STA2201H Methods of Applied Statistics II

Monica Alexander

Week 5: Bayesian regression and Stan

Annoucements

- Assignment 1 being graded
- Assignment 2 coming soon

Where are we at

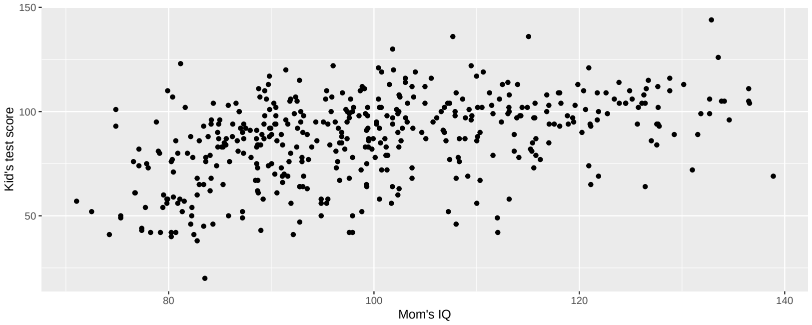
- Bayesian inference revolves around inference based on the posterior
- Posterior usually hard to write down in closed form
- But as long as we can get a set of samples from posterior, we can do inference
- For most problems, we can construct an MCMC algorithm that can be used to generate samples from 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

$$\alpha \sim N(0, 100^2)$$
$$\beta \sim N(0, 10^2)$$
$$\sigma \sim \text{Half-Normal}(0, 1)$$

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 $\tilde{y}$

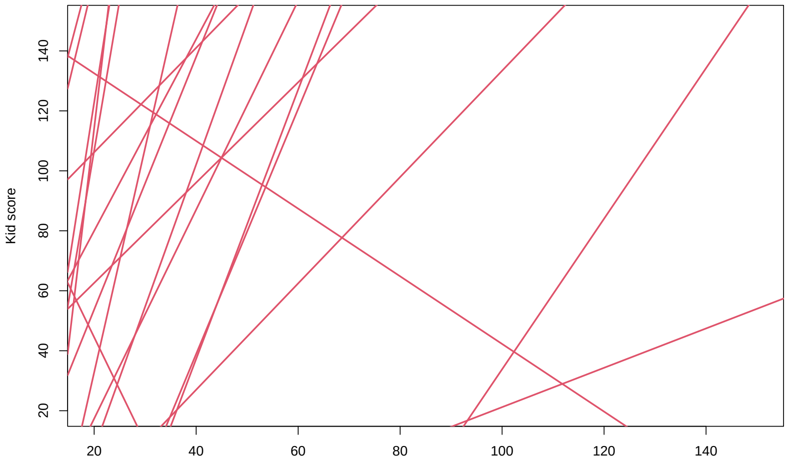
$$p(\tilde{y}) = \int_{\Theta} p(\tilde{y}, \theta) d\theta = \int_{\Theta} p(\tilde{y}|\theta) p(\theta) d\theta$$

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

```r
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

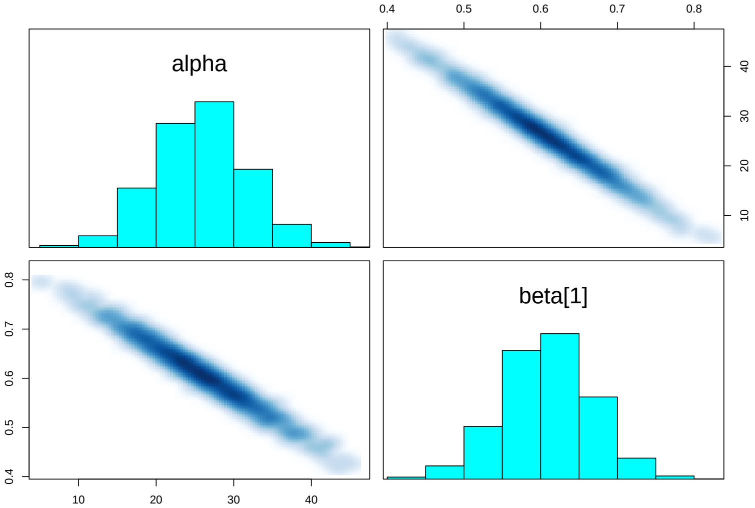
```
summary(fit)$summary[c("alpha", "beta[1]"),]
```

