STA2201H Methods of Applied Statistics II

Monica Alexander

Week 5: Bayesian regression and Stan

Annoucements

- Assignment 1 being graded
- Assignment 2 coming soon (Bayesian inference)

Where are we at

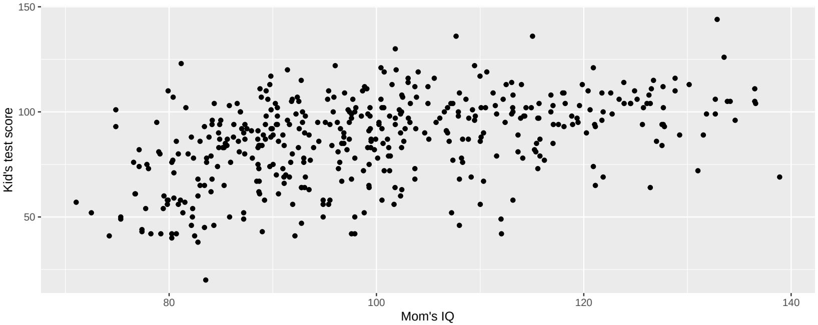
- Bayesian inference revolves around inference based on the posterior
- Posterior usually hard to write down in closed form
- But as long as we can get a set of samples from posterior, we can do inference
- For most problems, we can construct an MCMC algorithm that can be used to generate samples from posterior distributions
- lots of standard software to run MCMC so that we (usually) don't have to code it ourselves
- We will be using Stan, which fits models using a version of HMC

Bayesian inference for regression models

Kid's scores

- Outcome is Kid's test scores
- Let's introduce a covariate/explanatory variable of Mom's IQ

1) Question / goal : Describe the association between kid's test scores and Mom's IQ



Scientific model

2) What is the Scientific model (how are these observed data generated?)

How does Mom's IQ influence Kid's score? If we think about this relationship causally

$$X \rightarrow Y$$

- Changing Mom's IQ would change Kid's test score, but not the other way around
- This is a scientific claim

Scientific model

Adding another piece to our scientific model

$$X \to Y \leftarrow U$$

"Kid's score is a function of Mom's IQ and other stuff" This implies we need to find some function $Y = f(X, U)$. Let's assume Kid's score is a proportion of Mom's IQ plus the influence of unobserved causes

Statistical model

A reasonable model to consider is

$$y_i|\mu_i, \sigma \sim N\left(\mu, \sigma^2\right)$$
$$\mu_i = \alpha + \beta x_i$$

where X_i is mother's IQ score. This is a simple linear regression model. We are primarily interested in obtaining estimates for the regression coefficients, α and β.

We need to put priors on σ (as before) but also α and β. Let's put

$$\alpha \sim N(0, 100^2)$$
$$\beta \sim N(0, 10^2)$$
$$\sigma \sim \text{Half-Normal}(0, 10)$$

Bayesian regression

- OLS or MLE finds estimates of the parameters that best fit the data
- Bayesian inference incorporates prior information about the parameters
- In Bayesian inference, the estimates are a compromise between the prior info and the data

Bayesian inference for linear regression

What does Bayesian inference get us that MLE doesn't?

- **Inclusion of prior information**:
  - we usually know something
  - makes inferences more stable, as the estimates are typically somewhere between the prior and what would be obtained from the data alone
- **Propogation of uncertainty**:
  - least squares gives us a point estimate
  - in Bayesian inference, we can summarize uncertainty using simulations from the posterior distribution

Posterior distribution

$$\Pr(\alpha, \beta, \sigma \mid Y_i, X_i) = \frac{\Pr(Y_i \mid X_i, \alpha, \beta, \sigma)\Pr(\alpha, \beta, \sigma)}{Z}$$

- Note that α and β describe the line (conditional expectation) and σ describes the variation around the line

Prior predictive distributions

- Priors should express scientific knowledge, but “softly”
- Sigma must be positive
- Kid score on average increases with Mom IQ?
- ???

Idea of prior predictive distributions:

- We can understand the implications of priors through simulation: check that before the model sees data, it doesn’t hallucinate impossible things.
- We can force the model to make predictions even before data.

