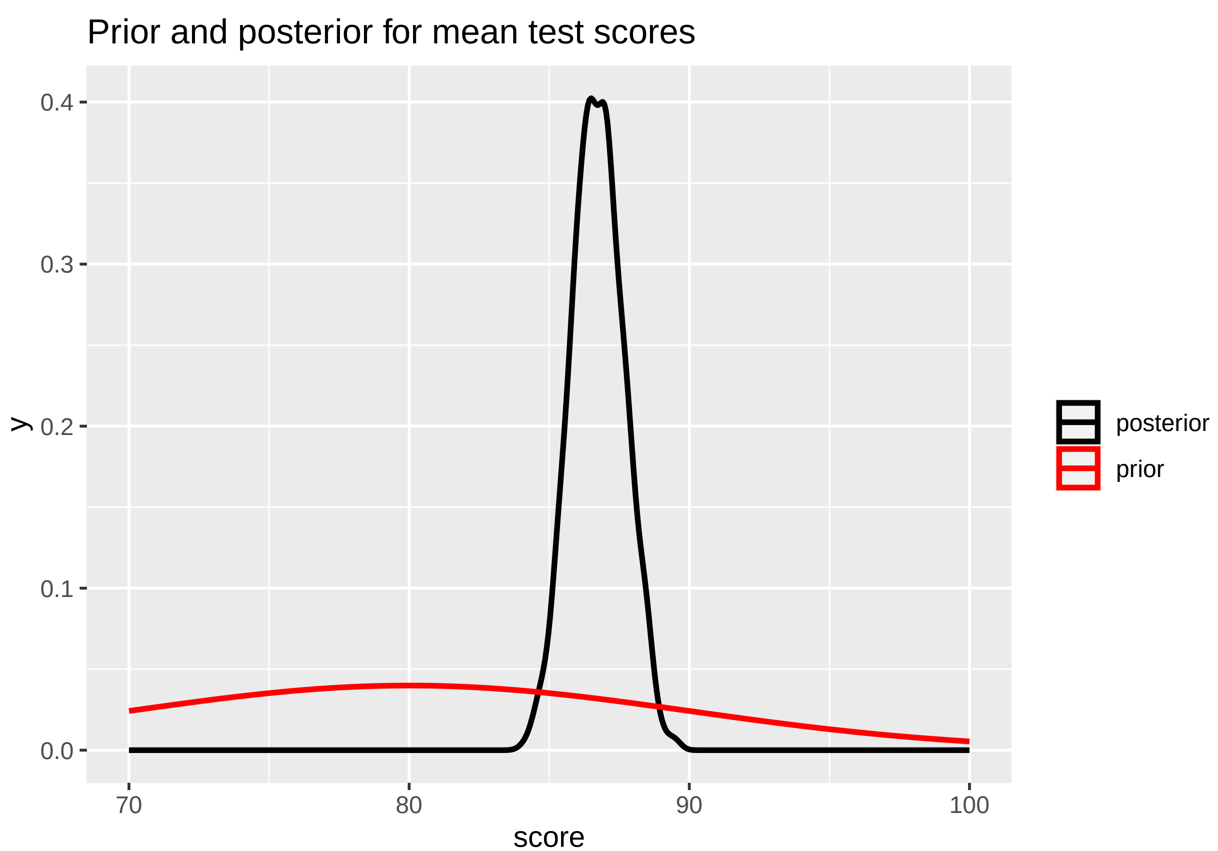
```
## # A tibble: 750 x 5
##    .chain .iteration .draw    mu sigma
##     <int>      <int> <int> <dbl> <dbl>
##  1      1          1     1  86.9  18.9
##  2      1          2     2  86.6  19.4
##  3      1          3     3  86.0  19.6
##  4      1          4     4  85.7  19.7
##  5      1          5     5  87.4  22.2
##  6      1          6     6  86.0  20.1
##  7      1          7     7  86.0  20.1
##  8      1          8     8  88.0  19.9
##  9      1          9     9  87.7  20.4
## 10      1         10    10  88.2  19.8
## # ... with 740 more rows
```

```r
# quickly calculate the quantiles using
dsamples |>
  median_qi(.width = 0.8)
```

```
## # A tibble: 2 x 7
##   .variable .value .lower .upper .width .point .interval
##   <chr>      <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>
## 1 mu          86.7   85.5   87.9    0.8 median qi
## 2 sigma       20.4   19.6   21.4    0.8 median qi
```

