

Linear Models

Lecture 02

Dr. Colin Rundel

Linear Models Basics

Pretty much everything we are going to see in this course will fall under the umbrella of either linear or generalized linear models.

In previous classes most of your time has likely been spent with the simple iid case,

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

$$\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$$

these models can also be expressed simply as,

$$y_i \stackrel{iid}{\sim} N(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \sigma^2)$$

Some notes on notation

- Observed values and scalars will usually be lower case letters, e.g. x_i, y_i, z_{ij} .
- Parameters will usually be greek symbols, e.g. μ, σ, ρ .
- Vectors and matrices will be shown in bold, e.g. $\boldsymbol{\mu}, \boldsymbol{X}, \boldsymbol{\Sigma}$.
- Elements of a matrix (or vector) will be referenced with {}s, e.g. $\{\boldsymbol{Y}\}_i, \{\boldsymbol{\Sigma}\}_{ij}$
- Random variables will be indicated by \sim , e.g. $x \sim \text{Norm}(0, 1), z \sim \text{Gamma}(1, 1)$
- Matrix / vector transposes will be indicated with $'$, e.g. $\boldsymbol{A}', (1 - \boldsymbol{B})'$

Linear model - matrix notation

We can also express a linear model using matrix notation as follows,

$$\begin{matrix} \mathbf{Y} \\ n \times 1 \end{matrix} = \begin{matrix} \mathbf{X} \\ n \times p \end{matrix} \begin{matrix} \boldsymbol{\beta} \\ p \times 1 \end{matrix} + \begin{matrix} \boldsymbol{\epsilon} \\ n \times 1 \end{matrix}$$
$$\begin{matrix} \boldsymbol{\epsilon} \\ n \times 1 \end{matrix} \sim N \left(\begin{matrix} \mathbf{0} \\ n \times 1 \end{matrix}, \sigma^2 \begin{matrix} \mathbf{1}_n \\ n \times n \end{matrix} \right)$$

or alternative as,

$$\begin{matrix} \mathbf{Y} \\ n \times 1 \end{matrix} \sim N \left(\begin{matrix} \mathbf{X} \boldsymbol{\beta} \\ n \times p \quad p \times 1 \end{matrix}, \sigma^2 \begin{matrix} \mathbf{1}_n \\ n \times n \end{matrix} \right)$$

Where possible I will include the dimensions of matrices and vectors as these provide a useful sanity check

that the dimensions conform.

Multivariate Normal Distribution - Review

For an n -dimension multivariate normal distribution with covariance Σ (positive semidefinite) can be written as

$$\underset{n \times 1}{\mathbf{Y}} \sim N\left(\underset{n \times 1}{\boldsymbol{\mu}}, \underset{n \times n}{\boldsymbol{\Sigma}}\right)$$

where $\{\boldsymbol{\Sigma}\}_{ij} = \rho_{ij}\sigma_i\sigma_j$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}, \begin{pmatrix} \rho_{11}\sigma_1\sigma_1 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1n}\sigma_1\sigma_n \\ \rho_{21}\sigma_2\sigma_1 & \rho_{22}\sigma_2\sigma_2 & \cdots & \rho_{2n}\sigma_2\sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1}\sigma_n\sigma_1 & \rho_{n2}\sigma_n\sigma_2 & \cdots & \rho_{nn}\sigma_n\sigma_n \end{pmatrix}\right)$$

Multivariate Normal Distribution - Density

For the n dimensional multivariate normal given on the last slide, its density is given by

$$f(\mathbf{Y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-n/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp \left(-\frac{1}{2} (\mathbf{Y}_{1 \times n} - \boldsymbol{\mu}_{n \times 1})' \boldsymbol{\Sigma}_{n \times n}^{-1} (\mathbf{Y}_{1 \times n} - \boldsymbol{\mu}_{n \times 1}) \right)$$

and its log density is given by

$$\log f(\mathbf{Y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{n}{2} \log 2\pi - \frac{1}{2} \log \det(\boldsymbol{\Sigma}) - \frac{1}{2} (\mathbf{Y}_{1 \times n} - \boldsymbol{\mu}_{n \times 1})' \boldsymbol{\Sigma}_{n \times n}^{-1} (\mathbf{Y}_{1 \times n} - \boldsymbol{\mu}_{n \times 1})$$

