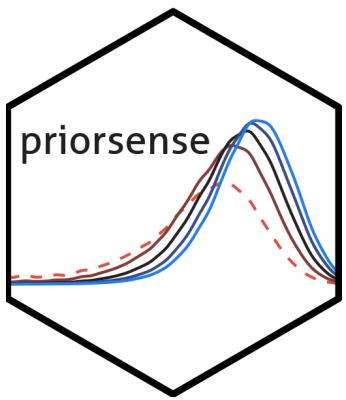
Prior and likelihood sensitivity checks with priorsense

- sensible prior specification is important, as otherwise priors can strongly and unintentionally influence results

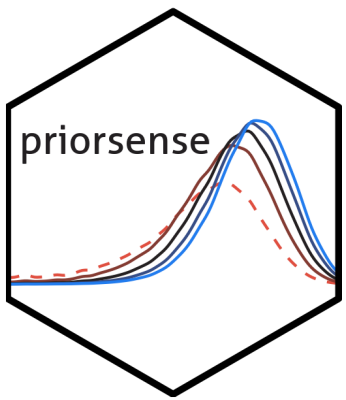
Prior and likelihood sensitivity checks with priorsense

- sensible prior specification is important, as otherwise priors can strongly and unintentionally influence results
- we recommend checking prior and likelihood sensitivity using priorsense (Kallioinen et al., 2023)



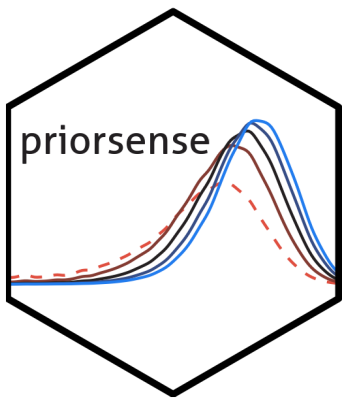
Prior and likelihood sensitivity checks with priorsense

- sensible prior specification is important, as otherwise priors can strongly and unintentionally influence results
- we recommend checking prior and likelihood sensitivity using priorsense (Kallioinen et al., 2023)
  - efficient prior and likelihood sensitivity checks



Prior and likelihood sensitivity checks with priorsense

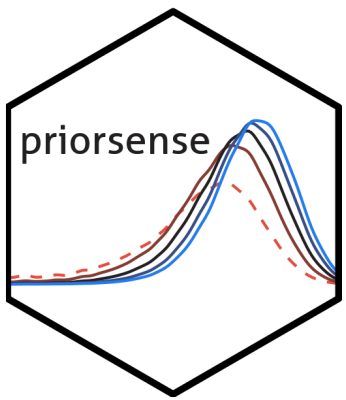
- sensible prior specification is important, as otherwise priors can strongly and unintentionally influence results
- we recommend checking prior and likelihood sensitivity using priorsense (Kallioinen et al., 2023)



- efficient prior and likelihood sensitivity checks
- provides numerical and graphical diagnostics

Prior and likelihood sensitivity checks with priorsense

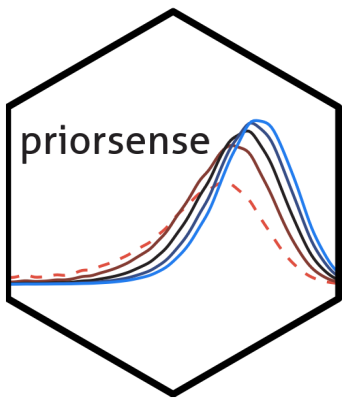
- sensible prior specification is important, as otherwise priors can strongly and unintentionally influence results
- we recommend checking prior and likelihood sensitivity using priorsense (Kallioinen et al., 2023)



- efficient prior and likelihood sensitivity checks
- provides numerical and graphical diagnostics
- compatible with `brms` and applicable to any model (excluding flat priors)

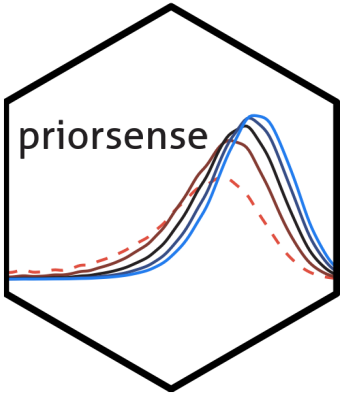
Prior and likelihood sensitivity checks with priorsense