```
##                mean     se_mean         sd       2.5%        25%        50%
## alpha   25.9021707 0.167995802 5.90152220 14.4842607 21.9706545 25.9697777
## beta[1]  0.6087277 0.001649548 0.05814357  0.4900049  0.5704404  0.6081894
##                 75%      97.5%    n_eff     Rhat
## alpha   29.8030582 37.6457406 1234.046 1.004106
## beta[1]  0.6476124  0.7212384 1242.434 1.004011
```

```
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

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

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

```
names(post_samples)
```

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

What do we get

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

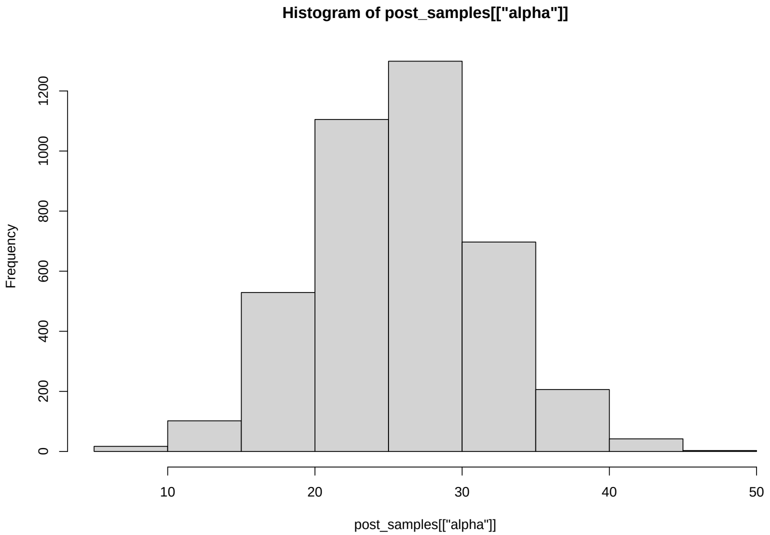
```
## [1] 4000
```

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

```
## [1] 31.40069 27.90220 30.84657 25.99818 25.61134
```

What do we get

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



Tidy version

```r
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       29.0
##  2      1          2     2 alpha       34.6
##  3      1          3     3 alpha       24.8
##  4      1          4     4 alpha       29.2
##  5      1          5     5 alpha       25.7
##  6      1          6     6 alpha       21.0
##  7      1          7     7 alpha       20.8
##  8      1          8     8 alpha       20.1
##  9      1          9     9 alpha       29.8
## 10      1         10    10 alpha       17.6
## # i 3,990 more rows
```

