Lab_5_Solved

Question 1

```r
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.3.2
```

```
Warning: package 'readr' was built under R version 4.3.2
```

```
Warning: package 'forcats' was built under R version 4.3.2
```

```
-- 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   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     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 becom
```

```r
library(rstan)
```

```
Warning: package 'rstan' was built under R version 4.3.2
```

```
Loading required package: StanHeaders
```

```
Warning: package 'StanHeaders' was built under R version 4.3.2
```

```
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)

Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file

Attaching package: 'rstan'

The following object is masked from 'package:tidyr':

    extract
```

```r
library(tidybayes)
```

```
Warning: package 'tidybayes' was built under R version 4.3.2
```

```r
library(here)
```

```
Warning: package 'here' was built under R version 4.3.2

here() starts at C:/Users/rudra/Documents/GitHub/STA2201WorkRudra/labs
```

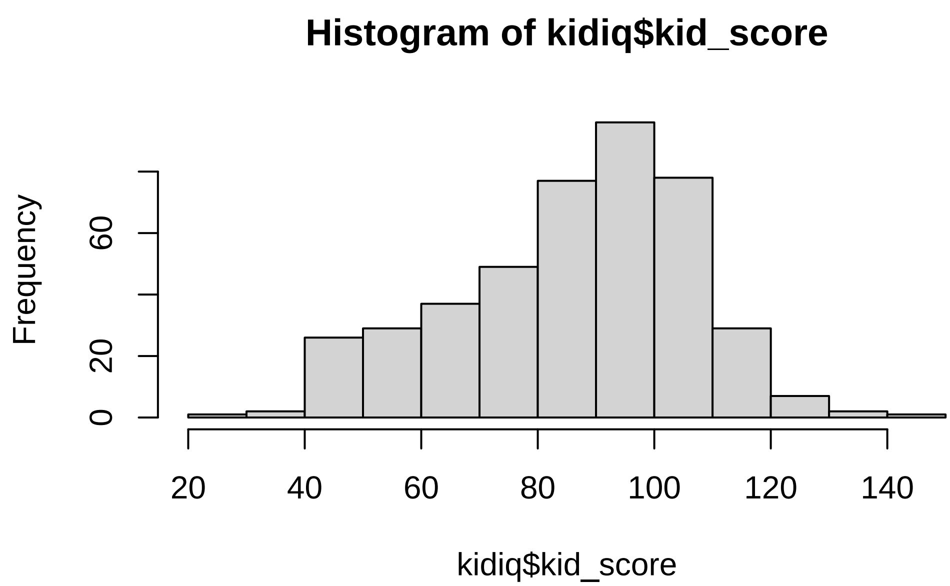
```r
kidiq <- read_rds(here("kidiq.RDS"))
kidiq
```

```
# 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
```

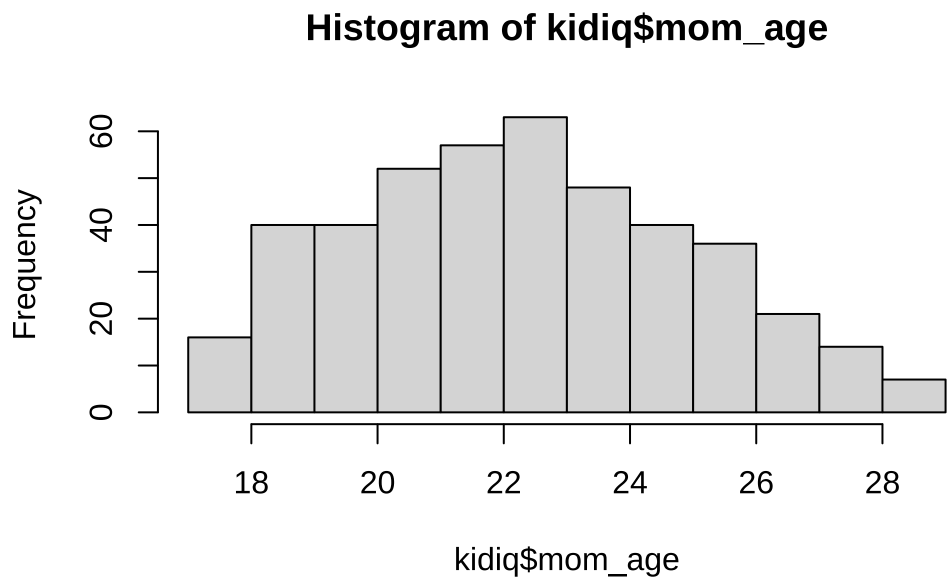
```
 4        83      1   99.4      25
 5       115      1   92.7      27
 6        98      0  108.       18
 7        69      1  139.       20
 8       106      1  125.       23
 9       102      1   81.6      24
10        95      1   95.1      19
# i 424 more rows
```