Prior predictive distributions

- If we specify proper priors for all parameters in the model, our model is **generative**
- Yields a joint prior distribution on the parameters and data, and hence a prior marginal distribution for the data

Prior predictive distribution for new $\tilde{y}$

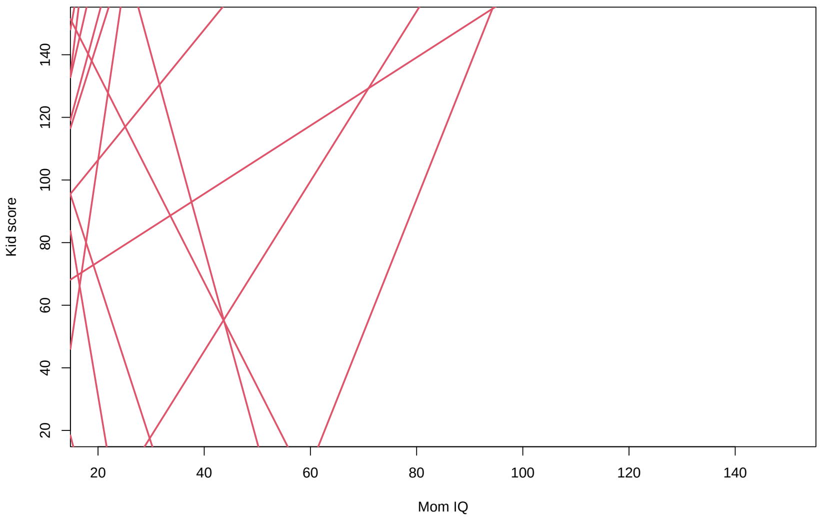
$$p(\tilde{y}) = \int_{\Theta} p(\tilde{y}, \theta) d\theta = \int_{\Theta} p(\tilde{y}|\theta) p(\theta) d\theta$$

In practice (in R) we can simulate values of θ from the prior distribution(s), and then simulate from the likelihood to generate values of $\tilde{y}$, and then look at the resulting distribution.

For now, I'm just going to generate values of the conditional expectation/linear predictor.

Make some lines

```
n <- 1000
alpha <- rnorm(n, 0, 100)
beta <- rnorm(n, 0, 10)
plot(NULL, xlim=c(20, 150), ylim = c(20, 150), xlab = "Mom IQ", ylab = "Kid score")
for (j in 1:50) abline(a = alpha[j], b = beta[j], col = 2, lwd = 2)
```



Sermon on priors (from Stat Rethinking)

- There are no correct priors, only scientifically justifiable priors
- Justify with information outside the data, like the rest of the model (eg the generative model)
- Priors are not so important in simple models
- Very important/useful in complex models
- Need to simulate and understand behavior

In Stan

```
data {
  int<lower=0> N;              // number of kids
  int<lower=0> K;              // number of covariates
  vector[N] y;                 // scores
  matrix[N, K] X;               // design matrix
}
parameters {
  real alpha;
  vector[K] beta;
  real<lower=0> sigma;
}
transformed parameters {
}
model {
  //priors
  alpha ~ normal(0, 100);
  beta ~ normal(0, 1);
  sigma ~ normal(0,1);

  //likelihood
  y ~ normal(alpha + X*beta, sigma);
}
```

Fits comparison

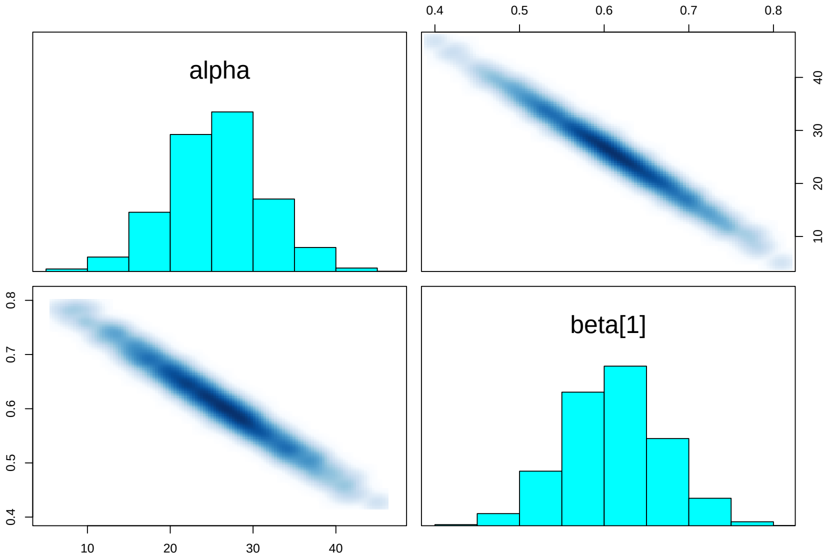
```
summary(fit)$summary[c("alpha", "beta[1]"),]
```

```
##                mean     se_mean         sd       2.5%        25%        50%
## alpha   25.6505410 0.160003162 5.86313286 13.9089387 21.7385364 25.7168057
## beta[1]  0.6113883 0.001591822 0.05795296  0.4977733  0.5744605  0.6107212
##                75%      97.5%    n_eff     Rhat
## alpha   29.3746990 37.3517609 1342.772 1.000457
## beta[1]  0.6493292  0.7283271 1325.446 1.000478
```

```
summary(lm(kid_score~mom_iq, data = kidiq))
```

