BDA Course 2026

Lecture 11

Outline

- Prior and likelihood sensitivity checks
- Variable selection

Sensitivity checks

- Priors influence on the posterior can vary from negligible to substantial
- Whether this influence is acceptable or not is context-dependent
- We can perform sensitivity checks to assess the impact
- The direct approach is to refit the model with different priors

Efficient prior and likelihood sensitivity checks

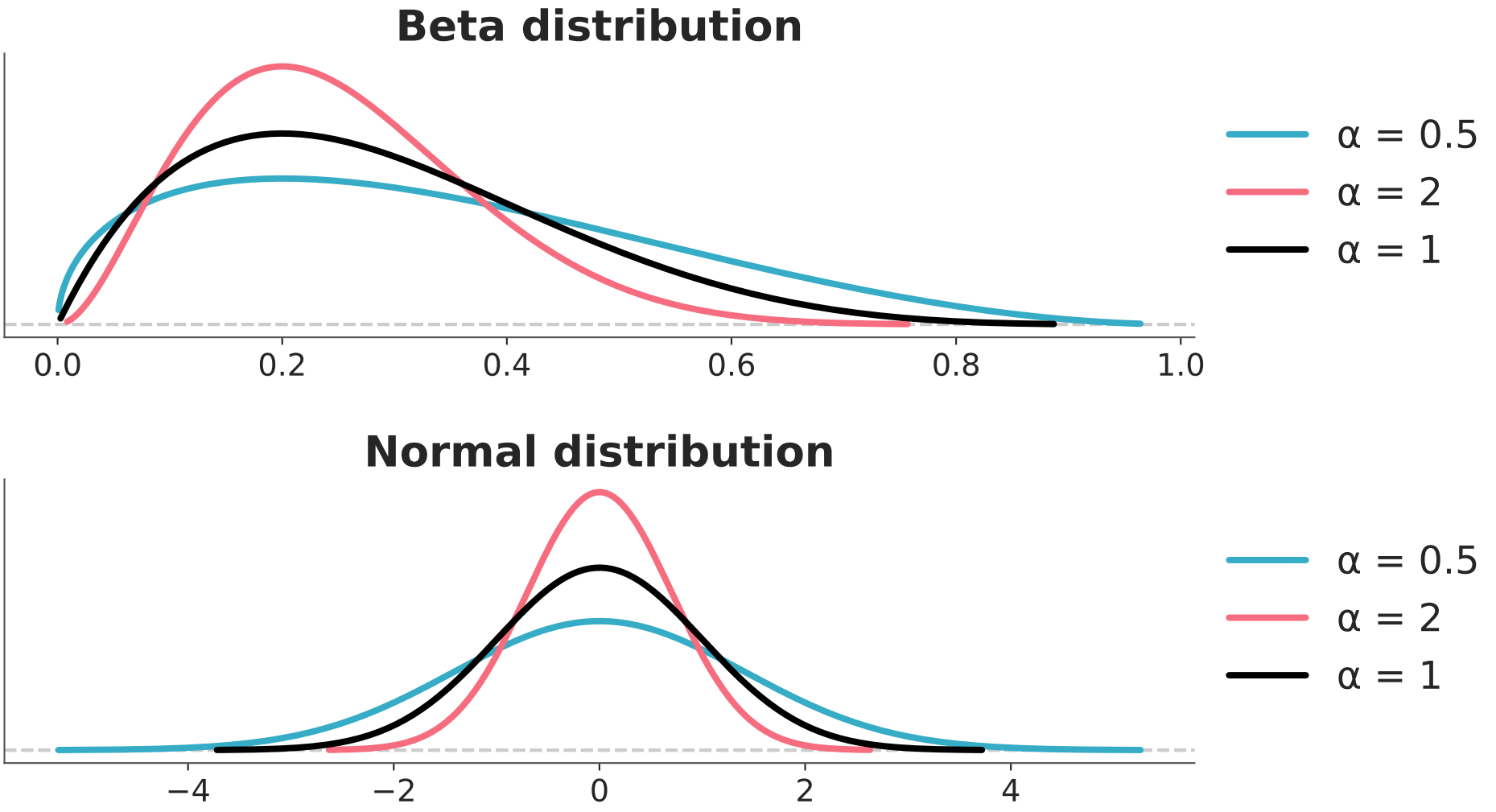
- We recommend checking prior and likelihood sensitivity following Kallioinen et al., 2023
  - Provides numerical and graphical diagnostics
  - No need to refit the model
  - In R, use priorsense package
  - In Python use the psense functions in ArviZ

Kallioinen et al. (2023). Detecting and diagnosing prior and likelihood sensitivity with power-scaling