Let's plot the density of the posterior samples for mu and add in the prior distribution

```
dsamples |>
  filter(.variable == "mu") |>
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) +
  xlim(c(70, 100)) +
  stat_function(fun = dnorm,
        args = list(mean = mu0,
                    sd = sigma0),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for mean test scores") +
  xlab("score")
```



Question 2

Change the prior to be much more informative (by changing the standard deviation to be 0.1). Rerun the model. Do the estimates change? Plot the prior and posterior densities.

```
set.seed(2201)
sigmaQ2 <- 0.1

dataQ2 <- list(y = y,
             N = length(y),
             mu0 = mu0,
             sigma0 = sigmaQ2)
```

```r
fitQ2 <- stan(file = "kids2.stan",
            data = dataQ2,
            chains = 3,
            iter = 500)
```

```r
summary(fit)[["summary"]]
```

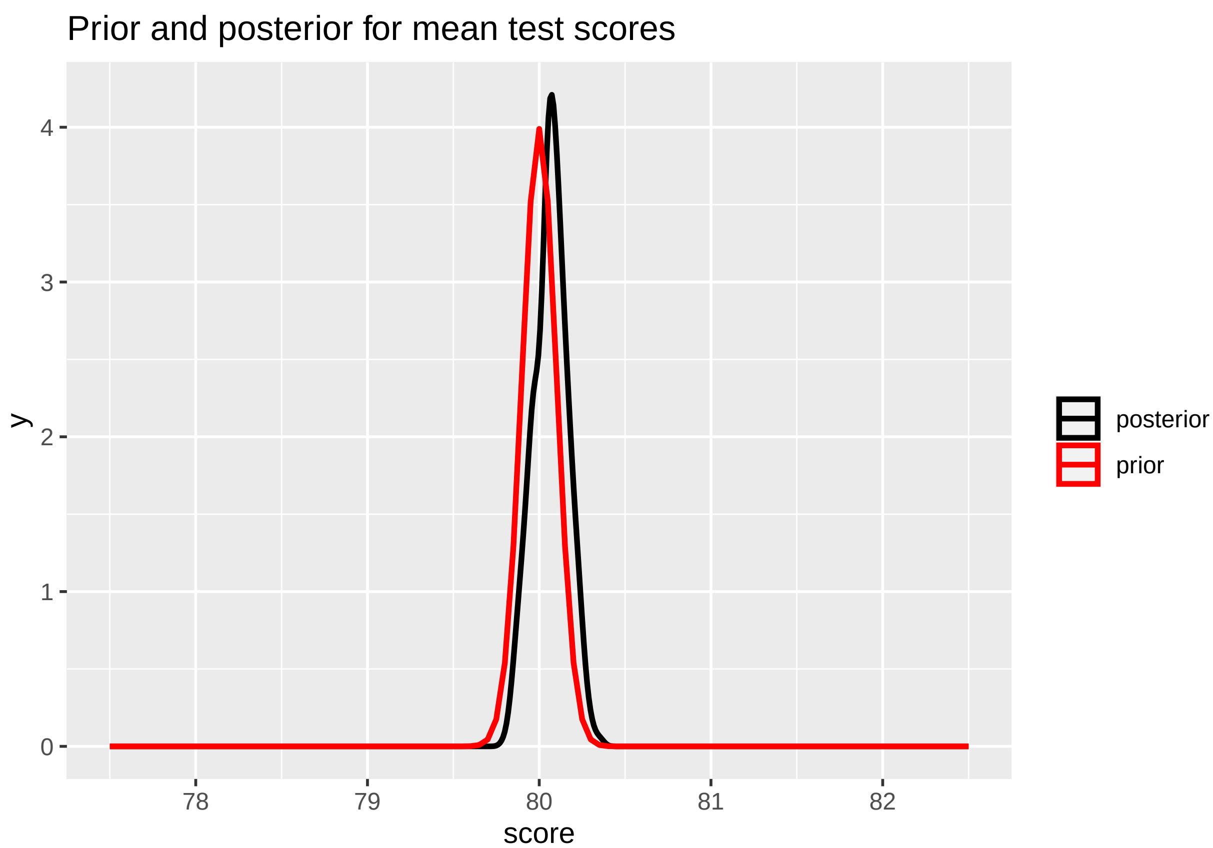
```
##               mean    se_mean        sd        2.5%         25%         50%
## mu        86.73473 0.03529160 0.9248996    84.97897    86.07700    86.70497
## sigma     20.42810 0.03812542 0.6949959    19.12198    19.95744    20.39745
## lp__   -1525.72215 0.04375584 0.9086878 -1527.90172 -1526.16823 -1525.45528
##                75%       97.5%    n_eff      Rhat
## mu        87.36371    88.50710 686.8258 0.9980855
## sigma     20.91177    21.75383 332.3038 1.0121093
## lp__   -1525.05461 -1524.77858 431.2781 1.0032148
```

```r
summary(fitQ2)[["summary"]]
```