- sensible prior specification is important, as otherwise priors can strongly and unintentionally influence results
- we recommend checking prior and likelihood sensitivity using priorsense (Kallioinen et al., 2023)



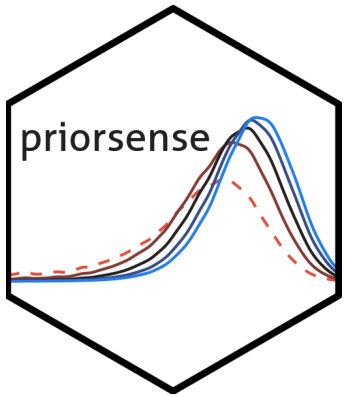
- efficient prior and likelihood sensitivity checks
- provides numerical and graphical diagnostics
- compatible with `brms` and applicable to any model (excluding flat priors)
- no need to refit

How it works



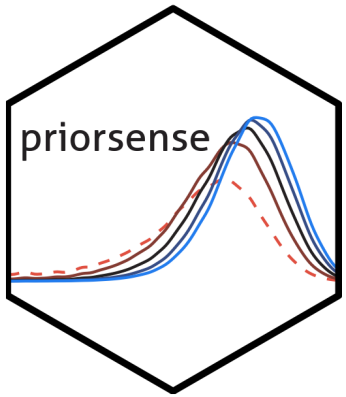
- modifies the posterior by power-scaling the **prior** or **likelihood** slightly

How it works



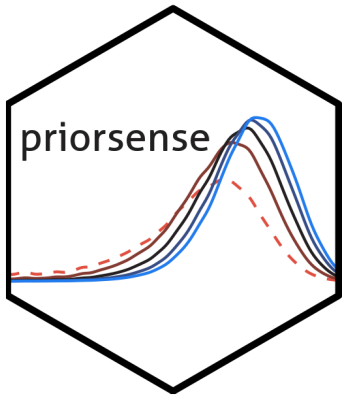
- modifies the posterior by power-scaling the **prior** or **likelihood** slightly
- original posterior:
  $p(\theta \mid y) \propto p(\theta)p(y \mid \theta)$

How it works



- modifies the posterior by power-scaling the **prior** or **likelihood** slightly
- original posterior: $p(\theta \mid y) \propto p(\theta)p(y \mid \theta)$
- modified posterior: $\boldsymbol{p(\theta)^{\alpha}}p(y \mid \theta)$ or $p(\theta)\boldsymbol{p(y \mid \theta)^{\alpha}}$

How it works



- modifies the posterior by power-scaling the **prior** or **likelihood** slightly
- original posterior: $p(\theta \mid y) \propto p(\theta)p(y \mid \theta)$
- modified posterior: $\boldsymbol{p(\theta)^{\alpha}}p(y \mid \theta)$ or $p(\theta)\boldsymbol{p(y \mid \theta)^{\alpha}}$
- uses Pareto-smoothed importance sampling to estimate modified posterior from original posterior draws $\theta^{(s)}$