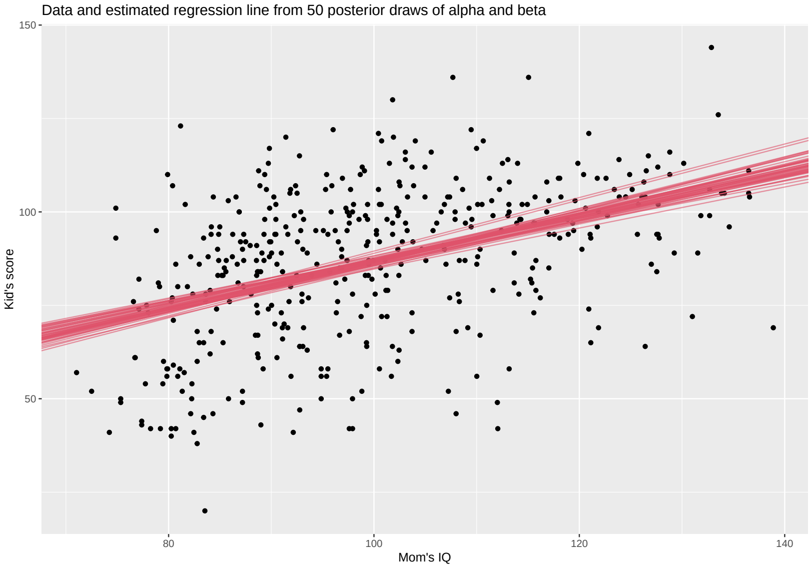
What can we do

- 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

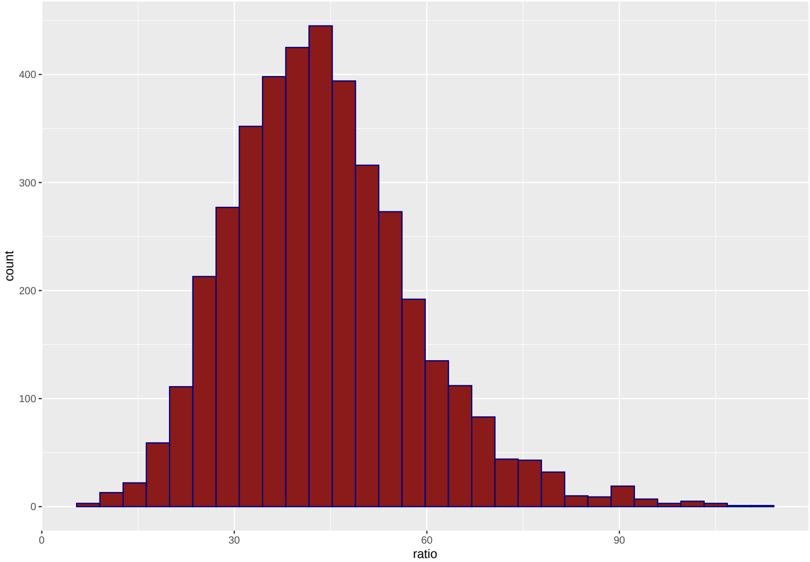
- Visualize uncertainty in the regression line
- Get uncertainty for functions of parameters
- Make predictions based on new data points

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 $\hat{\beta}$

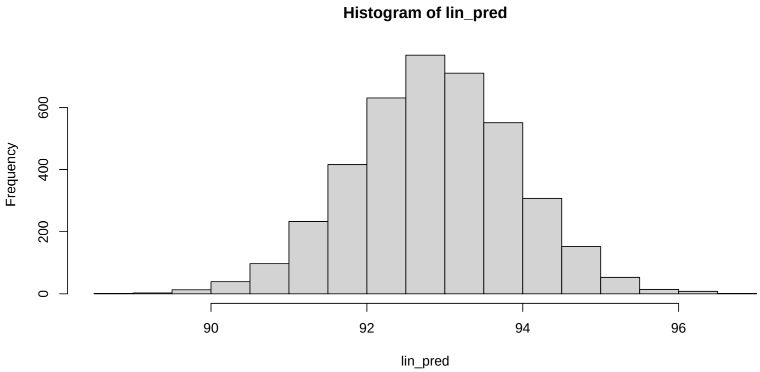
```
x_new <- 110
alpha_hat <- median(post_samples[["alpha"]])
beta_hat <- median(post_samples[["beta"]])

alpha_hat + beta_hat*x_new
```

```
## [1] 92.87062
```

Predictions

Linear predictor with uncertainty:

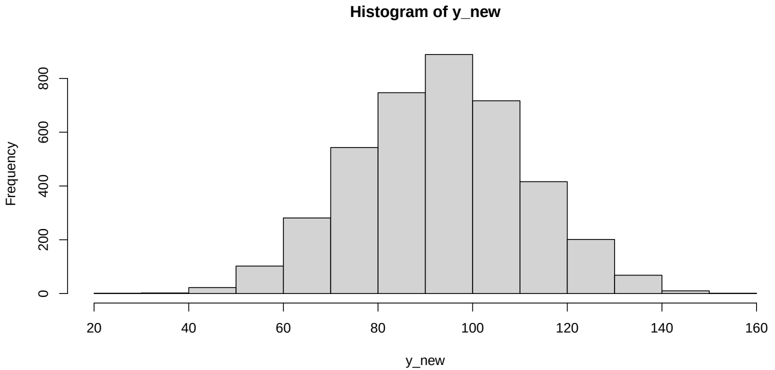
```
alpha <- post_samples[["alpha"]]
beta <- post_samples[["beta"]][,1]

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



Predictions

Predictive distribution for new observation:

