

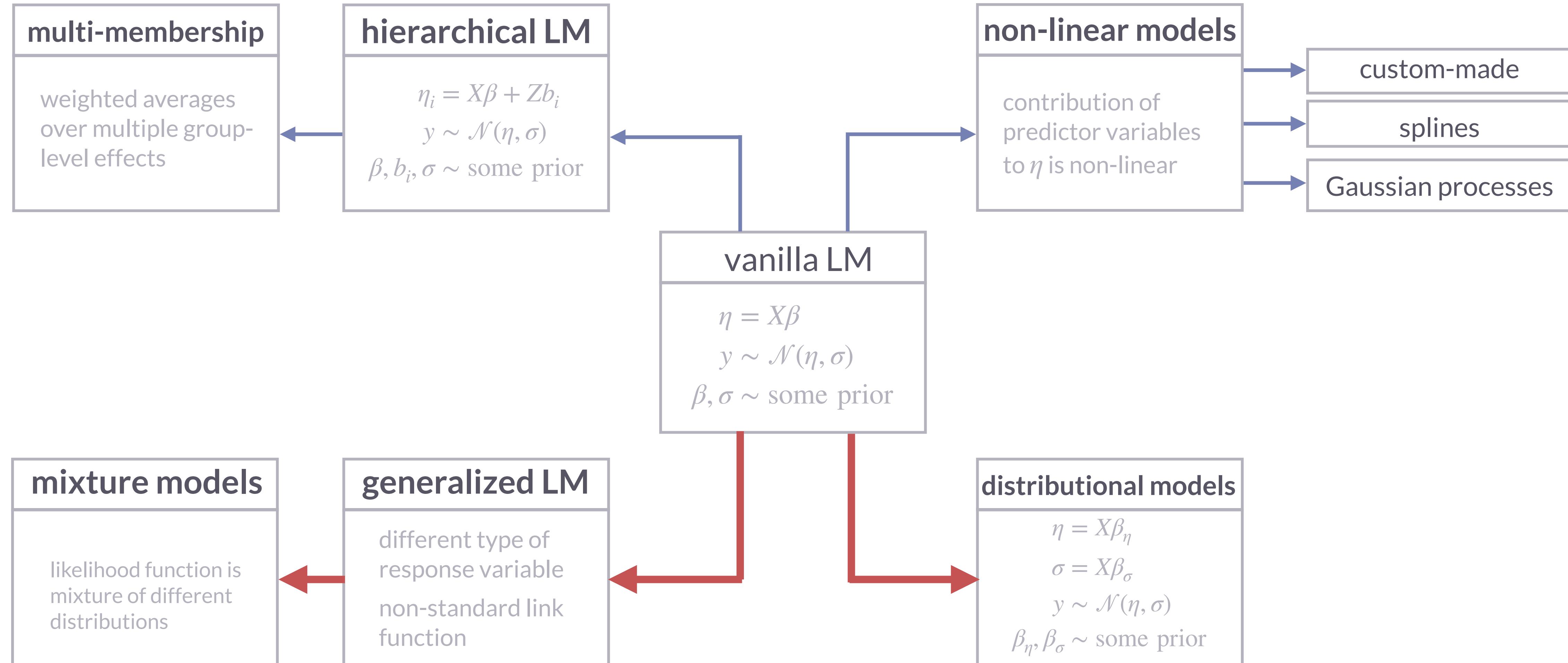
Bayesian regression modeling: Theory & practice

Part 4: Generalized linear models & beyond

Michael Franke

Roadmap “beyond vanilla”

common extensions of linear regression modeling

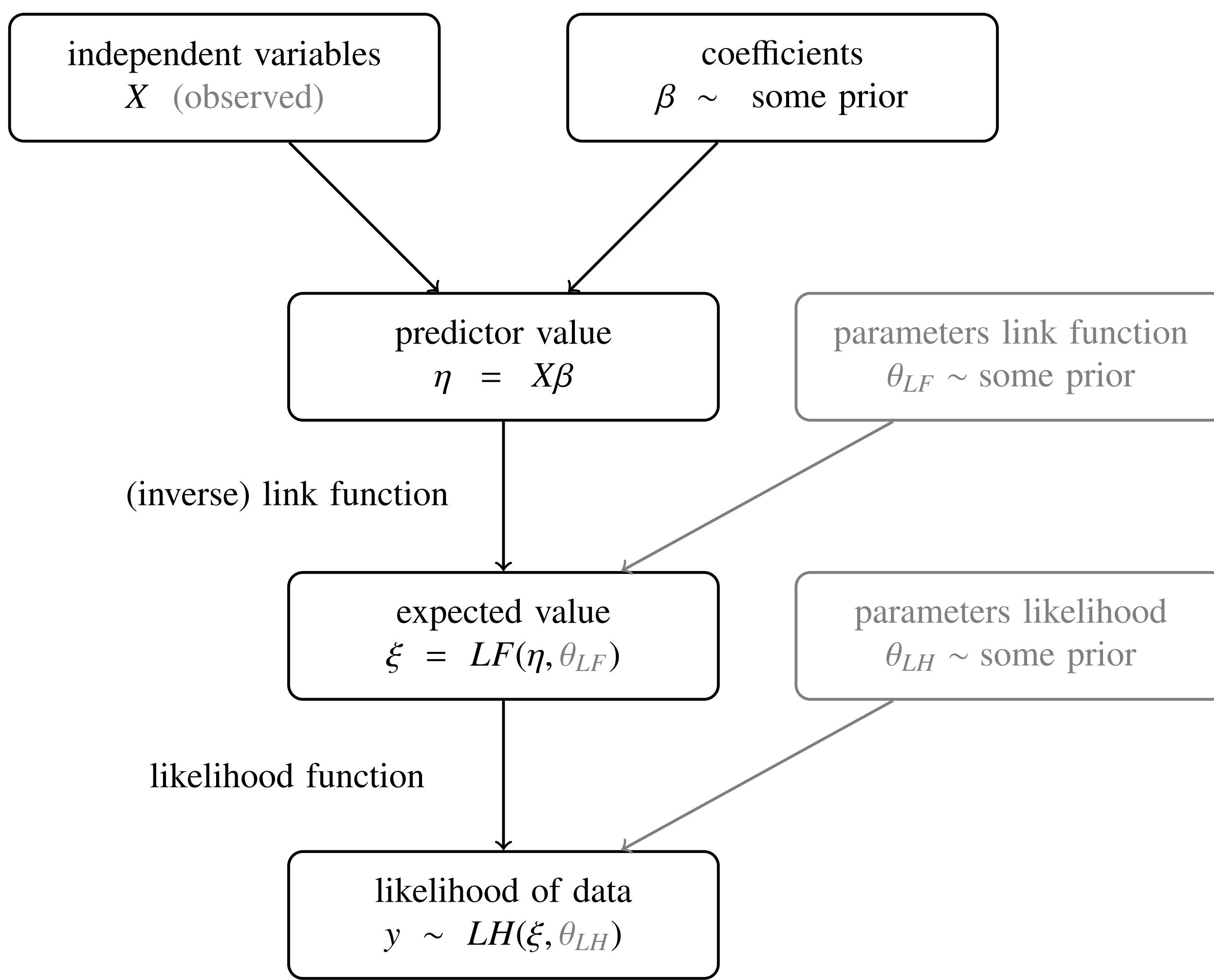




Generalized linear regression models

Generalized linear regression model

Coefficients → linear predictor → central tendency → likelihood



Simple linear regression

$$\eta_i = \mathbf{x}_i \cdot \beta$$

[linear predictor]

$$\xi_i = \eta_i$$

[predictor of central tendency]

$$y_i \sim \text{Normal}(\xi_i, \sigma)$$

[likelihood]

Logistic regression

$$\eta_i = \mathbf{x}_i \cdot \beta$$

[linear predictor]

$$\xi_i = \text{logistic}(\eta_i)$$

[predictor of central tendency]

$$y_i \sim \text{Bernoulli}(\xi_i)$$

[likelihood]

Poisson regression

$$\eta_i = \mathbf{x}_i \cdot \beta$$

[linear predictor]