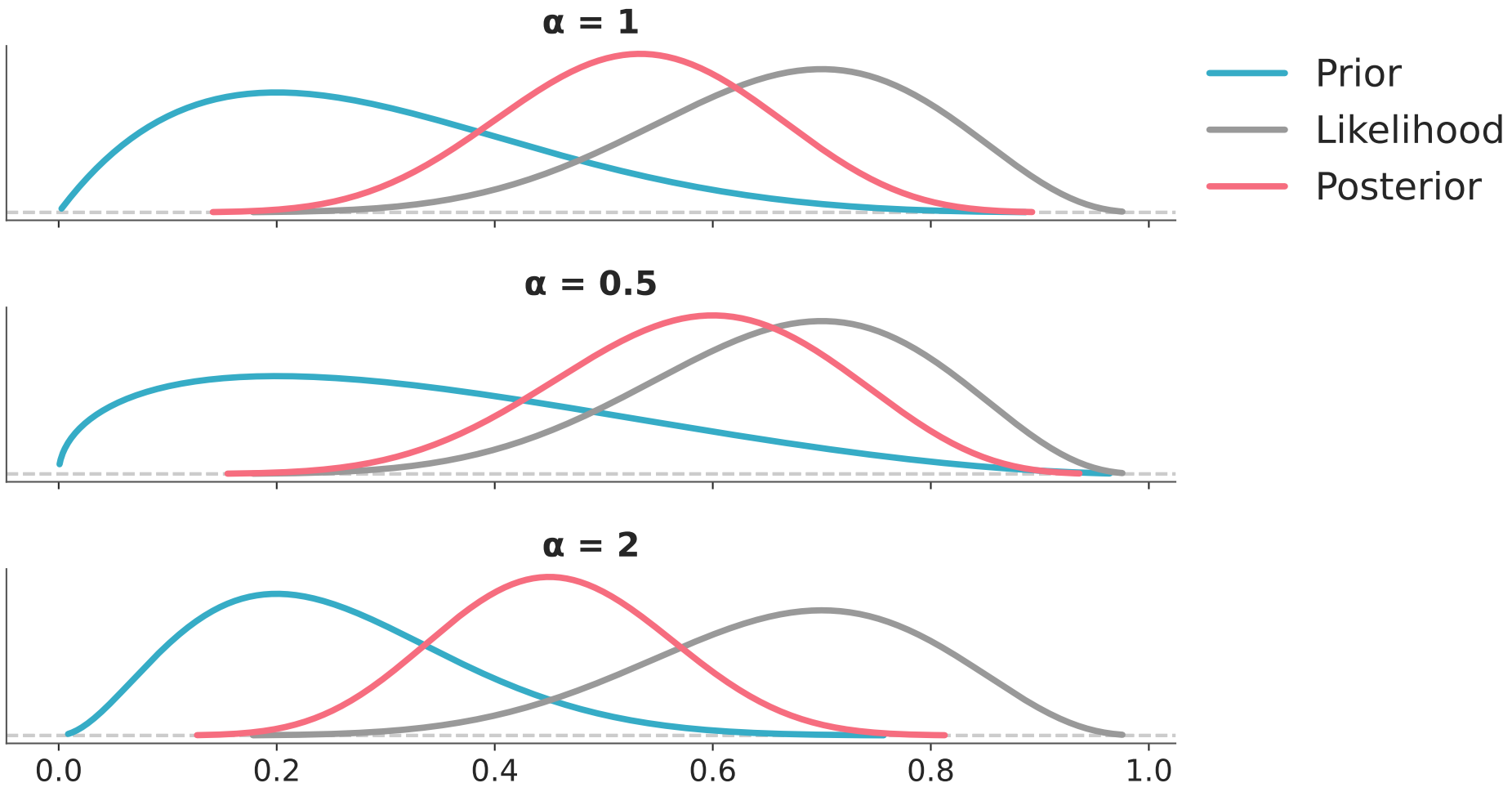
How it works

- The method it is based on two main ideas:
  - Importance sampling: avoid refitting the model
  - Power-scaling: we can write $p(\theta)^{\alpha}$ and use α the distribution more or less concentrated

Power-scaling distributions



Power-scaling the prior



Computation

For prior power-scaling:

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

Computation

For prior power-scaling:

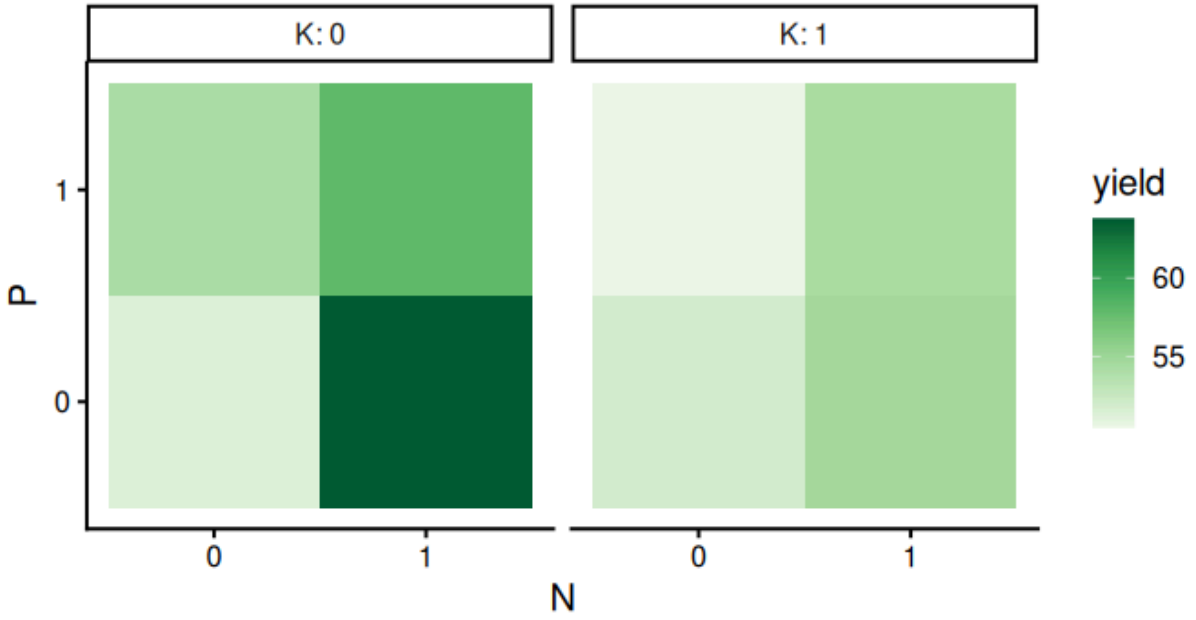
$$w_{\text{pri}}^{(s)} = \frac{p\left(\theta^{(s)}\right)^{\alpha} p\left(\theta^{(s)} \mid y\right)}{p\left(\theta^{(s)}\right) p\left(\theta^{(s)} \mid y\right)} = \frac{p\left(\theta^{(s)}\right)^{\alpha}}{p\left(\theta^{(s)}\right)} = p\left(\theta^{(s)}\right)^{\alpha-1}$$

For likelihood power-scaling:

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

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)



Mean yield per combination

Example

Model:

$$
\begin{aligned}
y_i &\sim N(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 \\
&\quad + \underbrace{\beta_{\mathrm{N}} N_i + \beta_{\mathrm{P}} P_i + \beta_{\mathrm{K}} K_i}_{\text{main effects}} \\
&\quad + \underbrace{\beta_{\mathrm{N}\times\mathrm{K}} N_i K_i + \beta_{\mathrm{N}\times\mathrm{P}} N_i P_i + \beta_{\mathrm{K}\times\mathrm{P}} K_i P_i}_{\text{2-way interactions}}
\end{aligned}
$$

