Regression and Other Stories: KidIQ

Prof. Kapitula

2020-10-22

Original Authors: Andrew Gelman, Jennifer Hill, Aki Vehtari

This code is shared in ~/Sharedprojects/Kapitula/STA631/MLR/KidIQ

Linear regression with multiple predictors. See Chapters 10, 11 and 12 in Regression and Other Stories.

```
library("rprojroot")
root<-has_dirname("MLR")$make_fix_file()
library("rstanarm")
library("ggplot2")
library("bayesplot")
theme_set(bayesplot::theme_default(base_family = "sans"))
library("foreign")
library("tidyverse")
library("skimr")
```

Load packages

```
kidiq <- read.csv("~/SharedProjects/Kapitula/STA631/ROSExamples/KidIQ/data/kidiq.csv")
head(kidiq)
```

Load children's test scores data

```
##   kid_score mom_hs    mom_iq mom_work mom_age
## 1        65      1 121.11753        4      27
## 2        98      1  89.36188        4      25
## 3        85      1 115.44316        4      27
## 4        83      1  99.44964        3      25
## 5       115      1  92.74571        4      27
## 6        98      0 107.90184        1      18
```

```
kidiq %>% skim_without_charts()
```

Table 1: Data summary

Name	Piped data
Number of rows	434
Number of columns	5

Column type frequency:

Table 1: Data summary

numeric	5
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
kid_score	0	1	86.80	20.41	20.00	74.00	90.00	102.00	144.00
mom_hs	0	1	0.79	0.41	0.00	1.00	1.00	1.00	1.00
mom_iq	0	1	100.00	15.00	71.04	88.66	97.92	110.27	138.89
mom_work	0	1	2.90	1.18	1.00	2.00	3.00	4.00	4.00
mom_age	0	1	22.79	2.70	17.00	21.00	23.00	25.00	29.00

A single predictor

A single binary predictor The option `refresh = 0` supresses the default Stan sampling progress output. This is useful for small data with fast computation. For more complex models and bigger data, it can be useful to see the progress.

```
fit_1 <- stan_glm(kid_score ~ mom_hs, data=kidiq, refresh = 0)
print(fit_1)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_hs
##  observations: 434
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 77.6    2.1
## mom_hs      11.8    2.3
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 19.9    0.7
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
lm_fit_1=lm(kid_score ~ mom_hs, data=kidiq)
summary(lm_fit_1)
```

```
##
## 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
```

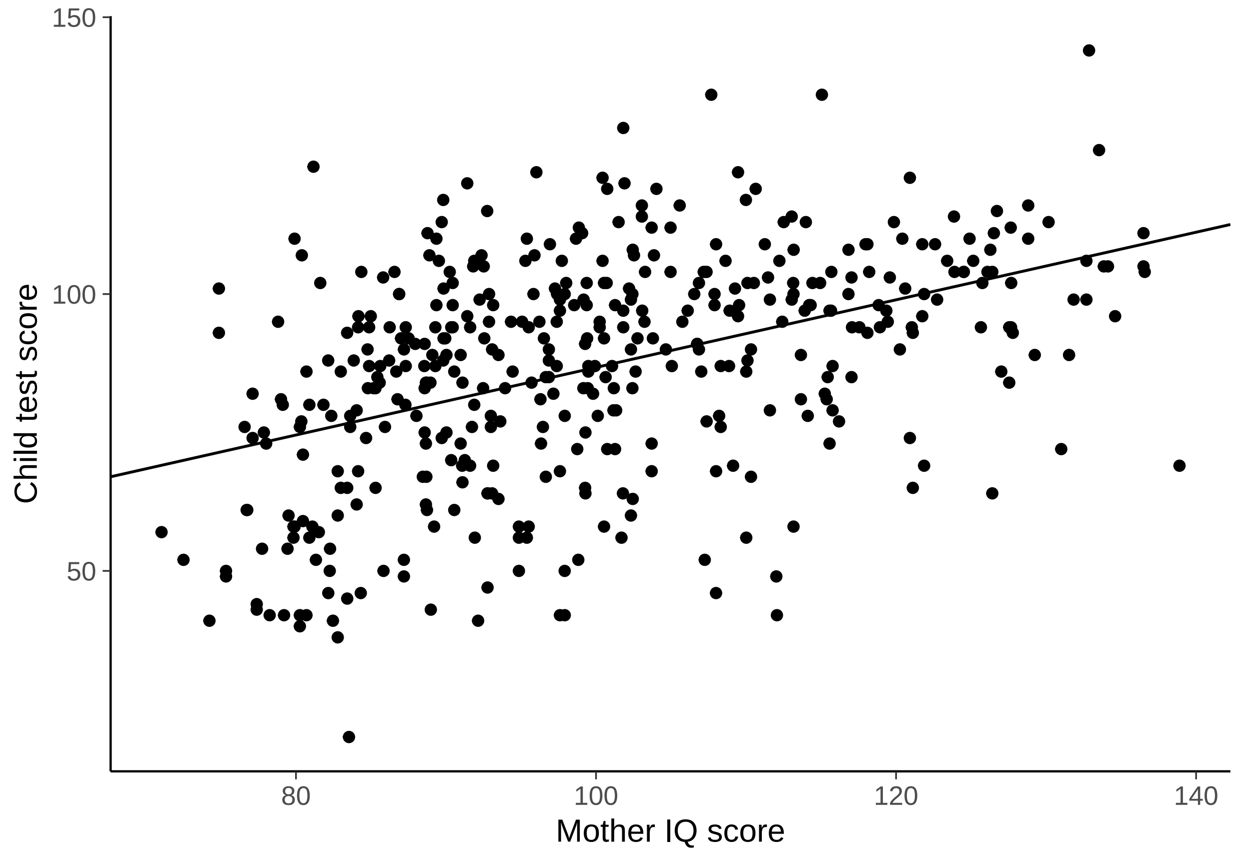
```r
fit_2 <- stan_glm(kid_score ~ mom_iq, data=kidiq, refresh = 0)
print(fit_2)
```