$$\xi_i = \exp(\eta_i)$$

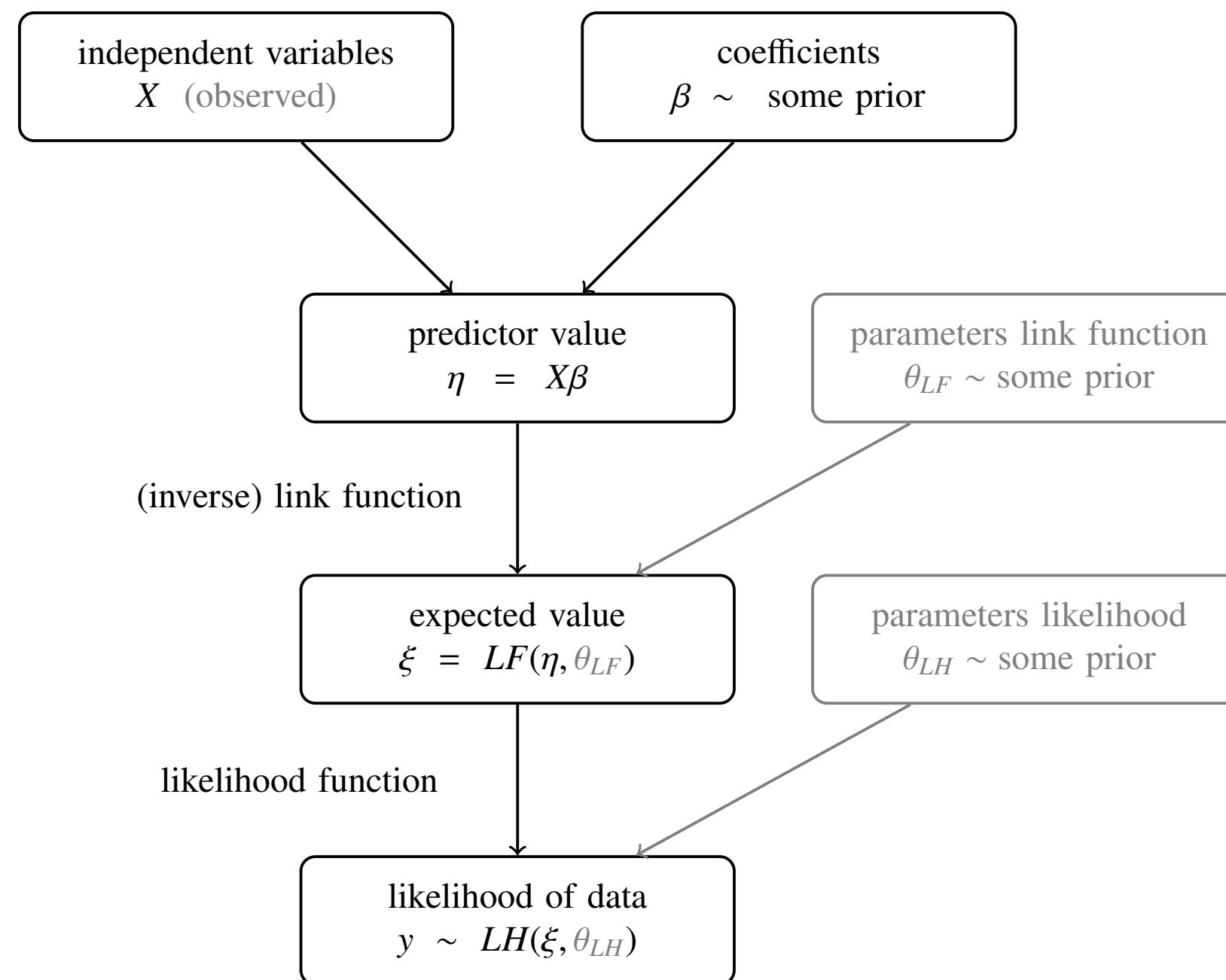
[predictor of central tendency]

$$y_i \sim \text{Poisson}(\xi_i)$$

[likelihood]

Generalized linear regression model

Coefficients → linear predictor → central tendency → likelihood



type of y	(inverse) link function	likelihood function
metric	$\xi = \eta$	$y \sim \text{Normal}(\xi; \sigma)$
binary	$\xi = \text{logistic}(\eta)$	$y \sim \text{Bernoulli}(\xi)$
nominal	$\xi = \text{soft-max}(\eta)$	$y \sim \text{Categorical}(\xi)$
ordinal	$\xi = \text{cumulative-logit}(\eta; \delta)$	$y \sim \text{Categorical}(\xi)$
count	$\xi = \exp(\eta)$	$y \sim \text{Poisson}(\xi)$

The BRMS “family of families”

- ▶ link- and likelihood function are set by family parameter in `brm` function
- ▶ requires an object of type `brms::brmsfamily`
 - many predefined families, listed [here](#)
 - instantiated by function calls like `cumulative()`
 - allow flexible parameterization
 - documentation of parameterization: [here](#)
 - creating custom families is possible, see [here](#)

```
fit_ordinal <- brm(  
  formula = prototype_label ~ MAD,  
  data = data_MT_preppe2,  
  family = cumulative()  
)
```



Logistic regression

Logistic regression

Definition

$$\eta_i = \mathbf{x}_i \cdot \beta$$

[linear predictor]

$$\xi_i = \text{logistic}(\eta_i)$$

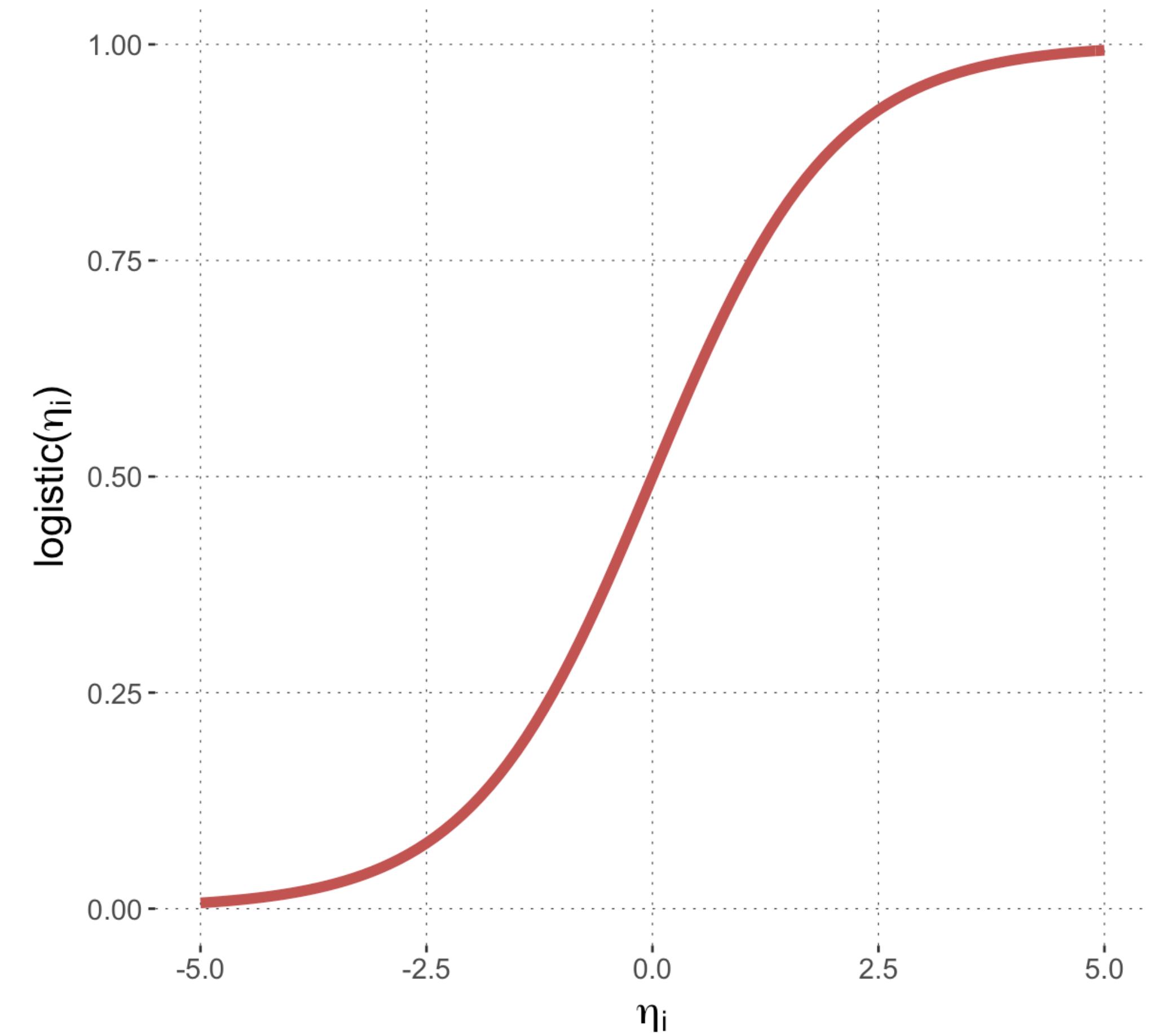
[predictor of central tendency]

$$y_i \sim \text{Bernoulli}(\xi_i)$$

[likelihood]

```
fit_logistic <- brm(  
  formula = correct ~ group * condition,  
  data = data_MT |>  
    mutate(correct = factor(ifelse(correct, "correct", "incorrect"),  
          levels = c("incorrect", "correct"))),  
  family = bernoulli(link="logit"))
```