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



What do we get

```r
post_samples <- extract(fit)
length(post_samples)
```

```
## [1] 4
```

```r
names(post_samples)
```

```
## [1] "alpha" "beta"  "sigma" "lp__"
```

What do we get

```r
dim(post_samples[["alpha"]])
```

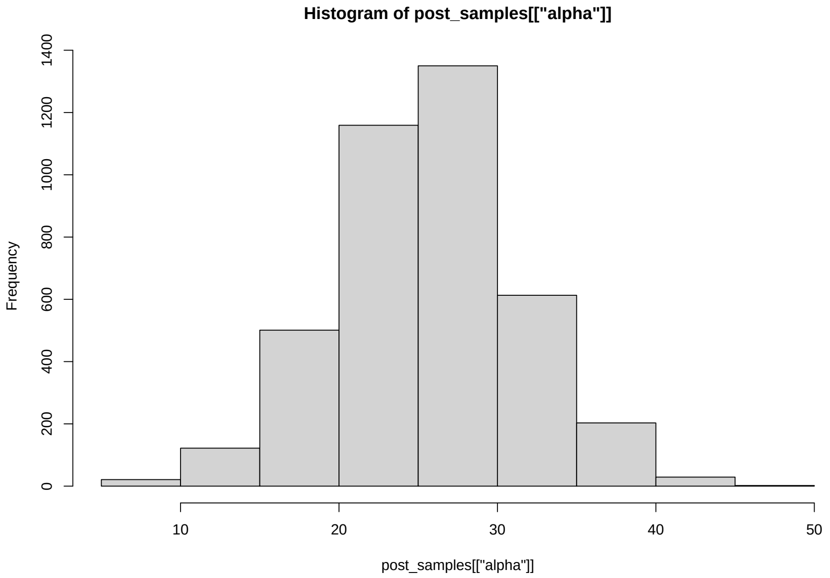
```
## [1] 4000
```

```r
post_samples[["alpha"]][1:5]
```

```
## [1] 27.58892 36.47942 17.22575 23.33283 28.06029
```

What do we get

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



Tidy version

```
library(tidybayes)
fit |>
  gather_draws(alpha)
```

```
## # A tibble: 4,000 x 5
## # Groups:   .variable [1]
##    .chain .iteration .draw .variable .value
##     <int>      <int> <int> <chr>      <dbl>
##  1      1          1     1 alpha       26.0
##  2      1          2     2 alpha       26.5
##  3      1          3     3 alpha       26.3
##  4      1          4     4 alpha       29.4
##  5      1          5     5 alpha       30.6
##  6      1          6     6 alpha       23.3
##  7      1          7     7 alpha       20.9
##  8      1          8     8 alpha       20.4
##  9      1          9     9 alpha       22.4
## 10      1         10    10 alpha       20.7
## # ... with 3,990 more rows
```

