Week 5 Lab Stat2201

Yeonjoon Choi

2023-02-10

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:

```
## # 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
```

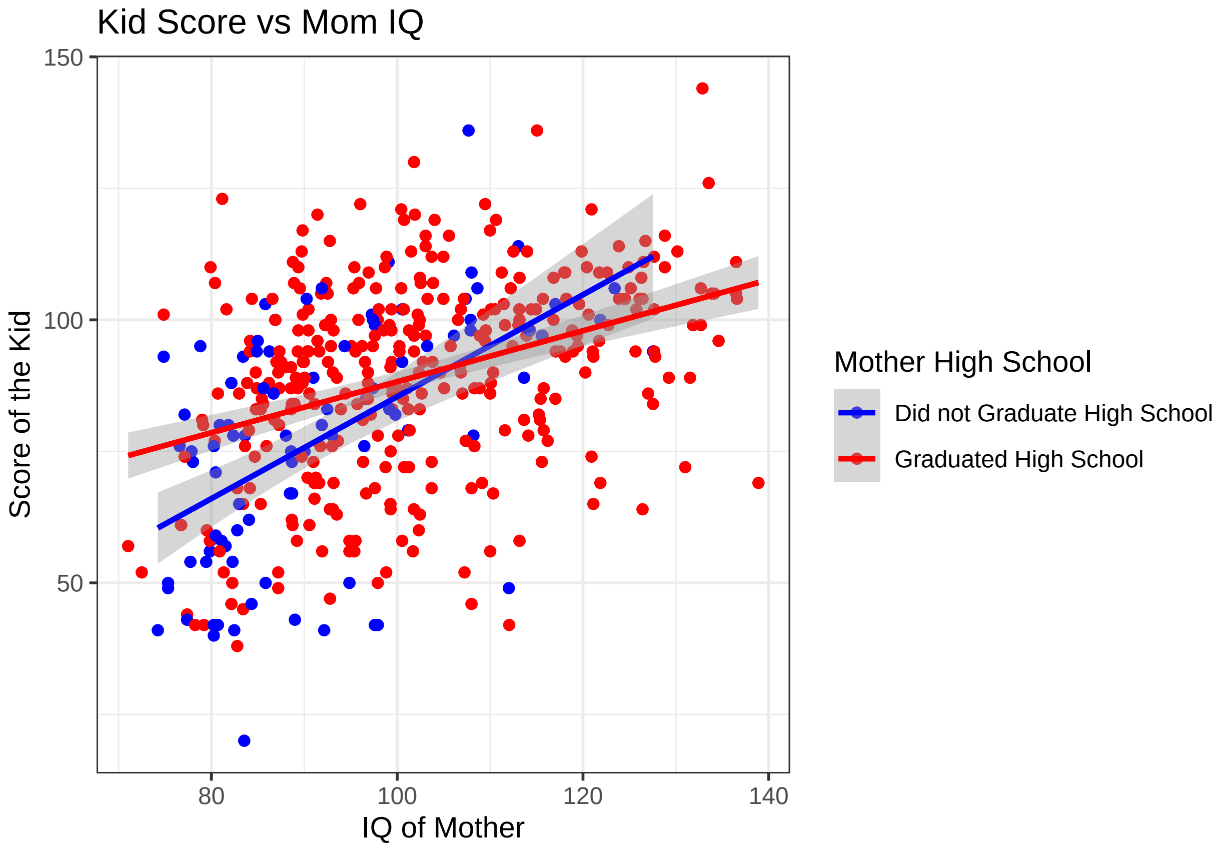
As well as the kid's test scores, we have a binary variable indicating whether or not the mother completed high school, the mother's IQ and age.

