Applied_Stat_Lab_5

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.1
-- 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)
```

```
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'
```

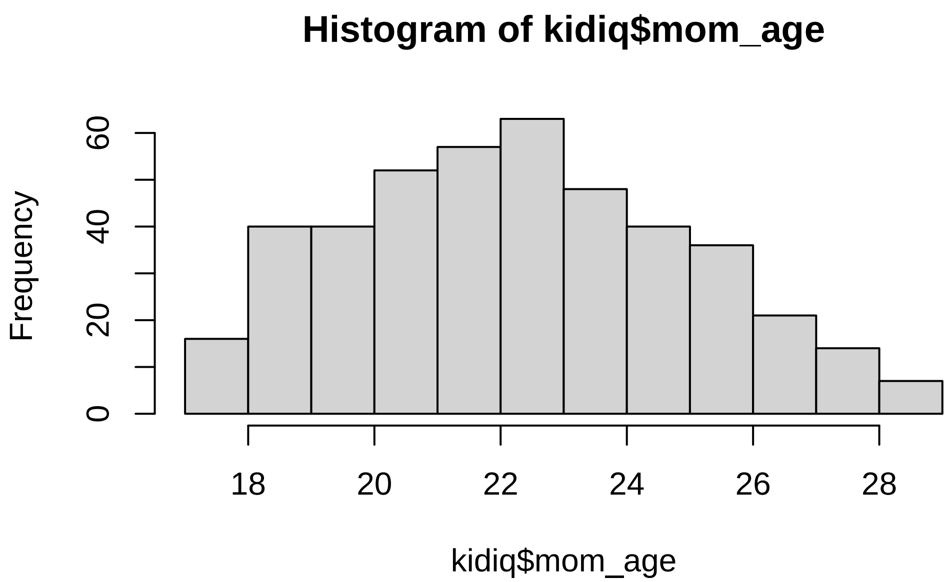
```
The following object is masked from 'package:tidyr':

    extract
```

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

```
here() starts at /Users/larskutschinski/Desktop/AppliedStats/AppliedStats22
```

Question 1

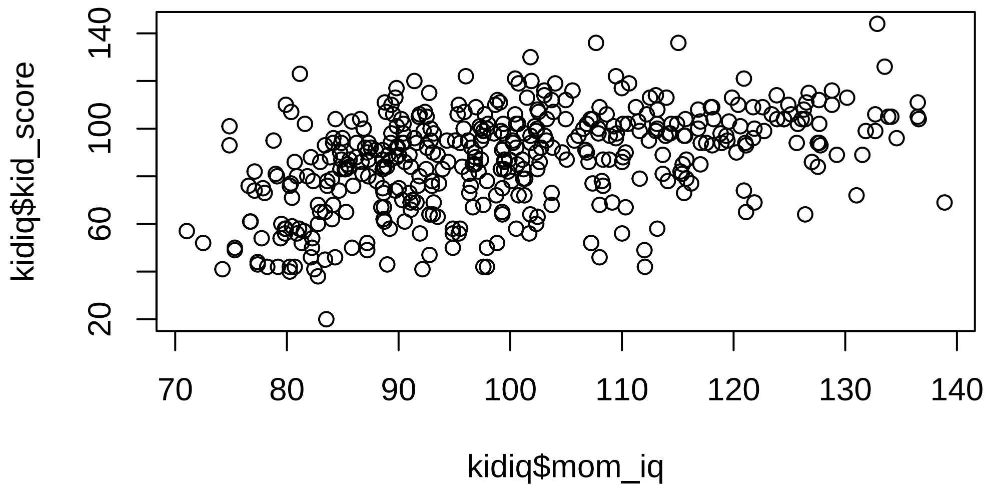
```r
kidiq <- read_rds(here("data","kidiq.RDS"))
hist(kidiq$mom_age)
```




```r
means <- kidiq |> group_by(mom_hs) |> summarize(means = mean(kid_score, na.rm =  TRUE))
means
```

```
# A tibble: 2 x 2
  mom_hs means
   <dbl> <dbl>
1      0  77.5
2      1  89.3
```