Power-scaling

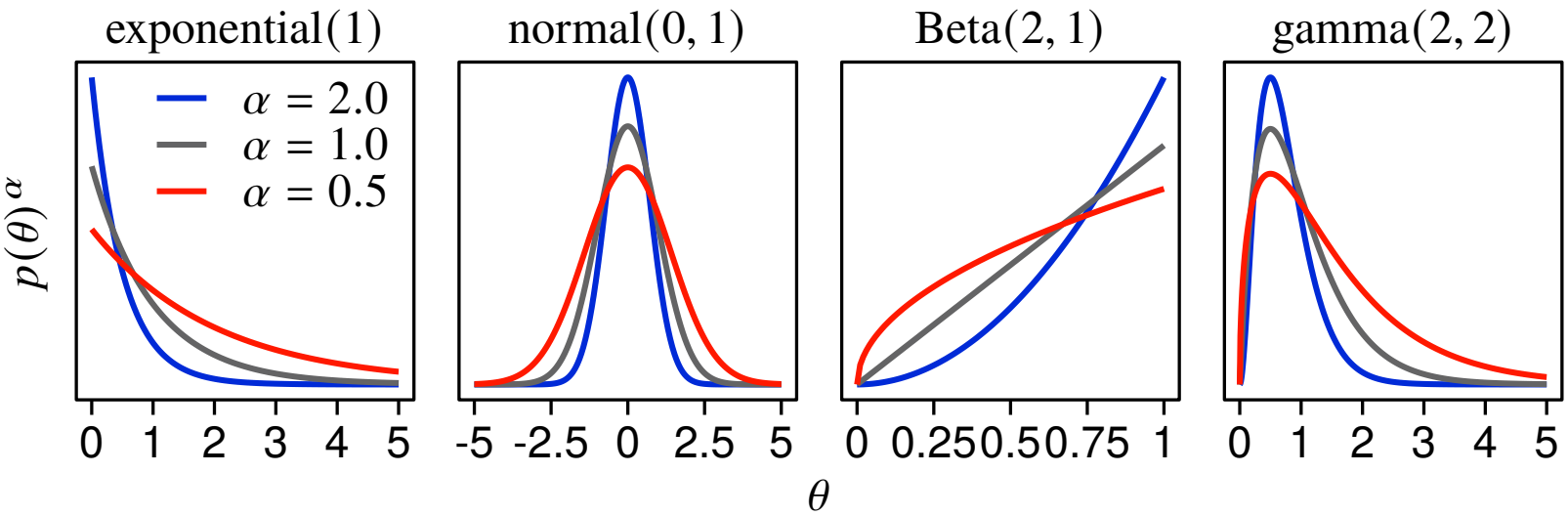


Figure: Example power-scaled distributions

Interpretation

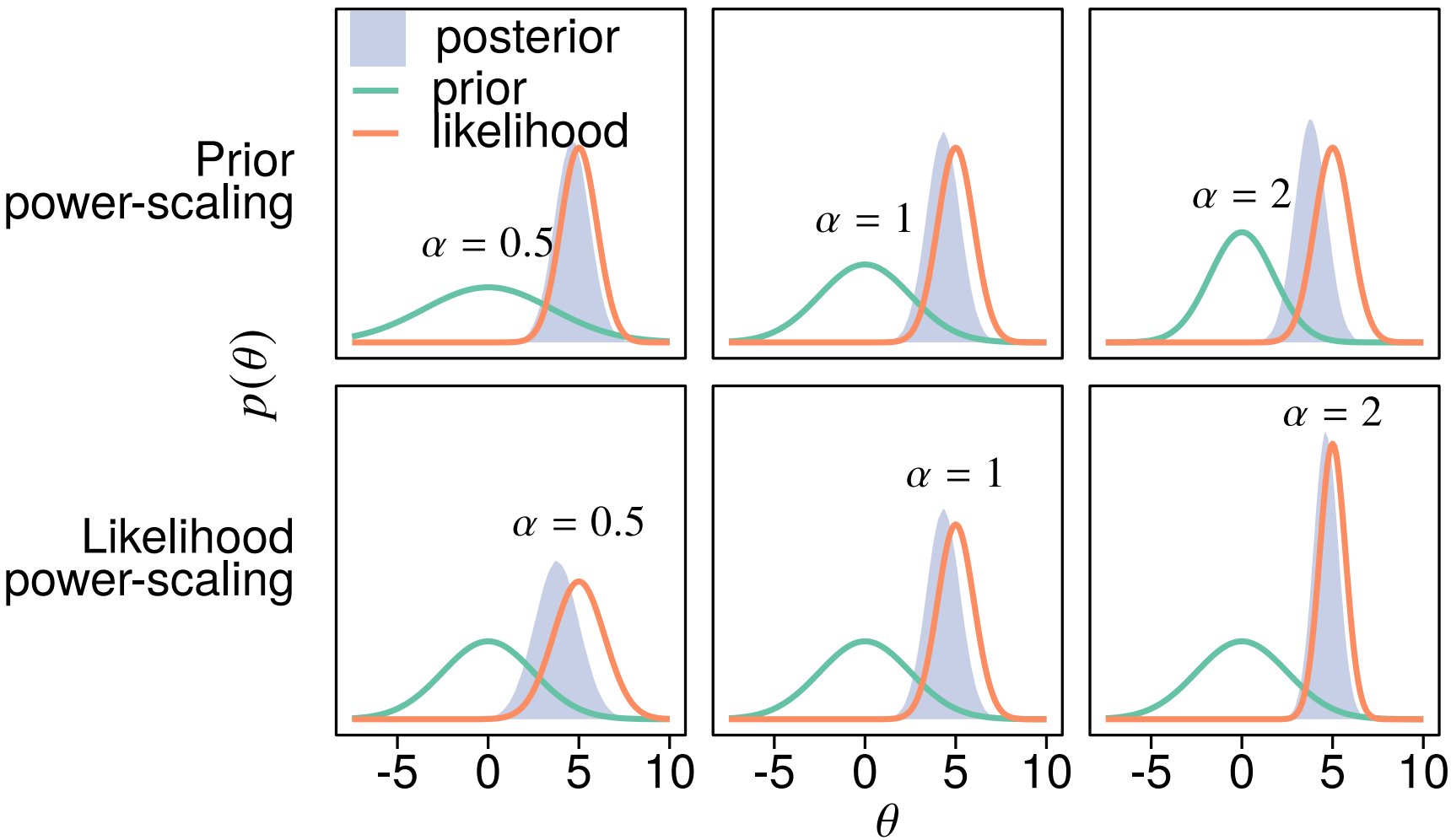


Figure: Prior-likelihood conflict

Computation

For prior power-scaling:

$$w_{pri}^{(s)} = \frac{p(\theta^{(s)})^{\alpha} p(\theta^{(s)} \mid y)}{p(\theta^{(s)}) p(\theta^{(s)} \mid y)} = \frac{p(\theta^{(s)})^{\alpha}}{p(\theta^{(s)})} = p(\theta^{(s)})^{(\alpha-1)}$$

Computation

For prior power-scaling:

$$w_{pri}^{(s)} = \frac{p(\theta^{(s)})^{\alpha} p(\theta^{(s)} \mid y)}{p(\theta^{(s)}) p(\theta^{(s)} \mid y)} = \frac{p(\theta^{(s)})^{\alpha}}{p(\theta^{(s)})} = p(\theta^{(s)})^{(\alpha-1)}$$

For likelihood power-scaling:

$$w_{lik}^{(s)} = \frac{p(\theta^{(s)}) p(\theta^{(s)} \mid y)^{\alpha}}{p(\theta^{(s)}) p(\theta^{(s)} \mid y)} = \frac{p(\theta^{(s)} \mid y)^{\alpha}}{p(y \mid \theta^{(s)})} = p(y \mid \theta^{(s)})^{(\alpha-1)}$$

Example

- Dataset: npk
- Goal: Model effect of Nitrogen, Phosphate and Potassium on pea crop yield
- Data: 24 observations of crop yield (3 observations of each combination)