Example: Initial model

Main effects only using `brms`

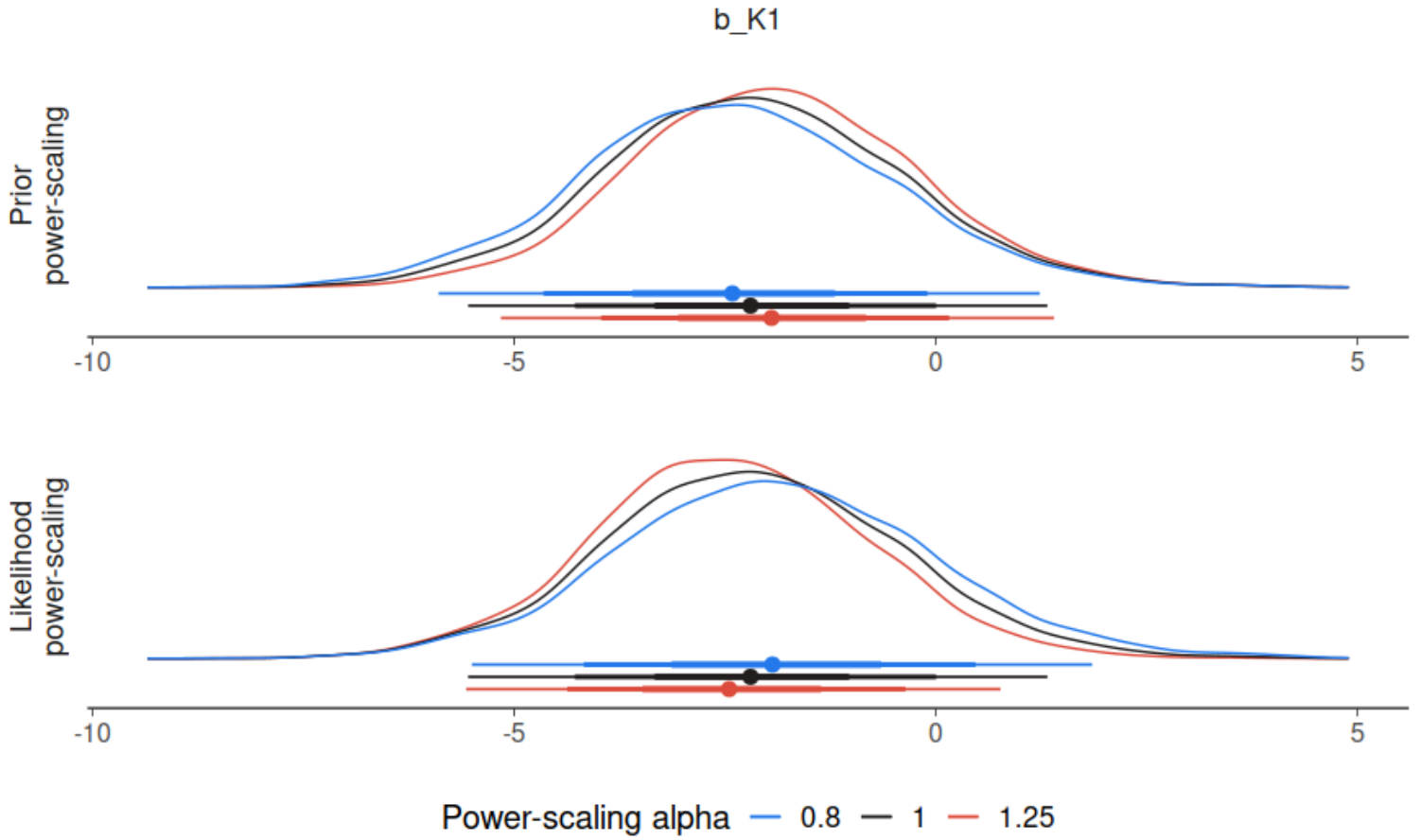
```
priors <- c(
  prior(normal(0, 2.5), coef = "N1", tag = "main"),
  prior(normal(0, 2.5), coef = "P1", tag = "main"),
  prior(normal(0, 2.5), coef = "K1", tag = "main")
)

fit <- brm(
  yield ~ N + P + K,
  data  = npk,
  prior = priors
)
```

We can later use the **tags** in priorsense to selectively apply power-scaling.