Descriptives

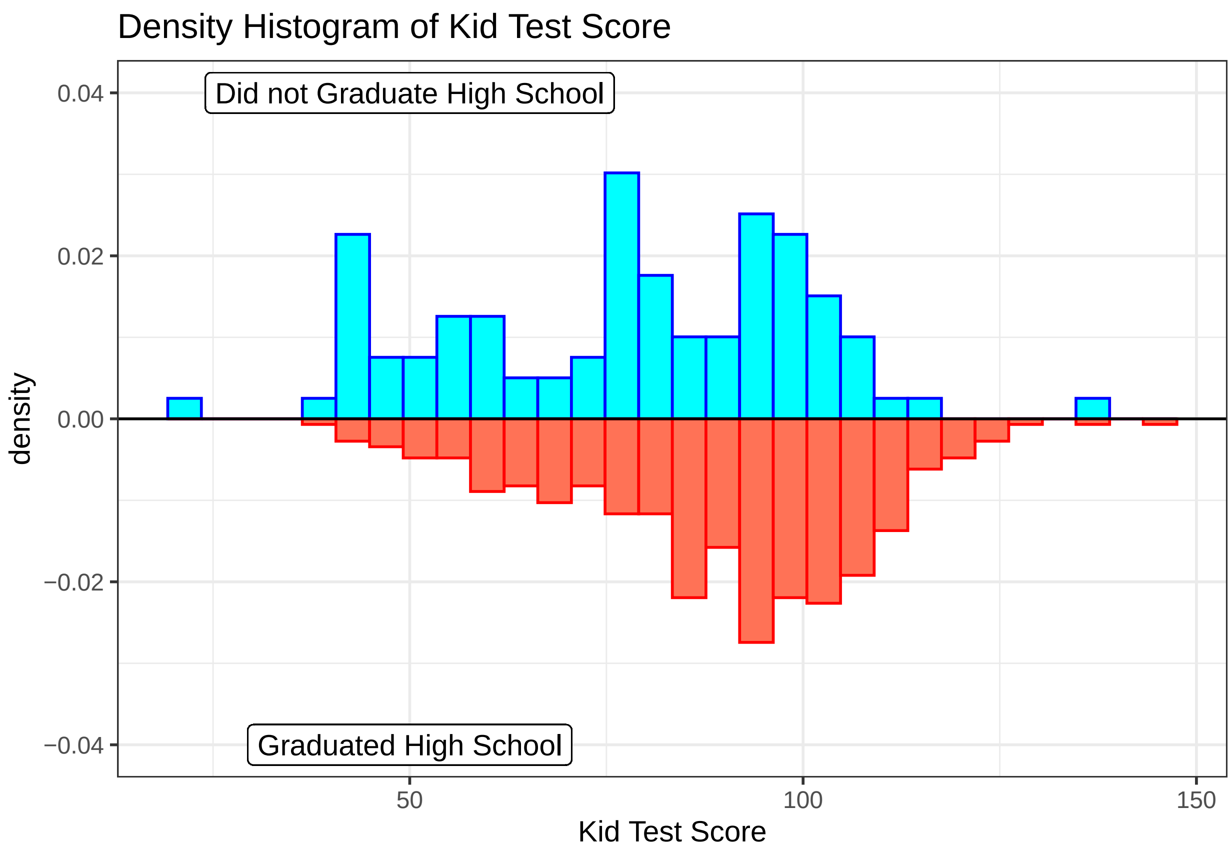
Question 1

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

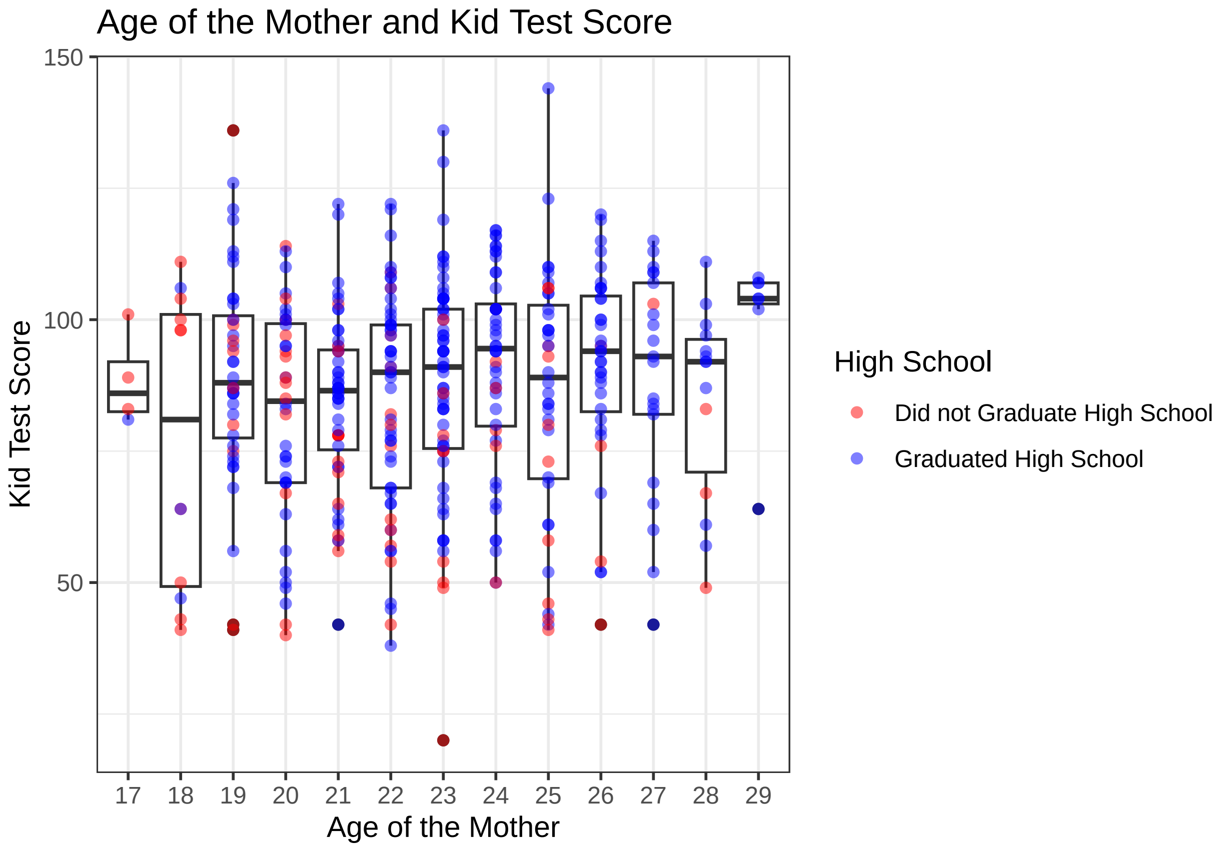
- Explain what your graph/ tables show
- Choose a graph type that's appropriate to the data type



We see in general that higher IQ score from the mother corresponded to higher test score from the kid. We see that IQ of the mother had a bigger positive effect on the score of the children for mothers who did not graduate from high school.



Comparing the two density histograms for did not graduate high school group and graduated high school group, we see that lower test score had higher density in did not graduate high school group (Compare the two group for test score of less than 50, and the difference is clear). And the graduated high school group has higher density for test scores greater than 100. In general, children with mother who graduated high school has higher test scores.



From the boxplot, there does not seem to be any relationship between age of the mother and the test score of the kids.

We may expect younger mother may be more likely to have not graduated from high school since unexpected teen pregnancy can hinder education. However, from the plot, such a pattern does not emerge. Majority of mothers aged 17 and 18 did not graduate from high school, but the sample size for that age group is small to make any conclusion. # 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.

```r
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. Please change the file directory as needed. You can see where my stan file is stored on my Github.

```
fit = stan(file = here("kids2.stan"),
            data = data,
            chains = 3,
            iter = 500, refresh = 0)
```

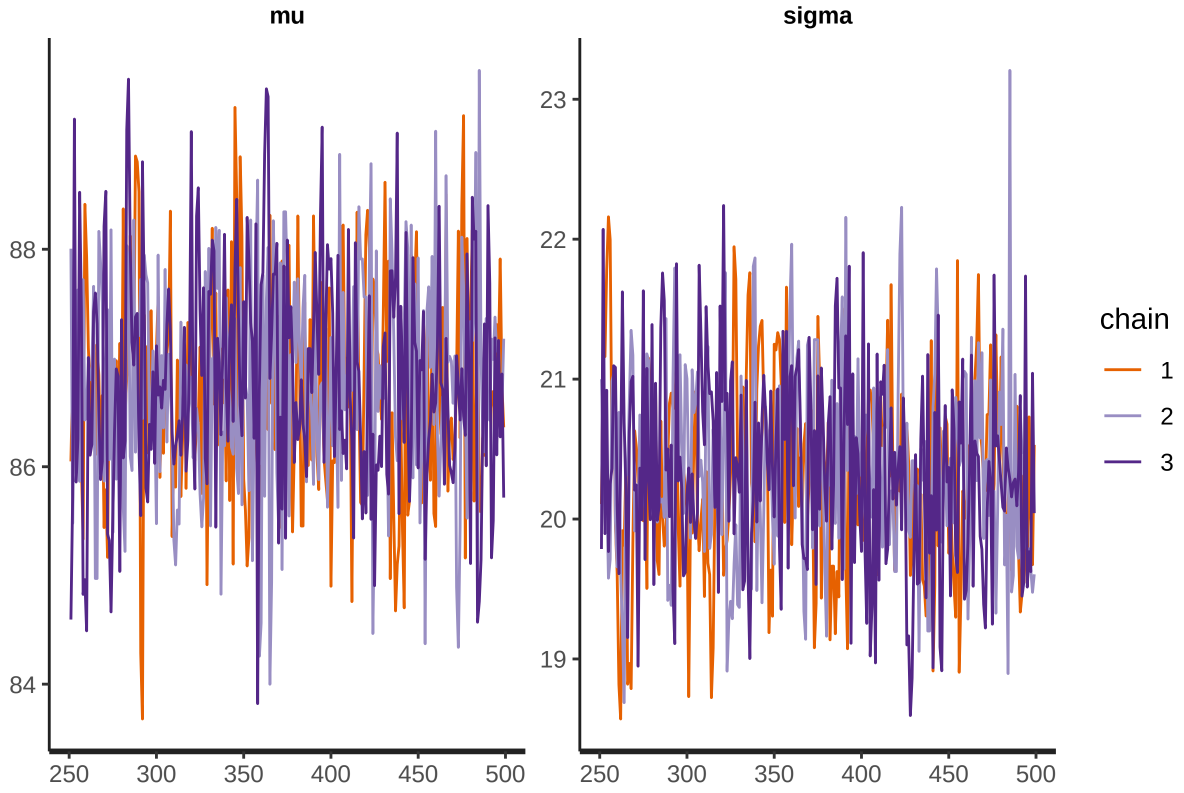
Look at the summary