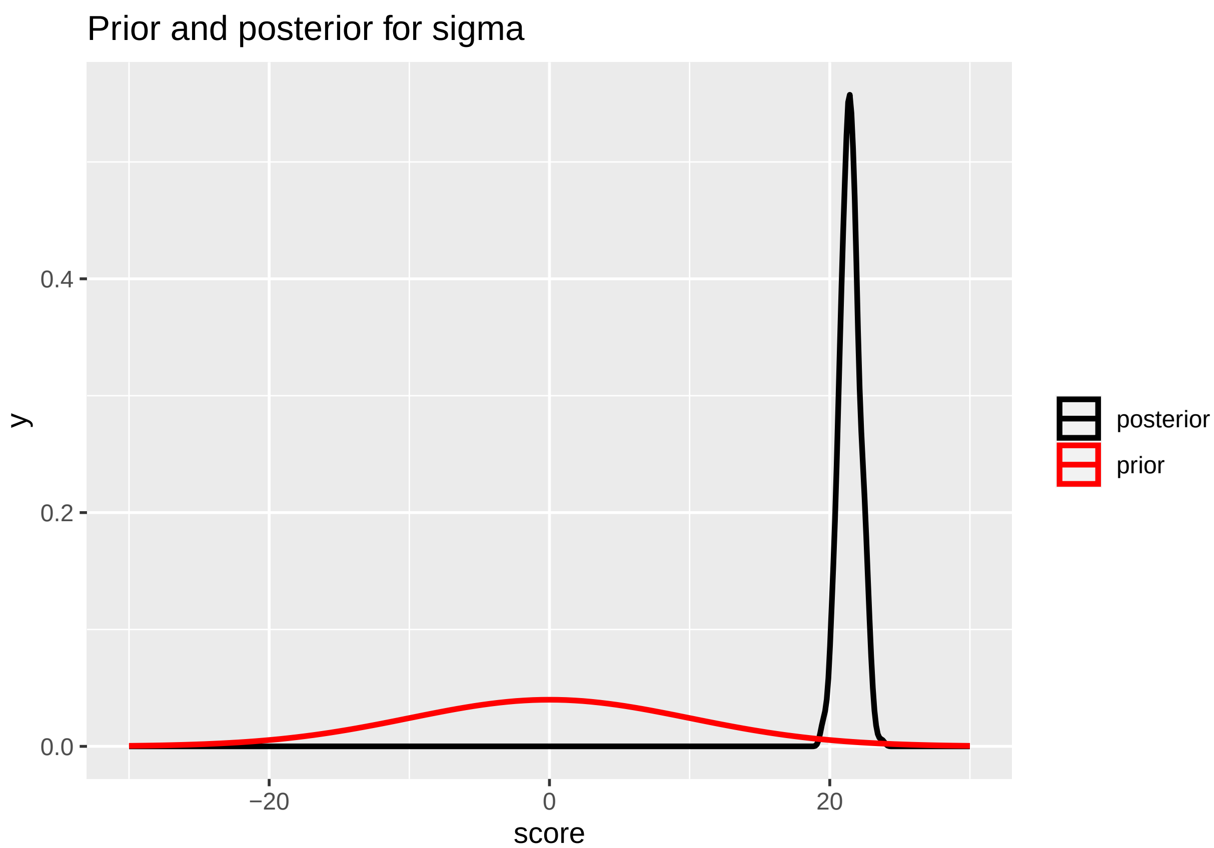
```
##               mean     se_mean         sd        2.5%         25%         50%
## mu        80.06479 0.004288688 0.09935999    79.87247    79.99557    80.06703
## sigma     21.43593 0.029341579 0.72229406    20.09633    20.94260    21.42293
## lp__   -1548.36688 0.053381309 0.94482297 -1550.93929 -1548.79645 -1548.08073
##                75%       97.5%    n_eff      Rhat
## mu        80.13114    80.25097 536.7526 1.0006517
## sigma     21.89191    22.81496 605.9839 0.9998215
## lp__   -1547.66389 -1547.40049 313.2726 0.9991593
```