```r
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 $\tilde{y}$

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

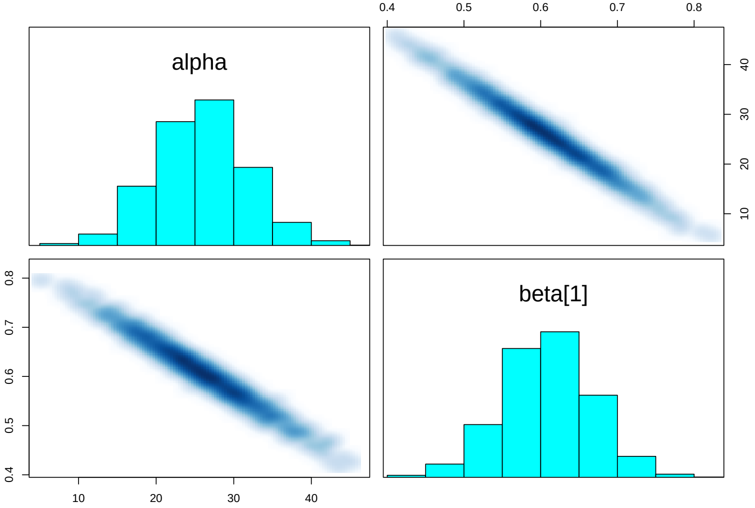
To obtain samples from this distribution, we need to

- 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

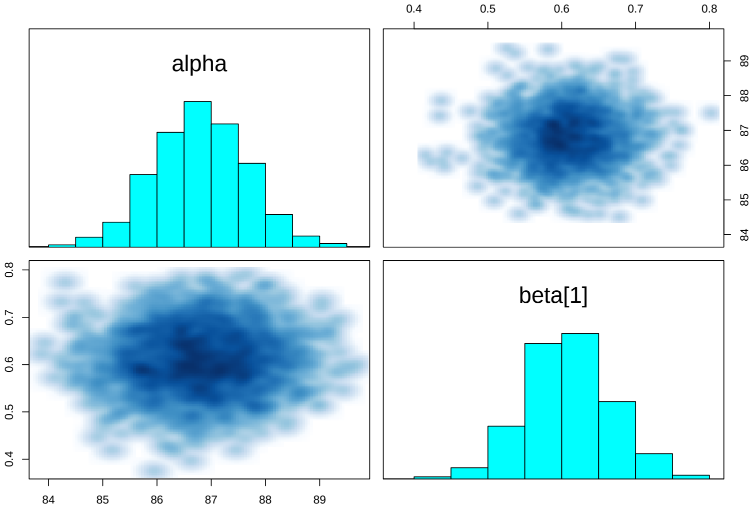
```r
summary(fit2)$summary[c("alpha", "beta[1]"),]
```

```
##                mean      se_mean         sd       2.5%        25%
## alpha    86.8044586 0.0134982924 0.85488732 85.1236984 86.2366701 86.
## beta[1]   0.6094195 0.0008783611 0.05806288  0.4934787  0.5730991  0.
##                 75%      97.5%    n_eff      Rhat
## alpha    87.3905097 88.4824846 4011.069 0.9994836
## beta[1]   0.6474439  0.7245516 4369.694 0.9997044
```

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:

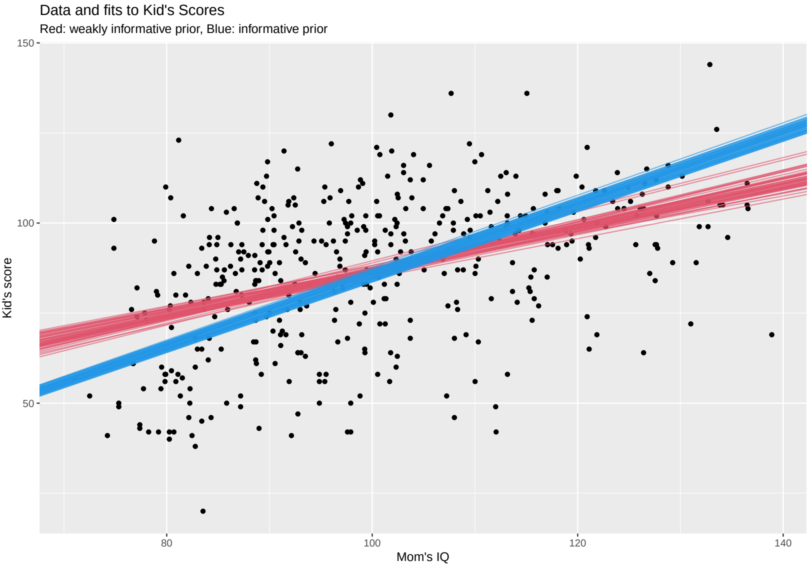
```
data <- list(y = y,
             N = length(y),
             K = 1,
             X = as.matrix(kidiq$mom_iq - mean(kidiq$mom_iq)))
fit3 <- stan(file = "kids5.stan",
            data = data)
```