```
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        86.75    0.04 1.00    84.83    86.09    86.71    87.44    88.79   670
## sigma     20.34    0.04 0.69    19.02    19.85    20.32    20.82    21.78   377
## lp__   -1525.79    0.05 1.02 -1528.24 -1526.23 -1525.49 -1525.06 -1524.77   380
##        Rhat
## mu        1
## sigma     1
## lp__      1
##
## Samples were drawn using NUTS(diag_e) at Sat Feb 11 16:57:33 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).
```

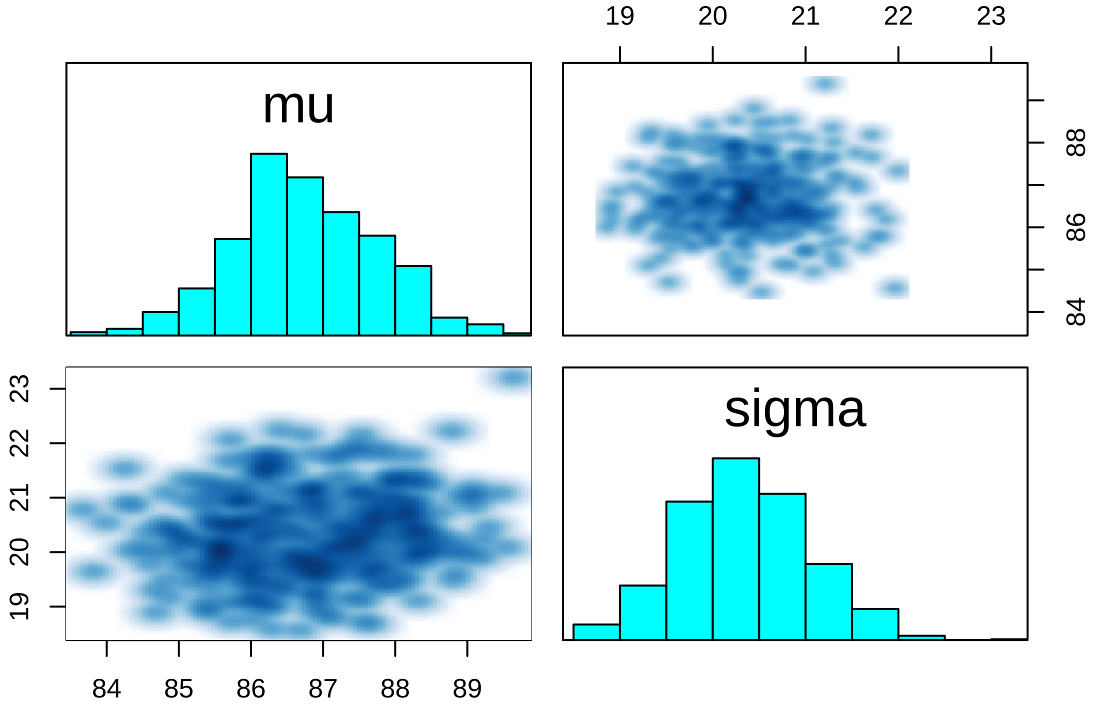
Traceplot

```
traceplot(fit)
```

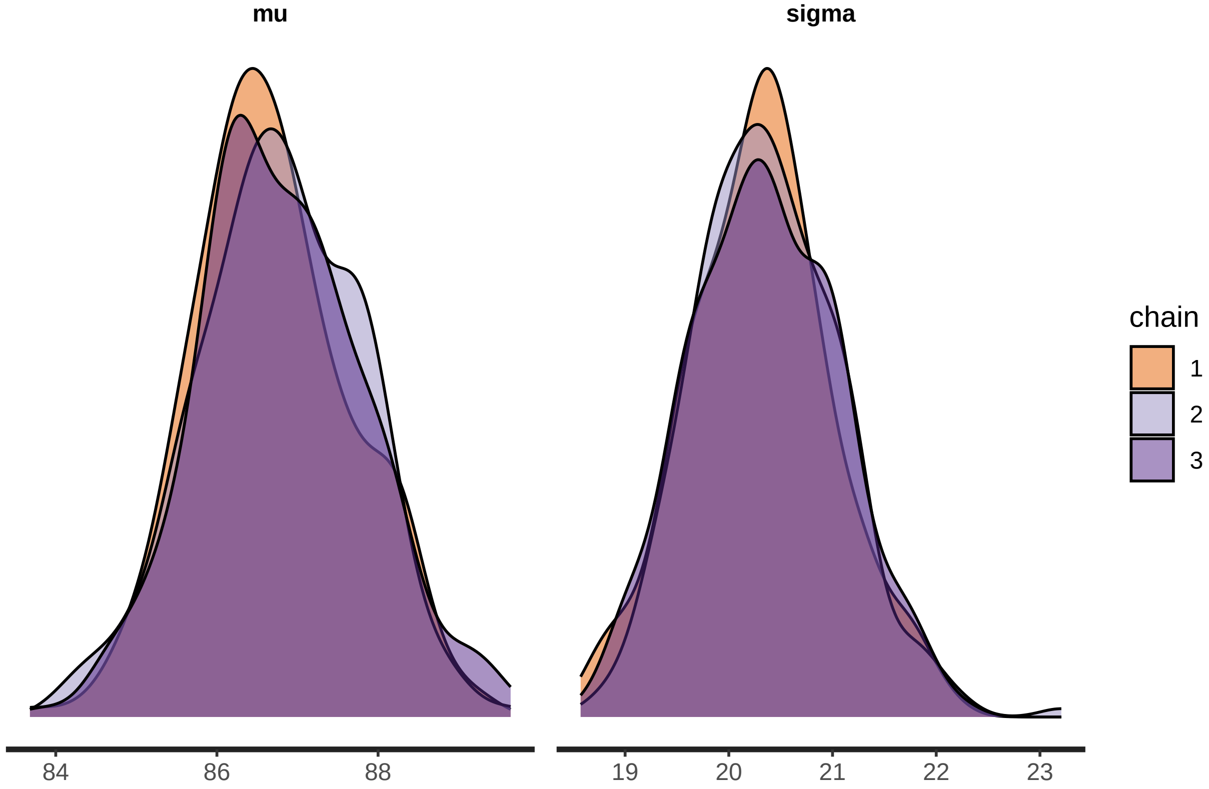


All looks fine.

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



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



Understanding output

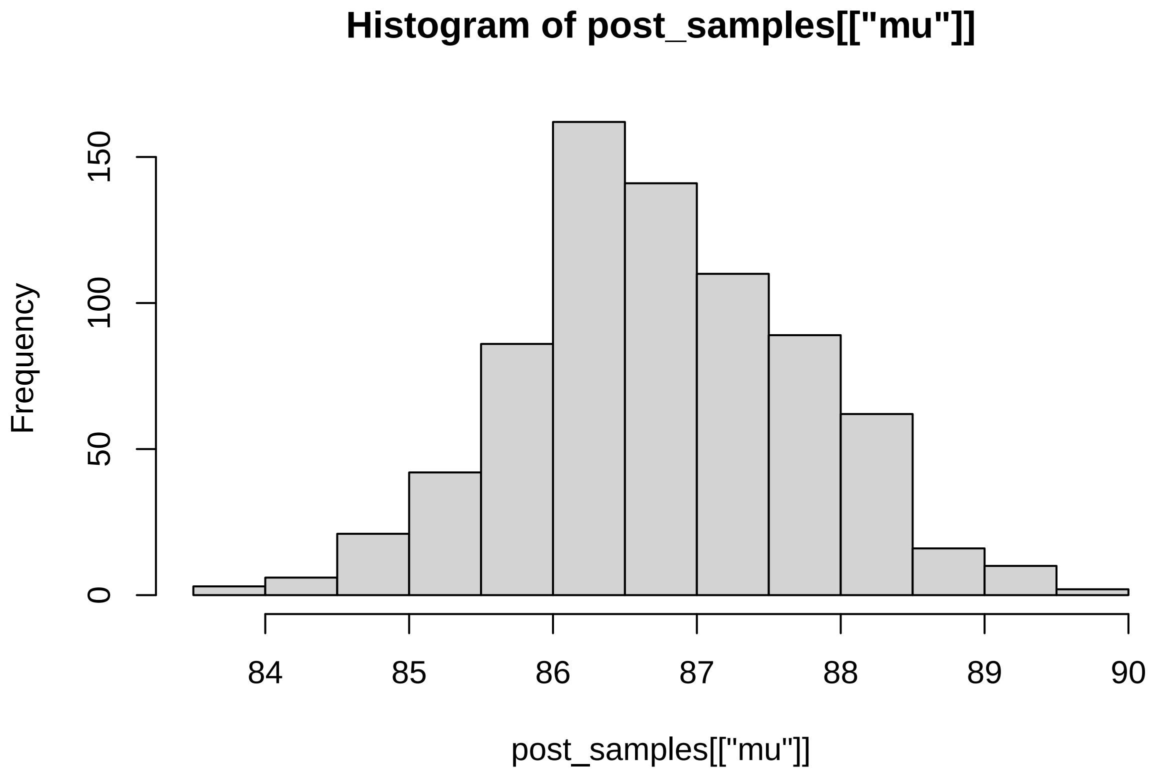
What does the model actually give us? A number of samples from the posteriors. To see this, we can use `extract` to get the samples.

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