We proceed to make the following plots about the data:

```
iq_plot <- hist(kidiq$kid_score)
```




```
mother_age <- hist(kidiq$mom_age)
```




```r
kid_mom_hs <- ggplot(data = kidiq, aes(x=kid_score,y=mom_iq,color=mom_hs))+
  geom_point()
iq_plot
```

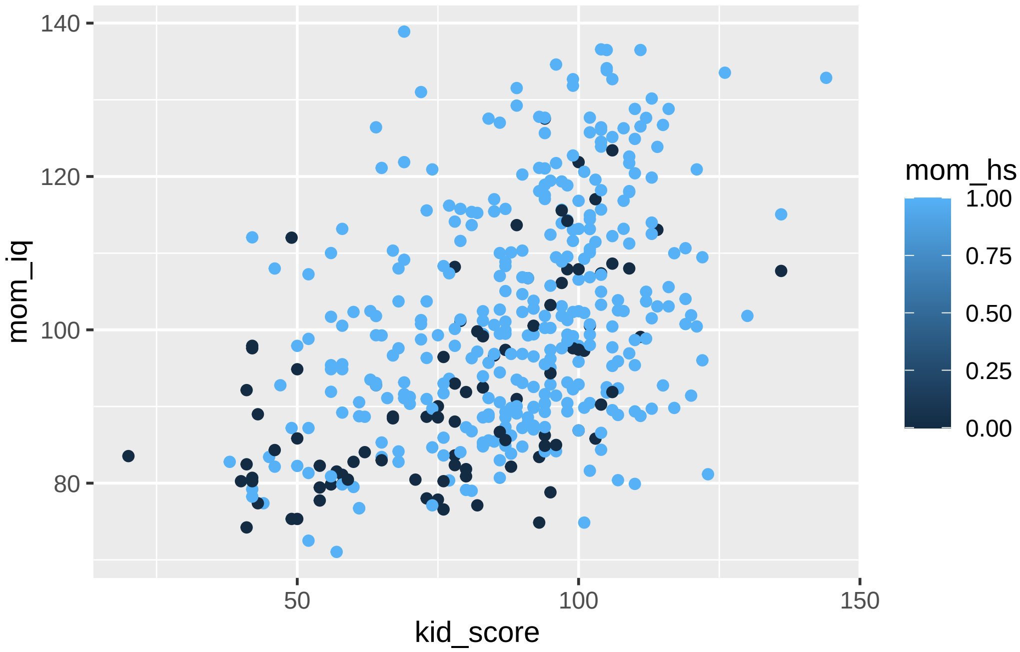
```
$breaks
 [1]  20  30  40  50  60  70  80  90 100 110 120 130 140 150

$counts
 [1]  1  2 26 29 37 49 77 96 78 29  7  2  1

$density
 [1] 0.0002304147 0.0004608295 0.0059907834 0.0066820276 0.0085253456
 [6] 0.0112903226 0.0177419355 0.0221198157 0.0179723502 0.0066820276
[11] 0.0016129032 0.0004608295 0.0002304147

$mids
 [1]  25  35  45  55  65  75  85  95 105 115 125 135 145

$xname
[1] "kidiq$kid_score"

$equidist
```

```
[1] TRUE

attr(,"class")
[1] "histogram"
```

```
mother_age
```

```
$breaks
 [1] 17 18 19 20 21 22 23 24 25 26 27 28 29

$counts
 [1] 16 40 40 52 57 63 48 40 36 21 14  7

$density
 [1] 0.03686636 0.09216590 0.09216590 0.11981567 0.13133641 0.14516129
 [7] 0.11059908 0.09216590 0.08294931 0.04838710 0.03225806 0.01612903

$mids
 [1] 17.5 18.5 19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5 28.5

$xname
[1] "kidiq$mom_age"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