A single continuous predictor

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_iq
##  observations: 434
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 25.7    5.7
## mom_iq       0.6    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.3    0.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Displaying a regression line as a function of one input variable Represent only one input variable.

```r
ggplot(kidiq, aes(mom_iq, kid_score)) +
  geom_point() +
  geom_abline(intercept = coef(fit_2)[1], slope = coef(fit_2)[2]) +
  labs(x = "Mother IQ score", y = "Child test score")
```



Two predictors

```r
fit_3 <- stan_glm(kid_score ~ mom_hs + mom_iq, data=kidiq, refresh = 0)
print(fit_3)
```

Linear regression

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_hs + mom_iq
##  observations: 434
##  predictors:   3
## ------
##             Median MAD_SD
## (Intercept) 25.8    5.9
## mom_hs       6.0    2.2
## mom_iq       0.6    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.2    0.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
summary(fit_3)
```

Alternative display

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_hs + mom_iq
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 434
##  predictors:   3
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) 25.7    5.9 18.0  25.8  33.3
## mom_hs       6.0    2.2  3.2   6.0   8.7
## mom_iq       0.6    0.1  0.5   0.6   0.6
## sigma       18.2    0.6 17.4  18.2  19.0
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 86.8    1.2 85.2  86.8  88.4
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.1  1.0  3962
## mom_hs        0.0  1.0  4369
## mom_iq        0.0  1.0  4108
## sigma         0.0  1.0  4340
## mean_PPD      0.0  1.0  4247
## log-posterior 0.0  1.0  1693
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

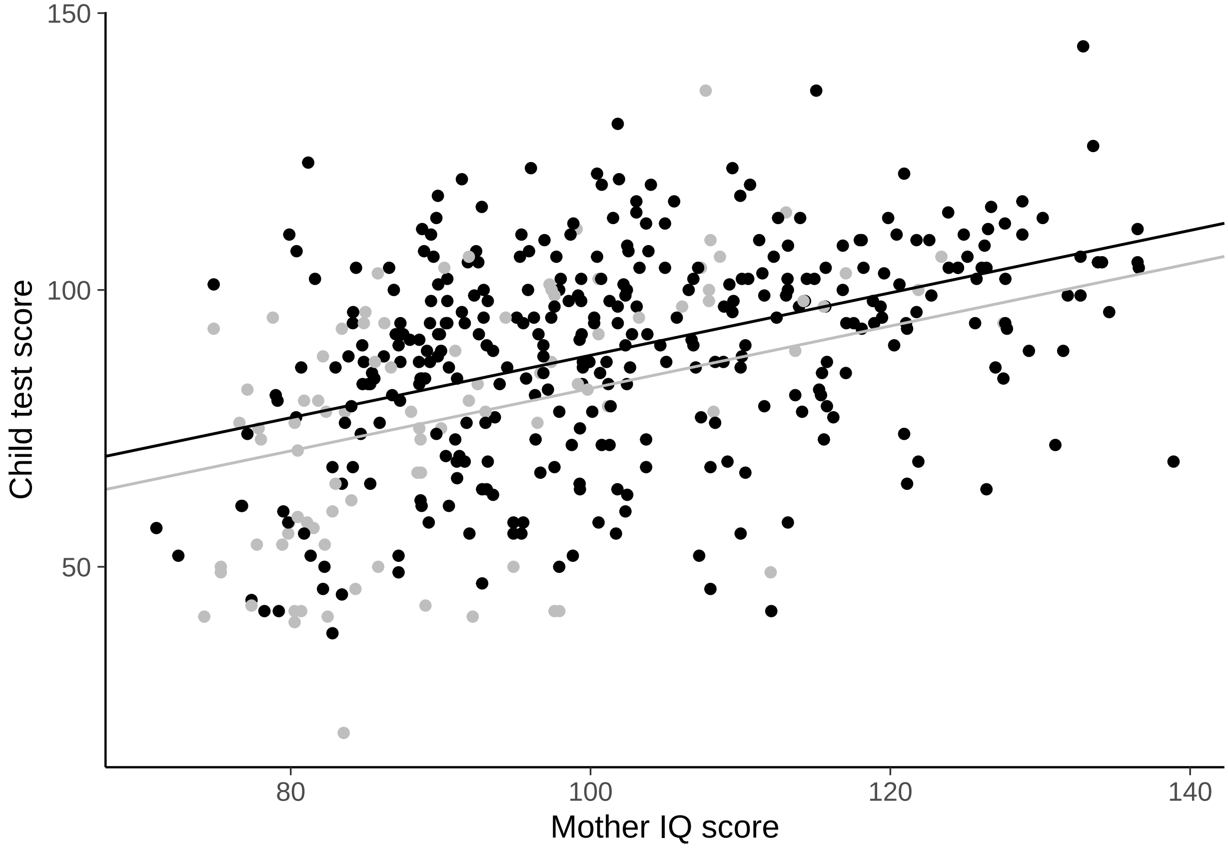
Graphical displays of data and fitted models

Two fitted regression lines – model with no interaction

ggplot version In the code below we bring in the estimated coefficients from the model fit above. We see we have no interaction.