```
## [1] 86.86500 86.10464 87.41335 85.34681 86.15718 86.84241
```

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

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



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

```
## [1] 86.70682
```

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

```
##    2.5%
## 84.8269
```

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

```
##    97.5%
## 88.79156
```

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:

```
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          86.0
##  2      1          2     2 mu          86.8
##  3      1          3     3 mu          86.8
##  4      1          4     4 mu          86.2
##  5      1          5     5 mu          86.7
##  6      1          6     6 mu          87.3
##  7      1          7     7 mu          86.0
##  8      1          8     8 mu          85.3
##  9      1          9     9 mu          88.4
## 10      1         10    10 mu          87.9
## # ... with 1,490 more rows
```

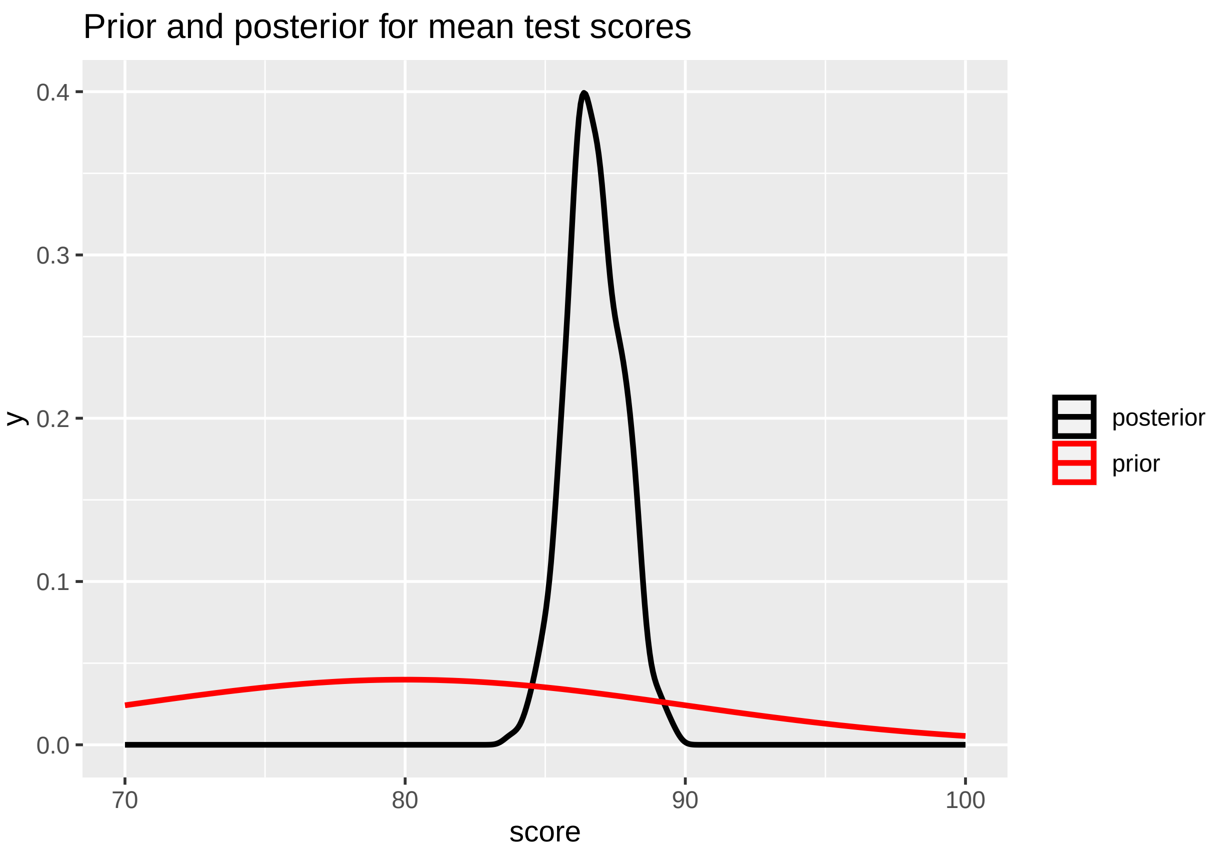
```
# wide format
fit  |>  spread_draws(mu, sigma)
```

```
## # A tibble: 750 x 5
##    .chain .iteration .draw    mu sigma
##     <int>      <int> <int> <dbl> <dbl>
##  1      1          1     1  86.0  20.1
##  2      1          2     2  86.8  21.2
##  3      1          3     3  86.8  21.2
##  4      1          4     4  86.2  21.8
##  5      1          5     5  86.7  22.2
##  6      1          6     6  87.3  22.0
##  7      1          7     7  86.0  20.5
##  8      1          8     8  85.3  20.9
##  9      1          9     9  88.4  20.0
## 10      1         10    10  87.9  19.6
## # ... with 740 more rows
```