Some useful identities

The following come from the [Matrix Cookbook](#) Chapters 1 & 2.

$$(\mathbf{AB})' = \mathbf{B}' \mathbf{A}'$$

$$(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$$

$$(\mathbf{A}')^{-1} = (\mathbf{A}^{-1})'$$

$$(\mathbf{ABC} \dots)^{-1} = \dots \mathbf{C}^{-1} \mathbf{B}^{-1} \mathbf{A}^{-1}$$

$$\det(\mathbf{A}') = \det(\mathbf{A})$$

$$\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$$

$$\det(c\mathbf{A}) = c^n \det(\mathbf{A})$$

$$\det(\mathbf{A}^n) = \det(\mathbf{A})^n$$

$$\partial \mathbf{A} = 0 \quad (\text{where } \mathbf{A} \text{ is constant})$$

$$\partial(a\mathbf{X}) = a(\partial \mathbf{A})$$

$$\partial(\mathbf{X} + \mathbf{Y}) = \partial \mathbf{X} + \partial \mathbf{Y}$$

$$\partial(\mathbf{XY}) = (\partial \mathbf{X})\mathbf{Y} + \mathbf{X}(\partial \mathbf{Y})$$

$$\partial(\mathbf{X}') = (\partial \mathbf{X})'$$

Maximum Likelihood - β (iid)

Maximum Likelihood - σ^2 (iid)

A Quick Example

Parameters -> Synthetic Data

Lets generate some simulated data where the underlying model is known and see how various regression procedures function.

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \epsilon_i$$

$$\beta_0 = 0.7, \quad \beta_1 = 1.5, \quad \beta_2 = -2.2, \quad \beta_3 = 0.1$$

$$\epsilon_i \sim N(0, 1)$$

```
1 set.seed(1234)
2 n = 100
3 beta = c(0.7, 1.5, -2.2, 0.1)
4 sigma = 1
5 eps = rnorm(n, 0, sd = sigma)
6
7 d = data_frame(
8   X1 = rt(n, df=5),
9   X2 = rt(n, df=5),
```


Model Matrix

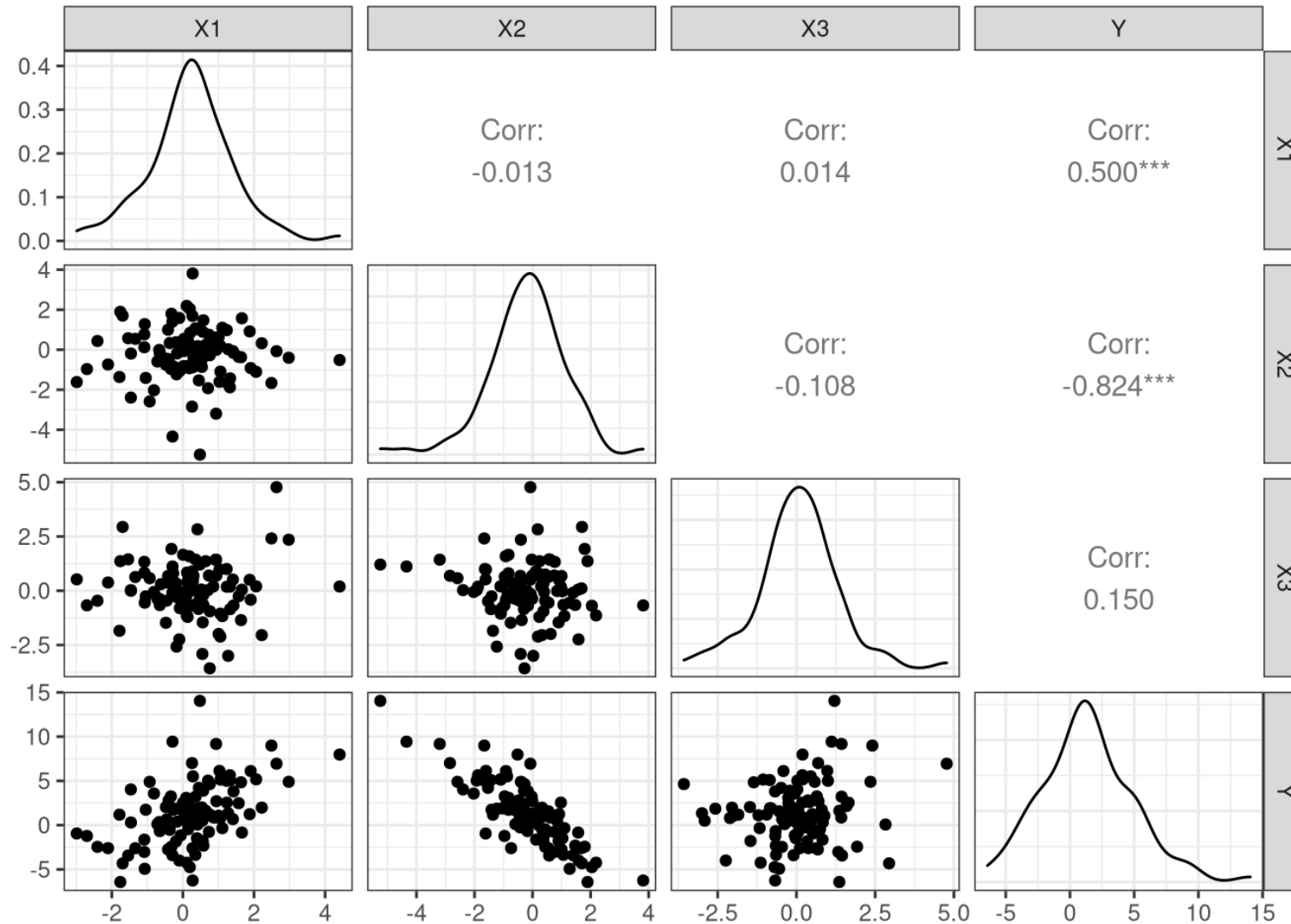
```
1 X = model.matrix(~X1+X2+X3, d)
2 tbl_df(X)
```