```r
ggplot(kidiq, aes(mom_iq, kid_score)) +
  geom_point(aes(color = factor(mom_hs)), show.legend = FALSE) +
  geom_abline(
    intercept = c(coef(fit_3)[1], coef(fit_3)[1] + coef(fit_3)[2]),
    slope = coef(fit_3)[3],
    color = c("gray", "black")) +
```

```
  scale_color_manual(values = c("gray", "black")) +
  labs(x = "Mother IQ score", y = "Child test score")
```



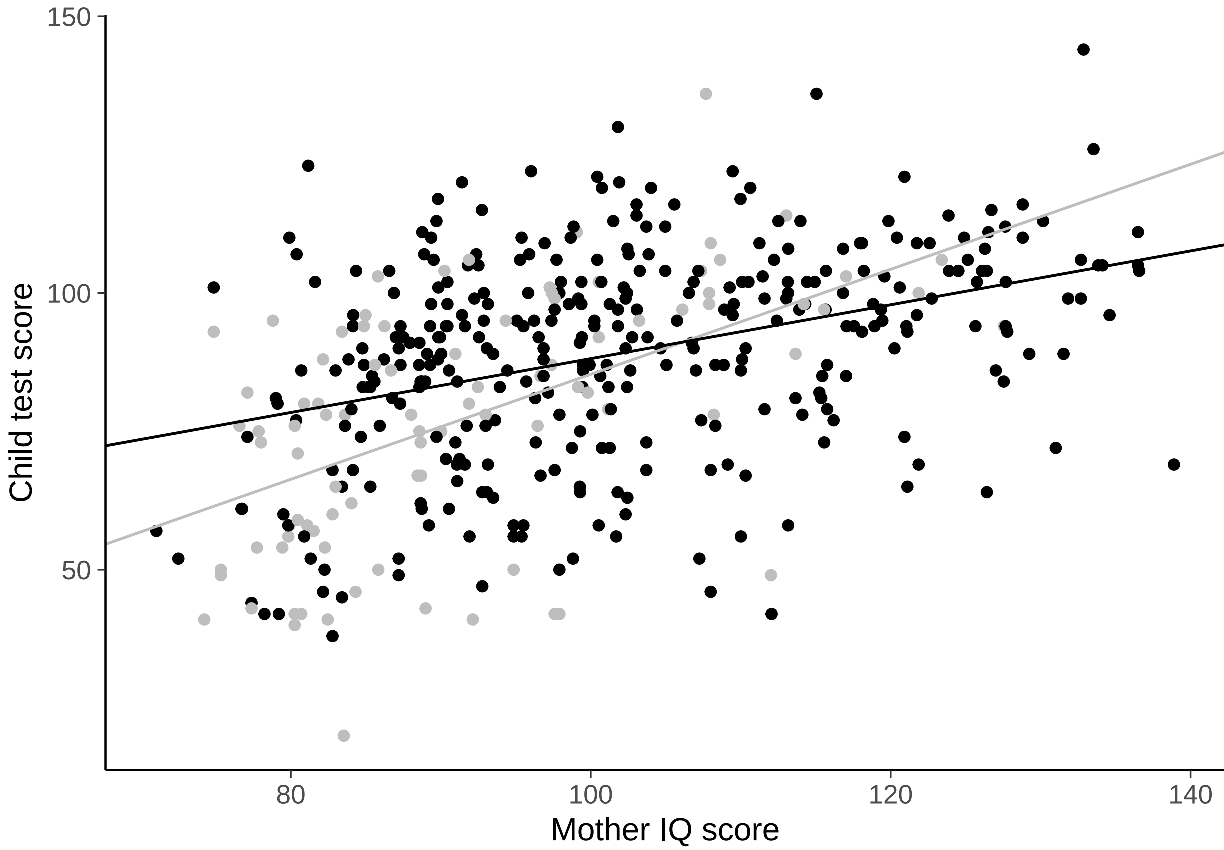