```r
plot(kidiq$mom_iq, kidiq$kid_score)
```



We created three different plots. In the histogram we plotted the mom's age frequency and notice that the age range is quite small with a maximum age of 28. The average age is at 22. Secondly we calculated the mean kids scores based on whether the mom obtained a high school degree. We notice that the mean is higher for mom_hs = 1. Lastly we plot the mom's iq against the kid's iq and notice that the definetely seems to be a positive correlation. The higher the moms iq, the higher the kid's iq.

Question 2

```r
y <- kidiq$kid_score
mu0 <- 80
sigma0 <- 0.1
# named list to input for stan function
data <- list(y = y,
N = length(y),
mu0 = mu0,
sigma0 = sigma0)

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

```
Warning in readLines(file, warn = TRUE): incomplete final line found on
'/Users/larskutschinski/Desktop/AppliedStats/AppliedStats22/stan/kids2.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"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/R
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
               ^
               ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#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 4e-06 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:   1 / 500 [  0%]  (Warmup)
Chain 1: Iteration:  50 / 500 [ 10%]  (Warmup)
Chain 1: Iteration: 100 / 500 [ 20%]  (Warmup)
Chain 1: Iteration: 150 / 500 [ 30%]  (Warmup)
Chain 1: Iteration: 200 / 500 [ 40%]  (Warmup)
Chain 1: Iteration: 250 / 500 [ 50%]  (Warmup)
```

```
Chain 1: Iteration: 251 / 500 [ 50%]  (Sampling)
Chain 1: Iteration: 300 / 500 [ 60%]  (Sampling)
Chain 1: Iteration: 350 / 500 [ 70%]  (Sampling)
Chain 1: Iteration: 400 / 500 [ 80%]  (Sampling)
Chain 1: Iteration: 450 / 500 [ 90%]  (Sampling)
Chain 1: Iteration: 500 / 500 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.002 seconds (Warm-up)
Chain 1:                0.001 seconds (Sampling)
Chain 1:                0.003 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.01 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:   1 / 500 [  0%]  (Warmup)
Chain 2: Iteration:  50 / 500 [ 10%]  (Warmup)
Chain 2: Iteration: 100 / 500 [ 20%]  (Warmup)
Chain 2: Iteration: 150 / 500 [ 30%]  (Warmup)
Chain 2: Iteration: 200 / 500 [ 40%]  (Warmup)
Chain 2: Iteration: 250 / 500 [ 50%]  (Warmup)
Chain 2: Iteration: 251 / 500 [ 50%]  (Sampling)
Chain 2: Iteration: 300 / 500 [ 60%]  (Sampling)
Chain 2: Iteration: 350 / 500 [ 70%]  (Sampling)
Chain 2: Iteration: 400 / 500 [ 80%]  (Sampling)
Chain 2: Iteration: 450 / 500 [ 90%]  (Sampling)
Chain 2: Iteration: 500 / 500 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.002 seconds (Warm-up)
Chain 2:                0.001 seconds (Sampling)
Chain 2:                0.003 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.01 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
```

```
Chain 3:
Chain 3: Iteration:   1 / 500 [  0%]  (Warmup)
Chain 3: Iteration:  50 / 500 [ 10%]  (Warmup)
Chain 3: Iteration: 100 / 500 [ 20%]  (Warmup)
Chain 3: Iteration: 150 / 500 [ 30%]  (Warmup)
Chain 3: Iteration: 200 / 500 [ 40%]  (Warmup)
Chain 3: Iteration: 250 / 500 [ 50%]  (Warmup)
Chain 3: Iteration: 251 / 500 [ 50%]  (Sampling)
Chain 3: Iteration: 300 / 500 [ 60%]  (Sampling)
Chain 3: Iteration: 350 / 500 [ 70%]  (Sampling)
Chain 3: Iteration: 400 / 500 [ 80%]  (Sampling)
Chain 3: Iteration: 450 / 500 [ 90%]  (Sampling)
Chain 3: Iteration: 500 / 500 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.002 seconds (Warm-up)
Chain 3:                0.001 seconds (Sampling)
Chain 3:                0.003 seconds (Total)
Chain 3:
```

```
Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and ta
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#tail-ess
```