By comparison between fit and fitQ2, we find the estimates change. Specifically, the mu estimate in fitQ2 decreases and gets closer to the mu0 which is 80. The associated standard error of the mu estimate in fitQ2 also decreases. For the other estimates, they change but not by a large margin.

```r
dsamplesQ2 <- fitQ2 %>%
  gather_draws(mu, sigma)
dsamplesQ2 %>%
  filter(.variable == "mu") %>%
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) +
  xlim(c(77.5, 82.5)) +
  stat_function(fun = dnorm,
        args = list(mean = mu0,
                    sd = sigmaQ2),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for mean test scores") +
  xlab("score")
```




```
dsamplesQ2 |>
  filter(.variable == "sigma") |>
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) + xlim(c(-30,30)) +
  stat_function(fun = dnorm,
                args = list(mean = 0,
                            sd = 10),
                aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for sigma") +
  xlab("score")
```



Adding covariates

Now let's see how kid's test scores are related to mother's education. We want to run the simple linear regression

$$Score = \alpha + \beta X$$

where $X = 1$ if the mother finished high school and zero otherwise.

`kid3.stan` has the stan model to do this. Notice now we have some inputs related to the design matrix X and the number of covariates (in this case, it's just 1).

Let's get the data we need and run the model.

```
set.seed(2201)
X <- as.matrix(kidiq$mom_hs, ncol = 1) # force this to be a matrix
K <- 1

data <- list(y = y, N = length(y),
             X =X, K = K)
fit2 <- stan(file = "kids3.stan",
            data = data,
            iter = 1000)
```

