

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 *iid* case,

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

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

these models can also be expressed as,

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

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,

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

or alternatively as,

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

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

Multivariate Normal Distribution - Review

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

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

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

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

Multivariate Normal Distribution - Density

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

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

$1 \times n \quad n \times n \quad n \times 1$

and its log density is given by

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

$1 \times n \quad n \times n \quad n \times 1$

Some useful matrix identities

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

$$(AB)' = B'A'$$

$$(A + B)' = A' + B'$$

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

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

$$\det(A') = \det(A)$$

$$\det(AB) = \det(A) \det(B)$$

$$\det(cA) = c^n \det(A)$$

$$\det(A^n) = \det(A)^n$$

$$\partial A = 0 \quad (\text{where } A \text{ is c}$$

$$\partial(aX) = a(\partial X)$$

$$\partial(X + Y) = \partial X + \partial Y$$

$$\partial(XY) = (\partial X)Y + X(\partial Y)$$

$$\partial(X') = (\partial X)'$$

$$\partial(X'AX) = (A + A')X$$

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.

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \epsilon_i$$

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

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

```
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 = tibble(
8    x1 = rt(n,df=5),
9    x2 = rt(n,df=5),
10   x3 = rt(n,df=5)
11 ) |>
12   mutate(
13     y = beta[1] + beta[2]*x1 + beta[3]*x2 + beta[4]*eps
14   )
```