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

fit |> spread_draws(mu, sigma)
```

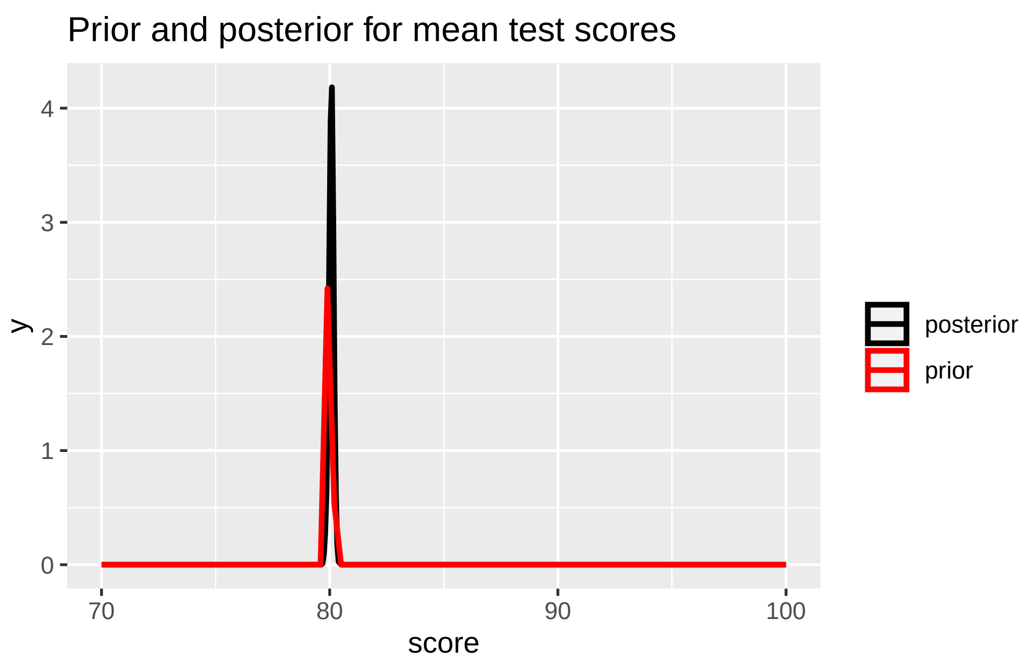
```
# A tibble: 750 x 5
   .chain .iteration .draw    mu sigma
    <int>      <int> <int> <dbl> <dbl>
 1      1          1     1  80.0  22.3
 2      1          2     2  80.1  21.6
 3      1          3     3  80.0  21.8
 4      1          4     4  80.1  21.3
 5      1          5     5  79.9  20.7
 6      1          6     6  80.0  21.9
 7      1          7     7  80.1  22.4
 8      1          8     8  80.0  21.3
 9      1          9     9  80.1  21.7
10      1         10    10  80.1  21.0
# i 740 more rows
```

```r
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          80.1   79.9   80.2    0.8 median qi
2 sigma       21.4   20.6   22.4    0.8 median qi
```

```r
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.
```



As we can see the estimates for μ and σ changed after choosing a more informative prior. We have $\hat{mu} = 80.06$ and $\widehat{sigma} = 21.41$ now.

Question 3

(a)

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

```
Warning in readLines(file, warn = TRUE): incomplete final line found on
'/Users/larskutschinski/Desktop/AppliedStats/AppliedStats22/stan/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"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/St
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
               ^
               ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/St
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#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 4.5e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.45 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: 0.08 seconds (Warm-up)
Chain 1:                0.039 seconds (Sampling)
Chain 1:                0.119 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 8e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.08 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: 0.089 seconds (Warm-up)
Chain 2:                0.034 seconds (Sampling)
Chain 2:                0.123 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 8e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.08 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: 0.077 seconds (Warm-up)
Chain 3:                0.032 seconds (Sampling)
Chain 3:                0.109 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 7e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.07 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: 0.066 seconds (Warm-up)
Chain 4:                0.038 seconds (Sampling)
Chain 4:                0.104 seconds (Total)
Chain 4:
```

```r
linear_model <- lm(kid_score ~ mom_hs, data = kidiq)