Question 3

a) Confirm that the estimates of the intercept and slope are comparable to results from `lm()`

```
summary(fit2)$summary[1:2,]
```

```
##              mean    se_mean       sd      2.5%      25%      50%     75%
## alpha    77.84325 0.08316184 2.116583 73.686743 76.44277 77.84921 79.2687
## beta[1]  11.33610 0.09500792 2.359045  6.736076  9.71153 11.34076 12.9986
##             97.5%    n_eff     Rhat
## alpha    81.95057 647.7722 1.004515
## beta[1]  15.97211 616.5280 1.005083
```

```
summary(lm(kidiq$kid_score ~ kidiq$mom_hs))
```

```
##
## Call:
## lm(formula = kidiq$kid_score ~ kidiq$mom_hs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57.55 -13.32   2.68  14.68  58.45
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    77.548      2.059  37.670  < 2e-16 ***
## kidiq$mom_hs   11.771      2.322   5.069 5.96e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.85 on 432 degrees of freedom
## Multiple R-squared:  0.05613,    Adjusted R-squared:  0.05394
## F-statistic: 25.69 on 1 and 432 DF,  p-value: 5.957e-07
```

```
# Stan estimate of the intercept and slope
summary(fit2)$summary[1:2,1]
```

```
##    alpha  beta[1]
## 77.84325 11.33610
```

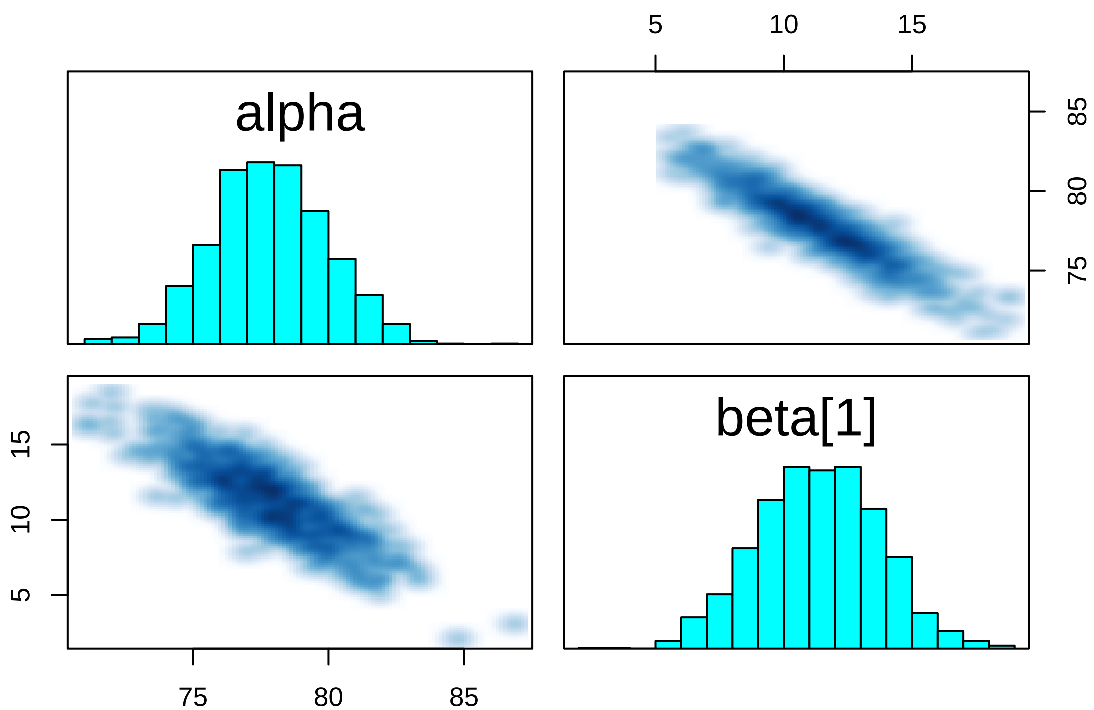
```
# lm estimate of the intercept and slope
summary(lm(kidiq$kid_score ~ kidiq$mom_hs))$coefficients[,"Estimate"]
```

```
##  (Intercept) kidiq$mom_hs
##     77.54839     11.77126
```

From the summaries of the fits above, we can confirm that the estimates of the intercept and slope are comparable for both fits.

b) Do a `pairs` plot to investigate the joint sample distributions of the slope and intercept. Comment briefly on what you see. Is this potentially a problem?

```
pairs(fit2, pars = c("alpha", "beta"))
```