A tibble: 100 × 4

	`(Intercept)`	X1	X2	X3
	<dbl>	<dbl>	<dbl>	<dbl>
1	1	0.557	0.897	-1.46
2	1	0.758	0.375	-0.945
3	1	0.273	3.81	-0.675
4	1	1.41	-0.0745	0.514
5	1	1.01	0.623	-1.99
6	1	0.942	-0.00618	0.700
7	1	1.66	1.57	0.0478
8	1	-1.09	0.766	1.33
9	1	-0.296	1.40	-0.0914

Pairs plot

```
1 GGally::ggpairs(d, progress = FALSE)
```



Least squares fit

Let $\hat{\mathbf{Y}}$ be our estimate for \mathbf{Y} based on our estimate of $\boldsymbol{\beta}$,

$$\hat{\mathbf{Y}} = \hat{\beta}_0 + \hat{\beta}_1 \mathbf{X}_1 + \hat{\beta}_2 \mathbf{X}_2 + \hat{\beta}_3 \mathbf{X}_3 = \mathbf{X} \hat{\boldsymbol{\beta}}$$

The least squares estimate, $\hat{\boldsymbol{\beta}}_{ls}$, is given by

$$\operatorname{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^n (Y_i - \mathbf{X}_{i.} \boldsymbol{\beta})^2$$

Previously we showed that,

$$\hat{\boldsymbol{\beta}}_{ls} = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Y}$$

Frequentist Fit

```
1 (beta_hat = solve(t(X) %*% X, t(X)) %*% d$Y)
```

```
      [,1]
```

```
(Intercept) 0.5522298
```

```
X1          1.4708769
```

```
X2          -2.1761159
```

```
X3          0.1535830
```

```
1 l = lm(Y ~ X1 + X2 + X3, data=d)
```

```
2 l$coefficients
```