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      78.00    0.08 1.99    74.30    76.63    77.98    79.33    82.02
beta[1]    11.16    0.08 2.24     6.61     9.63    11.22    12.76    15.21
sigma      19.81    0.02 0.66    18.56    19.35    19.80    20.23    21.19
lp__    -1514.34    0.05 1.25 -1517.48 -1514.89 -1514.01 -1513.42 -1512.97
        n_eff Rhat
alpha     684 1.00
beta[1]   740 1.00
sigma    1105 1.00
lp__      675 1.01

Samples were drawn using NUTS(diag_e) at Fri Feb 16 01:37:38 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).
```

```r
summary(linear_model)
```

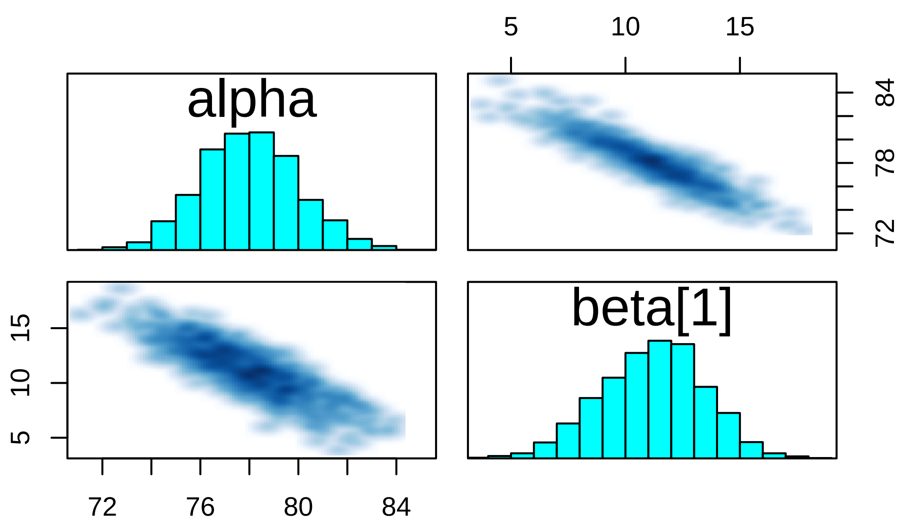
```
Call:
lm(formula = kid_score ~ mom_hs, data = kidiq)

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
```

The estimate for the intercept in the linear model is 77.548 which is very close to the posterior mean of the intercept (78.06). Furthermore the slope estimate lies at 11.771 in the linear model which is also close to the posterior mean of the slope at 11.11.

(b)

```
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
```



There appears to be a negative linear relationship between the intercept and the slope coefficient. This could imply multicollinearity, which would be an issue in the estimation of the coefficients.

Question 4

```
kidiq$mom_iq_centered <- kidiq$mom_iq - mean(kidiq$mom_iq)
X <- as.matrix(kidiq[, c("mom_hs", "mom_iq_centered")])
K <- 2

data<- list(y = y, N = length(y), X = X, K=K)
```

```r
fit3 <- stan(here("stan", "kids3.stan"), data = data, iter = 1000)
```

```
Warning in readLines(file, warn = TRUE): incomplete final line found on
'/Users/larskutschinski/Desktop/AppliedStats/AppliedStats22/stan/kids3.stan'



SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.2e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.12 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: 0.108 seconds (Warm-up)
Chain 1:                0.047 seconds (Sampling)
Chain 1:                0.155 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.1 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: 0.076 seconds (Warm-up)
Chain 2:                0.05 seconds (Sampling)
Chain 2:                0.126 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: 0.067 seconds (Warm-up)
Chain 3:                0.046 seconds (Sampling)
Chain 3:                0.113 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: 0.078 seconds (Warm-up)
Chain 4:                0.046 seconds (Sampling)
Chain 4:                0.124 seconds (Total)
Chain 4:
```

```
print(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      82.20    0.06 1.92    78.46    80.91    82.15    83.51    85.90
beta[1]     5.84    0.07 2.14     1.65     4.44     5.92     7.28     9.94
beta[2]     0.57    0.00 0.06     0.45     0.53     0.56     0.61     0.68
sigma      18.10    0.02 0.60    16.96    17.67    18.09    18.50    19.33
lp__    -1474.36    0.05 1.39 -1478.03 -1475.01 -1474.02 -1473.36 -1472.66
        n_eff Rhat
alpha     997    1
beta[1]   950    1
beta[2]  1374    1
sigma    1468    1
lp__      851    1
```