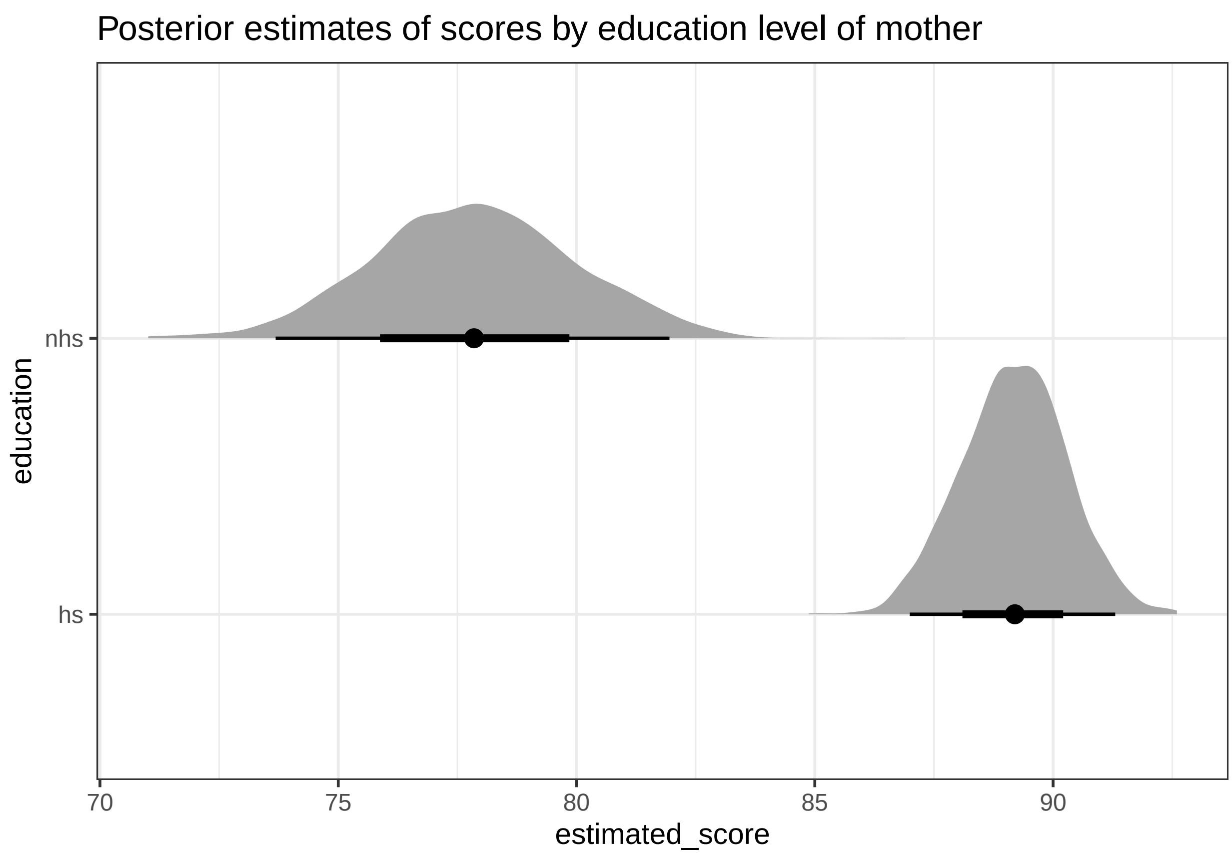


From the pairs plot, we can see that there is a strong negative relationship between the intercept and the slope which means changes in the slope would induce the opposite change in the intercept. This is potentially a problem since this would bring difficulties to interpret the intercepts. At the same time, the correlation between the intercept and the slope seems to be close to -1 which makes it harder to sample. In this situation, centering may be a choice to tackle the problem.