(Intercept)	X1	X2	X3
0.5522298	1.4708769	-2.1761159	0.1535830

Bayesian regression model

Basics of Bayes

brms

We will be using a package called `brms` for most of our Bayesian model fitting

- it has a convenient model specification syntax
- mostly sensible prior defaults
- supports most of the model types we will be exploring
- uses Stan behind the scenes

brms + linear regression

```
1 b = brms::brm(Y ~ X1 + X2 + X3, data=d, chains = 2)
```

```
Running /usr/lib64/R/bin/R CMD SHLIB foo.c
```

```
gcc -m64 -I"/usr/include/R" -DNDEBUG -
```

```
I"/usr/lib64/R/library/Rcpp/include/" -
```

```
I"/usr/lib64/R/library/RcppEigen/include/" -
```

```
I"/usr/lib64/R/library/RcppEigen/include/unsupported" -
```

```
I"/usr/lib64/R/library/BH/include" -
```

```
I"/usr/lib64/R/library/StanHeaders/include/src/" -
```

```
I"/usr/lib64/R/library/StanHeaders/include/" -
```

```
I"/usr/lib64/R/library/RcppParallel/include/" -
```

```
I"/usr/lib64/R/library/rstan/include" -DEIGEN_NO_DEBUG -
```

```
DBOOST_DISABLE_ASSERTS -DBOOST_PENDING_INTEGER_LOG2_HPP -DSTAN_THREADS -
```

```
DBOOST_NO_AUTO_PTR -include
```


Model results

1 b

Family: gaussian

Links: mu = identity; sigma = identity

Formula: Y ~ X1 + X2 + X3

Data: d (Number of observations: 100)

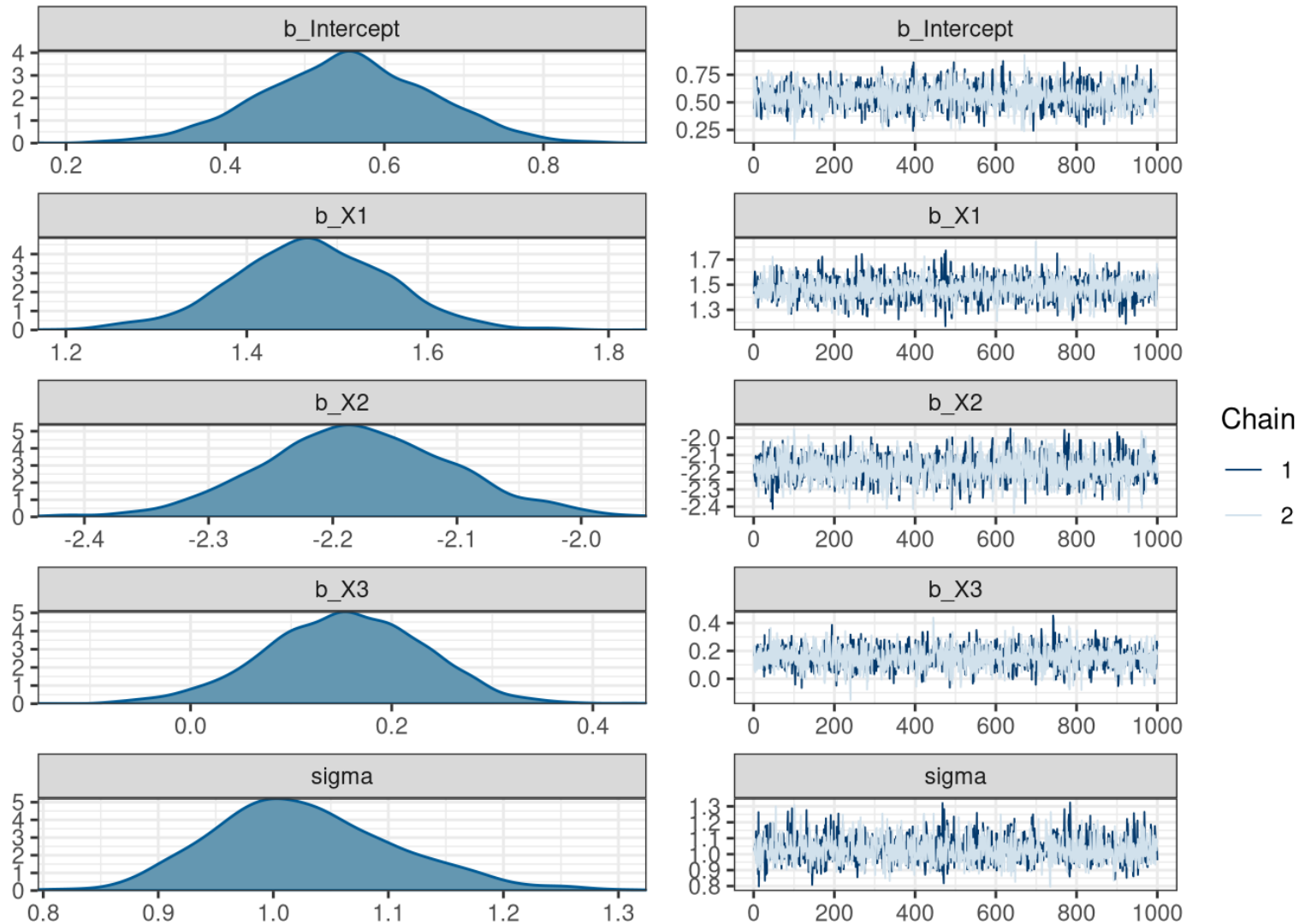
Draws: 1 chains, each with iter = 2000; warmup = 1000; thin = 1;
total post-warmup draws = 1000