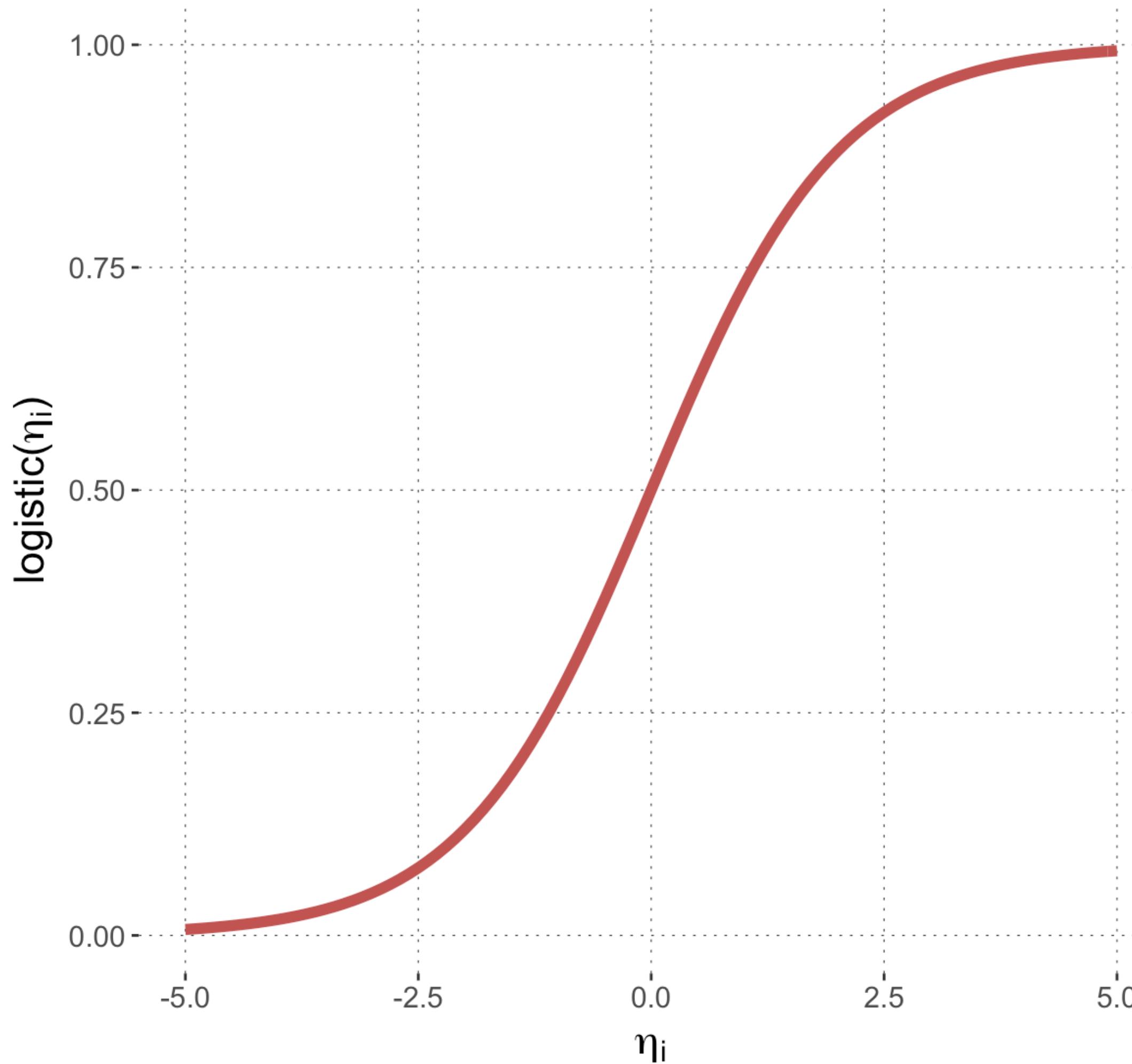
Link function: logistic



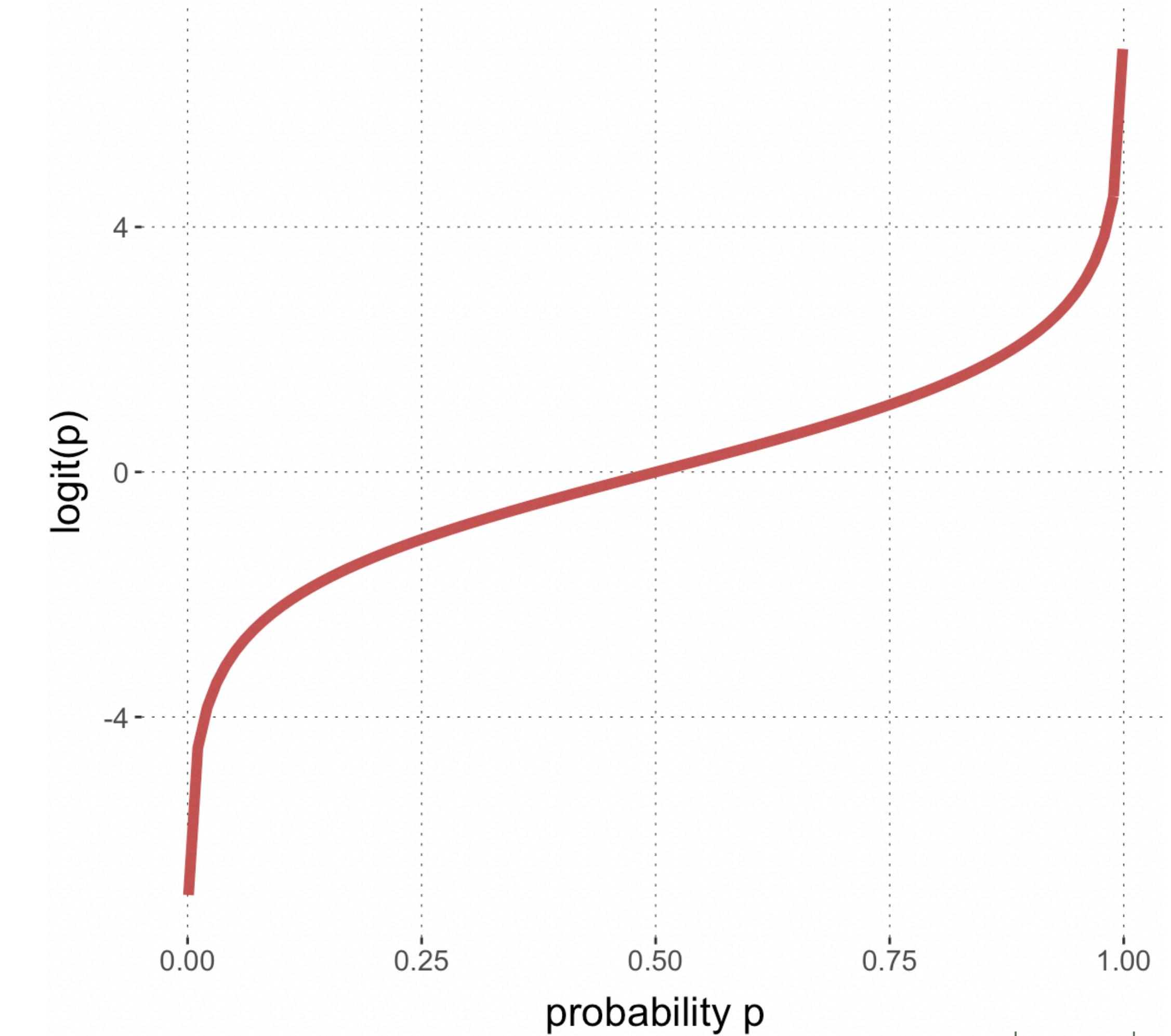
Logistic regression

Link & inverse link function

Link function: logistic



“Inverse” link function: logit



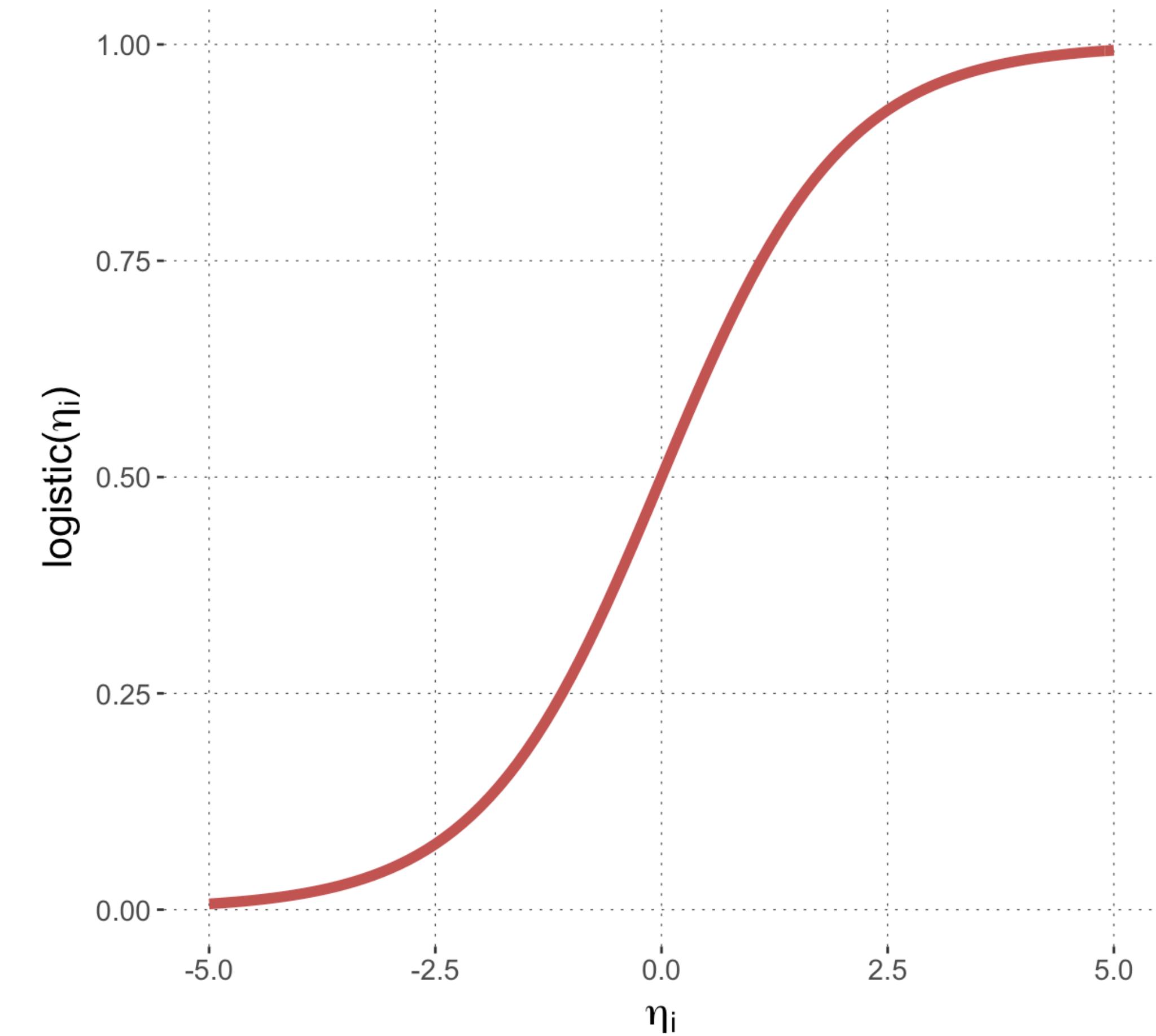
Logistic regression

interpretation

- ▶ linear predictor η encodes the log-odds
 - probability of “heads” vs “tails” (1 vs 0)
- ▶ unit change of η can be seen as a change in beliefs corresponding to a Bayes factor of ~ 2.72

$$\begin{aligned}\eta_1 - \eta_2 &= \log \frac{\xi_1}{1 - \xi_1} - \log \frac{\xi_2}{1 - \xi_2} = \log \left(\frac{\xi_1}{1 - \xi_1} \frac{1 - \xi_2}{\xi_2} \right) \\ \Leftrightarrow \frac{\xi_1}{1 - \xi_1} &= \exp(\eta_1 - \eta_2) \frac{\xi_2}{1 - \xi_2}\end{aligned}$$

Link function: logistic





Multinomial regression

Multinomial regression

- ▶ we want to predict probabilities $\mathbf{p} = \langle p_1, \dots, p_k \rangle$
 - p_j is the prediction for category j probability of
- ▶ it suffices to estimate $k - 1$ probabilities
 - probabilities sum to one
 - fix a reference category (similar to treatment coding!)
- ▶ (non-normalized) weights s_j from linear predictors:
- ▶ probabilities from soft-max:

$$s_j = \mathbf{x}_i \cdot \boldsymbol{\beta}^j$$
$$p_j = \frac{\exp s_j}{\sum_{j'=1}^k \exp s'_j}$$

Interpretation

- ▶ linear predictor predictor s_j represents $\log(p_j/p_1)$
- ▶ Think of this as ...