Example: Initial model checks

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



Example: Initial model checks

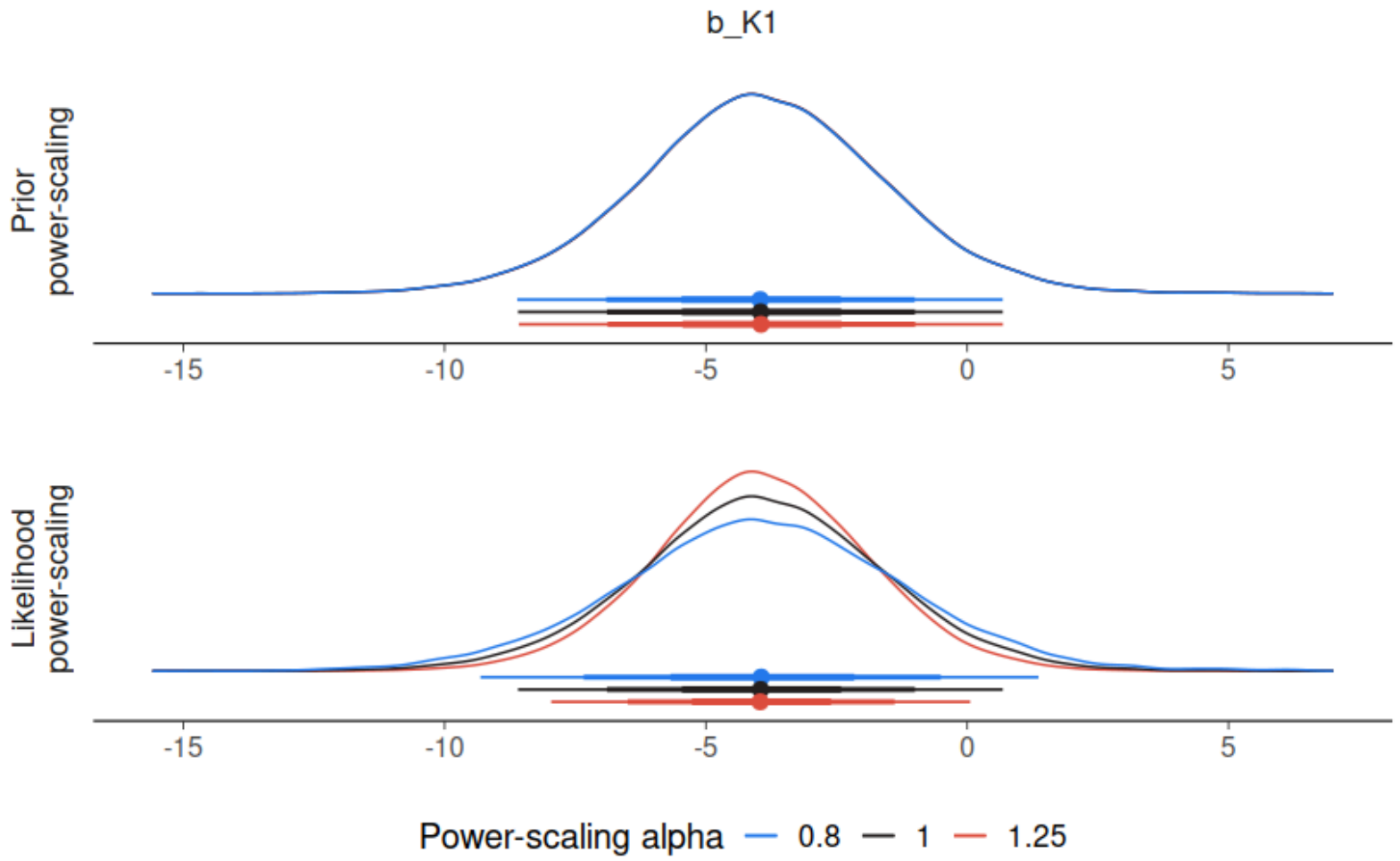
```
powerscale_sensitivity(fit, prior_selection = "main")
```

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

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

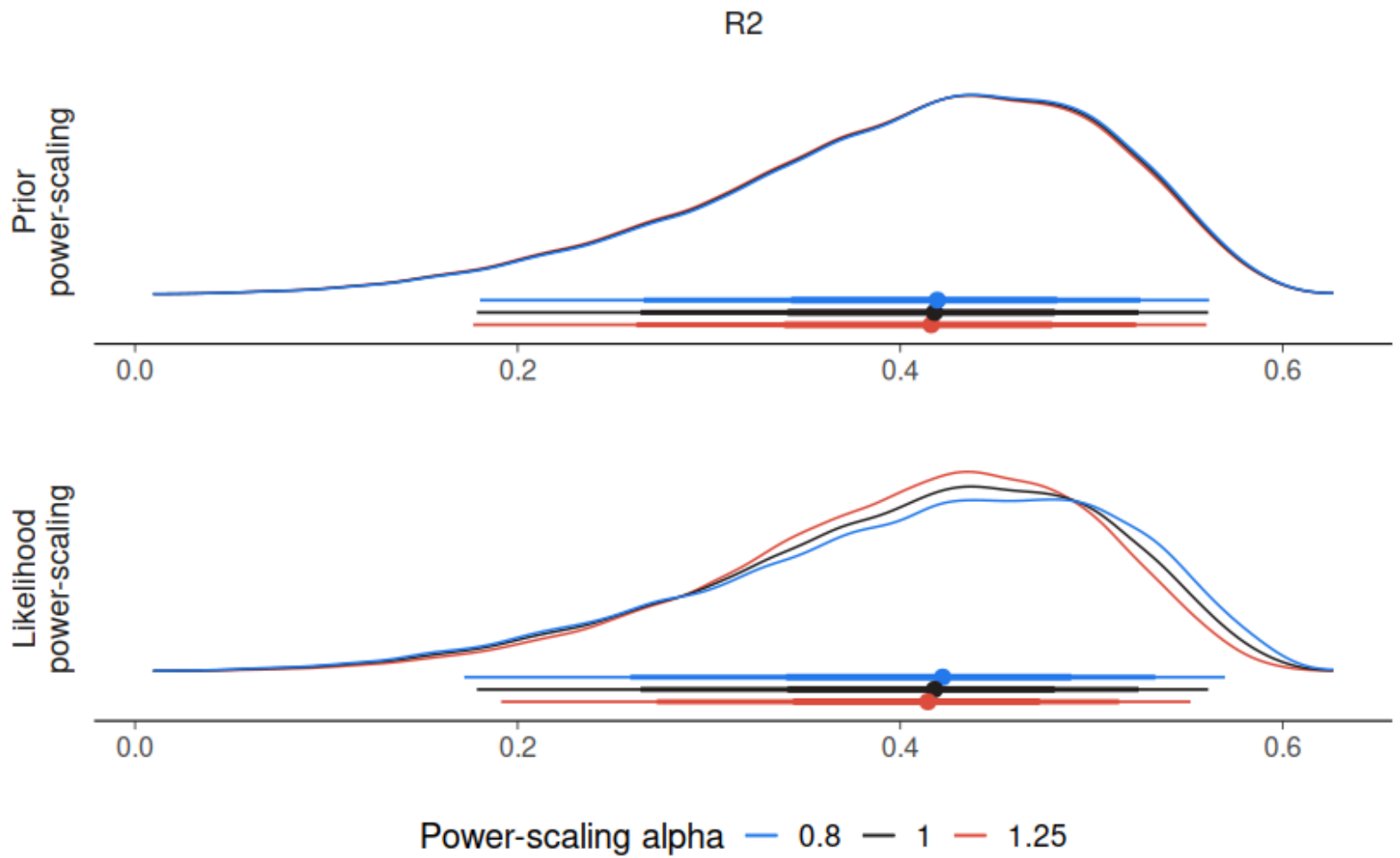
variable	prior sens.	likelihood sens.	diagnosis
b_N1	0.00	0.10	-
b_P1	0.00	0.09	-
b_K1	0.00	0.10	-