Model Matrix

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

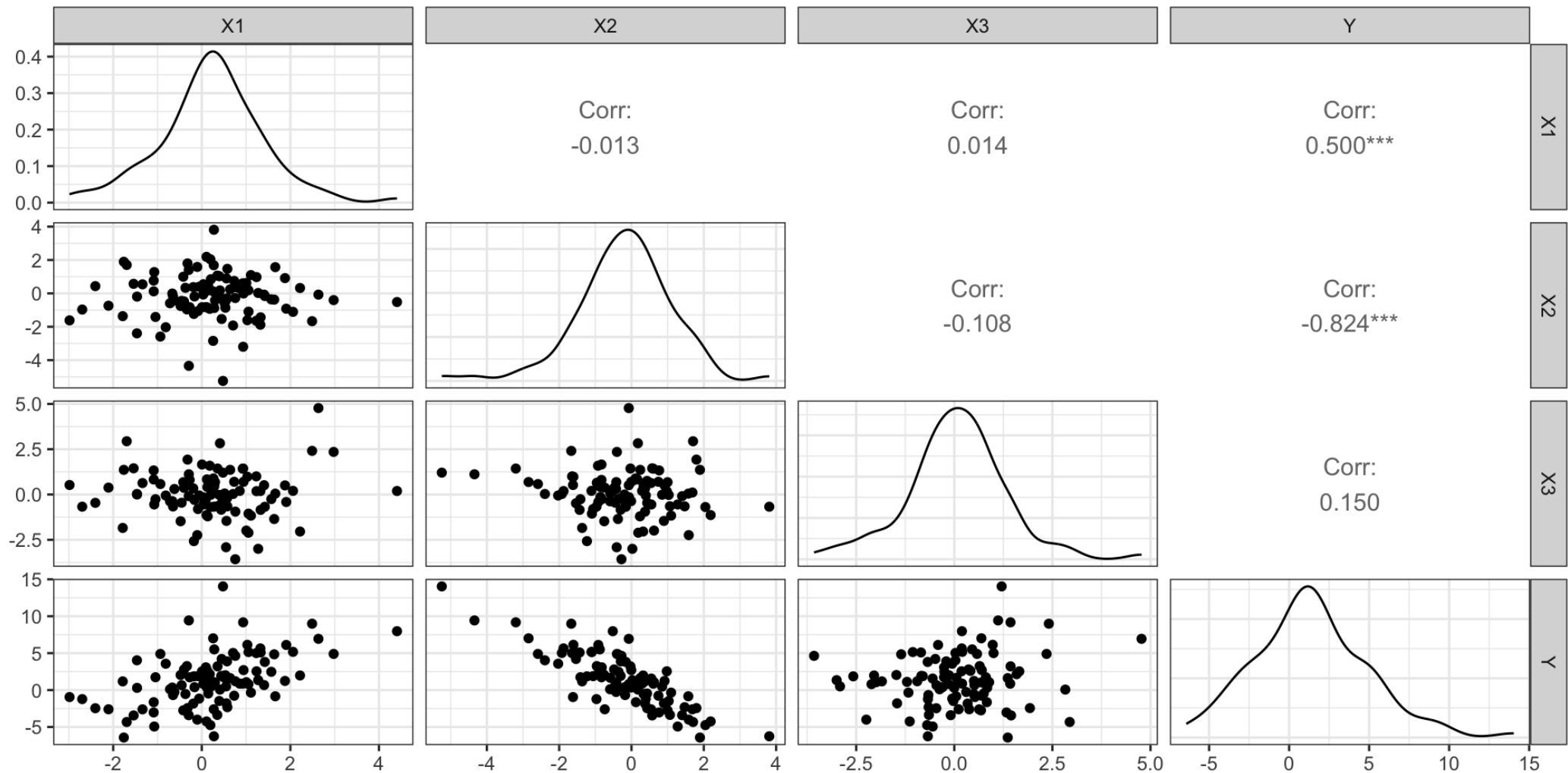
```
# A tibble: 100 × 4
```

	`(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
10	1	-0.0604	0.396	-0.0527

```
# i 90 more rows
```

Pairs plot

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



Least squares fit

Let \hat{Y} be our estimate for Y based on our estimate of β ,

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 = X \hat{\beta}$$

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

$$\hat{\beta}_{ls} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n (Y_i - X_i \cdot \beta)^2$$

Previously we showed that,

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

Beta estimate

```
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

We will be fitting the same model as described above, we just need to provide some additional information in the form of a prior for our model parameters (the β s and σ^2).

$$\begin{aligned} f(\theta|\mathbf{x}) &= \frac{f(\mathbf{x}|\theta) \pi(\theta)}{\int f(\mathbf{x}|\theta) d\theta} \\ &\propto f(\mathbf{x}|\theta) \pi(\theta) \end{aligned}$$

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, silent = 2)
```

```
Running /opt/homebrew/Cellar/r/4.3.1/lib/R/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 14.0.3 (clang-1403.0.22.14.1)'
using SDK: ''
clang -I"/opt/homebrew/Cellar/r/4.3.1/lib/R/include" -DNDEBUG -
I"/Users/rundel/Library/R/arm64/4.3/library/Rcpp/include/" -
I"/Users/rundel/Library/R/arm64/4.3/library/RcppEigen/include/" -
I"/Users/rundel/Library/R/arm64/4.3/library/RcppEigen/include/unsupported" -
I"/Users/rundel/Library/R/arm64/4.3/library/BH/include" -
I"/Users/rundel/Library/R/arm64/4.3/library/StanHeaders/include/src/" -
I"/Users/rundel/Library/R/arm64/4.3/library/StanHeaders/include/" -
I"/Users/rundel/Library/R/arm64/4.3/library/RcppParallel/include/" -
I"/Users/rundel/Library/R/arm64/4.3/library/rstan/include" -DEIGEN_NO_DEBUG -DBOOST_DISABLE_ASSERTS
-DBOOST_PENDING_INTEGER_LOG2_HPP -DSTAN_THREADS -DBOOST_NO_AUTO_PTR -include
'/Users/rundel/Library/R/arm64/4.3/library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp' -
D_REENTRANT -DRCPP_PARALLEL_USE_TBB=1 -I/opt/homebrew/include -fPIC -I/opt/homebrew/include -c
foo.c -o foo.o
In file included from <built-in>:1:
In file included from
/Users/rundel/Library/R/arm64/4.3/library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:22:
In file included from /Users/rundel/Library/R/arm64/4.3/library/RcppEigen/include/Eigen/Dense:1:
In file included from /Users/rundel/Library/R/arm64/4.3/library/RcppEigen/include/Eigen/Core:88:
/Users/rundel/Library/R/arm64/4.3/library/RcppEigen/include/Eigen/Core:111:1: warning: brace around brace-closed expression [-Wbraced]
```