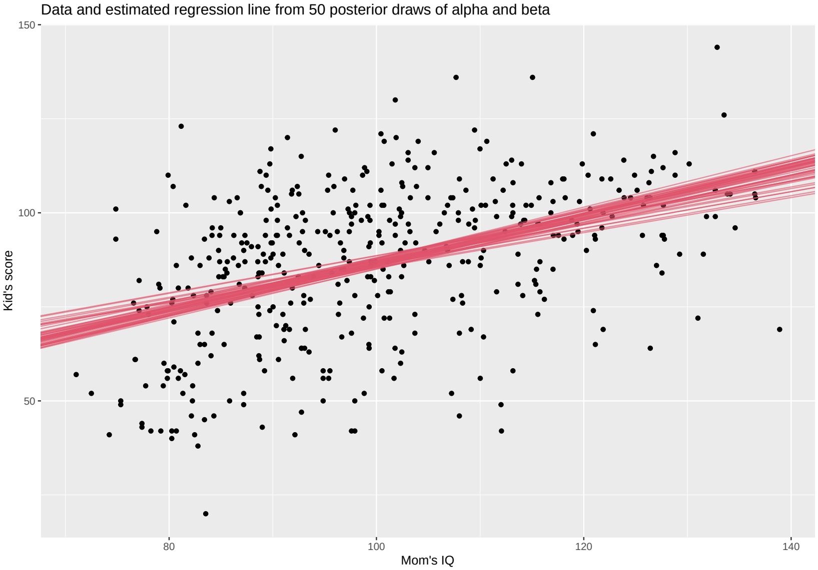
What can we do

- The data and model are combined to form a posterior distribution, which we typically summarize by a set of simulations of the parameters in the model
- We can propagate uncertainty in this distribution, that is, we can get simulation-based prediction for unobserved or future outcomes that accounts for uncertainty in the model parameters

With simulations, we can

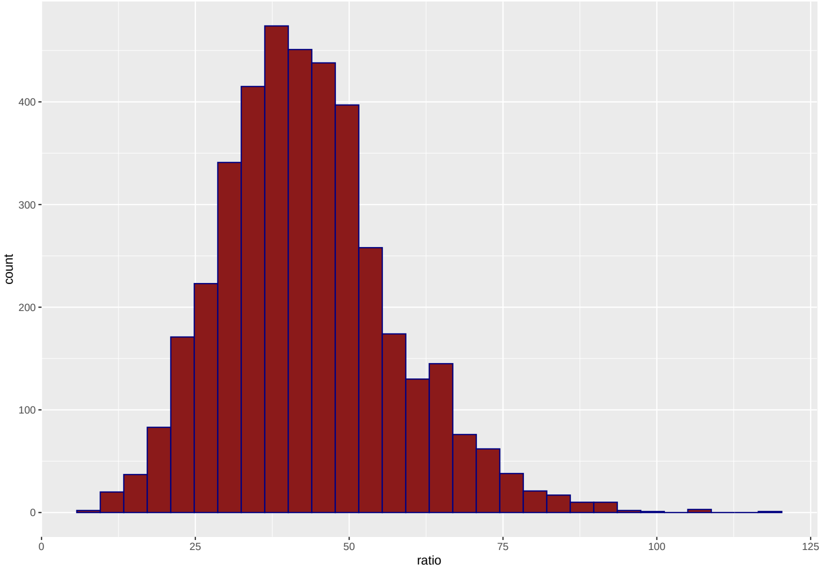
- Visualize uncertainty in the regression line
- Get uncertainty for functions of parameters
- Make predictions based on new data points

The posterior is full of lines



Uncertainty about a function of parameters

For example, posterior samples for the ratio of α and β



Making predictions

Consider making a prediction of kid's score with a new observation of mother's IQ, x^{new}. We have

- the point prediction $\hat{\alpha} + \hat{\beta}x^{\text{new}}$
- the linear predictor with uncertainty $\alpha + \beta x^{\text{new}}$
  - propagates uncertainty in regression coefficients
  - represents the distribution of uncertainty about the expected value of y for new data points x^{new}
- the predictive distribution for a new observation $\alpha + \beta x^{\text{new}} +$ error
  - represents uncertainty about a new observation y with predictor x^{new}

Predictions

Consider a new mother with an IQ of 110.