Summary of fit

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

```
##                mean      se_mean          sd       2.5%        25%        50%
## alpha   86.7887933 0.0124096015 0.762758716 85.2809431 86.2911873 86.7894400
## beta[1]  0.9848027 0.0001461138 0.009825454  0.9658015  0.9781669  0.9848722
##                 75%     97.5%    n_eff      Rhat
## alpha   87.3024986 88.255837 3777.972 1.0000040
## beta[1]  0.9914099  1.004173 4521.919 0.9999344
```

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?

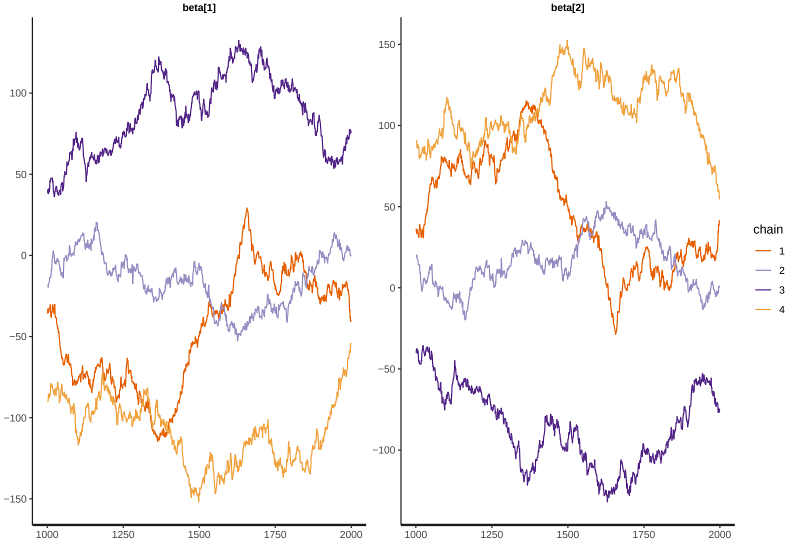
Break the model

$$y_i | \mu_i, \sigma \sim N\left(\mu, \sigma^2\right)$$
$$\mu_i = \alpha + \beta_1 x_i + \beta_2 x_i$$

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      26.51    0.92  4.74    17.83    23.17    26.27    29.49    36.84
## beta[1]   -21.07   49.83 75.08  -134.71   -86.50   -22.52    30.97   118.41
## beta[2]    21.67   49.83 75.08  -117.87   -30.42    23.14    87.10   135.35
## sigma      14.86    0.10  0.39    14.07    14.63    14.81    15.06    15.72
## lp__    -1606.96    0.19  1.33 -1610.24 -1607.70 -1606.59 -1605.91 -1605.42
##         n_eff Rhat
## alpha      26 1.16
## beta[1]     2 4.39
## beta[2]     2 4.39
## sigma      15 1.43
## lp__       48 1.08
##
## Samples were drawn using NUTS(diag_e) at Wed Feb  7 08:50:52 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).
```



beta[1]
beta[2]
100
50
0
−50
−100
−150
150
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)
```

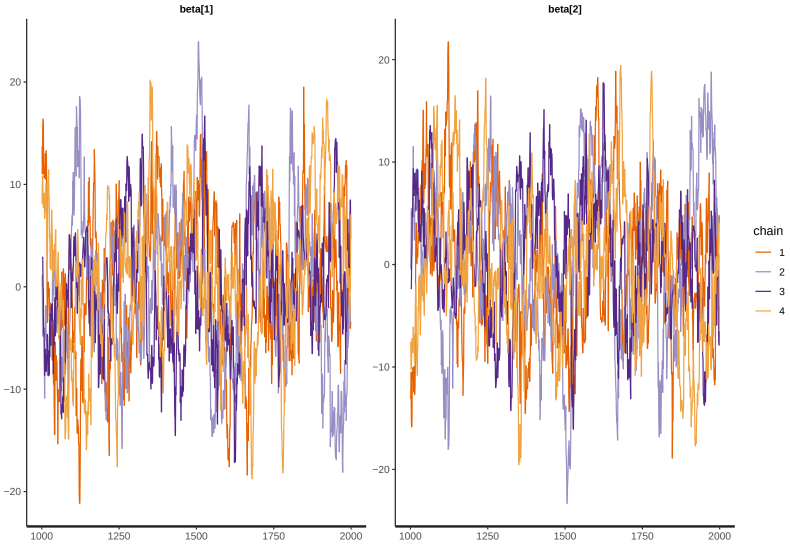
What do you think will happen?