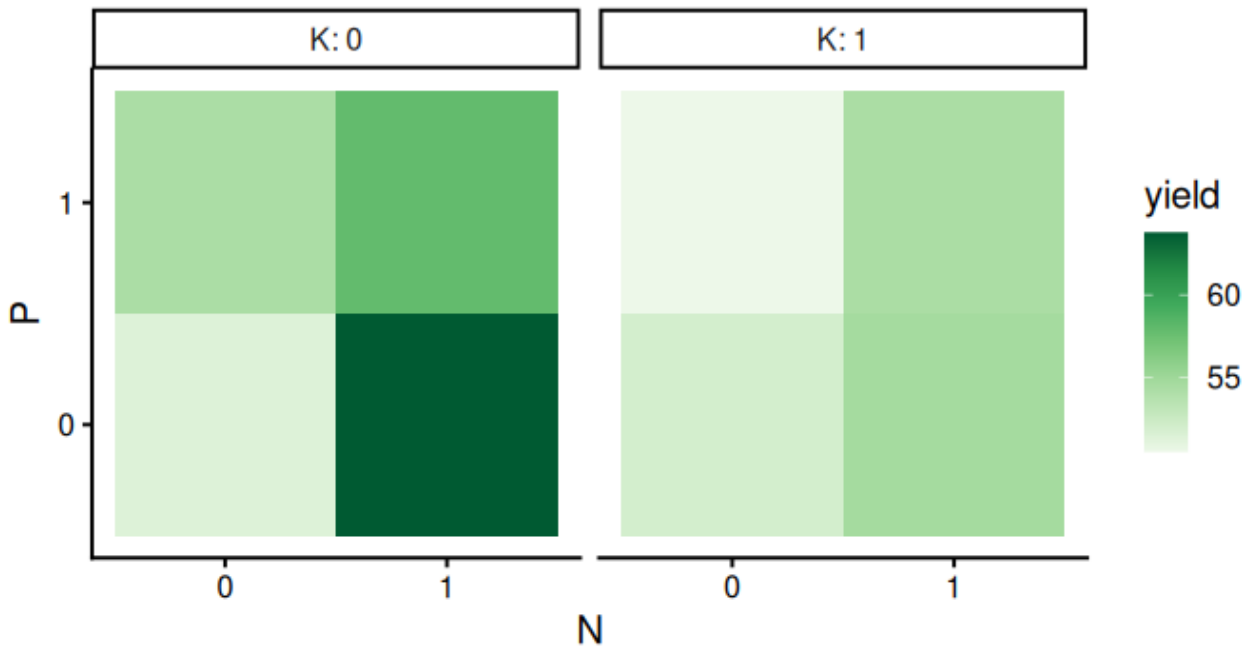


Figure: Mean yield per combination

Example

Model:

$$y_i \sim \mathrm{N}(\mu_i, \sigma^2)$$

Example

Model:

$$y_i \sim \mathrm{N}(\mu_i, \sigma^2)$$

$$\mu_i = \beta_{\text{Intercept}}$$

Example

Model:

$$y_i \sim \mathrm{N}(\mu_i, \sigma^2)$$

$$\begin{aligned}
\mu_i =& \beta_{\text{Intercept}} \\
&+ \beta_{\text{N}} N_i && \text{(main effects)} \\
&+ \beta_{\text{P}} P_i \\
&+ \beta_{\text{K}} K_i
\end{aligned}$$

Example

Model:

$$y_i \sim \mathrm{N}(\mu_i, \sigma^2)$$

$$\begin{aligned}
\mu_i =& \beta_{\text{Intercept}} \\
&+\beta_{\text{N}} N_i && \text{(main effects)} \\
&+\beta_{\text{P}} P_i \\
&+\beta_{\text{K}} K_i \\
&+\beta_{\text{N}\times\text{K}} N_i K_i && \text{(2-way interactions)} \\
&+\beta_{\text{N}\times\text{P}} N_i P_i \\
&+\beta_{\text{K}\times\text{P}} K_i P_i
\end{aligned}$$

Example: Initial model

Main effects only

`brms` formula: `yield ~ N + P + K`

Example: Initial model

Main effects only

`brms` formula: `yield ~ N + P + K`

```
prior(normal(0, 2.5), coef = "b_N1", tag = "main")
prior(normal(0, 2.5), coef = "b_K1", tag = "main")
prior(normal(0, 2.5), coef = "b_P1", tag = "main")
```