Plotting results

It might be nice to plot the posterior samples of the estimates for the non-high-school and high-school mothered kids. Here's some code that does this: notice the `beta[condition]` syntax. Also notice I'm using `spread_draws`, because it's easier to calculate the estimated effects in wide format

```
fit2 |>
  spread_draws(alpha, beta[k], sigma) |>
     mutate(nhs = alpha, # no high school is just the intercept
          hs = alpha + beta) |>
  select(nhs, hs) |>
  pivot_longer(nhs:hs, names_to = "education", values_to = "estimated_score") |>
  ggplot(aes(y = education, x = estimated_score)) +
  stat_halfeye() +
  theme_bw() +
  ggtitle("Posterior estimates of scores by education level of mother")
```



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.

```
set.seed(2201)
X <- cbind(kidiq$mom_hs, kidiq$mom_iq - mean(kidiq$mom_iq))
K <- 2

dataQ4 <- list(y = y, N = length(y),
             X =X, K = K)
fitQ4 <- stan(file = "kids3.stan",
            data = dataQ4,
            iter = 1000)
```

```
summary(fitQ4)$summary[1:3,]
```

```
##               mean     se_mean         sd       2.5%        25%        50%
## alpha    82.366633 0.061018851 1.94944698 78.6475446 81.0178733 82.2670895
## beta[1]   5.638320 0.068290843 2.16645142  1.3143624  4.1237902  5.7353308
## beta[2]   0.567819 0.001729209 0.06091704  0.4546417  0.5245463  0.5672048
##                75%      97.5%    n_eff     Rhat
## alpha    83.6798341 86.3096766 1020.692 1.000535
## beta[1]   7.1313593  9.7249860 1006.405 1.000877
## beta[2]   0.6095783  0.6894157 1241.031 1.000524
```

Interpretation: For every one unit increase in the centered mom's IQ, the expected kid's test score increases around 0.57, with all other variables being the same (i.e. the high school status remains the same).

Question 5

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

```
# Result from Stan
summary(fitQ4)$summary[1:3,]
```

```
##              mean     se_mean         sd       2.5%        25%        50%
## alpha   82.366633 0.061018851 1.94944698 78.6475446 81.0178733 82.2670895
## beta[1]  5.638320 0.068290843 2.16645142  1.3143624  4.1237902  5.7353308
## beta[2]  0.567819 0.001729209 0.06091704  0.4546417  0.5245463  0.5672048
##                75%      97.5%    n_eff     Rhat
## alpha   83.6798341 86.3096766 1020.692 1.000535
## beta[1]  7.1313593  9.7249860 1006.405 1.000877
## beta[2]  0.6095783  0.6894157 1241.031 1.000524
```

```
# Result from lm
kidiq$mom_iq_c <- kidiq$mom_iq - mean(kidiq$mom_iq)
summary(lm(kidiq$kid_score ~ kidiq$mom_hs + kidiq$mom_iq_c))
```