Point prediction: use medians of posterior samples for $\hat{\alpha}$ and $b\hat{e}ta$

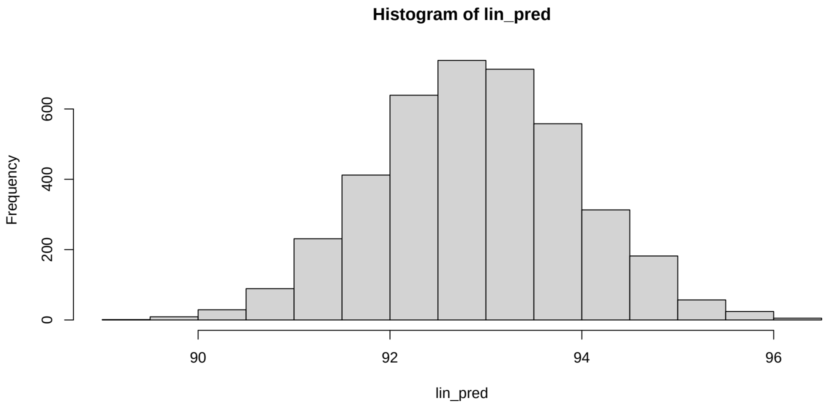
```r
x_new <- 110
alpha_hat <- median(post_samples[["alpha"]])
beta_hat <- median(post_samples[["beta"]])

alpha_hat + beta_hat*x_new
```

```
## [1] 92.89614
```

Predictions

Linear predictor with uncertainty:

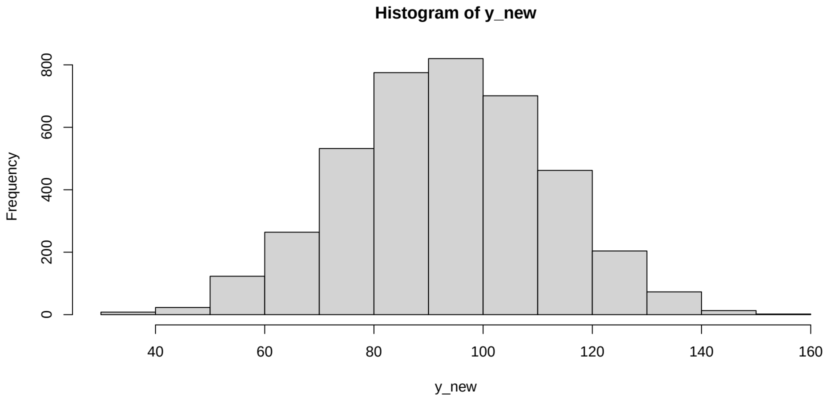
```
alpha <- post_samples[["alpha"]]
beta <- post_samples[["beta"]][,1]

lin_pred <- alpha + beta*x_new
hist(lin_pred)
```



Predictions

Predictive distribution for new observation:

```
sigma <- post_samples[["sigma"]]
y_new <- rnorm(n = length(sigma),mean = lin_pred, sd = sigma)
hist(y_new)
```



Can also do this within Stan

Can get posterior predictive distribution samples using the generated quantities block:

```
generated quantities{
  real y_new[1];
  y_new = normal_rng(alpha + x_new*beta, sigma);
}
```

Posterior predictive distribution

$$p(\tilde{y}|y) = \int_{\Theta} p(\tilde{y}|\theta, y)p(\theta|y)d\theta$$

- After we have seen the data and obtained the posterior distributions of the parameters, we can now use the posterior distributions to generate new data from the model.
- Given the posterior distributions of the parameters of the model, the posterior predictive distribution gives us some indication of what new data might look like, given the data and model.
- We can avoid performing the integration explicitly by generating samples from the posterior predictive distribution.

Posterior predictive distributions also important for model checking. More next week.

Posterior predictive distribution

Posterior predictive distribution for new $\tilde{y}$

$$p(\tilde{y}|y) = \int_{\Theta} p(\tilde{y}|\theta, y)p(\theta|y)d\theta$$

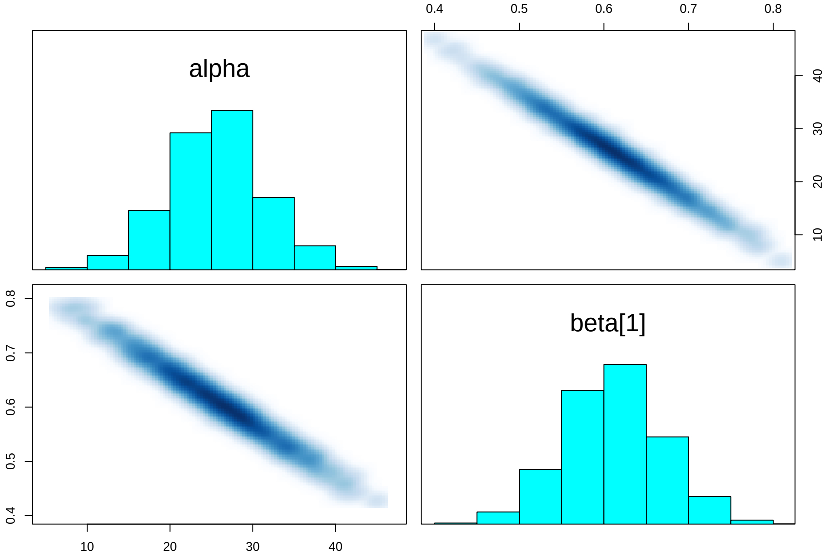
To obtain samples from this distribution, we need to