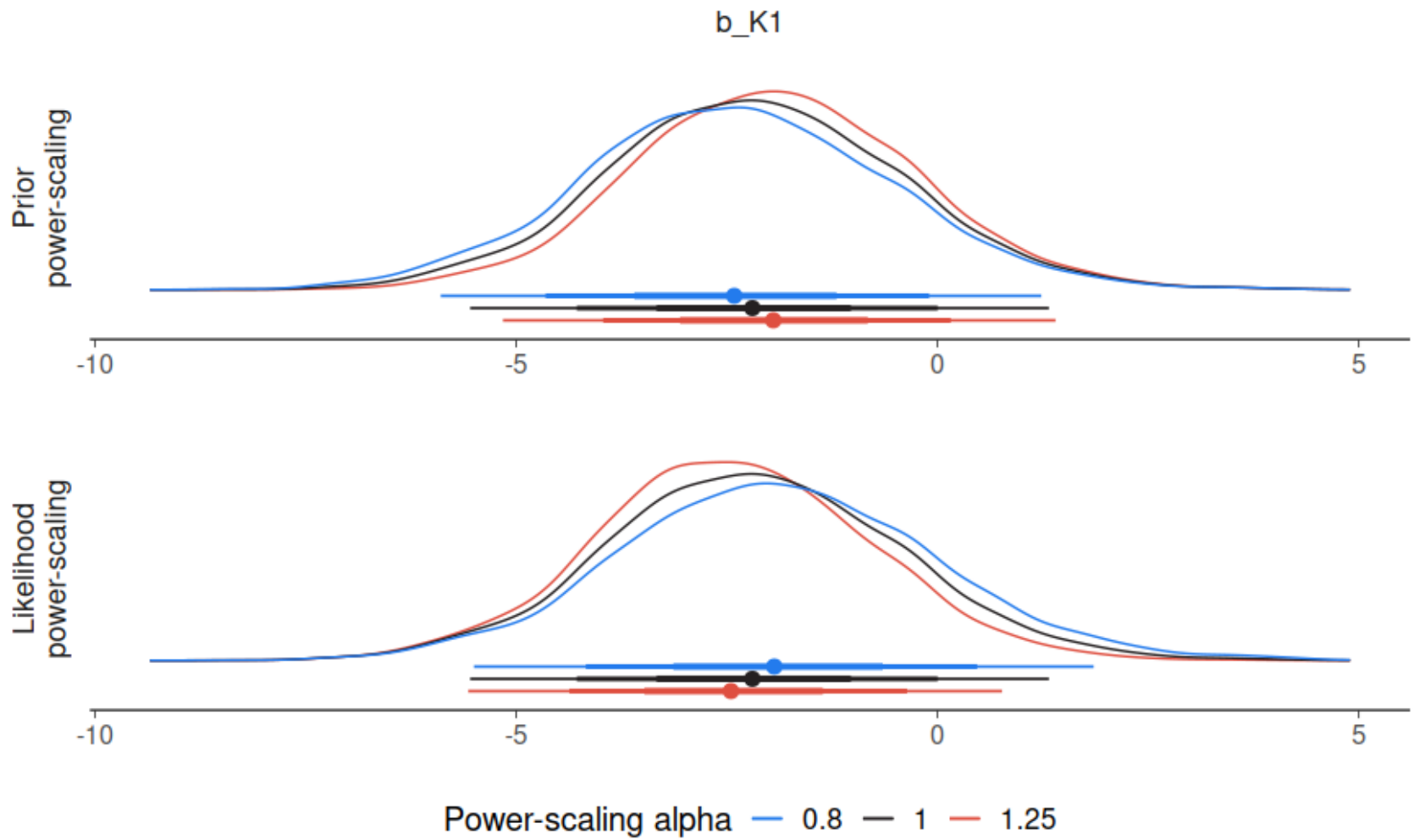
tagging a group of priors allows for selective power-scaling

Example: Initial model checks

```
powerscale_plot_dens(fit, prior_selection = "main")
```

Example: Initial model checks

```
powerscale_plot_dens(fit, prior_selection = "main")
```



Example: Initial model checks

```
powerscale_sensitivity(fit, prior_selection = "main")
```

Example: Initial model checks

```
powerscale_sensitivity(fit, prior_selection = "main")
```