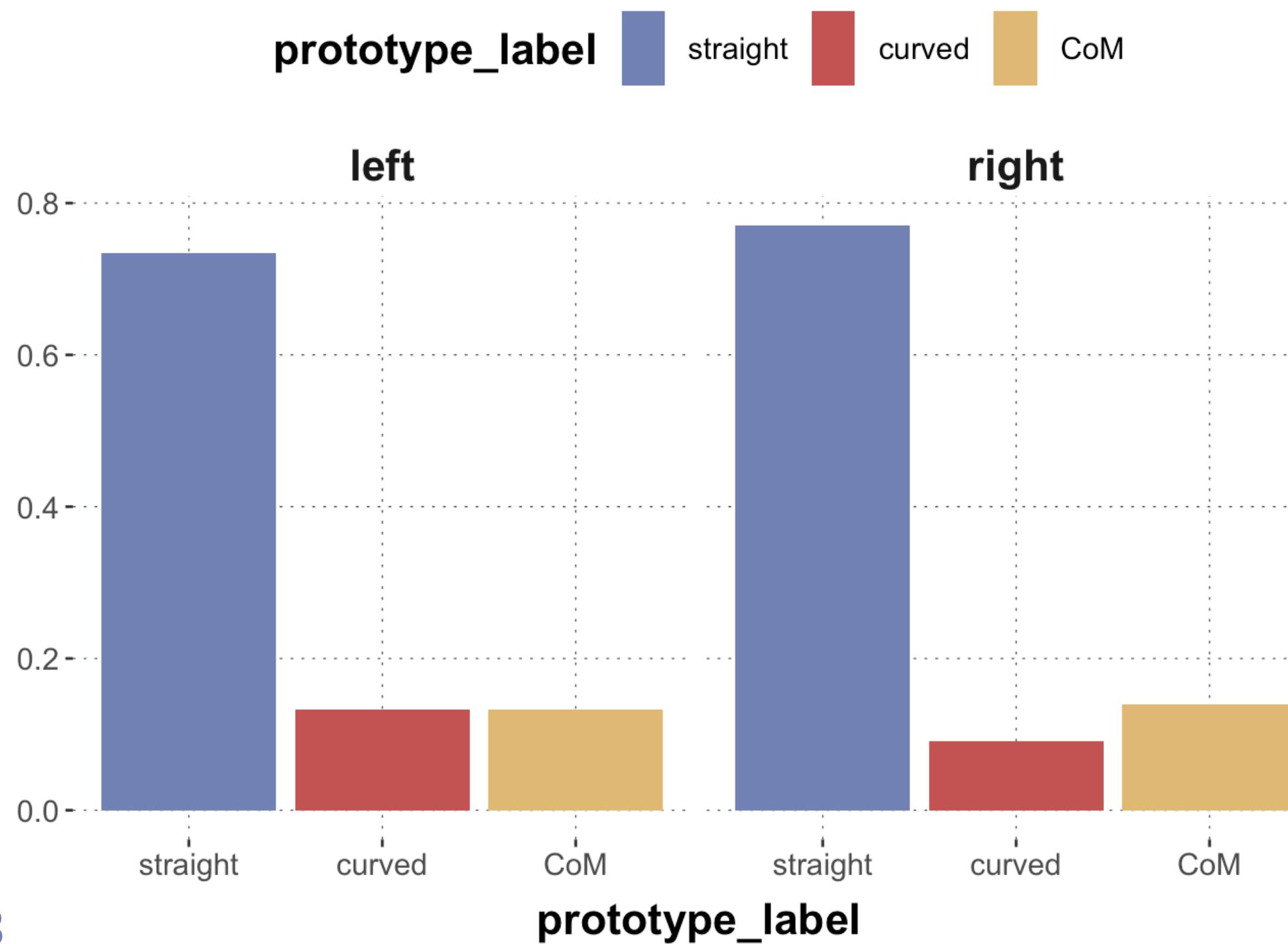
$k - 1$ parallel logistic regressions

```
fit_multinom <- brm(  
  formula = prototype_label ~ target_position,  
  data = data_MT_prepped,  
  family = categorical()  
)
```

Multinomial regression

```
fit_multinom <- brm(  
  formula = prototype_label ~ target_position,  
  data = data_MT_preppeD,  
  family = categorical()  
)
```

```
Family: categorical  
Links: mucurved = logit; muCoM = logit  
Formula: prototype_label ~ target_position  
Data: data_MT_prep (Number of observations: 2052)  
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;  
       total post-warmup draws = 4000
```



Population-Level Effects

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat
mucurved_Intercept	-1.71	0.09	-1.89	-1.53	1.00
muCoM_Intercept	-1.71	0.09	-1.90	-1.53	1.00
mucurved_target_positionright	-0.44	0.14	-0.72	-0.16	1.00
muCoM_target_positionright	-0.00	0.13	-0.26	0.25	1.00
	Bulk_ESS	Tail_ESS			
mucurved_Intercept	4595	3205			
muCoM_Intercept	3696	2886			
mucurved_target_positionright	3925	3178			
muCoM_target_positionright	3910	3066			

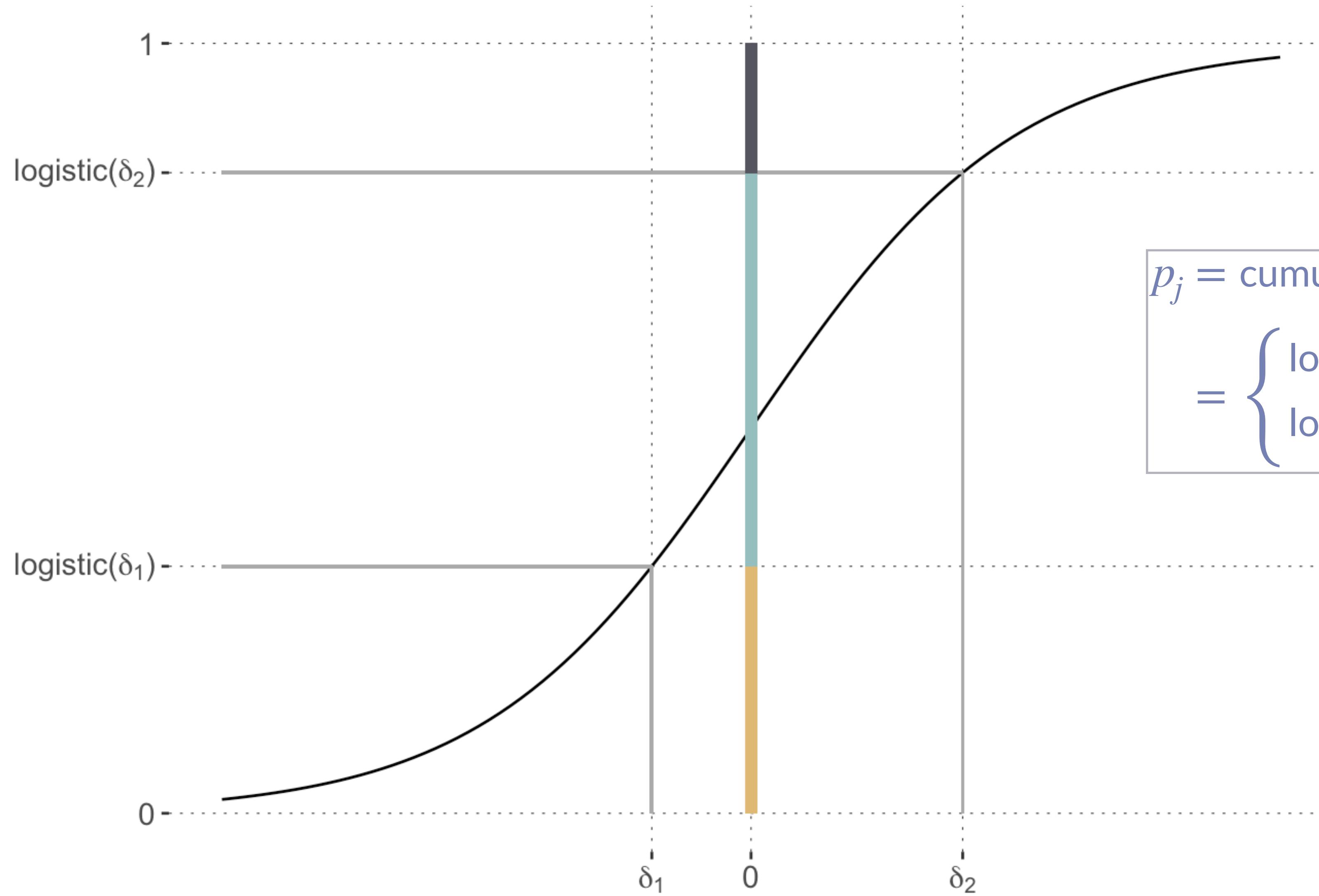
Draws were sampled using `sampling(NUTS)`. For each parameter, `Bulk_ESS` and `Tail_ESS` are effective sample size measures, and `Rhat` is the potential scale reduction factor on split chains (at convergence, `Rhat = 1`).