Model results

```
1 b
```

```
Family: gaussian
```

```
Links: mu = identity; sigma = identity
```

```
Formula: Y ~ X1 + X2 + X3
```

```
Data: d (Number of observations: 100)
```

```
Draws: 2 chains, each with iter = 2000; warmup = 1000; thin = 1;  
total post-warmup draws = 2000
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	0.55	0.10	0.36	0.75	1.00	2908	1697
X1	1.47	0.09	1.29	1.64	1.00	2361	1437
X2	-2.18	0.07	-2.32	-2.02	1.00	2366	1528
X3	0.15	0.08	-0.01	0.30	1.00	2681	1479

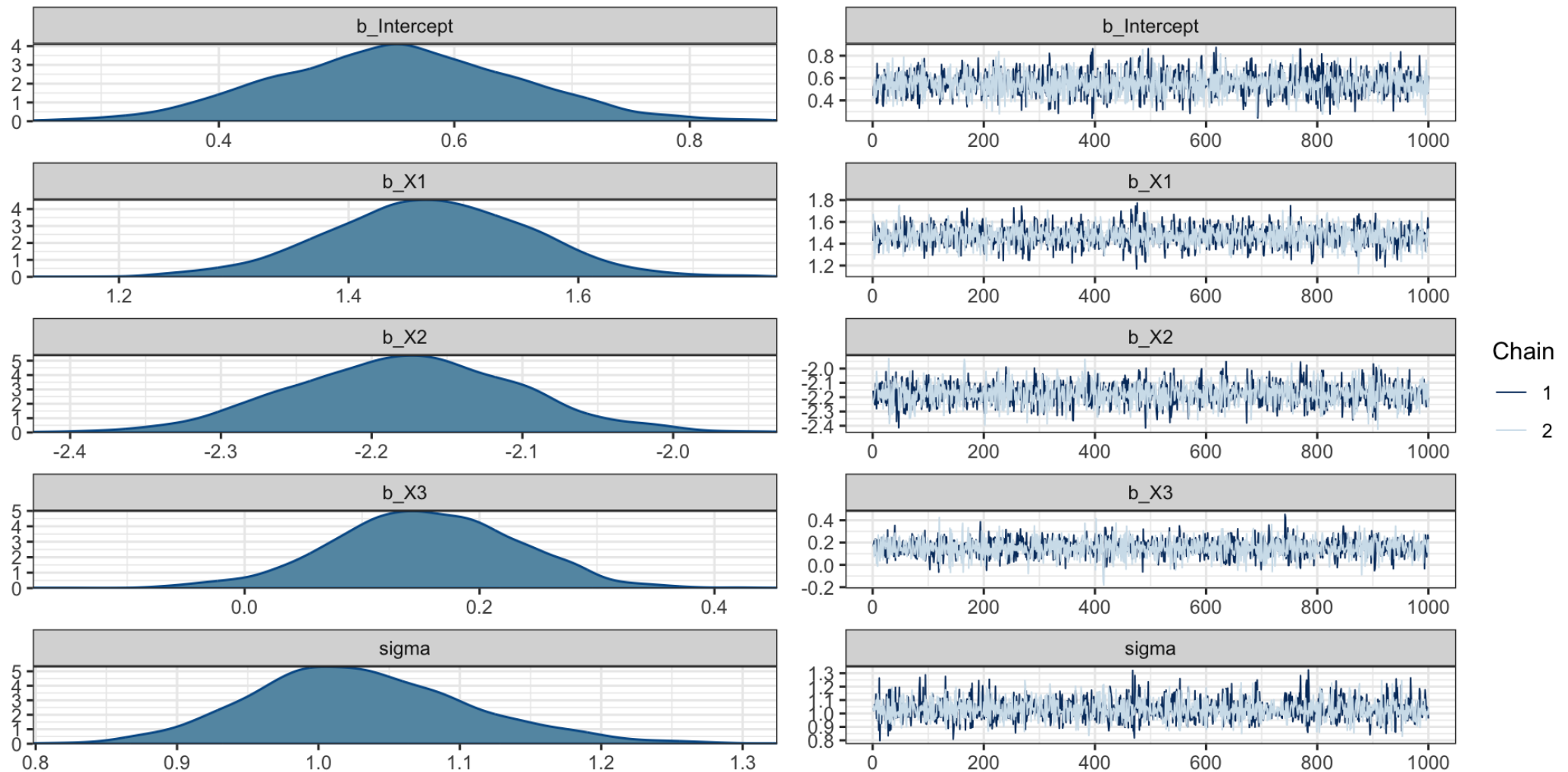
Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	1.03	0.08	0.89	1.19	1.00	2691	1361

Draws were sampled using `sampling(NUTS)`. For each parameter, `Bulk_ESS` and `Tail_ESS` are effective sample size measures, and `Rhat` is the potential scale reduction factor on split chains (at convergence, `Rhat` = 1).

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	x3							(vectorized)
student_t(3, 1.1, 3.1)		Intercept								default
student_t(3, 0, 3.1)		sigma						0		default

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
9	1	9	9	b_Intercept	0.379
10	1	10	10	b_Intercept	0.514

```
# i 9,990 more rows
```


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     .groups = "drop"
7   )
8 )
```

A tibble: 10 × 4

	.variable <chr>	.chain <int>	post_mean <dbl>	post_median <dbl>
1	b_Intercept	1	0.556	0.555
2	b_Intercept	2	0.546	0.546
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
8	b_X3	2	0.151	0.149
9	sigma	1	1.03	1.02
10	sigma	2	1.03	1.02

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$coefficients,
7               sd(l$residuals))
8   ) |>
9   select(
10     .variable, truth,
11     ols, post_mean
12   )
13 )
```

```
# A tibble: 5 × 4
  .variable    truth    ols post_mean
  <chr>      <dbl> <dbl>    <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

```
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_to = "variable", values_to = "value"),
8     aes(xintercept = value, color=pt_est),
9     alpha = 0.5, linewidth=1.5
10  )
```

Comparing Approaches