```
fit_4 <- stan_glm(kid_score ~ mom_hs + mom_iq + mom_hs:mom_iq, data=kidiq,
                  refresh = 0)
print(fit_4)
```

Two fitted regression lines – model with interaction

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_hs + mom_iq + mom_hs:mom_iq
##  observations: 434
##  predictors:   4
## ------
##               Median MAD_SD
## (Intercept)   -9.6   13.3
## mom_hs        49.1   14.7
## mom_iq         0.9    0.1
## mom_hs:mom_iq -0.5    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.0    0.6
##
```

```
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
ggplot(kidiq, aes(mom_iq, kid_score)) +
  geom_point(aes(color = factor(mom_hs)), show.legend = FALSE) +
  geom_abline(
    intercept = c(coef(fit_4)[1], sum(coef(fit_4)[1:2])),
    slope = c(coef(fit_4)[3], sum(coef(fit_4)[3:4])),
    color = c("gray", "black")) +
  scale_color_manual(values = c("gray", "black")) +
  labs(x = "Mother IQ score", y = "Child test score")
```



ggplot version

Displaying uncertainty in the fitted regression

Since when using stan_glm we simulate from the distribution for our estimated regression coefficients, we can use these simulatons to display this inferential uncertainty graphically. Consider the simple model with only mom_iq as a predictor.

```
print(fit_2)
```

A single continuous predictor

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_iq
##  observations: 434
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 25.7    5.7
## mom_iq       0.6    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.3    0.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