```
# 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   88.1    0.8 median qi
## 2 sigma       20.3   19.5   21.2    0.8 median qi
```

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



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.

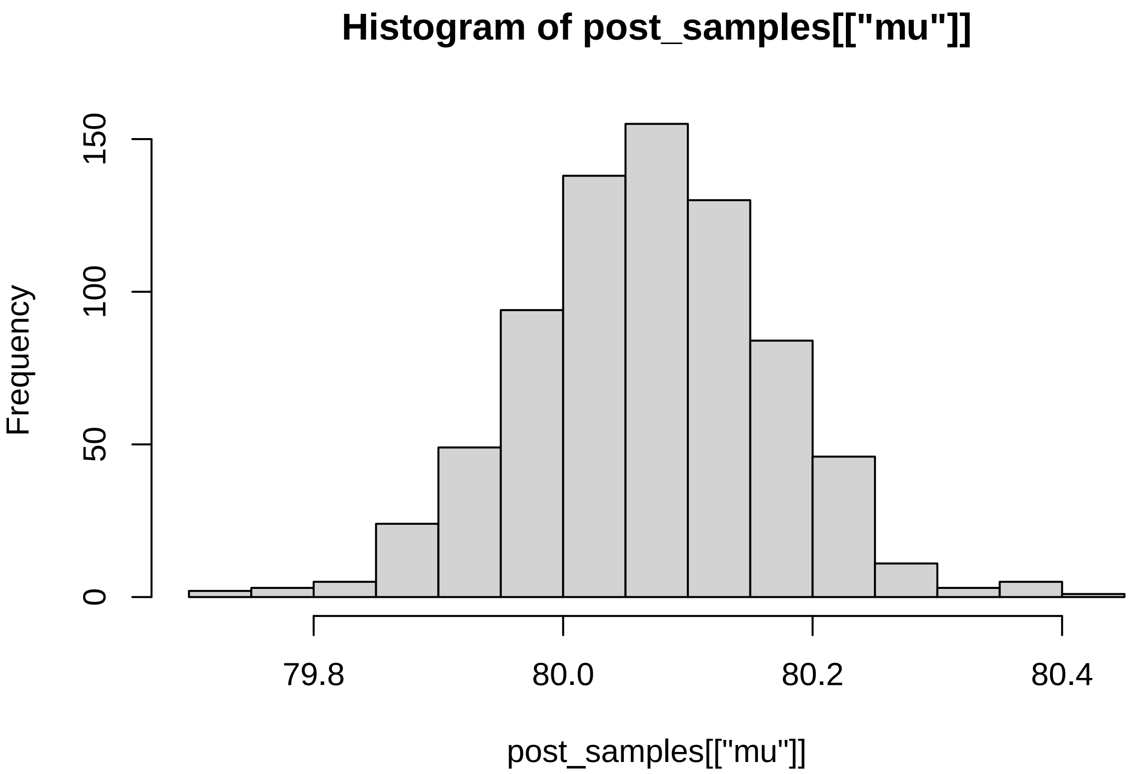
The new model is the following:

```
fit_new
```

```
## 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        80.07    0.01 0.10    79.87    80.01    80.07    80.13    80.25   371
## sigma     21.41    0.03 0.73    19.95    20.91    21.41    21.86    22.95   656
## lp__   -1548.38    0.06 1.09 -1551.45 -1548.76 -1548.07 -1547.64 -1547.39   346
##        Rhat
## mu        1
## sigma     1
## lp__      1
##
## Samples were drawn using NUTS(diag_e) at Sat Feb 11 16:57:35 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).
```

```
post_samples = extract(fit_new)

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



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

```
## [1] 80.06977
```

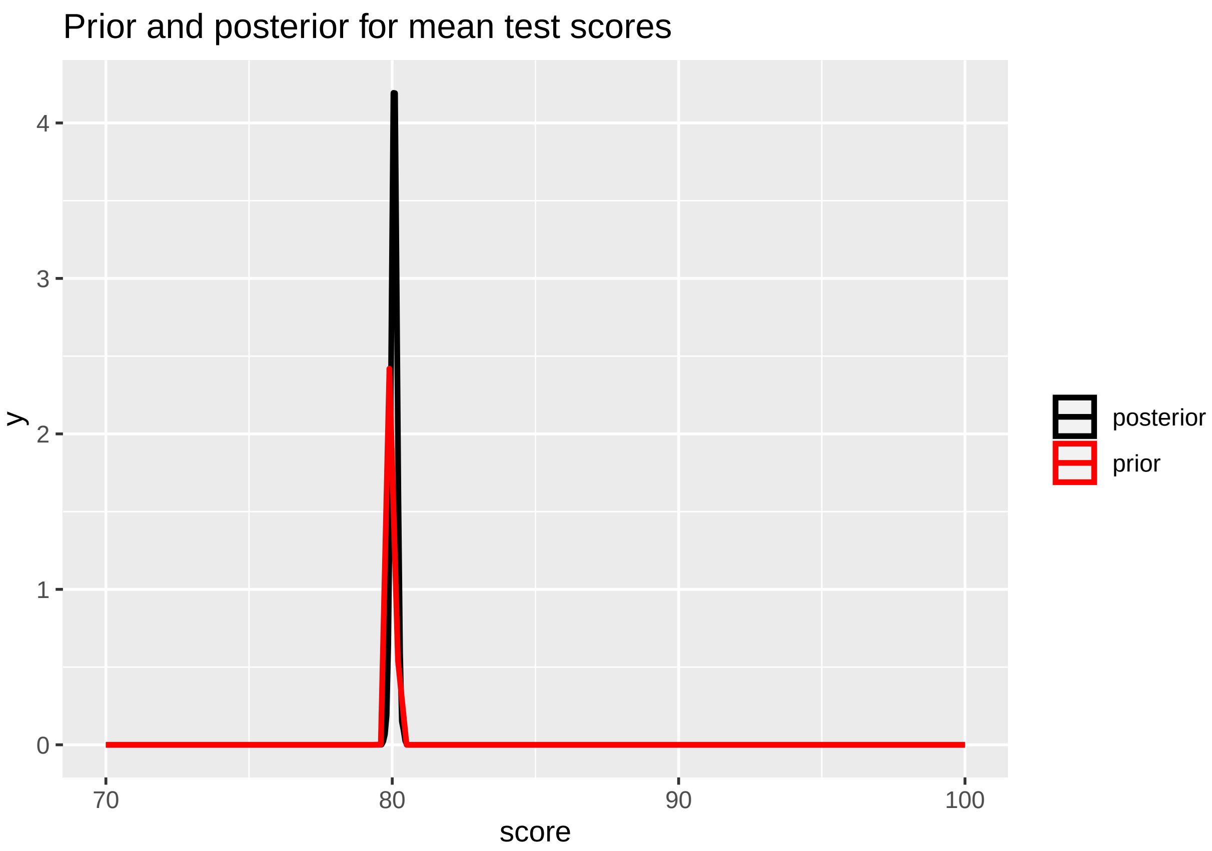
```
# 95% bayesian credible interval
quantile(post_samples[["mu"]], 0.025)
```

```
##     2.5%
## 79.87181
```

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

```
##    97.5%
## 80.25006
```

We can see that since our prior was very informative, even after seeing the data, our point estimate is still close to 80 compared to the model with less informative prior. The estimate for σ^2 did not change much as σ_0^2 is not related to the prior for σ_0^2. However, we see that our point estimate for μ is now approximately 80, when before the point estimate was approximate 87.