Ordinal regression

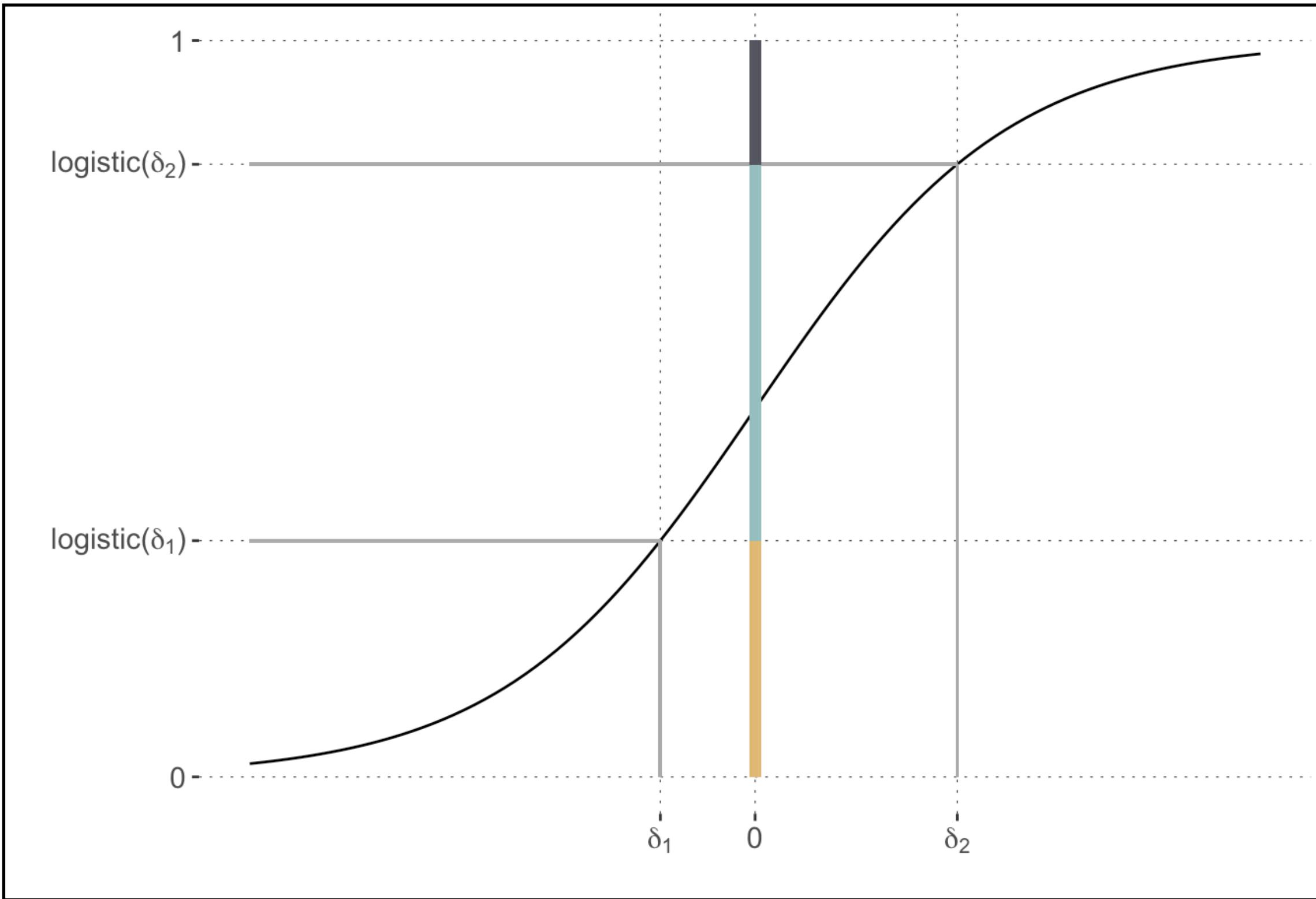
Ordinal regression

cumulative logit



Ordinal regression

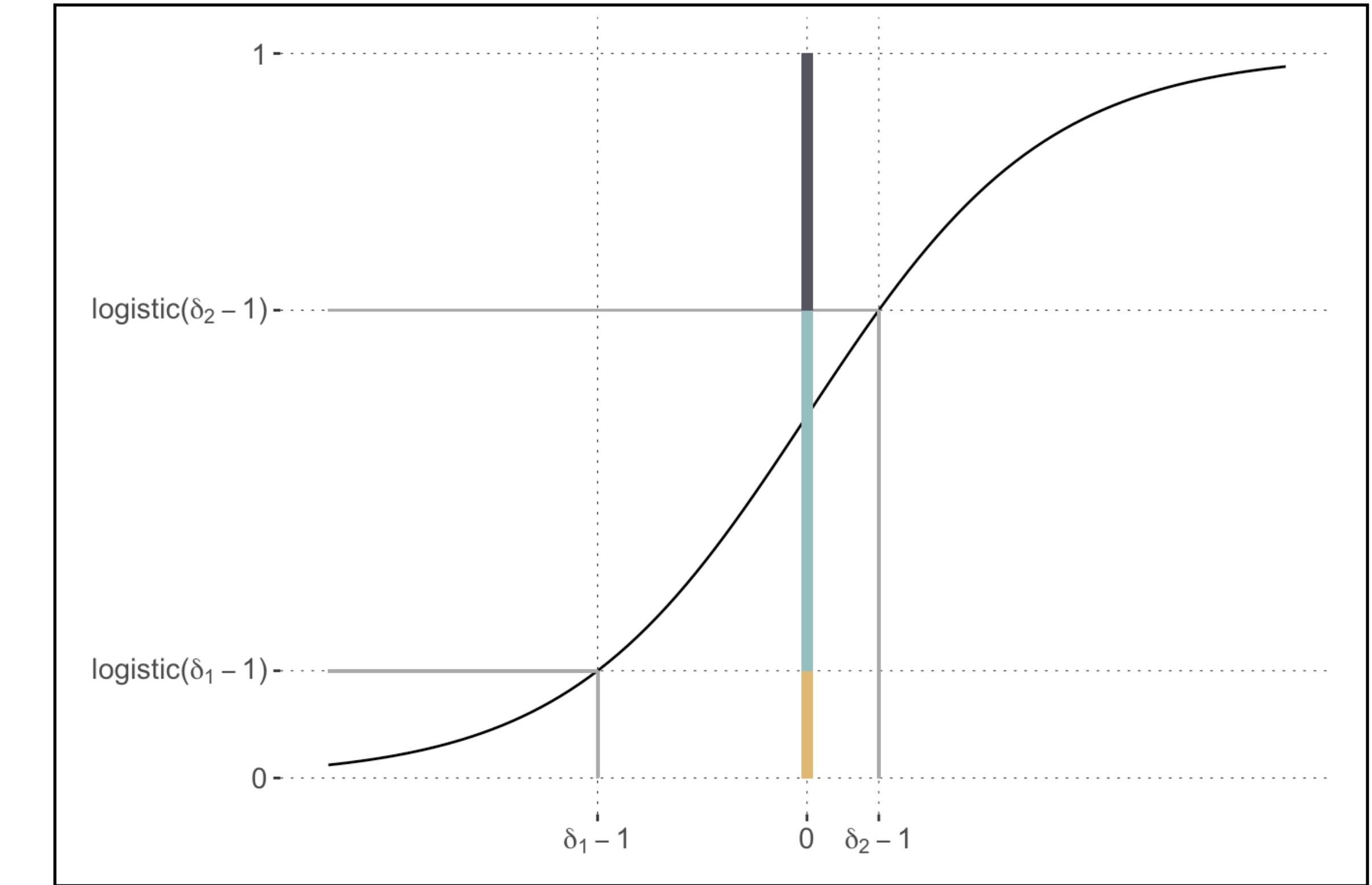
cumulative logit



$$\eta_i = \mathbf{x}_i \cdot \beta$$

$$\xi_i = \text{cumulative-logit}(\eta_i; \delta)$$

$$y_i \sim \text{Categorical}(\xi_i)$$



[linear predictor]

[predictor of central tendency]

[likelihood]

Ordinal regression

cumulative logit

```
fit_ordinal <- brm(  
  formula = prototype_label ~ MAD,  
  data = data_MT_preppe2,  
  family = cumulative()  
)
```

Family: cumulative
Links: mu = logit; disc = identity
Formula: prototype_label ~ MAD
Data: data_MT_preppe2 (Number of observations: 2052)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
total post-warmup draws = 4000

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept[1]	4.05	0.18	3.71	4.40	1.00	2216	2247
Intercept[2]	9.52	0.51	8.56	10.54	1.00	2003	1979
MAD	0.02	0.00	0.02	0.03	1.00	2543	2514