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	0.55	0.11	0.34	0.76	1.00	1269	627
X1	1.47	0.09	1.29	1.64	1.00	1273	723
X2	-2.18	0.08	-2.32	-2.02	1.00	1178	663

Model visual summary

1 `plot(b)`



What about the priors?

```
1 brms::prior_summary(b)
```

	prior		class	
coef group resp dpar nlpar	lb	ub		
source				
	(flat)		b	
default				
	(flat)		b	
X1				
(vectorized)				
	(flat)		b	
X2				
(vectorized)				
	(flat)		b	

```
1 brms::get_prior("Y ~ X1 + X2 + X3")
```

	prior		class	
coef group resp dpar nlpar	lb	ub		
source				
	(flat)		b	
default				
	(flat)		b	
X1				
(vectorized)				
	(flat)		b	
X2				
(vectorized)				
	(flat)		b	

tidybayes

```
1 (post = b %>%  
2   tidybayes::gather_draws(b_Intercept, b_X1, b_X2, b_X3, sigma)  
3 )
```

A tibble: 10,000 × 5

Groups: .variable [5]

	.chain	.iteration	.draw	.variable	.value
	<int>	<int>	<int>	<chr>	<dbl>
1	1	1	1	b_Intercept	0.440
2	1	2	2	b_Intercept	0.524
3	1	3	3	b_Intercept	0.544
4	1	4	4	b_Intercept	0.502
5	1	5	5	b_Intercept	0.567
6	1	6	6	b_Intercept	0.569
7	1	7	7	b_Intercept	0.373
8	1	8	8	b_Intercept	0.731

tidybayes + posterior summaries

```
1 (post_sum = post %>%
2   group_by(.variable, .chain) %>%
3   summarize(
4     post_mean = mean(.value),
5     post_median = median(.value)
6   )
7 )
```