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       25.20    0.33 5.77    13.68    21.25    25.24    29.06    36.50
## beta[1]      0.22    0.66 6.95   -13.44    -4.55     0.36     4.88    14.02
## beta[2]      0.39    0.66 6.95   -13.40    -4.26     0.24     5.17    14.09
## sigma       18.28    0.04 0.61    17.11    17.85    18.27    18.70    19.52
## lp__     -1477.54    0.06 1.39 -1481.19 -1478.27 -1477.22 -1476.47 -1475.82
##          n_eff Rhat
## alpha      308 1.01
## beta[1]    111 1.01
## beta[2]    111 1.01
## sigma      222 1.01
## lp__       472 1.01
##
## Samples were drawn using NUTS(diag_e) at Wed Feb  7 08:52:46 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).
```

Traceplots



Shrinkage priors

Additional reading for this part

- Piironen and Vehtari, 2017. ‘Sparsity information and regularization in the horseshoe and other shrinkage priors’. Electron. J. Statist. 11(2): 5018-5051 (2017). DOI: 10.1214/17-EJS1337SI
- Nice but long case study by Michael Betancourt: https://betanalpha.github.io/assets/case_studies/modeling_sparsity.html

Penalized regression models

- In many contexts, may have lots of possible covariates (p), which is big in relation to the number of observations (n)
- Common problem in genomic studies, imaging, etc
- In general, when there are many possible covariates, overfitting can be a problem
- (You may have seen) LASSO/Ridge regression (in frequentist-based inference), which maximize an objective function defined as the log-likelihood minus a penalty term.

Penalization in Bayesian models

- In a Bayesian model, we can use induce sparsity through priors on regression coefficients, which assume only a small number of covariates are non-zero.
- One way of doing this is to choose a prior on the coefficient that has a sharp peak at 0, but also has a heavy tail, allowing some coefficient estimates to 'escape'

Horseshoe prior

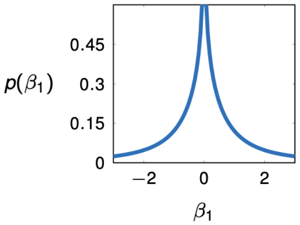
The horseshoe prior (Carvalho, Polson, and Scott 2009) sets the scale for each component to the product of a global scale, τ and a local scale, λ_j which are themselves unknown parameters:

$$\beta_j | \lambda_j, \tau \sim N(0, \tau^2 \lambda_j^2)$$
$$\lambda_j \sim \mathrm{C}^+(0, 1)$$
$$\tau \sim \mathrm{C}^+(0, \tau_0)$$

- where C^+ is the half-Cauchy distribution
- Concept of **global** and **local** scales: the global scale (τ) shrinks all coefficients to zero, while the local scale (λ_j) allows some coefficients to escape shrinkage
- Different levels of sparsity can be achieved by changing the value of τ

Horseshoe prior notes

- This is a continuous version of the spike-and-slab prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993)
- It has a **hierarchical** structure, in that the local scales are assumed to be exhangeable (shelve this, more later)
- Compared to a normal prior, horseshoe prior has more density at 0, but also more density for extreme values.
- Thus, for coefficients with very weak evidence, the regularizing prior will shrink it to zero, whereas for coefficients with strong evidence, the shrinkage will be very small.

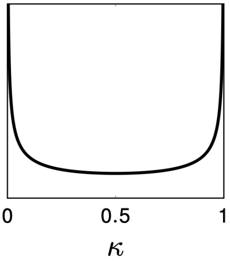


Why horseshoe?

Given the hyperparameters, the posterior mean satisfies approximately

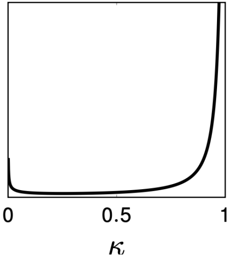
$$\bar{\beta}_j = (1 - \kappa_j)\,\beta_j^{\mathrm{ML}}, \quad \kappa_j = \frac{1}{1 + n\sigma^{-2}\tau^2\lambda_j^2}$$

where κ_j is the shrinkage factor. With $\lambda_j \sim \mathrm{C}^+(0,1)$ the prior on κ_j looks like a U. Below is when $n\sigma^{-2}\tau^2 = 0.9$



Changing τ

When $n\sigma^{-2}\tau^2 = 0.1$ (i.e. small τ), there's more shrinkage of the coefficients to zero.



In Stan