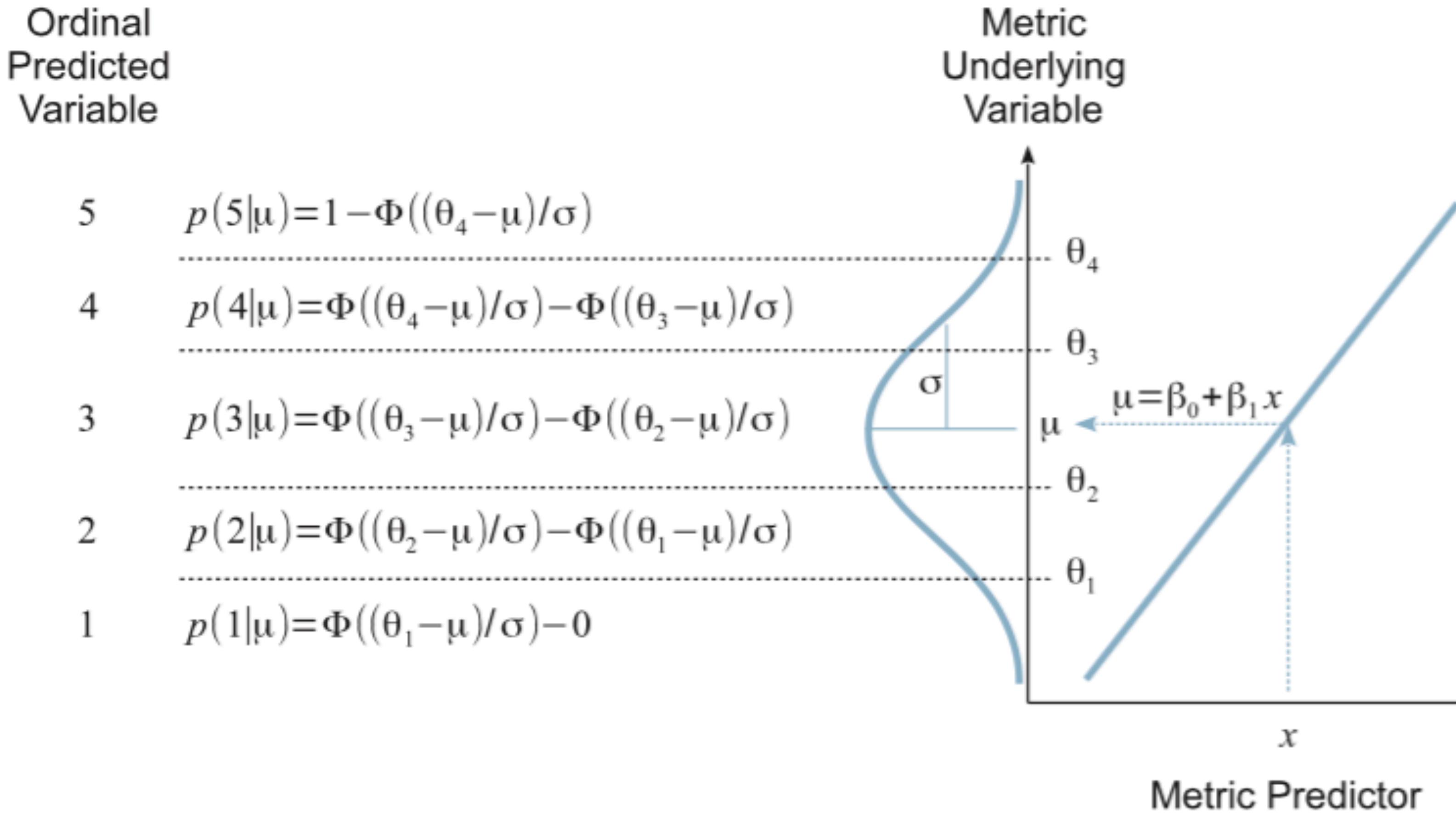
Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
disc	1.00	0.00	1.00	1.00	NA	NA	NA

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).