We see that the posterior distribution did not change much compared to the prior distribution.

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.

```r
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 = here("kids3.stan"),
            data = data,
            iter = 1000, refresh = 0)
```

Question 3

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

```
lm_model = lm(kid_score~as.factor(mom_hs), data = kidiq)
summary(lm_model)
```

```
##
## Call:
## lm(formula = kid_score ~ as.factor(mom_hs), data = kidiq)
##
## 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 ***
## as.factor(mom_hs)1   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
```

```
fit2
```

```
## 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      78.04    0.07 2.03    73.87    76.72    78.16    79.41    81.77
## beta[1]    11.13    0.08 2.26     7.02     9.60    11.01    12.58    15.92
## sigma      19.83    0.02 0.68    18.56    19.35    19.82    20.27    21.24
## lp__    -1514.38    0.04 1.22 -1517.58 -1514.94 -1514.04 -1513.47 -1512.99
##         n_eff Rhat
## alpha     818 1.01
## beta[1]   809 1.01
## sigma    1040 1.00
## lp__      857 1.01
##
## Samples were drawn using NUTS(diag_e) at Sat Feb 11 16:58:32 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).
```

We do see that the point estimate of the intercept and the β_1 are very similar, as well as the estimate for σ, since residual standard error is the standard point estimate for σ in the framework of linear regression.

```r
post_samples = extract(fit2)

quantile(post_samples[["beta"]], 0.025)
```

```
##     2.5%
## 7.019432
```

```r
quantile(post_samples[["beta"]], 0.975)
```

```
##    97.5%
## 15.92458
```

```r
quantile(post_samples[["alpha"]], 0.025)
```

```
##     2.5%
## 73.86822
```

```r
quantile(post_samples[["alpha"]], 0.975)
```

```
##  97.5%
## 81.771
```

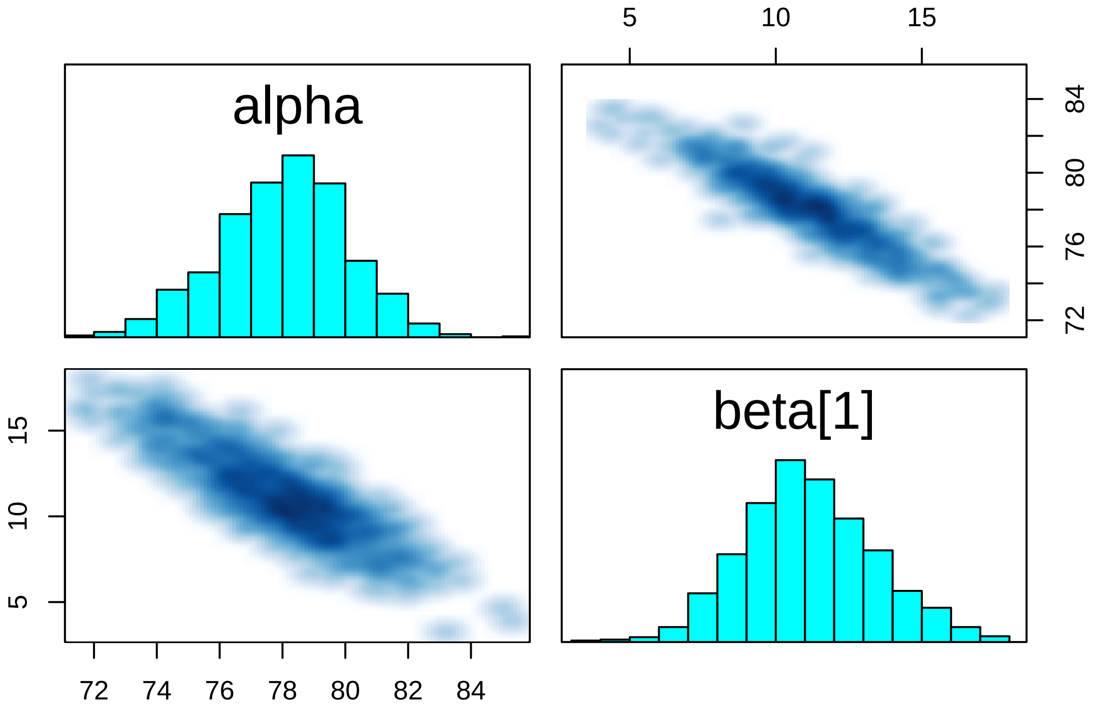
```r
confint(lm_model)
```

```
##                         2.5 %   97.5 %
## (Intercept)        73.502246 81.59453
## as.factor(mom_hs)1  7.206598 16.33592
```

Even though the interpretation is drastically different, the credible interval for α and β and the confidence interval for the two parameters are also similar.

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?

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



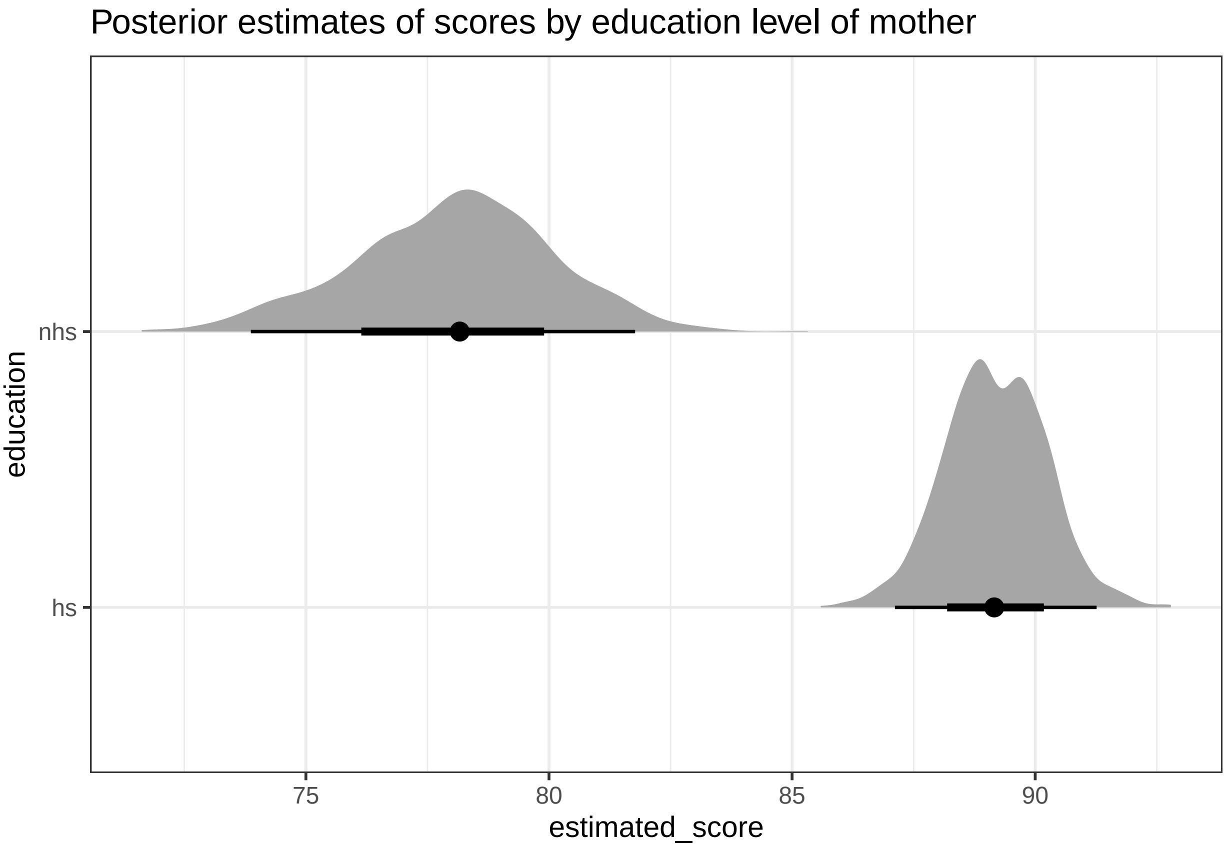
We see that in the joint distribution there are some strong correlations between the slope and the intercept. In particular, if the slope is small, the intercept is big, and if the slope is big, the intercept is small.