- Get posterior samples of our parameters $\theta^{(s)}$ (MCMC output!)
- For each posterior sample, we obtain one replicated dataset $\tilde{y}^{(s)}$ by sampling from the likelihood $p(\tilde{y}|\theta^{(s)})$. Can do this in R or within Stan.

Centering predictors to improve posterior geometries

Remember this

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



Centering

```
data <- list(y = y,
             N = length(y),
             K = 1,
             X = as.matrix(kidiq$mom_iq - mean(kidiq$mom_iq)))
fit2 <- stan(file = "kids3.stan",
            data = data)
```

Summary of fit

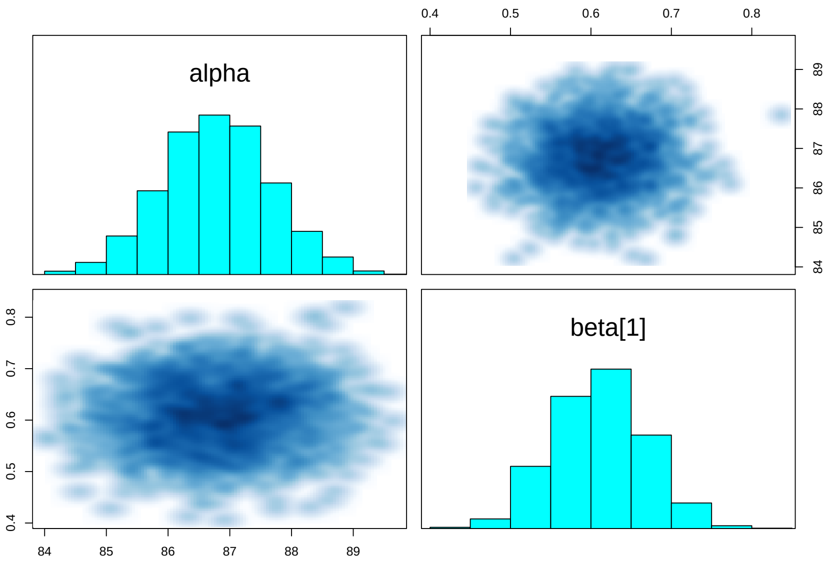
```
summary(fit2)$summary[c("alpha", "beta[1]"),]
```

```
##                mean      se_mean         sd       2.5%        25%
## alpha    86.7892184 0.0132815304 0.88159102 85.0690380 86.186347 86.7
## beta[1]   0.6110352 0.0009157762 0.05767691  0.5021104  0.570862  0.6
##                 75%      97.5%    n_eff      Rhat
## alpha    87.3671882 88.5772355 4405.935 0.9993431
## beta[1]   0.6503686  0.7207103 3966.663 1.0000865
```

What's different? What's the same?

Now look at joint posteriors

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



What do you notice? Why does this matter?

Centering predictors

- When the mean of the predictors is far away from zero, changes in the slope induce the opposite change in the intercept
- Hard to interpret what intercepts mean
- Harder to sample: reducing correlation may speed up convergence

Changing prior information

Changing prior information

What if we knew with relative certainty that there's a 1:1 correspondence between kid's score and mother's IQ? How would we encode this information?