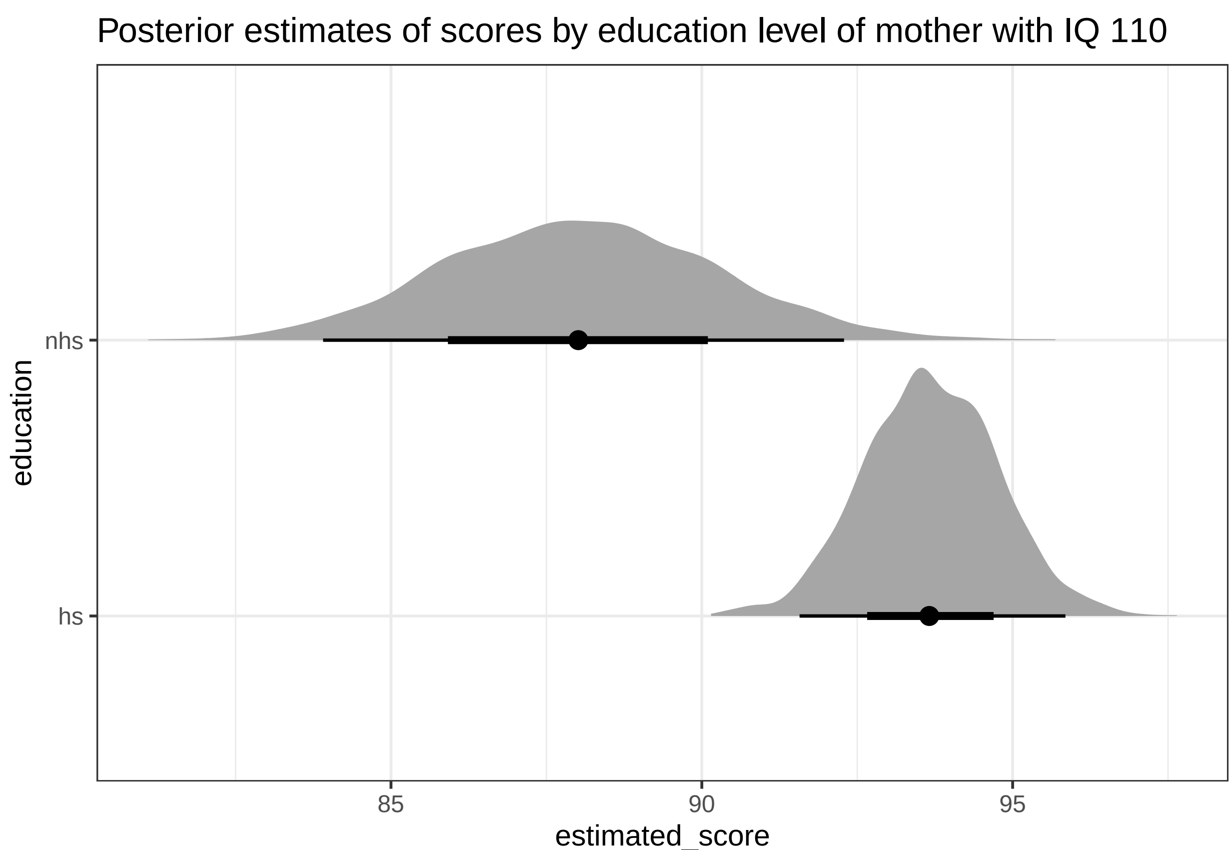
```
##
## Call:
## lm(formula = kidiq$kid_score ~ kidiq$mom_hs + kidiq$mom_iq_c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.873 -12.663   2.404  11.356  49.545
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    82.12214    1.94370  42.250  < 2e-16 ***
## kidiq$mom_hs    5.95012    2.21181   2.690  0.00742 **
## kidiq$mom_iq_c  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
```

Based on the above summaries, we can confirm the results from Stan are similar with the results from `lm()`.

Question 6

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

```r
fitQ4 %>%
  spread_draws(alpha, beta[k], sigma)  %>%
  pivot_wider(names_from = k, names_prefix = "beta", values_from = beta)   %>%
  mutate(nhs = alpha + beta2 * (110-mean(kidiq$mom_iq)),
         hs = alpha + beta1 + beta2 * (110-mean(kidiq$mom_iq)))  %>%
  select(nhs, hs)  %>%
  pivot_longer(nhs:hs, names_to = "education", values_to = "estimated_score")  %>%
  ggplot(aes(y = education, x = estimated_score)) +
  stat_halfeye() +
  theme_bw() +
  ggtitle("Posterior estimates of scores by education level of mother with IQ 110")
```



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.

```r
set.seed(2201)
post_samplesQ7 <- extract(fitQ4)
alpha <- post_samplesQ7$alpha
beta1 <- post_samplesQ7$beta[,1]
beta2 <- post_samplesQ7$beta[,2]
sigma <- post_samplesQ7$sigma
```

```
lin_pred <- alpha + beta1 + (95- mean(kidiq$mom_iq)) * beta2
y_new <- rnorm(n = length(sigma), mean = lin_pred, sd = sigma)

hist(y_new)
```