The potential problem here is that this will make the posterior sampling possibly inefficient.

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.

The returned model is the following.

```
fit3
```

```
## 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       82.34    0.06 1.85    78.72    81.09    82.32    83.58    85.89
## beta[1]      5.67    0.06 2.12     1.59     4.26     5.64     7.10     9.79
## beta[2]      8.43    0.02 0.89     6.66     7.85     8.44     9.02    10.16
## sigma       18.10    0.02 0.62    16.96    17.66    18.08    18.50    19.39
## lp__     -1474.76    0.05 1.39 -1478.11 -1475.45 -1474.49 -1473.72 -1473.01
##          n_eff Rhat
## alpha    1111    1
## beta[1]  1150    1
## beta[2]  1656    1
## sigma    1272    1
## lp__      901    1
```

```
##
## Samples were drawn using NUTS(diag_e) at Sat Feb 11 16:58:33 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).
```

We will use the mean of 8.41 as the point estimate for the coefficient on the IQ of the mother. The interpretation is that increase corresponding to one standard deviation(which is 15 in this case) results in increase of 8.41 in the expected score of the children.

Question 5

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

```
lm_model = lm(kid_score~as.factor(mom_hs)+scale(mom_iq), data = kidiq)

summary(lm_model)
```

```
##
## Call:
## lm(formula = kid_score ~ as.factor(mom_hs) + scale(mom_iq), data = kidiq)
##
## 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)         82.1221     1.9437  42.250  < 2e-16 ***
## as.factor(mom_hs)1   5.9501     2.2118   2.690  0.00742 **
## scale(mom_iq)        8.4586     0.9086   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
```

We can see that point estimate for the intercept, β_1, and β_2, and σ are all very similar to that of the results from Stan.

```
confint(lm_model)
```

```
##                        2.5 %   97.5 %
## (Intercept)        78.301832 85.94245
## as.factor(mom_hs)1  1.602837 10.29740
## scale(mom_iq)       6.672731 10.24445
```

```
post_samples = extract(fit3)

#beta_3
quantile(post_samples[["beta"]][,1], 0.025)
```

```
##     2.5%
## 1.585901
```

```
quantile(post_samples[["beta"]][,1], 0.975)
```

```
##    97.5%
## 9.789855
```

```
#beta_2
quantile(post_samples[["beta"]][,2], 0.025)
```

```
##     2.5%
## 6.657967
```

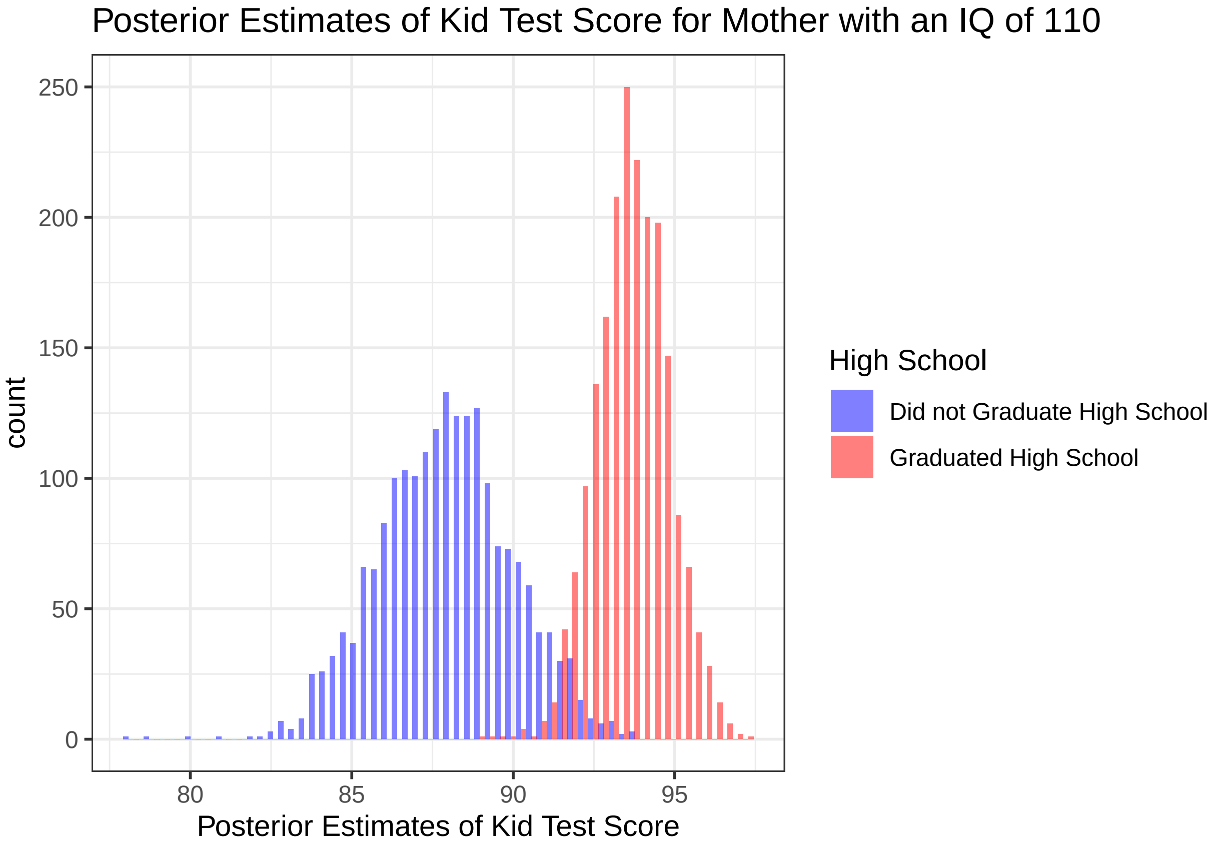
```
quantile(post_samples[["beta"]][,2], 0.975)
```

```
##    97.5%
## 10.16259
```

Even though the interpretation is different, we also see that the 95% from the linear model and the 95% credible intervals are also similar.