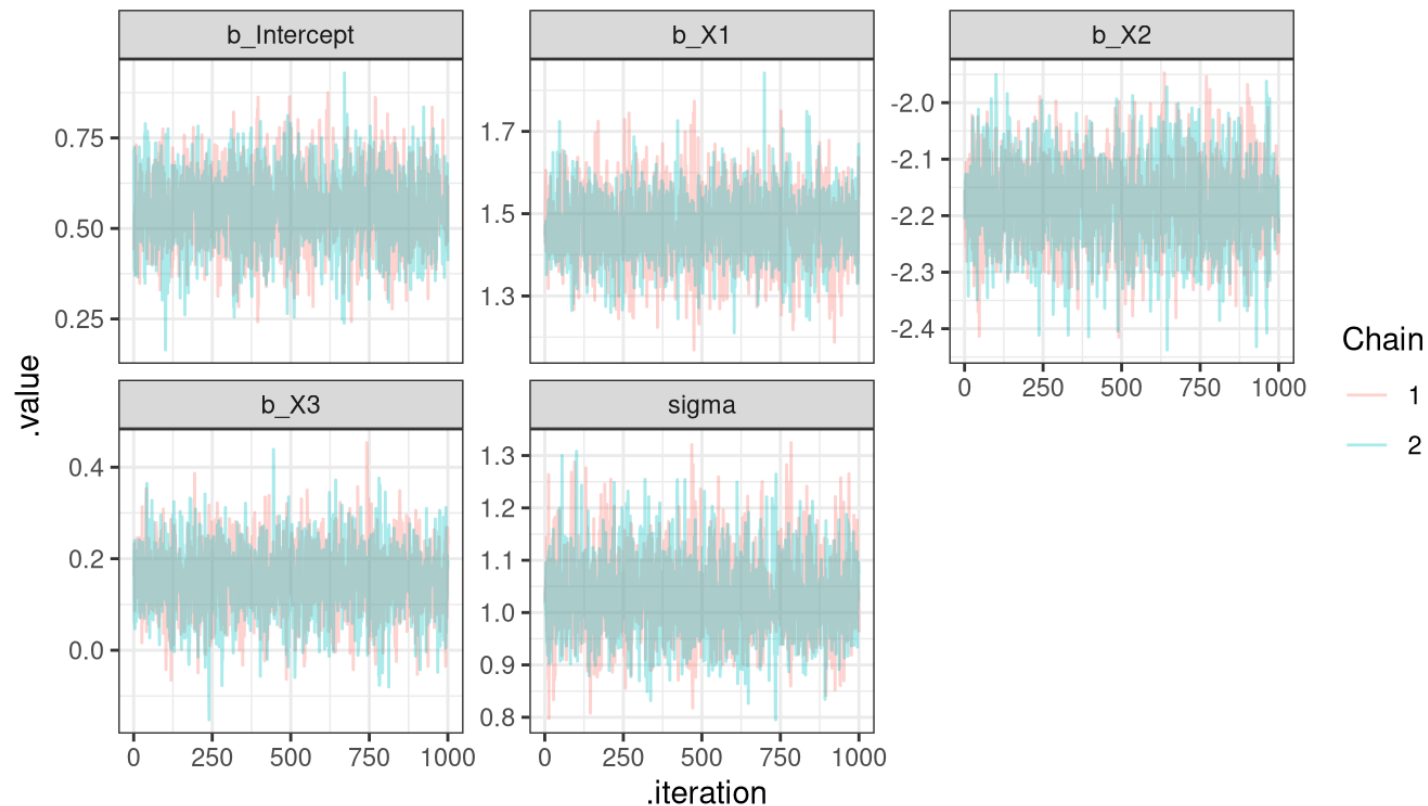
A tibble: 10 × 4

Groups: .variable [5]

	.variable	.chain	post_mean	post_median
	<chr>	<int>	<dbl>	<dbl>
1	b_Intercept	1	0.556	0.555
2	b_Intercept	2	0.550	0.553
3	b_X1	1	1.47	1.47
4	b_X1	2	1.47	1.47
5	b_X2	1	-2.18	-2.18
6	b_X2	2	-2.18	-2.18
7	b_X3	1	0.157	0.157