```
Samples were drawn using NUTS(diag_e) at Fri Feb 16 01:37:39 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).
```

The coefficient for mom's iq has the following interpretation: An unit increase in mom's iq amounts to a 0.57 increase in kid's iq on average.

Question 5

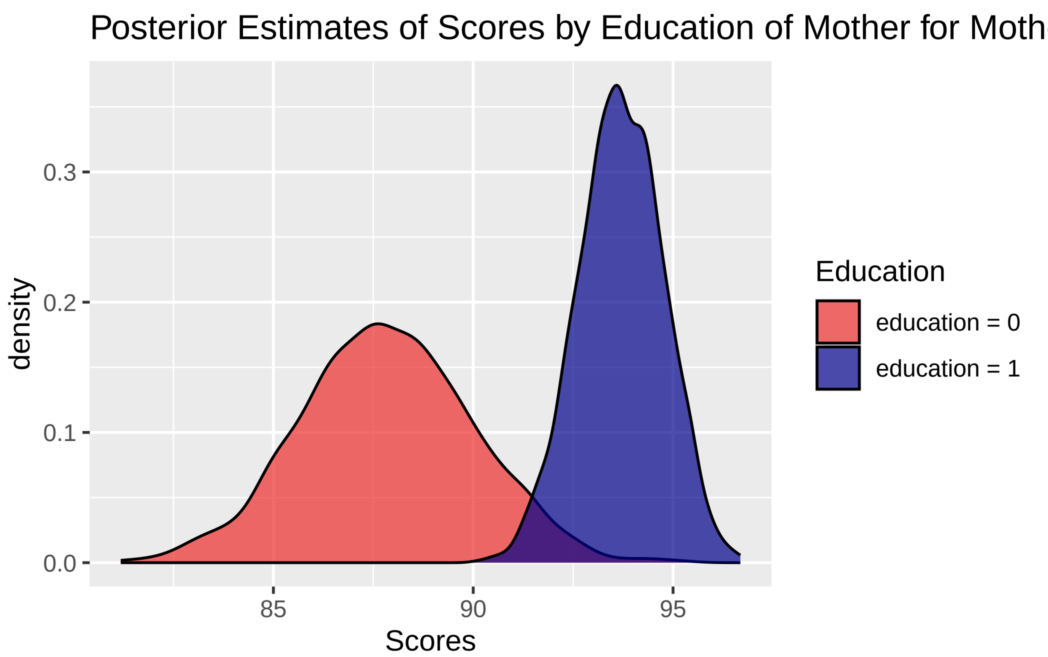
```r
linear_model_2 <- lm(kid_score ~ mom_hs + mom_iq_centered, data = kidiq)
summary(linear_model_2)
```

```
Call:
lm(formula = kid_score ~ mom_hs + mom_iq_centered, 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_centered  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 results of the linear model agree with Stan as the coefficient estimate for the centered mom iq is at 0.56391.

Question 6

```
ext_fit <- extract(fit3)
alpha_post <- ext_fit$alpha
beta_post <- ext_fit$beta
sigma<-ext_fit$sigma
b1<-beta_post[,1]
b2<-beta_post[,2]

posterior0 <- alpha_post  +  b1 * 0 + b2  * (110 - mean(kidiq$mom_iq))
posterior1 <- alpha_post  + b1 * 1 + b2 * (110 - mean(kidiq$mom_iq))

df <- data.frame(
  Scores = c(posterior0, posterior1),
  Education = rep(c("education = 0", "education = 1"), each = length(posterior0))
)
ggplot(df, aes(x = Scores, fill = Education)) +
  geom_density(alpha = 0.7) +
  labs(title = "Posterior Estimates of Scores by Education of Mother for Mothers with iq o
  scale_fill_manual(values = c("firebrick2", "blue4"))
```



Question 7

```
posterior <- alpha_post  + b1 * 1 + b2 * (95 - mean(kidiq$mom_iq)) + sigma
  data <- data.frame(Scores = posterior)
  ggplot(data, aes(x = Scores)) +
    geom_histogram(fill = "firebrick4", color = "black")
```

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