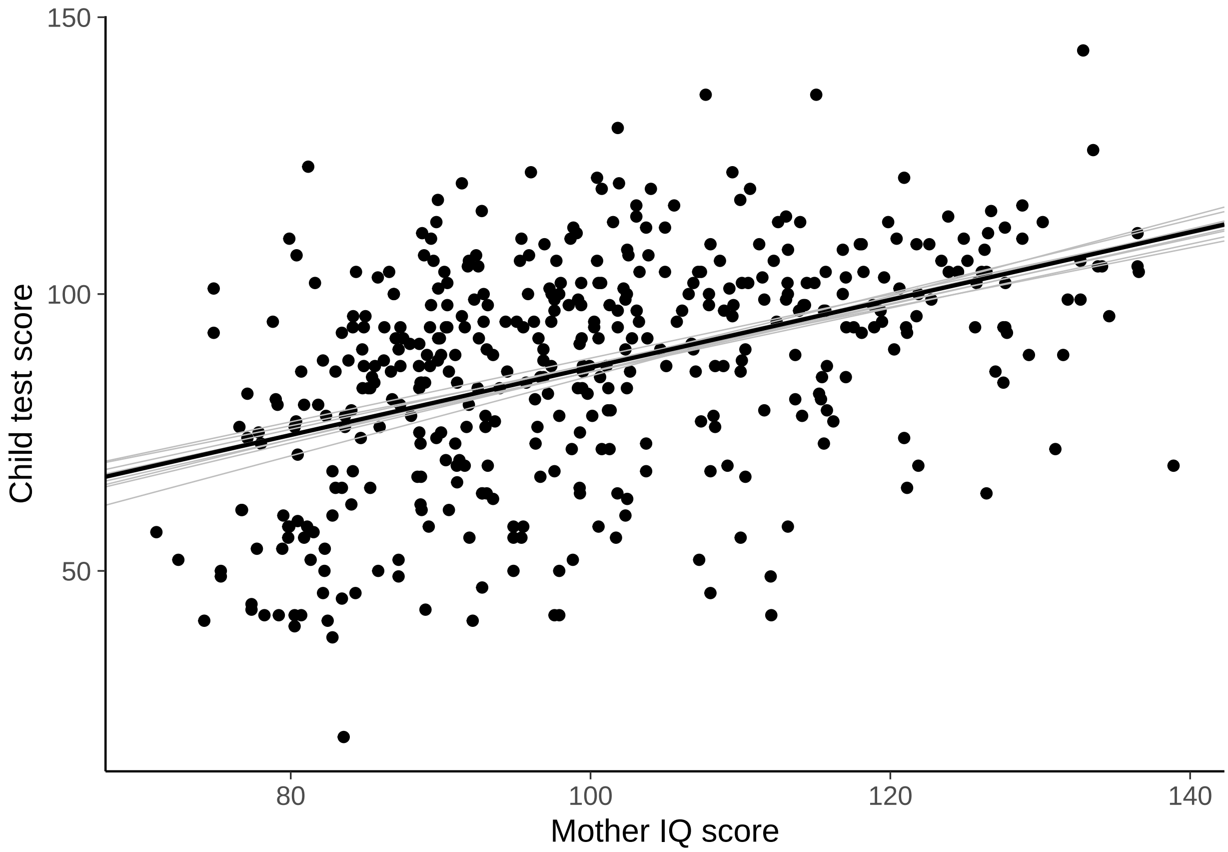
```r
sims_2 <- as.matrix(fit_2)
n_sims_2 <- nrow(sims_2)
subset <- sample(n_sims_2, 10) #random sample of 10
subset
```

```
##  [1]  964 2975 2888 3260 3819 2649  662 2019  751 1836
```

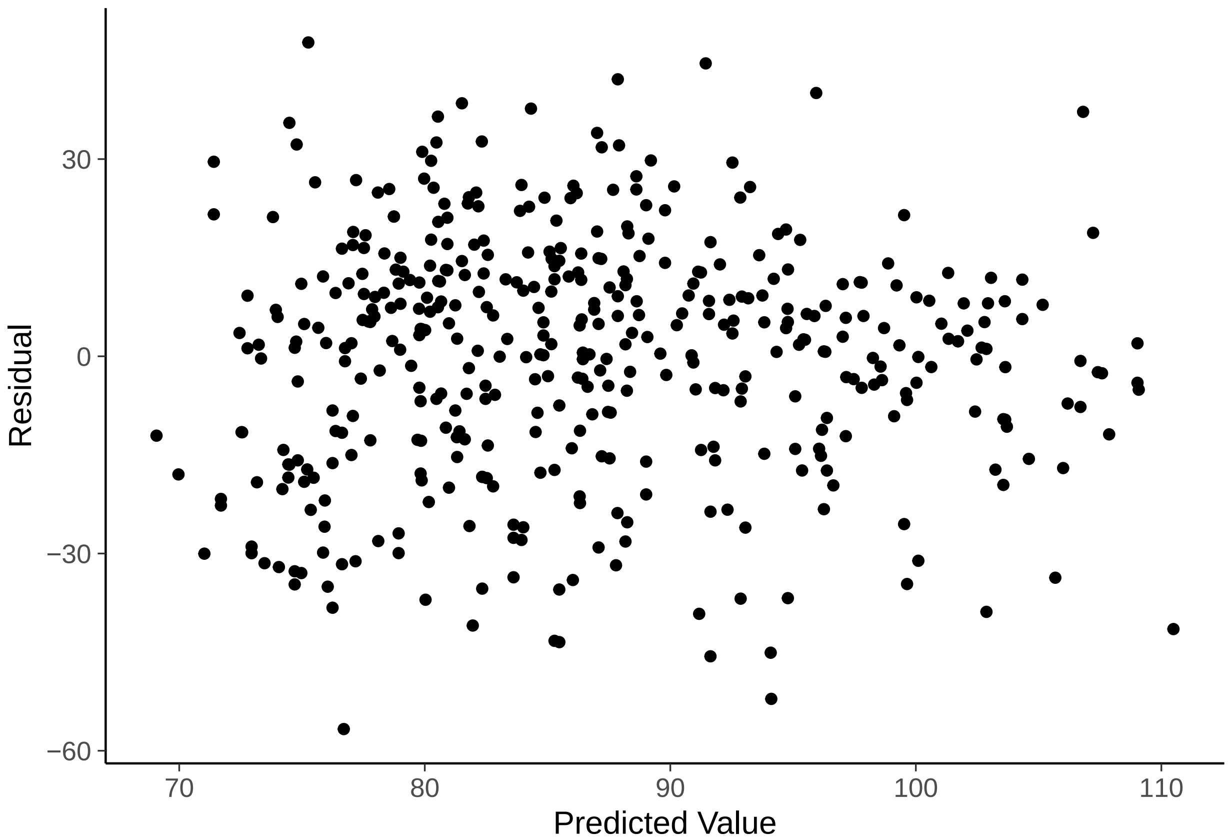
Data and regression of child's test score on maternal IQ with the solid line showing the fitted regression model and the light lines indicating uncertainty in the fitted regression line.