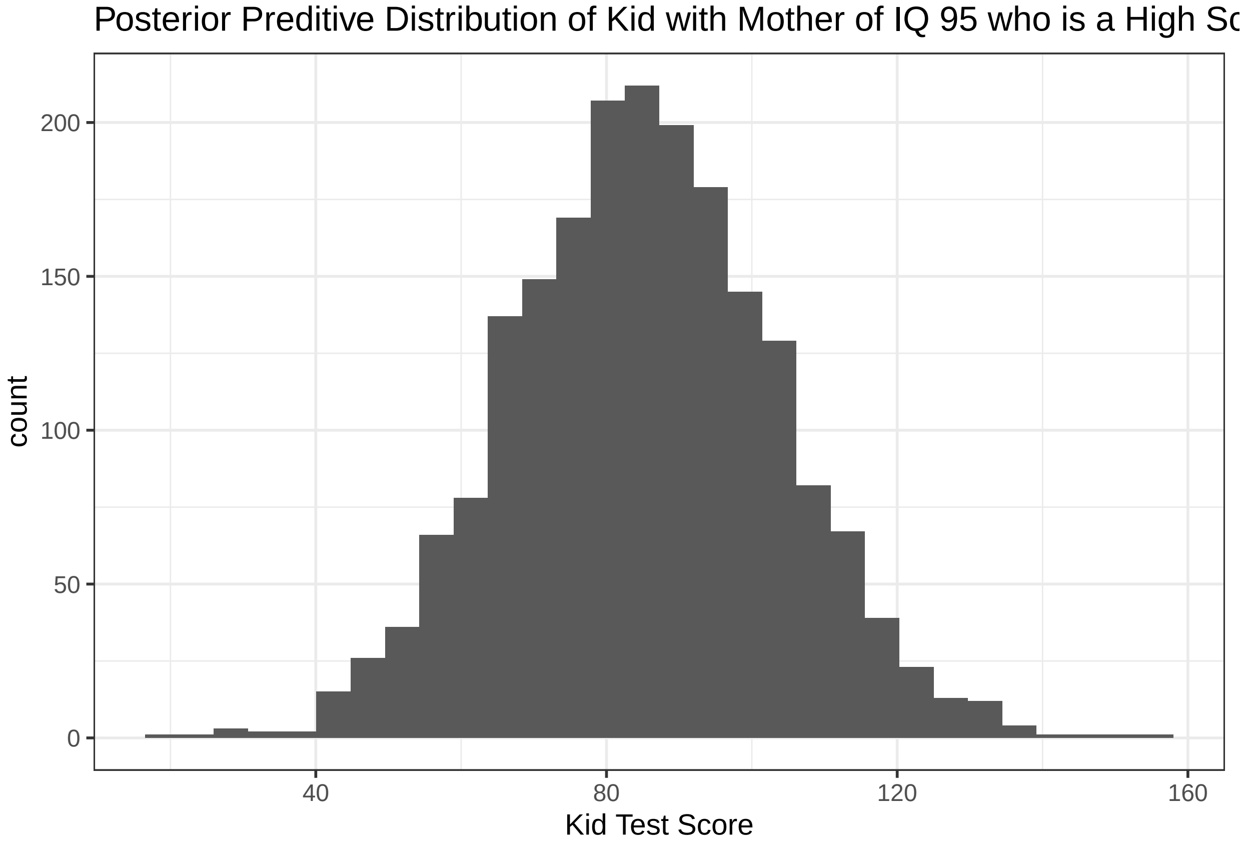
Question 6

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

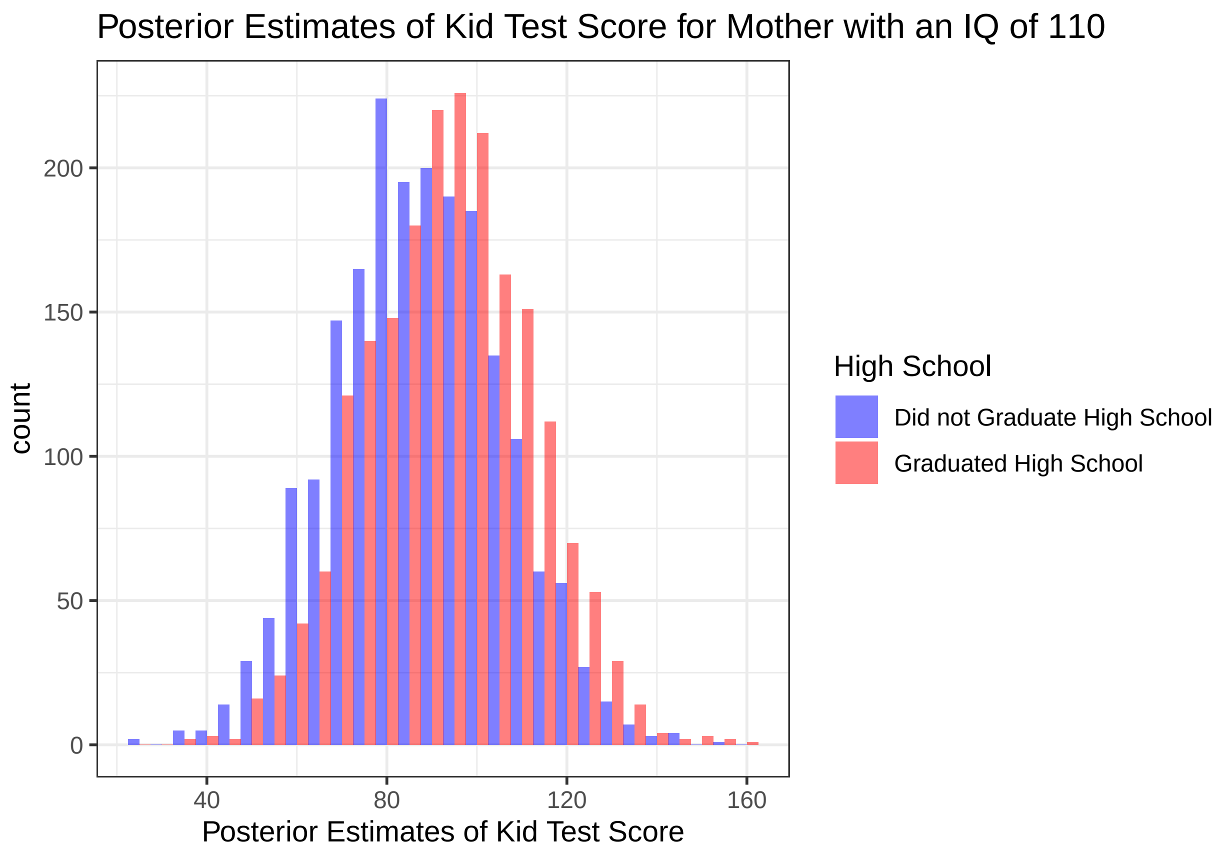


Question 7

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.



For question 6, it was not clear to me if the question wanted posterior predictive distribution or not. Below, I plot the posterior predictive distribution for question 6.



With the added uncertainty, the distribution is not as separated compared to what we did first in question 6.