Changing prior information

$$\beta \sim N(1, 0.01^2)$$

Let's fit this:

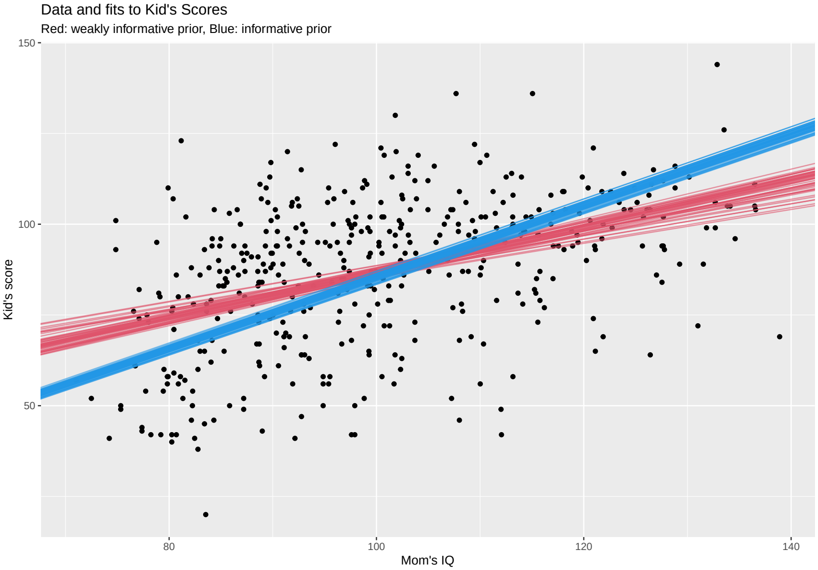
```
data <- list(y = y,
             N = length(y),
             K = 1,
             X = as.matrix(kidiq$mom_iq - mean(kidiq$mom_iq)))
fit3 <- stan(file = "kids5.stan",
            data = data)
```

Summary of fit

```
summary(fit3)$summary[c("alpha", "beta[1]"),]
```

```
##                mean      se_mean          sd       2.5%        25%        50%
## alpha   86.7825087 0.0118031292 0.728263738 85.3494689 86.2973942 86.7970300
## beta[1]  0.9845948 0.0001374247 0.009826061  0.9657328  0.9778366  0.9846424
##                75%     97.5%    n_eff      Rhat
## alpha   87.2743635 88.225777 3807.002 1.0004073
## beta[1]  0.9912788  1.003631 5112.450 0.9995584
```

Comparison with weakly informative priors



Comments

- Okay, maybe this was a bad decision in this context, but when might we want to consider more informative priors?
- Measurement error?
- Less data?
- Previous evidence?

Break the model

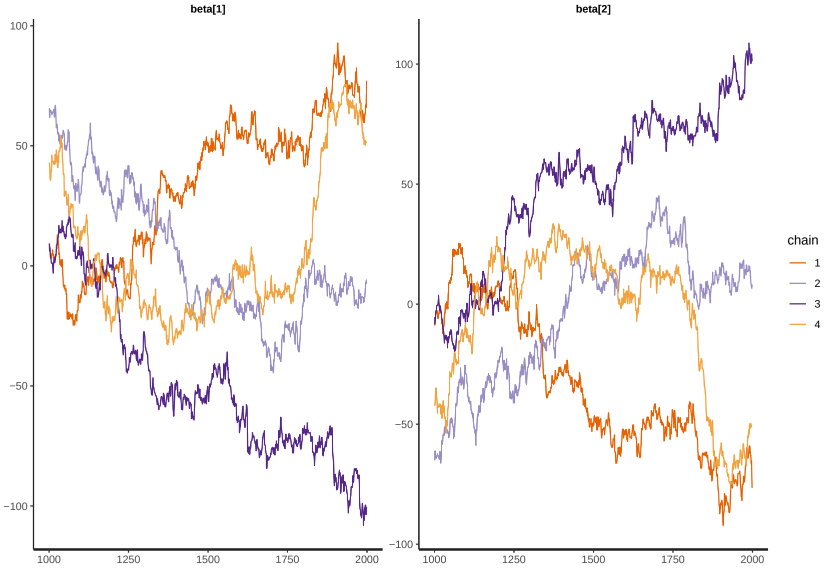
$$y_i|\mu_i, \sigma \sim N\left(\mu, \sigma^2\right)$$