The gray lines are close to the line because we are illustrating variability in the estimation of the line, not individual level variability.

```r
ggplot(kidiq, aes(mom_iq, kid_score)) +
  geom_point() +
  geom_abline(
    intercept = sims_2[subset, 1],
    slope = sims_2[subset, 2],
    color = "gray",
    size = 0.25) +
  geom_abline(
    intercept = coef(fit_2)[1],
    slope = coef(fit_2)[2],
    size = 0.75) +
  labs(x = "Mother IQ score", y = "Child test score")
```




```
#predicted2 is the y-hats, predicted values, using model 2
#then we calculate the residuals as the actual value - the predicted value.
kidiq2 <- kidiq %>%
  mutate(predicted2=predict(fit_2), resid2=kid_score-predict(fit_2))
ggplot(kidiq2, aes(predicted2, resid2)) +
  geom_point() +
    labs(x = "Predicted Value", y = "Residual")
```



Notice it looks really cloud like which is what we want. We see no evidence of non-constant variance of systematic lack of fit.

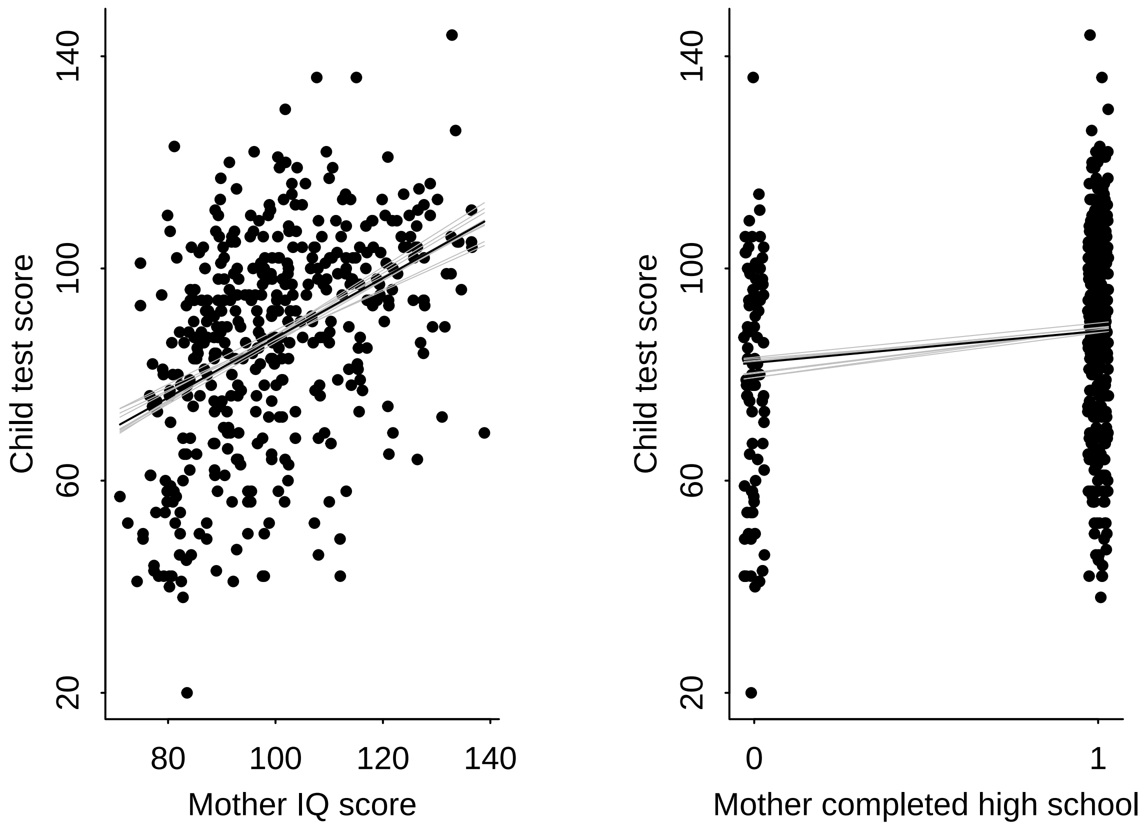
Two predictors In the plots below we use individual plots to illustrate the differences in one variable at the average value of the other.