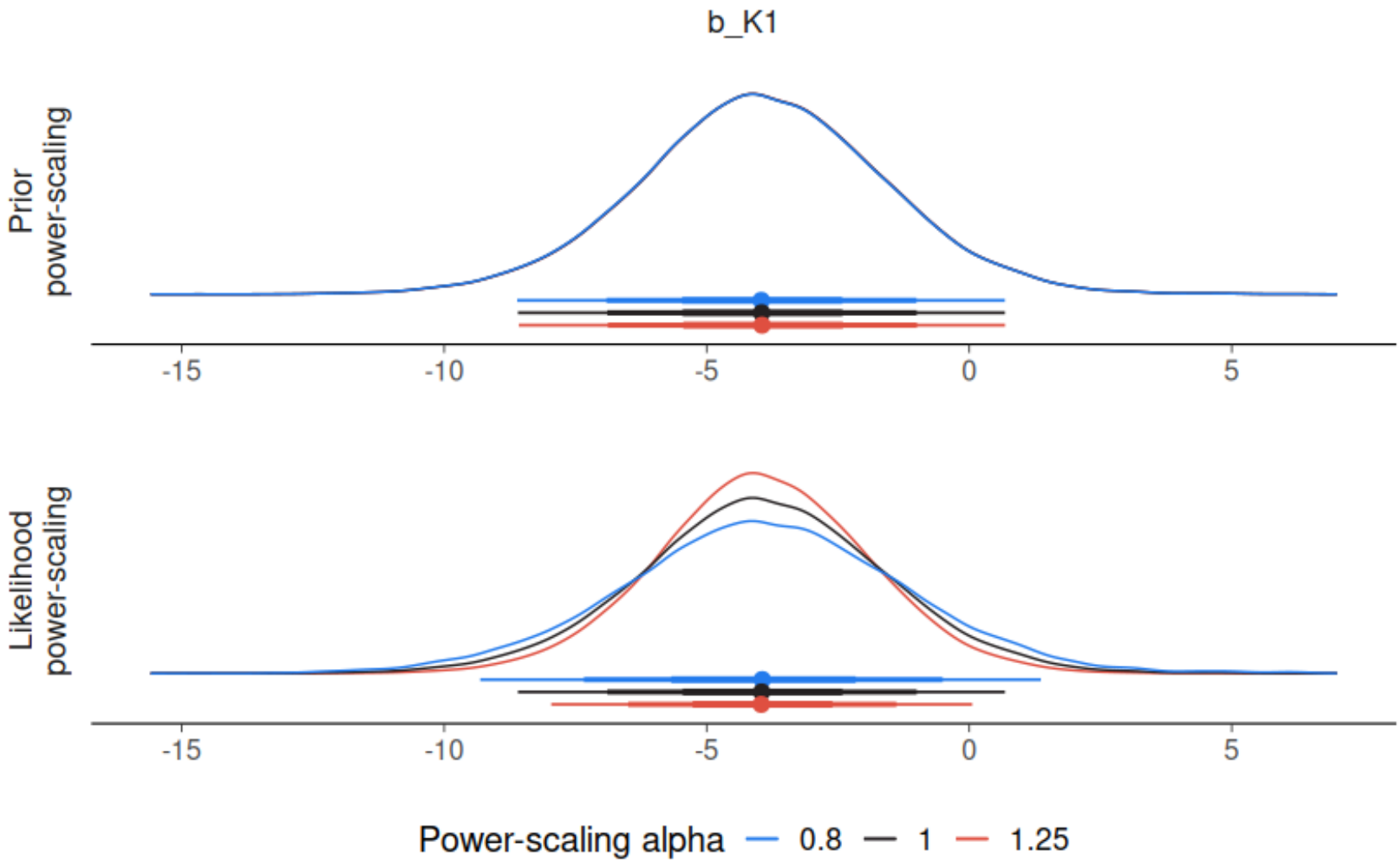
variable	prior sens.	likelihood sens.	diagnosis
b_N1	0.15	0.18	potential prior-data conflict
b_P1	0.06	0.07	potential prior-data conflict
b_K1	0.11	0.14	potential prior-data conflict

Table: Sensitivity diagnostics

Example: Adjusted prior

```
prior(normal(0, 25), coef = "b_N1", tag = "main")
prior(normal(0, 25), coef = "b_K1", tag = "main")
prior(normal(0, 25), coef = "b_P1", tag = "main")
```

Example: Adjusted prior checks



Example: Adjusted prior checks

variable	prior sens.	likelihood sens.	diagnosis
b_N1	0.00	0.10	-
b_P1	0.00	0.09	-
b_K1	0.00	0.10	-