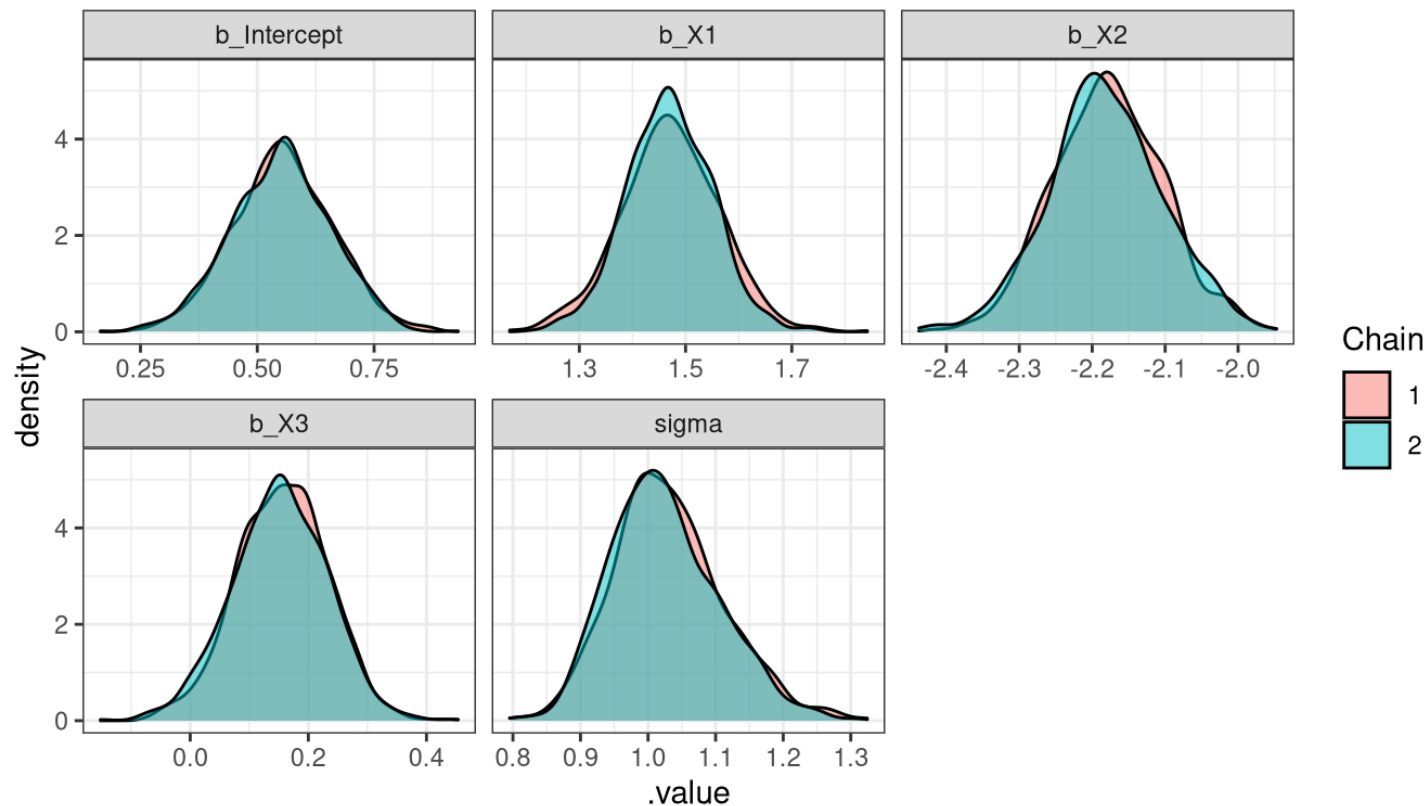
tidybayes + ggplot - traceplot

```
1 post %>%  
2   ggplot(aes(x=.iteration, y=.value, color=as.character(.chain))) +  
3   geom_line(alpha=0.33) +  
4   facet_wrap(~.variable, scale="free_y") +  
5   labs(color="Chain")
```



Tidy Bayes + ggplot - Density plot

```
1 post %>%  
2   ggplot(aes(x=.value, fill=as.character(.chain))) +  
3   geom_density(alpha=0.5) +  
4   facet_wrap(~.variable, scale="free_x") +  
5   labs(fill="Chain")
```



Comparing Approaches

```
1 (pt_est = post_sum %>%
2   filter(.chain == 1) %>%
3   ungroup() %>%
4   mutate(
5     truth = c(beta, sigma),
6     ols   = c(l$coefficients, sd(l$residuals))
7   ) %>%
8   select(.variable, truth, ols, post_mean)
9 )
```

A tibble: 5 × 4

	.variable <chr>	truth <dbl>	ols <dbl>	post_mean <dbl>
1	b_Intercept	0.7	0.552	0.556
2	b_X1	1.5	1.47	1.47
3	b_X2	-2.2	-2.18	-2.18
4	b_X3	0.1	0.154	0.157
5	sigma	1	1.00	1.03

Comparing Approaches - code

```
1 post %>%
2   filter(.chain == 1) %>%
3   ggplot(aes(x=.value)) +
4   geom_density(alpha=0.5, fill="lightblue") +
5   facet_wrap(~.variable, scale="free_x") +
6   geom_vline(
7     data = pt_est %>% tidyr::pivot_longer(cols = truth:post_mean, names_
8     aes(xintercept = value, color=pt_est),
9     alpha = 0.5, size=2
10  )
```

Comparing Approaches - plot