$$\mu_i = \alpha + \beta_1 x_i + \beta_2 x_i$$

Priors on β are improper: $p(\beta) \propto 1$

```
data <- list(y = y,
             N = length(y),
             K = 2,
             X = cbind(as.matrix(kidiq$mom_iq), as.matrix(kidiq$mom_iq)))
fit4 <- stan(file = "kid6.stan",
            data = data)
```

```
## Inference for Stan model: kid6.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##              mean se_mean    sd     2.5%      25%      50%      75%    97.5%
## alpha       25.20    0.85  4.96    15.61    21.81    25.42    28.21    36.22
## beta[1]     -1.47   19.41 42.31   -84.98   -22.88    -5.13    32.12    71.61
## beta[2]      2.09   19.40 42.29   -70.99   -31.47     5.72    23.50    85.67
## sigma       14.93    0.05  0.30    14.35    14.72    14.93    15.14    15.52
## lp__     -1606.74    0.16  1.18 -1609.82 -1607.20 -1606.42 -1605.93 -1605.46
##         n_eff Rhat
## alpha      34 1.08
## beta[1]     5 2.08
## beta[2]     5 2.08
## sigma      40 1.11
## lp__       52 1.09
##
## Samples were drawn using NUTS(diag_e) at Wed Feb  8 08:07:13 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).
```


beta[1]
beta[2]
100
50
0
−50
−100
1000
1250
1500
1750
2000
chain
1
2
3
4

Compare to weakly informative priors

Priors on β are $\beta \sim N(0, 1)$

```
data <- list(y = y,
             N = length(y),
             K = 2,
             X = cbind(as.matrix(kidiq$mom_iq), as.matrix(kidiq$mom_iq)))
fit5 <- stan(file = "kids3.stan",
            data = data)
```

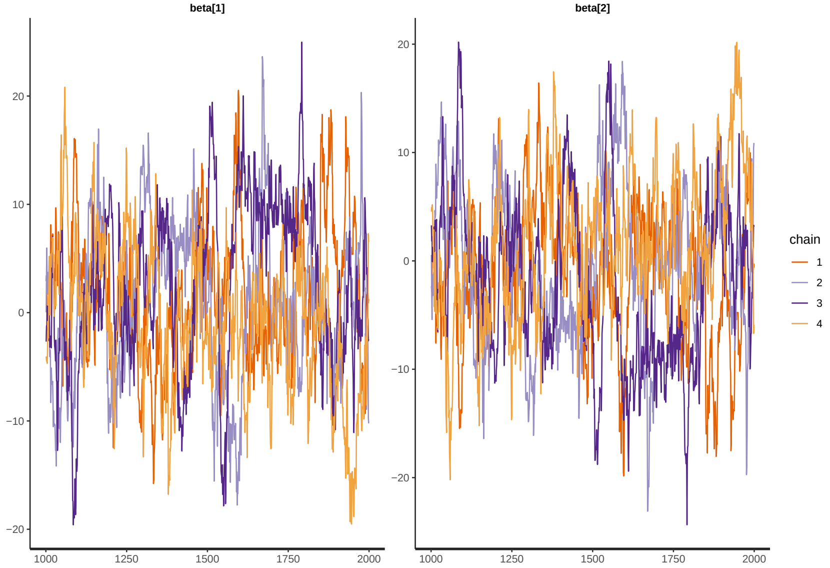
What do you think will happen?

Results

What is identifiable given the observed data?

```
## Inference for Stan model: kids3.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##              mean se_mean   sd     2.5%      25%      50%      75%    97.5%
## alpha       25.81    0.55 5.79    13.80    22.01    25.92    29.67    36.87
## beta[1]      1.09    0.80 6.97   -12.55    -3.62     0.99     5.94    14.66
## beta[2]     -0.48    0.79 6.97   -14.03    -5.31    -0.39     4.22    13.18
## sigma       18.27    0.06 0.65    17.08    17.83    18.22    18.66    19.65
## lp__     -1477.58    0.08 1.44 -1481.29 -1478.23 -1477.23 -1476.55 -1475.84
##          n_eff Rhat
## alpha      112 1.05
## beta[1]     77 1.07
## beta[2]     77 1.07
## sigma      127 1.04
## lp__       337 1.01
##
## Samples were drawn using NUTS(diag_e) at Wed Feb  8 08:09:11 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).
```

Traceplots



Summary

In any modeling problem:

1) Question/Goal
2) Scientific model
3) Statistical model

Bayesian inference for linear regression

- Focus on simulation-based inference and prediction, rather than point estimates
- Can simulate predictions even before seeing data
- Easy to propagate uncertainty to predictions, functions of estimated parameters

Lab: practice with kids dataset