```
kid_mom_hs
```



We observe the distributions of kid iq and and the ages of the mothers, and we observe that most mothers in the dataset are young. Not to mention, the plot of kid_score and mom iq against each other parametrized by whether the mother visited high school shows us that most kids in the dataset have mothers who attended high school

```
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)

fit <- stan(file = here("code/models/kids2.stan"),
            data = data,
            # reducing the iterations a bit to speed things up
            chains = 3,
            iter = 500)
```

Question 2

We proceed to implement the new updated sigma in our model:

```
y <- kidiq$kid_score
mu0 <- 80
sigma0 <- 0.1
data <- list(y = y,
             N = length(y),
             mu0 = mu0,
             sigma0 = sigma0)

fit1 <- stan(file = here("code/models/kids2.stan"),
            data = data,
            chains = 3,
            iter = 500)
```

```
print(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 0.88    85.10    86.09    86.74    87.34    88.33   564
sigma   20.40    0.03 0.69    19.18    19.92    20.39    20.81    21.80   610
lp__ -1525.67    0.05 0.90 -1528.36 -1526.02 -1525.41 -1525.03 -1524.79   284
      Rhat
mu    1.00
sigma 1.01
lp__  1.00

Samples were drawn using NUTS(diag_e) at Fri Feb 16 15:26:42 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).
```

```
print(fit1)
```

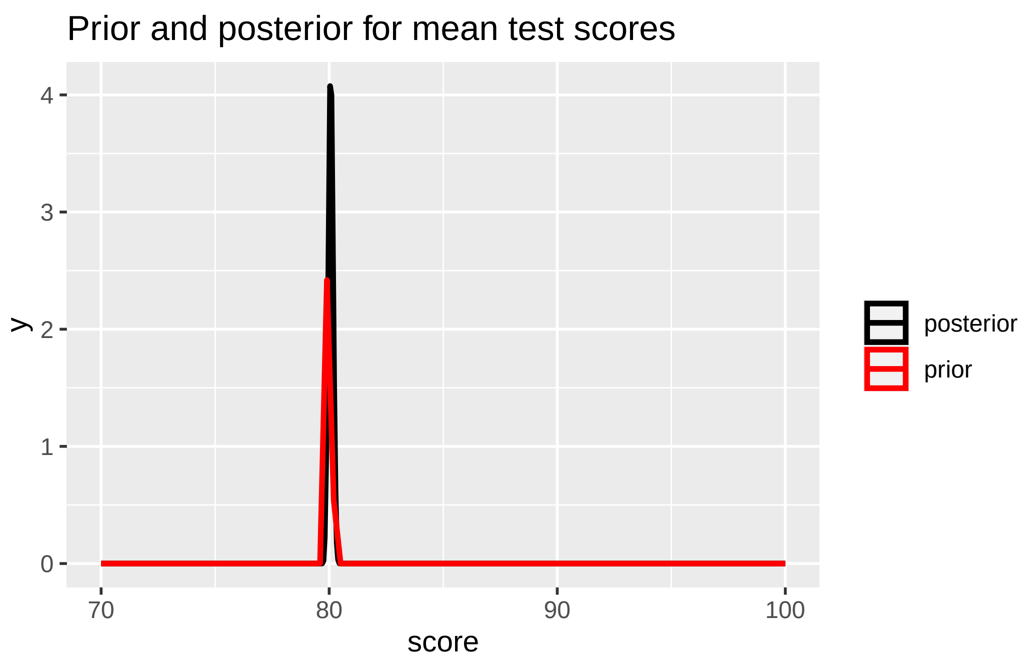
```
Inference for Stan model: anon_model.
```