```
  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;
  real<lower=0> tau;
  real<lower=0> lambda[K];
}
transformed parameters {
}
model {
  //priors
  alpha ~ normal(0, 100);
  for(k in 1:K){
      beta[k] ~ normal(0,tau*lambda[k]);
  }
  sigma ~ normal(0,10);
  lambda ~ cauchy(0,1);
  tau ~ cauchy(0,1);

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

Regularized horseshoe

- Horseshoe prior does not regularize slopes that are far from zero at all
- The regularized horseshoe (or Finnish horseshoe, or pony horseshoe???) introduces an additional layer to further regularize larger coefficients:

$$
\begin{aligned}
\beta_j|\lambda_j, \tau &\sim N\left(0, \tau^2 \tilde{\lambda}_j^2\right) \\
\tilde{\lambda}_j &= \frac{c\lambda_j}{\sqrt{c^2 + \tau^2 \lambda_j^2}} \\
\lambda_j &\sim \mathrm{C}^+(0, 1) \\
c^2 &\sim \text{Inv-Gamma}\left(\nu/2, \nu/2s^2\right) \\
\tau &\sim \mathrm{C}^+\left(0, \tau_0\right)
\end{aligned}
$$

Regularized horseshoe

$$
\begin{aligned}
\beta_j | \lambda_j, \tau &\sim N\left(0, \tau^2 \tilde{\lambda}_j^2\right) \\
\tilde{\lambda}_j &= \frac{c\lambda_j}{\sqrt{c^2 + \tau^2 \lambda_j^2}} \\
\lambda_j &\sim \mathrm{C}^+(0, 1) \\
c^2 &\sim \text{Inv-Gamma}\left(\nu/2, \nu/2s^2\right) \\
\tau &\sim \mathrm{C}^+\left(0, \tau_0\right)
\end{aligned}
$$

- New scale variable c, which controls the distribution of coefficients far from zero ("slab width")
- When $\tau^2\lambda_j^2 \ll c^2$, meaning the coefficient is close to zero, then $\tilde{\lambda}^2 \to \lambda^2$
- When $\tau^2\lambda_j^2 \gg c^2$ then $\tilde{\lambda}^2 \to c^2/\tau^2$ and the prior on the coefficient approaches $N(0, c^2)$.

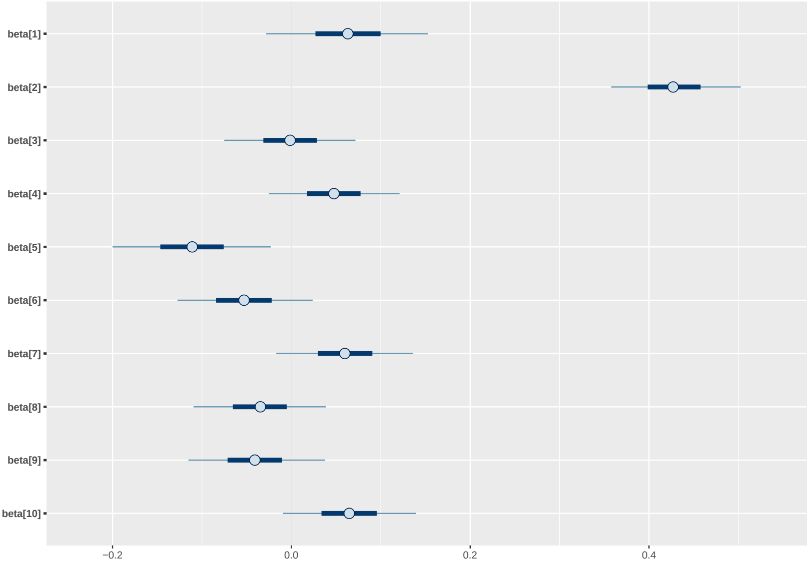
Regularized horseshoe

- Often performs better computationally than normal horseshoe (helps to reduce the number of divergences)
- Be wary of sensitivities to hyperparameter choice (i.e. τ_0, ν and s)
- Example Stan code is up on github