Example: Second model

Main effects and two-way interaction with `brms`

```
fit <- brm(
  yield ~ (N + P + K)^2,
  data = npk,
  prior = c(
    prior(normal(0, 25), coef = "N1", tag = "main"),
    prior(normal(0, 25), coef = "K1", tag = "main"),
    prior(normal(0, 25), coef = "P1", tag = "main"),
    prior(normal(0, 25), coef = "N1:P1", tag = "twoway"),
    prior(normal(0, 25), coef = "N1:K1", tag = "twoway"),
    prior(normal(0, 25), coef = "P1:K1", tag = "twoway")
  )
)
```

Example: Second model checks



Summary

- Priors can be challenging to specify without domain knowledge
- Prior predictive checking and prior/likelihood sensitivity checking can help identify potential issues
- If checks indicate strong prior sensitivity, prior-data conflict, or weak likelihood, further thought should be put into the prior
- see more examples and documentation:
  - R https://mc-stan.org/priorsense/
  - Python arviz-devs.github.io/EABM/Chapters/Sensitivity_checks.html

Variable selection

- The process of identifying the most relevant variables in a model from a larger set of predictors.
- We assume variables contribute unevenly to the outcome.
  - We may want to identify the most/least “important”ones.
  - Sometimes we also want to rank them.

Motivation for variable selection

- We can always include all available variables in a model
- Theory says this is a good idea, in particular if we use predictively consistent priors
- However, sometimes we need to reduce the number of variables
  - measurement cost in covariates
  - interpretability of the model
  - running cost of predictive model

Rihan et al. (2026). To select or not to select: predictively consistent priors instead of model selection

The problem with variable selection

- The number of potential models is 2^p, where p is the number of variables
- Evaluating all models can be computationally infeasible even for moderate p
- The process is prone to overfitting

How to overcome the problem?

- We recommend to use a technique called projection predictive inference
  - In R, it can be done with `brms` + `projpred`
  - In Python, it can be done with `bambi` + `kulprit`

Variable selection with projpred/kulprit

- The main advantage is that it reduces overfitting
- Other advantages are:
  - Automatic model building and fitting process.
  - Reduced number of models we need to fit.
  - Reduced time it takes to fit each model.

Main concepts

- Reference model:
  - A model that includes all available variables and describes the data well.
- Search strategy:
  - A method for searching through the model space.
- Projection:
  - A way to estimate the posterior distribution of a model given a reference model.

Using a reference model is not a novel idea: Lindley (1968), Goutis & Robert (1998). Related approaches are known as gold standard, preconditioning, teacher and student, and distilling.

Why use a reference model?

- Intuitively: data = signal + noise
- Selecting variables directly from the data also picks up the noise
- A reference model is our best attempt to estimate the signal

How projection is done

- Projection can be framed as an **optimization problem**
- **Reference model**: parameters θ (a posterior draw)
- **Submodel**: parameters $\theta_\perp$
- $\tilde{y}$: samples from the reference model's posterior predictive distribution, $p(\tilde{y} \mid \theta)$
- Find $\theta_\perp$ so that the submodel's predictive distribution $q(\tilde{y} \mid \theta_\perp)$ is **as close as possible** to $p(\tilde{y} \mid \theta)$

How projection is done

Use Kullback-Leibler divergence:

$$\mathrm{KL}(p(\tilde{y} \mid \theta) \parallel q(\tilde{y} \mid \theta_\perp)) = \mathbb{E}_{\tilde{y} \sim p(\tilde{y} \mid \theta)} \left[\log \frac{p(\tilde{y} \mid \theta)}{q(\tilde{y} \mid \theta_\perp)} \right]$$

Reordering:

$$\mathrm{KL} = \underbrace{\mathbb{E}_{\tilde{y} \sim p(\tilde{y} \mid \theta)} [\log p(\tilde{y} \mid \theta)]}_{\text{constant}} - \mathbb{E}_{\tilde{y} \sim p(\tilde{y} \mid \theta)} [\log q(\tilde{y} \mid \theta_\perp)]$$

The first term does not depend on $\theta_\perp$, so

$$\text{minimize KL} \quad \Leftrightarrow \quad \text{maximize} \quad \mathbb{E}_{\tilde{y} \sim p(\tilde{y} \mid \theta)} [\log q(\tilde{y} \mid \theta_\perp)]$$

Take q to be the **submodel**'s likelihood (same observation model as the reference).

The projection is the maximum likelihood fit of the **submodel** to data $\tilde{y}$ simulated from the **reference model**

Searching for submodels

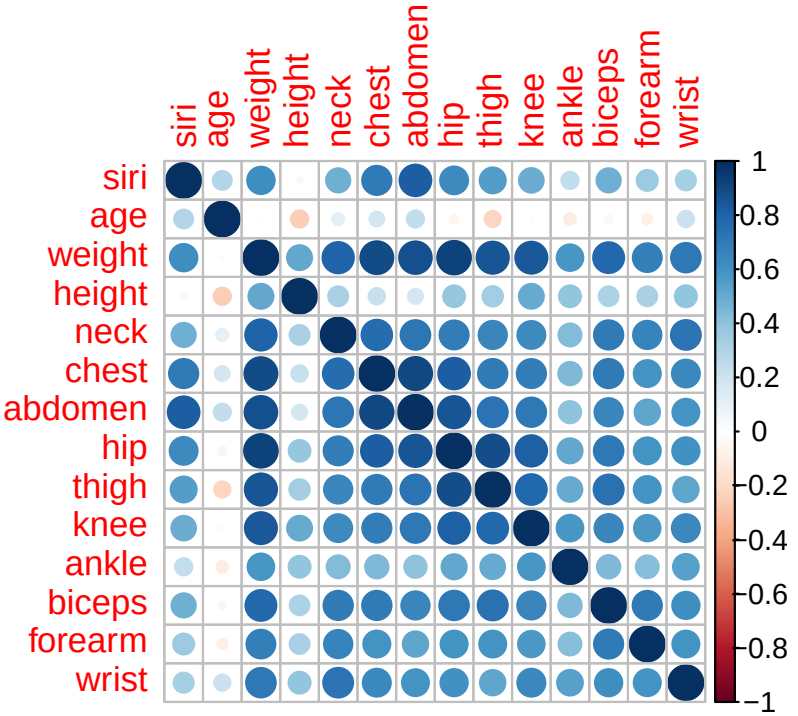
- We may not be able to evaluate the 2^p combinations
- Search heuristics, e.g.
  - Forward search
  - L_1-penalization (as in Lasso)
  - ...
- For a given model size, choose feature combination with minimal projective loss
- Use cross-validation to select the appropriate model size
  - In some cases like, $p \gg n$, we need to cross-validate over the search paths