```
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       80.06    0.00 0.10    79.87    80.00    80.07    80.13    80.26   606
sigma    21.37    0.03 0.74    19.96    20.89    21.35    21.86    22.85   735
lp__  -1548.39    0.05 1.00 -1551.15 -1548.84 -1548.08 -1547.66 -1547.39   363
      Rhat
mu    1.01
sigma 1.00
lp__  1.00

Samples were drawn using NUTS(diag_e) at Fri Feb 16 15:26:42 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).
```

As we see from the summaries of the models, in the new one, the estimate for mu shifts downward heavily, decreasing by 6 points, while the estimate for sigma increases by 1 point. We get the prior and posterior distribution plots in the next chunk:

```r
dsamples <- fit1 |>
            gather_draws(mu, sigma)
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")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



Question 3

```
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("code/models/kids3.stan"),
data = data,
iter = 1000)
```

a)

We evaluate and compare the results of our fit with a linear model in the next code chunk:

```
model <- lm(kid_score ~ mom_hs, data=kidiq)
summary(model)
```

```
Call:
lm(formula = kid_score ~ 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 ***
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
```

```r
print(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      77.98    0.07 2.01    74.09    76.67    77.95    79.28    81.99
beta[1]    11.22    0.08 2.28     6.62     9.71    11.21    12.73    15.59
sigma      19.83    0.02 0.72    18.47    19.32    19.82    20.31    21.26
lp__    -1514.49    0.05 1.30 -1517.83 -1515.13 -1514.18 -1513.50 -1512.99
        n_eff Rhat
alpha     876    1
beta[1]   849    1
sigma     974    1
lp__      771    1

Samples were drawn using NUTS(diag_e) at Fri Feb 16 15:27:28 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).
```

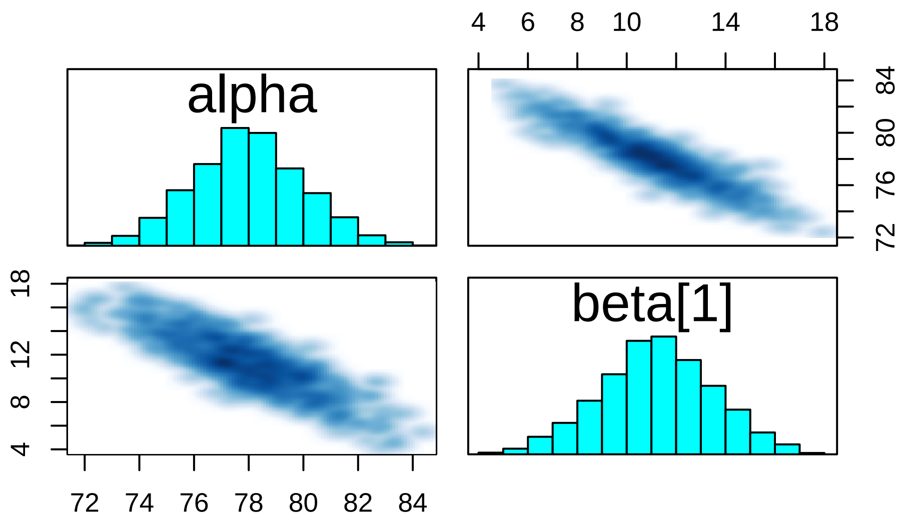
We see that the coefficient estimates for the intercept and beta 1 by the linear model are very close to our fit object.

b)

We get the pairs plot from the following chunk:

```
pairs(fit2, pars = c("alpha", "beta[1]"))
```

```
Warning in par(usr): argument 1 does not name a graphical parameter

Warning in par(usr): argument 1 does not name a graphical parameter
```



We see that there is a negative linear relation between our intercept and the coefficient of mom_hs. A possible explanation of this phenomenon might be multicollinearity present in the dataset.