Example: Second model

Main effects and two-way interactions

`brms` formula:

```
yield ~ (N + P + K)^2
```

Example: Second model

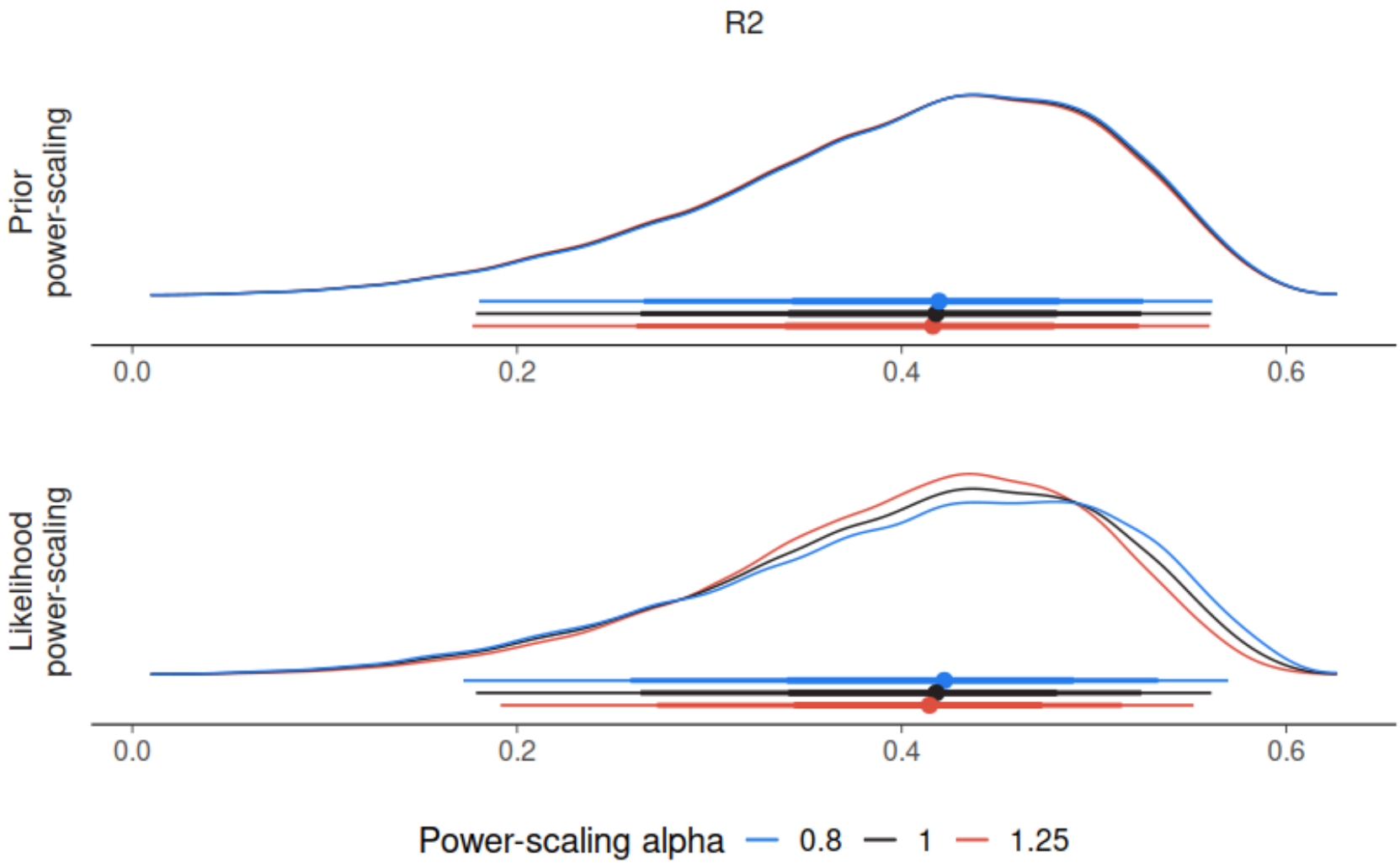
Main effects and two-way interactions

`brms` formula:

```
yield ~ (N + P + K)^2
```

```
prior(normal(0, 25), coef = "b_N1:P1", tag = "twoway")
prior(normal(0, 25), coef = "b_N1:K1", tag = "twoway")
prior(normal(0, 25), coef = "b_P1:K1", tag = "twoway")
```

Example: Second model checks



Summary

- priors can be challenging to specify without domain knowledge

Summary

- priors can be challenging to specify without domain knowledge
- prior and likelihood sensitivity checking is recommended

Summary

- priors can be challenging to specify without domain knowledge
- prior and likelihood sensitivity checking is recommended
- `priorsense` can make sensitivity checking faster

Summary

- priors can be challenging to specify without domain knowledge
- prior and likelihood sensitivity checking is recommended
- `priorsense` can make sensitivity checking faster
- with many parameters, use tags to select priors and look at predictions or model fit measures

Summary

- priors can be challenging to specify without domain knowledge
- prior and likelihood sensitivity checking is recommended
- `priorsense` can make sensitivity checking faster
- with many parameters, use tags to select priors and look at predictions or model fit measures
- if checks indicate strong prior sensitivity, prior-data conflict, or weak likelihood, further thought should be put into the prior

Summary

- priors can be challenging to specify without domain knowledge
- prior and likelihood sensitivity checking is recommended
- `priorsense` can make sensitivity checking faster
- with many parameters, use tags to select priors and look at predictions or model fit measures
- if checks indicate strong prior sensitivity, prior-data conflict, or weak likelihood, further thought should be put into the prior
- see more at n-kall.github.io/priorsense