Ordinal regression

cumulative profit

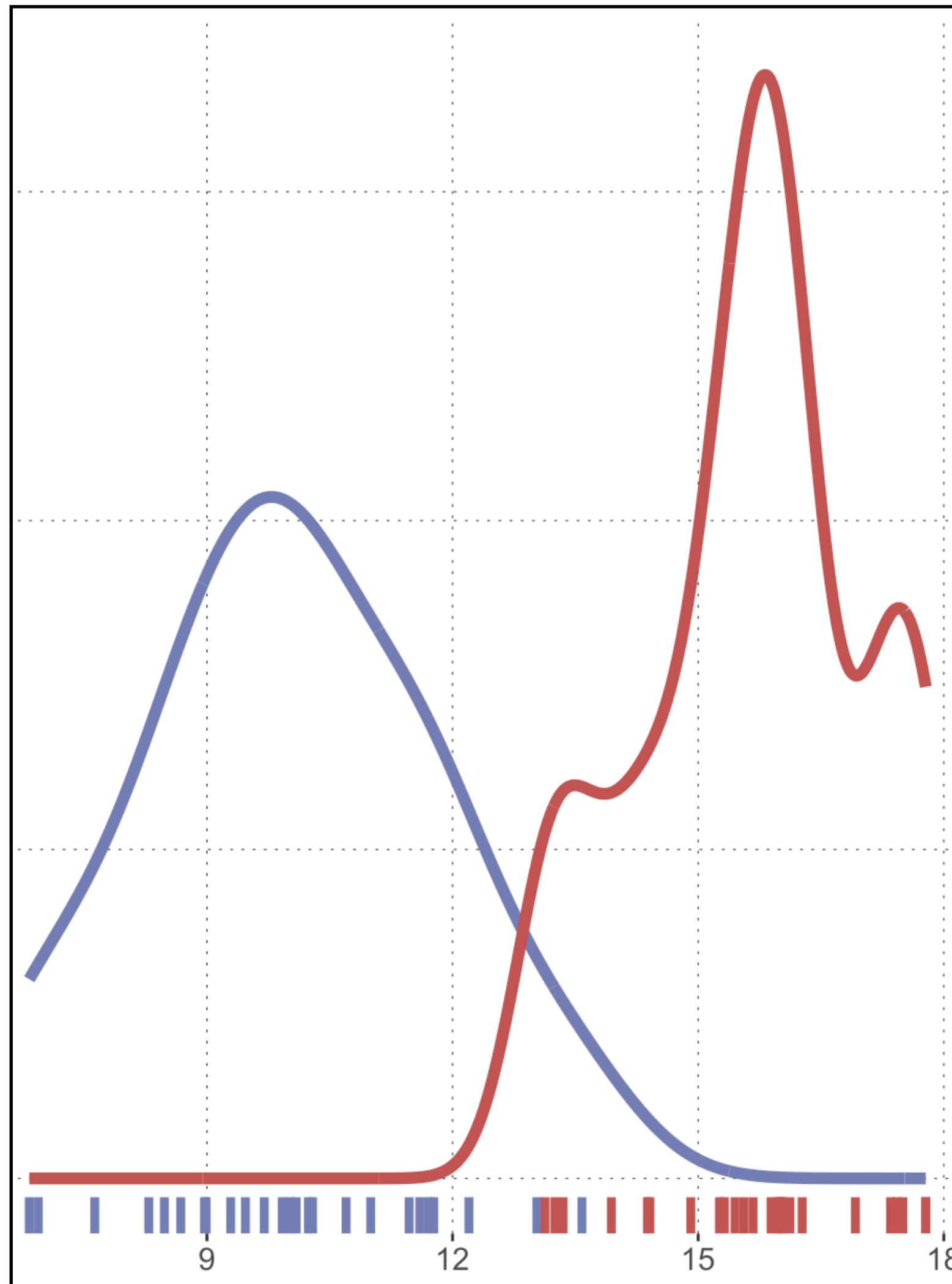




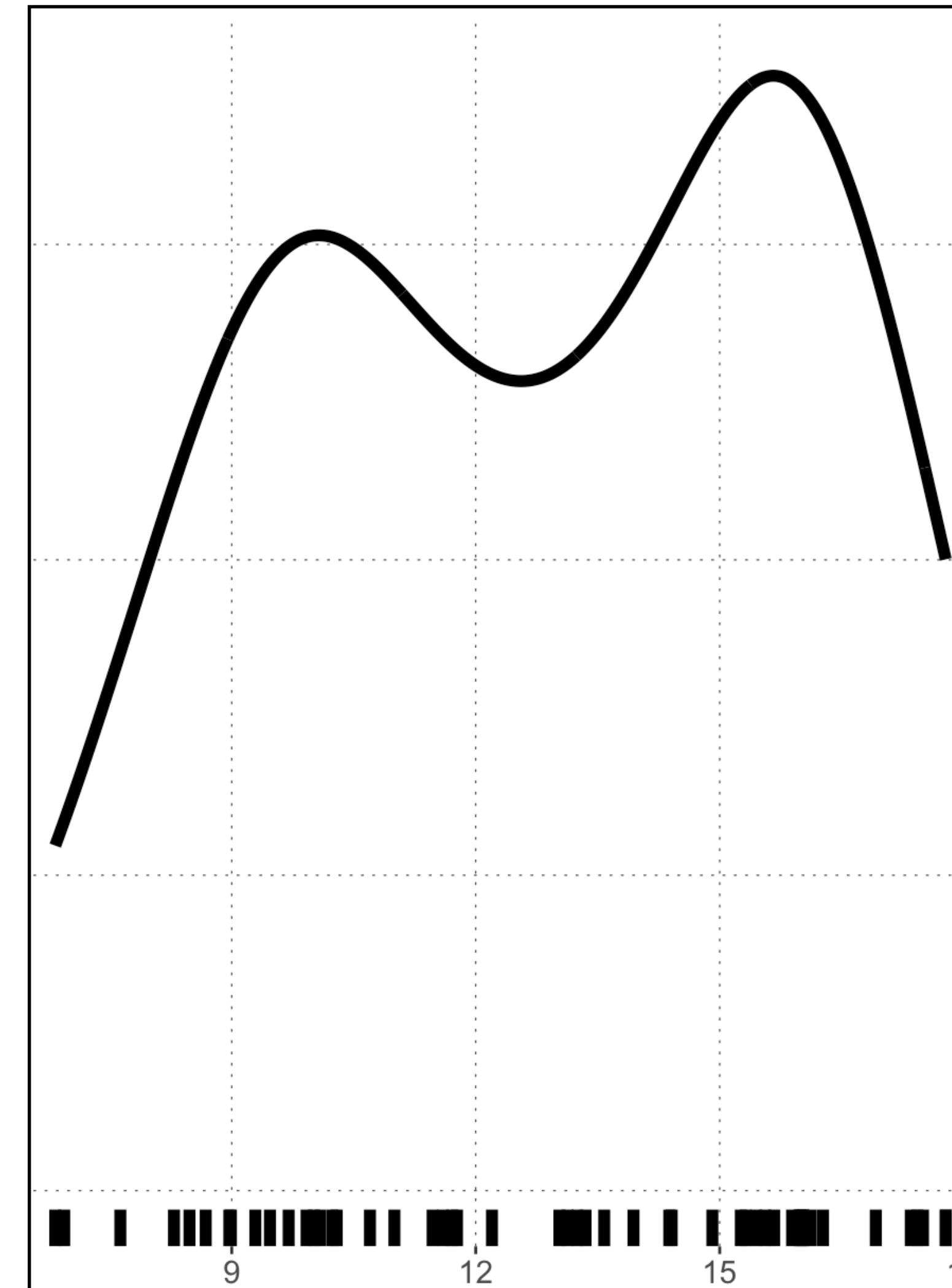
finite mixture models

Multi-modal response distributions

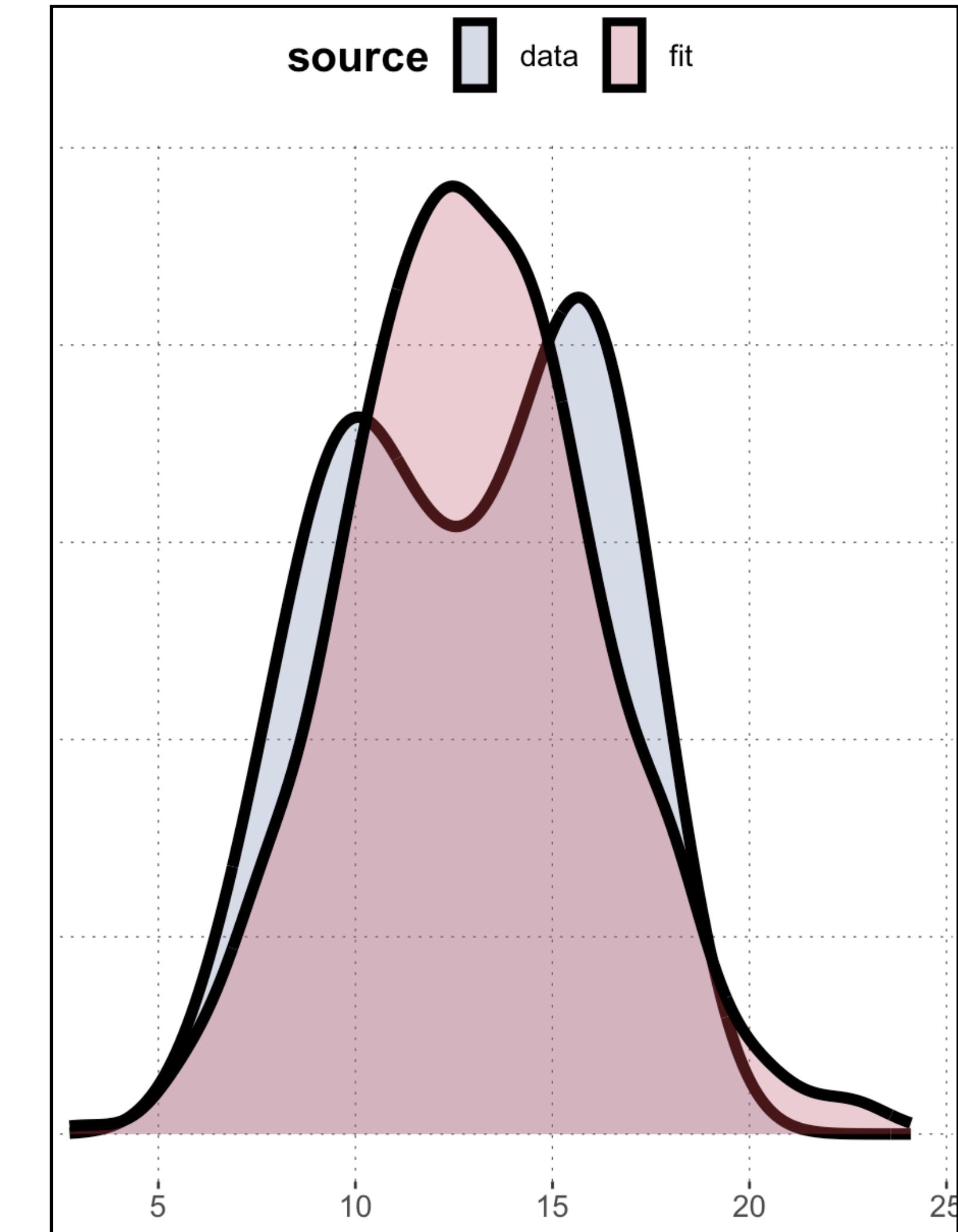
Two-component data



Treated as one for analysis



Posterior predictive ... failure



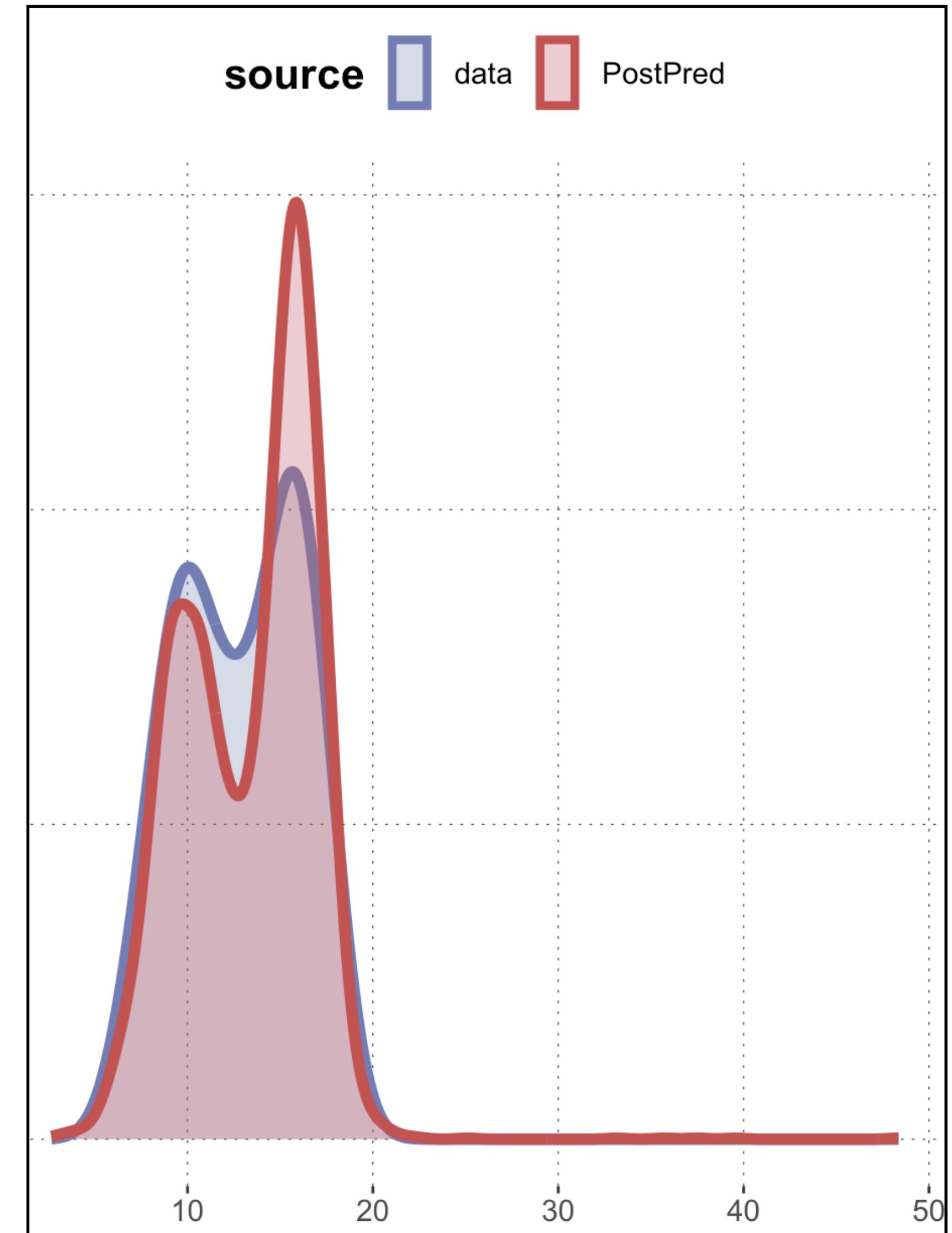
Mixture models

mixing multiple components in the LH function

Let $\langle f_1, \dots, f_k \rangle$ be k likelihood functions for data Y . The k -mixture model for Y explains the data as a weighted combination, with mixture weights α (a probability vector). The mixture likelihood function is:

$$f^{\text{MM}}(y_i) = \alpha_{k(i)} f_{k(i)}$$

where $k(i)$ is the mixture component associated with observation i .