Question 4

We create a new column containing centered mom iqs and then create a new fit object using that as a covariate in the next code chunk:

```r
kidiq$mom_iq_cent <- kidiq$mom_iq - mean(kidiq$mom_iq)
X <- as.matrix(kidiq[, c("mom_hs", "mom_iq_cent")])
K <- 2
data3 <- list(y = y,
N = length(y),
X = X,
K = K
)
fit4 <- stan(file = "code/models/kids3.stan",
data = data3,
iter = 1000)
```

Question 5

We create a linear model in the next code chunk:

```r
model1 <-lm(kid_score ~ mom_hs + mom_iq_cent, data=kidiq)
```

We now check the summary of that model with fit4:

```r
summary(model1)
```

```
Call:
lm(formula = kid_score ~ mom_hs + mom_iq_cent, 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) 82.12214    1.94370  42.250  < 2e-16 ***
mom_hs       5.95012    2.21181   2.690  0.00742 ** 
mom_iq_cent  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
```

```r
print(fit4)
```

```
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    0.06 1.92    78.56    80.94    82.24    83.55    85.85
beta[1]     5.77    0.07 2.17     1.58     4.27     5.77     7.21    10.15
beta[2]     0.56    0.00 0.06     0.44     0.52     0.56     0.61     0.68
sigma      18.09    0.02 0.64    16.86    17.65    18.08    18.52    19.40
lp__    -1474.49    0.06 1.47 -1478.15 -1475.22 -1474.17 -1473.41 -1472.69
        n_eff Rhat
alpha     979 1.00
beta[1]   991 1.00
beta[2]  1555 1.00
sigma    1315 1.00
lp__      701 1.01

Samples were drawn using NUTS(diag_e) at Fri Feb 16 15:27:29 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).
```

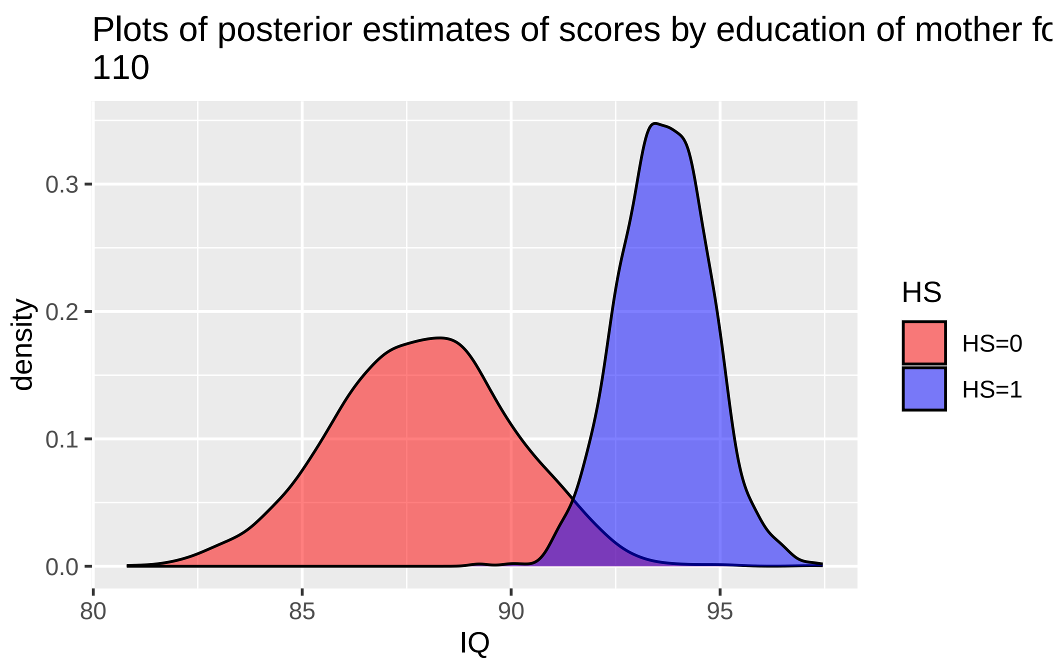
Again, we see that the estimates of our linear model are very close to that of our stan model.