Data and regression of child's tests score on maternal IQ and hight school completion, shown as a functions of each of the two input variables with the other held at its average value. Light lines indicate uncertainty in the regressions. Values for mother's high school completion have been jittered to make the points more distinct.

```
sims_3 <- as.matrix(fit_3)
n_sims_3 <- nrow(sims_3)

par(mar=c(3,3,1,3), mgp=c(1.7, .5, 0), tck=-.01)
par(mfrow=c(1,2))
plot(kidiq$mom_iq, kidiq$kid_score, xlab="Mother IQ score", ylab="Child test score", bty="l", pch=20, xa
axis(1, seq(80, 140, 20))
axis(2, seq(20, 140, 40))
mom_hs_bar <- mean(kidiq$mom_hs)
subset <- sample(n_sims_3, 10)
for (i in subset){
  curve(cbind(1, mom_hs_bar, x) %*% sims_3[i,1:3], lwd=.5,
     col="gray", add=TRUE)
}
curve(cbind(1, mom_hs_bar, x) %*% coef(fit_3), col="black", add=TRUE)
jitt <- runif(nrow(kidiq), -.03, .03)
plot(kidiq$mom_hs + jitt, kidiq$kid_score, xlab="Mother completed high school", ylab="Child test score"
```

```
axis(1, c(0,1))
axis(2, seq(20, 140, 40))
mom_iq_bar <- mean(kidiq$mom_iq)
for (i in subset){
  curve(cbind(1, x, mom_iq_bar) %*% sims_3[i,1:3], lwd=.5,
     col="gray", add=TRUE)
}
curve(cbind(1, x, mom_iq_bar) %*% coef(fit_3), col="black", add=TRUE)
```



Why are the lines so tight to the estimates when the data are so variable?

```
kidiq$c_mom_hs <- kidiq$mom_hs - mean(kidiq$mom_hs)
kidiq$c_mom_iq <- kidiq$mom_iq - mean(kidiq$mom_iq)
fit_4c <- stan_glm(kid_score ~ c_mom_hs + c_mom_iq + c_mom_hs:c_mom_iq,
                   data=kidiq, refresh = 0)
print(fit_4c)
```

Center predictors to have zero mean

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ c_mom_hs + c_mom_iq + c_mom_hs:c_mom_iq
##  observations: 434
##  predictors:   4
## ------
##                   Median MAD_SD
## (Intercept)       87.6    0.9
```

```
## c_mom_hs           2.9    2.4
## c_mom_iq           0.6    0.1
## c_mom_hs:c_mom_iq -0.5    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.0    0.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
kidiq$c2_mom_hs <- kidiq$mom_hs - 0.5
kidiq$c2_mom_iq <- kidiq$mom_iq - 100
fit_4c2 <- stan_glm(kid_score ~ c2_mom_hs + c2_mom_iq + c2_mom_hs:c2_mom_iq,
                    data=kidiq, refresh = 0)
print(fit_4c2)
```

Center predictors based on a reference point

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ c2_mom_hs + c2_mom_iq + c2_mom_hs:c2_mom_iq
##  observations: 434
##  predictors:   4
## ------
##                     Median MAD_SD
## (Intercept)         86.8    1.2
## c2_mom_hs            2.9    2.4
## c2_mom_iq            0.7    0.1
## c2_mom_hs:c2_mom_iq -0.5    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.0    0.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
fit_5 <- stan_glm(kid_score ~ as.factor(mom_work), data=kidiq, refresh = 0)
print(fit_5)
```