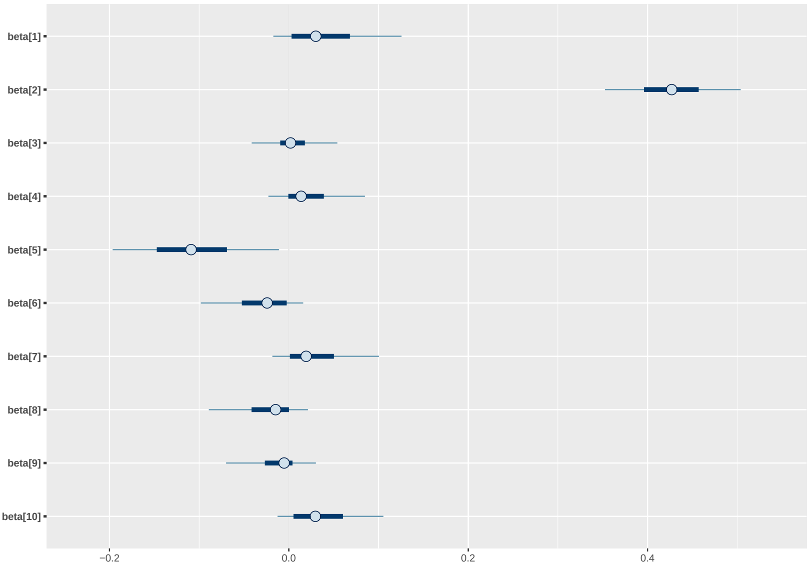
Kid's scores with all possible covariates and interactions

- Covariates are mother's education (high school y/n), age, IQ, and work status
- Included all interactions
- NOTE: regularization treats variables which vary on a larger scale as more relevant, so should scale variables such that all have unit variance before fitting model

Standard regression

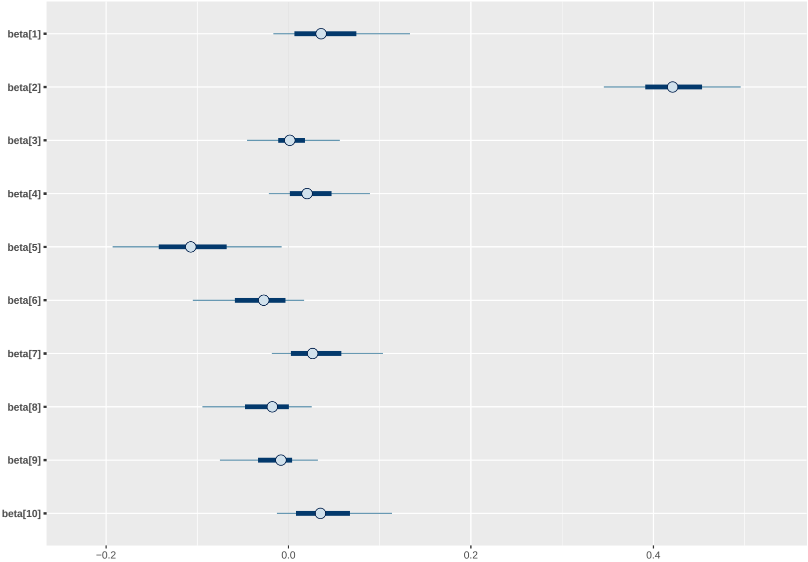


Horseshoe



beta[1]
beta[2]
beta[3]
beta[4]
beta[5]
beta[6]
beta[7]
beta[8]
beta[9]
beta[10]
−0.2
0.0
0.2
0.4

Regularized horseshoe



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

Summary

Inducing sparsity

- Rather than inducing sparsity through a penalty term, we induce sparsity through priors
- Horseshoe priors shrink everything to zero, with a local scale variable allowing some coefficients to escape shrinkage
- Regularized horseshoe controls the scale of non-zero coefficients, which is often better computationally

Lab: practice with kids dataset