Question 6

We proceed to extract the fit posterior object and then get posterior estimates for alpha and beta to get the required estimates of scores:

```r
fit_obj <- rstan::extract(fit4)
alpha_posterior <- fit_obj$alpha
beta_posterior <- fit_obj$beta
sigma_post <- fit_obj$sigma
beta_1 <- beta_posterior[,1]
beta_2 <- beta_posterior[,2]

post_non_hs <- alpha_posterior +  beta_2 * (110 - mean(kidiq$mom_iq))
post_hs <- alpha_posterior + beta_1 * 1 + beta_2 * (110 - mean(kidiq$mom_iq))
```

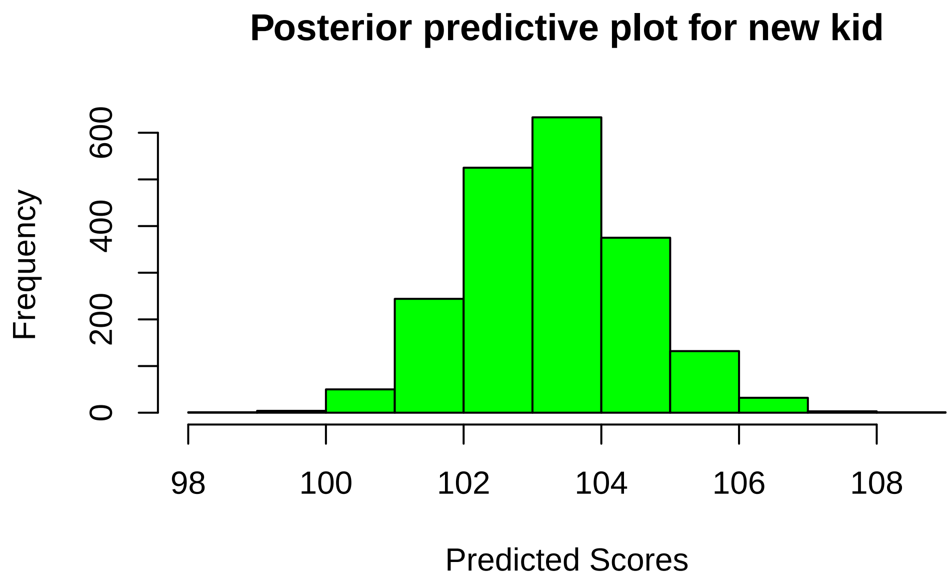
```r
df<- data.frame(
  IQ = c(post_non_hs,post_hs),
  HS = rep(c("HS=0","HS=1"),each = length(post_non_hs))
)
ggplot(df, aes(x= IQ, fill=HS))+
  geom_density(alpha=0.5)+
  labs(title = "Plots of posterior estimates of scores by education of mother for mothers
110")+
  scale_fill_manual(values=c("red","blue"))
```



Question 7

We proceed to generate a histogram plot for posterior predictive samples of the case for a new kid with a mother who graduated high school and has a IQ of 95

```r
posterior_95 <- alpha_posterior + beta_1 + beta_2*(95-mean(kidiq$mom_iq)) + sigma_post

hist(posterior_95,main = "Posterior predictive plot for new kid", xlab="Predicted Scores",
```



Posterior predictive plot for new kid
Frequency
0
200
400
600
98
100
102
104
106
108
Predicted Scores