Bodyfat: small p example of projection predictive

- Predict bodyfat `siri`. The reference value is obtained by immersing person in water. $n = 251$.
- We also have measurements of various body parts: age, weight, abdomen, wrist, hip, thigh, ankle, forearm, height, neck, knee, biceps, chest

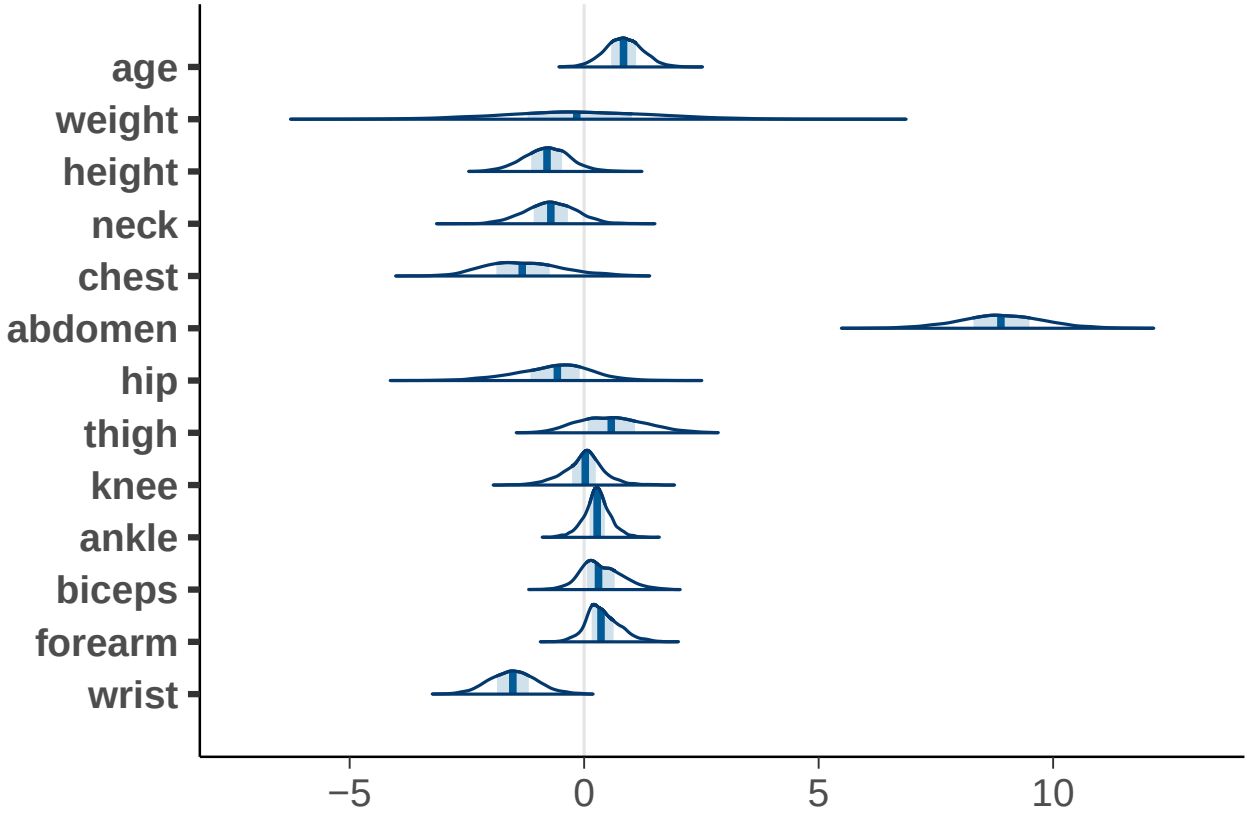
Bodyfat: small p example of projection predictive

- Predict bodyfat `siri`. The reference value is obtained by immersing person in water. $n = 251$.
- We also have measurements of various body parts: age, weight, abdomen, wrist, hip, thigh, ankle, forearm, height, neck, knee, biceps, chest