Predict using working status of mother

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ as.factor(mom_work)
##  observations: 434
##  predictors:   4
## ------
##                      Median MAD_SD
```

```
## (Intercept)          82.0    2.3
## as.factor(mom_work)2  3.8    3.1
## as.factor(mom_work)3 11.5    3.5
## as.factor(mom_work)4  5.3    2.7
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 20.3    0.7
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
print(fit_2)
```

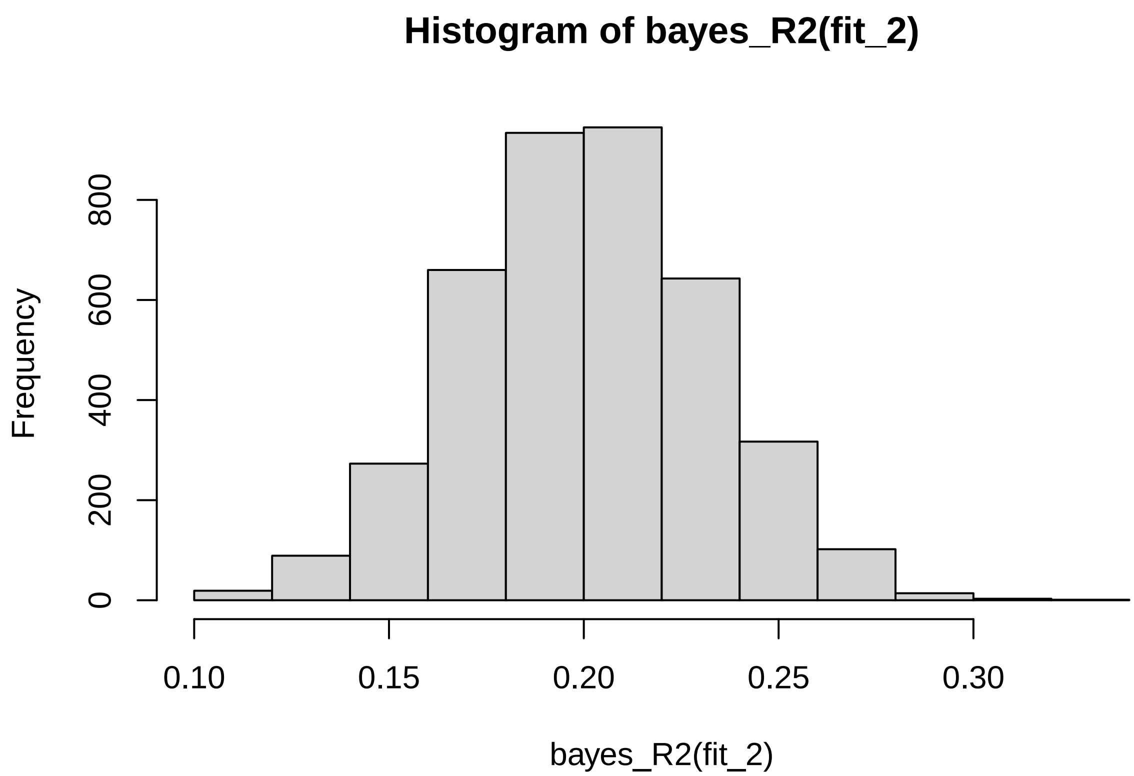
What about R-square if we use stan_glm

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      kid_score ~ mom_iq
##  observations: 434
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 25.7    5.7
## mom_iq       0.6    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 18.3    0.6
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
sims=as.matrix(fit_2)
quantile(sims[,2],c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.4989200 0.7208003
```

```r
hist(bayes_R2(fit_2))
```




```r
median(bayes_R2(fit_2))
```

```
## [1] 0.2005748
```

```r
lm_fit_2 <- lm(kid_score ~ mom_iq , data=kidiq)
summary(lm_fit_2) #traditional output
```

```
##
## Call:
## lm(formula = kid_score ~ mom_iq, data = kidiq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.753 -12.074   2.217  11.710  47.691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.79978    5.91741    4.36 1.63e-05 ***
## mom_iq       0.60997    0.05852   10.42  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.27 on 432 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.1991
## F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

```r
confint(lm_fit_2)
```

```
##                  2.5 %     97.5 %
## (Intercept) 14.1692789 37.4302768
## mom_iq       0.4949534  0.7249957
```