Gaussian mixture models

in BRMS

```
brms_fit_2e_GMM <- brm(  
  # intercept only model  
  formula = y ~ 1,  
  data = data_GMM,  
  # declare that the likelihood should be a mixture  
  family = mixture(gaussian, gaussian),  
  # use weakly informative priors on mu  
  prior = c(  
    prior(normal(12, 10), Intercept, dpar = mu1),  
    prior(normal(12, 10), Intercept, dpar = mu2)  
)  
)
```

special syntax for mixture LH

one intercept for each component

Gaussian mixture models

in BRMS

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI
mu1_Intercept	10.43	1.25	8.84	12.74
mu2_Intercept	15.56	0.80	13.28	16.66

means estimated for each component

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI
sigma1	2.23	0.76	1.16	3.66
sigma2	1.59	0.85	0.66	3.31

SDs estimated for each component

	Estimate	Est.Error	l-95% CI	u-95% CI
theta1	0.54	0.16	0.18	0.88
theta2	0.46	0.16	0.12	0.82

estimated weights of each component

Zero-inflation models

zeros can be generated by two independent paths

If f is a likelihood function for data y , the **zero-inflated likelihood function** is:

$$f^{0\text{inf}}(y; \theta, z) = \begin{cases} z + (1 - z) f(y; \theta) & \text{if } y = 0 \\ (1 - z) f(y; \theta) & \text{otherwise} \end{cases}$$

both mixture components contribute
to likelihood of “zero”

Zero-hurdle models

zeros are generated by one independent path

If f is a likelihood function for data y , the **zero-hurdle likelihood function** is:

$$f^{0\text{hur}}(y; \theta, z) = \begin{cases} z & \text{if } y = 0 \\ (1 - z) \frac{f(y; \theta)}{1 - f(0; \theta)} & \text{otherwise} \end{cases}$$

only one mixture component contributes to likelihood of “zero”; the other is truncated

Zero/one-inflation models

zeros and can be generated by independent paths

If f is a likelihood function for data y , the **zero/one-inflated likelihood function** is:

$$f^{0/1\text{inf}}(y ; \theta, \alpha, \beta) = \begin{cases} \alpha & \text{if } y = 1 \\ \alpha (1 - \gamma) & \text{if } y = 0 \\ (1 - \alpha) f(y ; \theta) & \text{otherwise} \end{cases}$$

Hurdle and inflation models in BRMS

as of March 2023

```
hurdle_poisson(link = "log")

hurdle_negbinomial(link = "log", link_shape = "log", link_hu = "logit")

hurdle_gamma(link = "log", link_shape = "log", link_hu = "logit")

hurdle_lognormal(link = "identity", link_sigma = "log", link_hu = "logit")

zero_inflated_beta(link = "logit", link_phi = "log", link_zi = "logit")

zero_one_inflated_beta(
  link = "logit",
  link_phi = "log",
  link_zoi = "logit",
  link_coi = "logit"
)

zero_inflated_poisson(link = "log", link_zi = "logit")

zero_inflated_negbinomial(link = "log", link_shape = "log", link_zi = "logit")

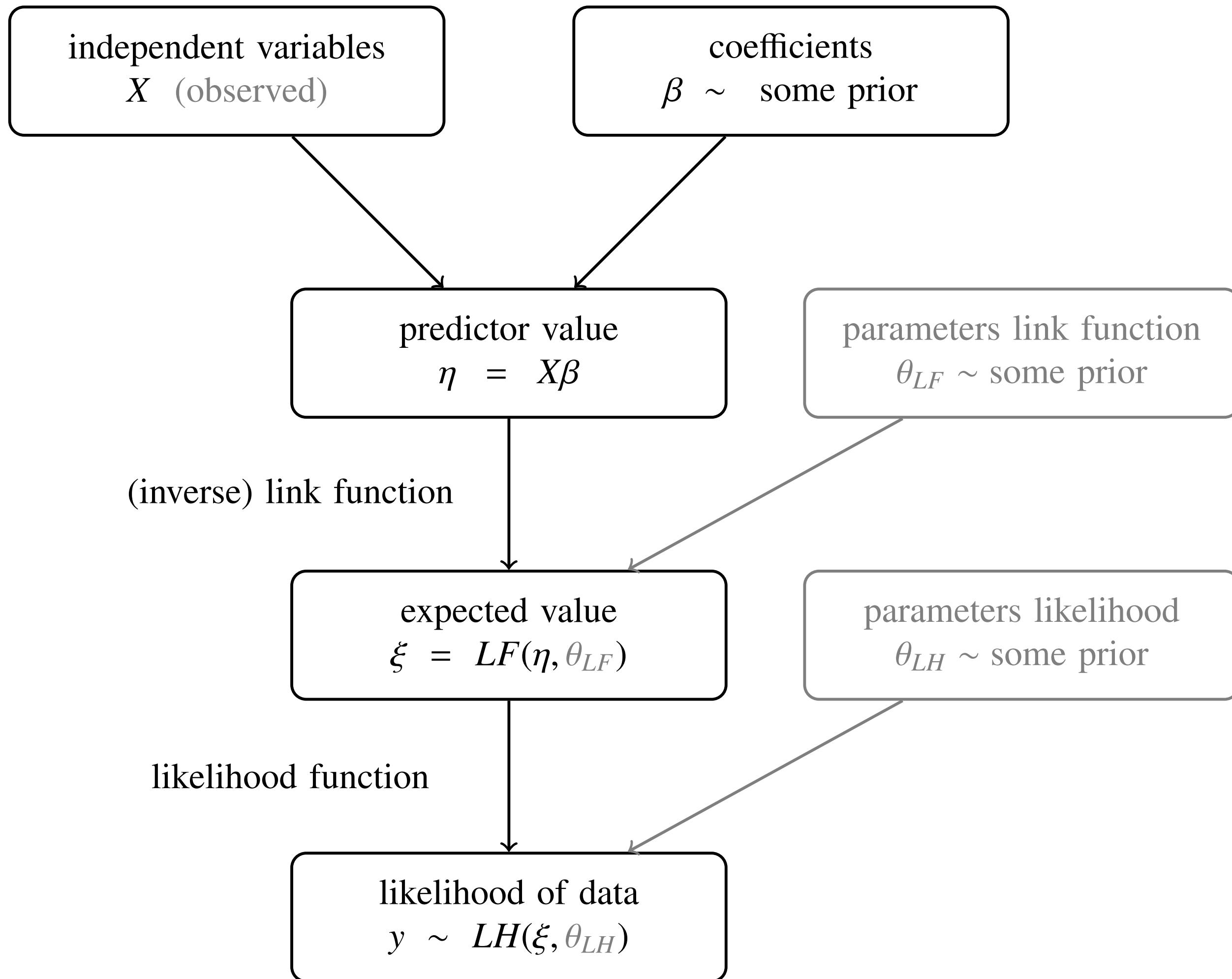
zero_inflated_binomial(link = "logit", link_zi = "logit")
```



distributional models

Distributional models

linear predictors for parameters of the link- and likelihood functions



Normal GLM:

$$\eta = X\beta$$

$$\theta_{LF}, \theta_{LH} \sim \text{some prior}$$

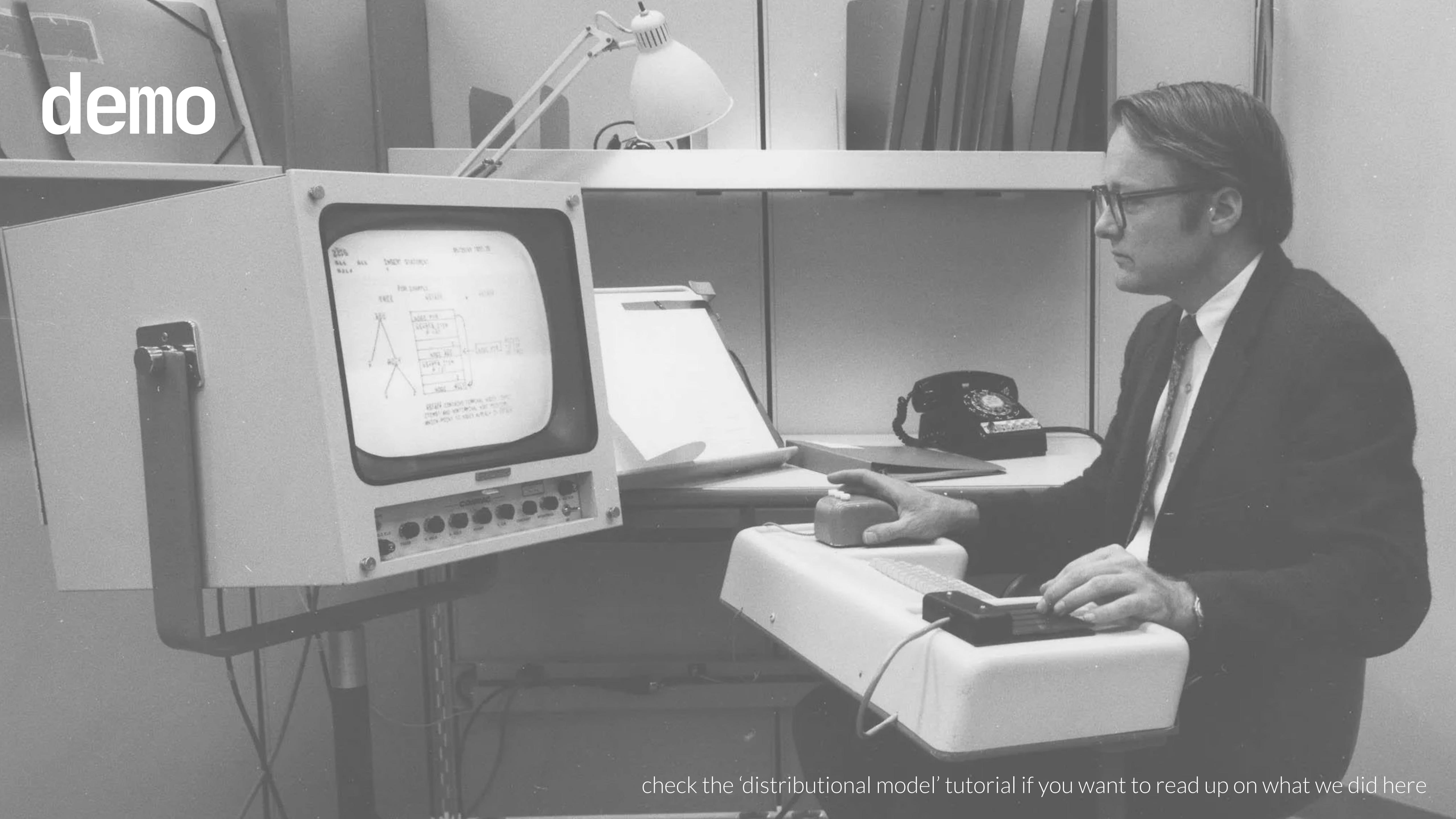
Distributional GLM:

$$\eta = X\beta$$

$$\theta_{LF} = F(X'\beta')$$

$$\theta_{LH} = F(X''\beta'')$$

demo



check the 'distributional model' tutorial if you want to read up on what we did here