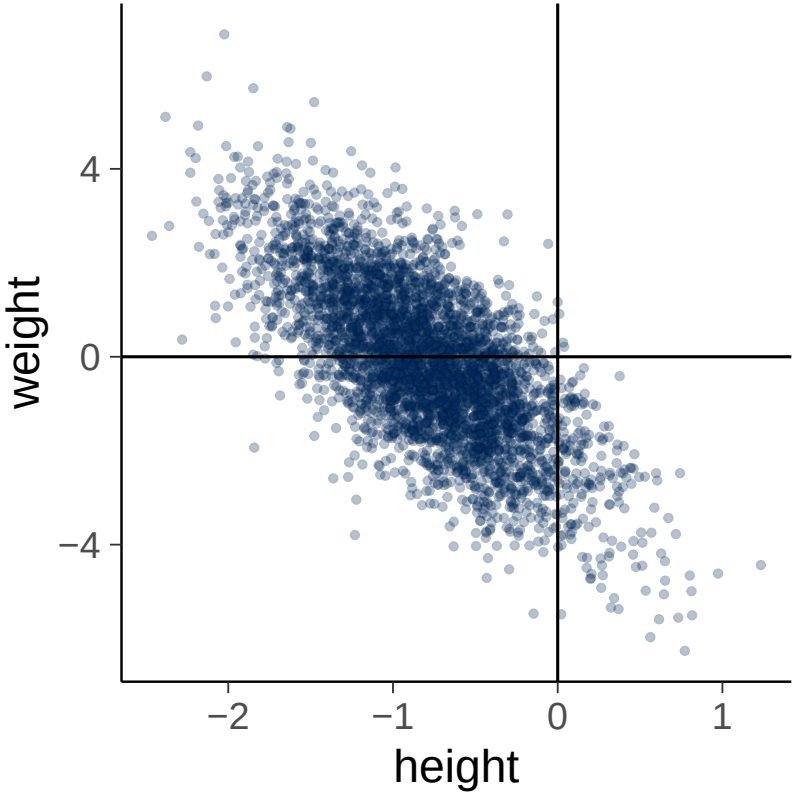
Bodyfat

Marginal posteriors of coefficients



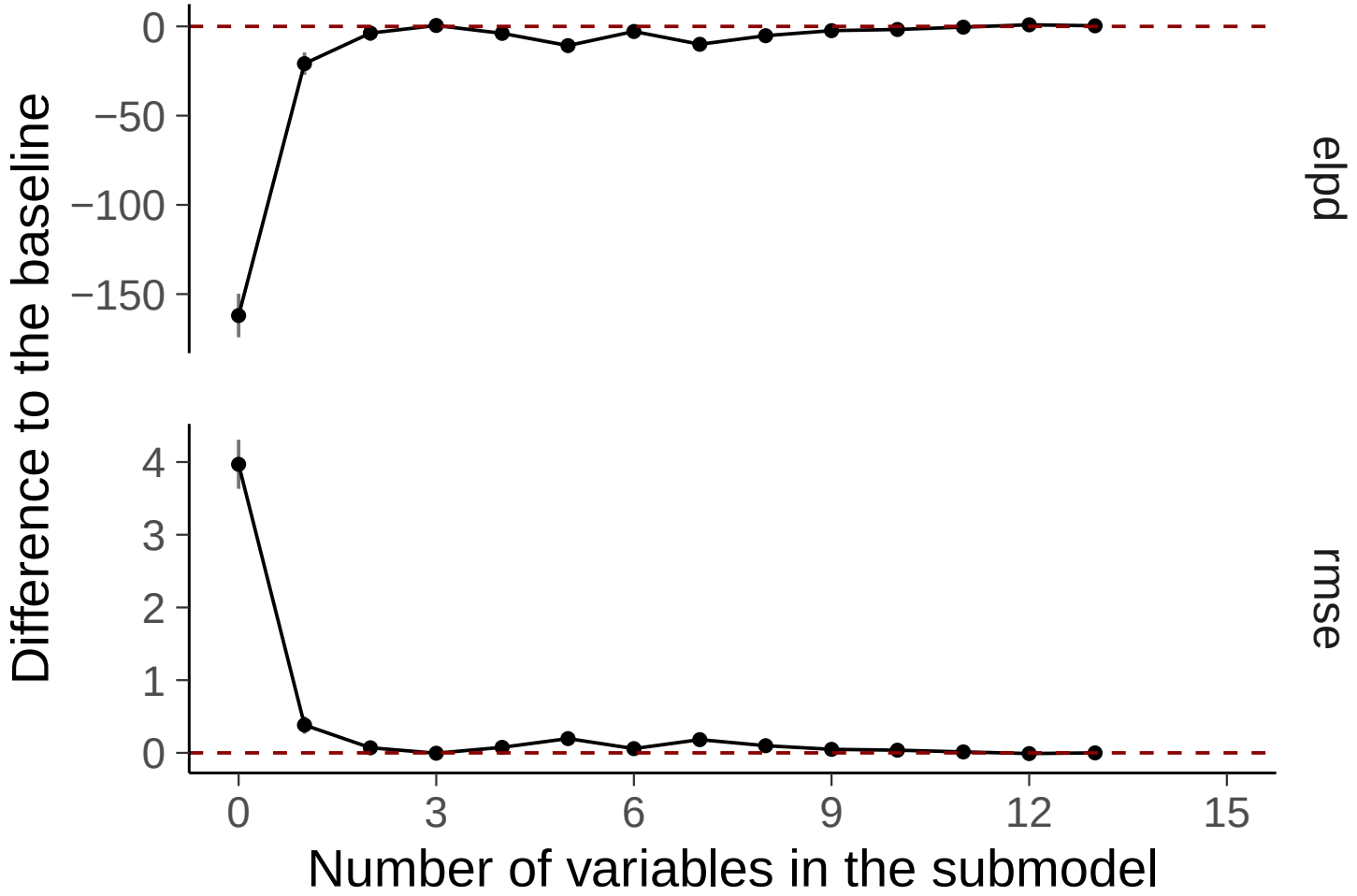
Bodyfat

Bivariate marginal of weight and height



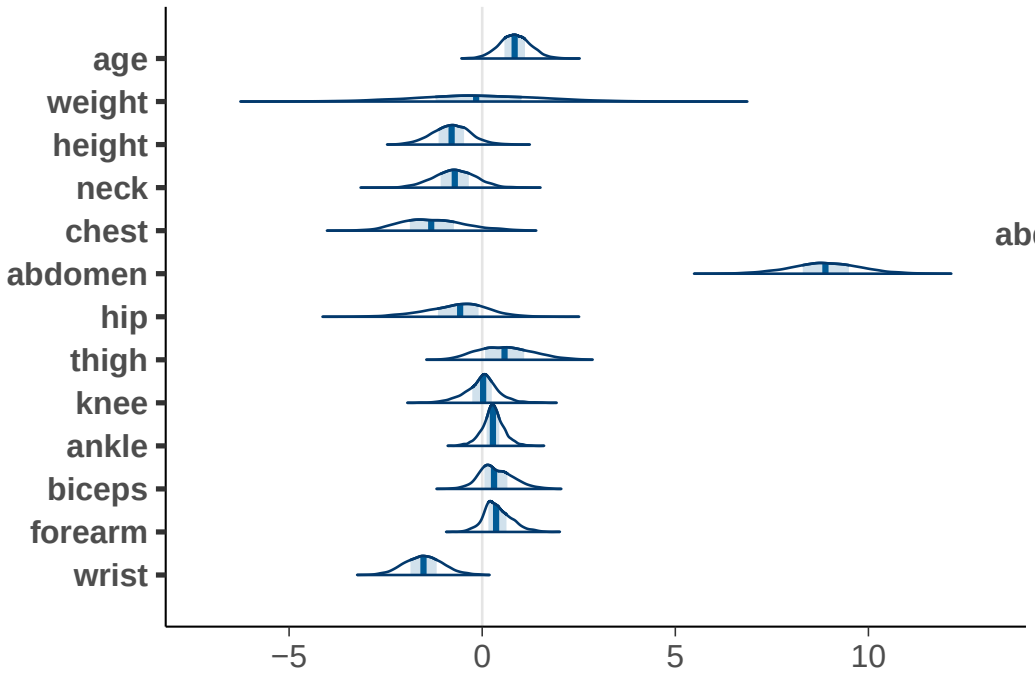
Bodyfat

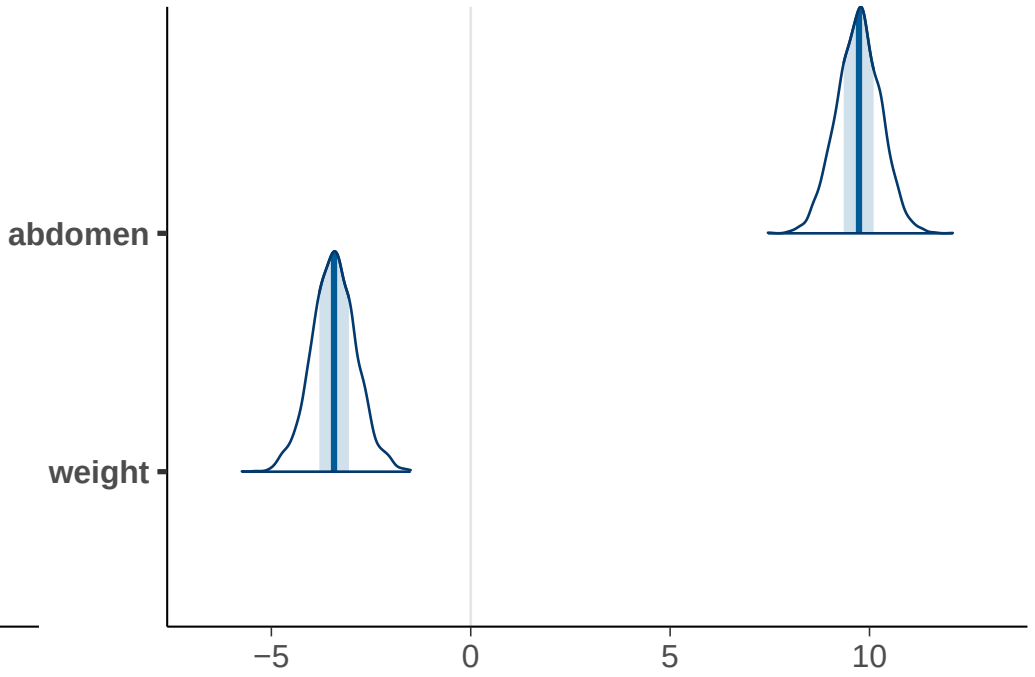
The predictive performance of the full and submodels



Bodyfat

Marginals of the reference and projected posterior



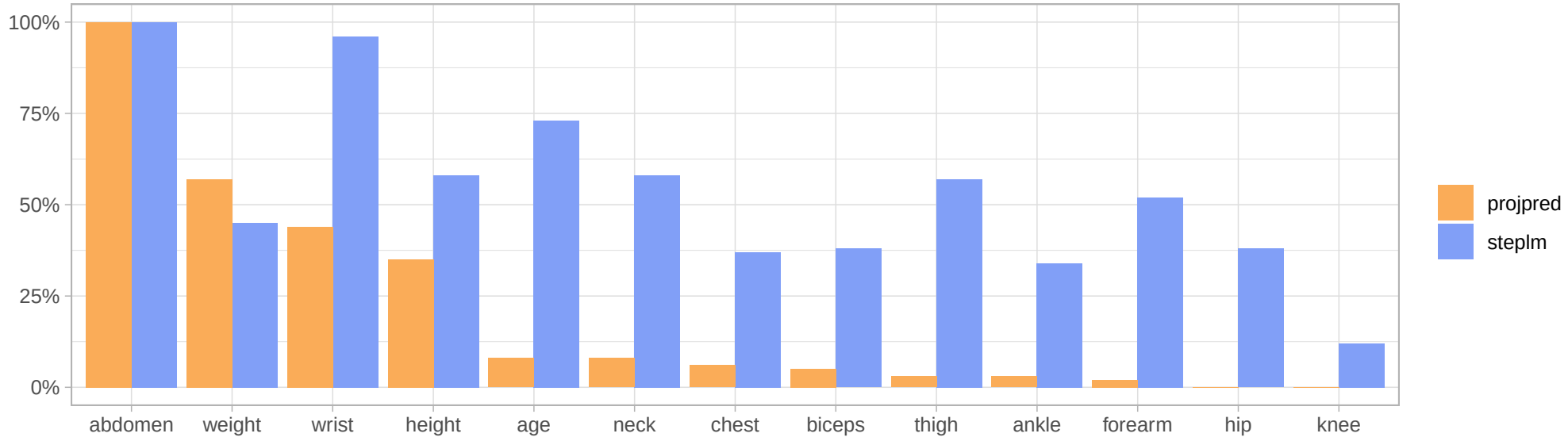


Predictive performance vs. selected variables

- The initial aim: find the minimal set of variables providing similar predictive performance as the reference model
- Some keep asking can it find the true variables
  - What do you mean by true variables?
- The goal is predictive performance, not causal inference

Variability under data perturbation

Comparing projection predictive variable selection (projpred) and stepwise maximum likelihood over bootstrapped datasets



Intro paper and examples

- McLatchie, Rögnvaldsson, Weber, and Aki Vehtari (2024). Advances in projection predictive inference. *Statistical Science*.
- https://mc-stan.org/projpred/articles/projpred.html
- https://users.aalto.fi/~ave/casestudies.html
- https://kulprit.readthedocs.io