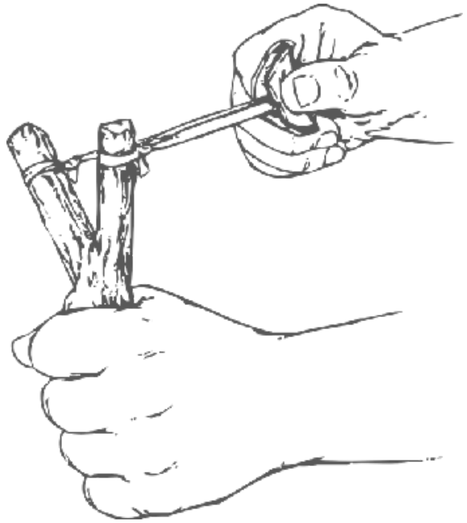
Bayesian data analysis: Theory & practice

Part 3a: Categorical predictors & generalized linear models

Michael Franke

Main learning goals
for this part

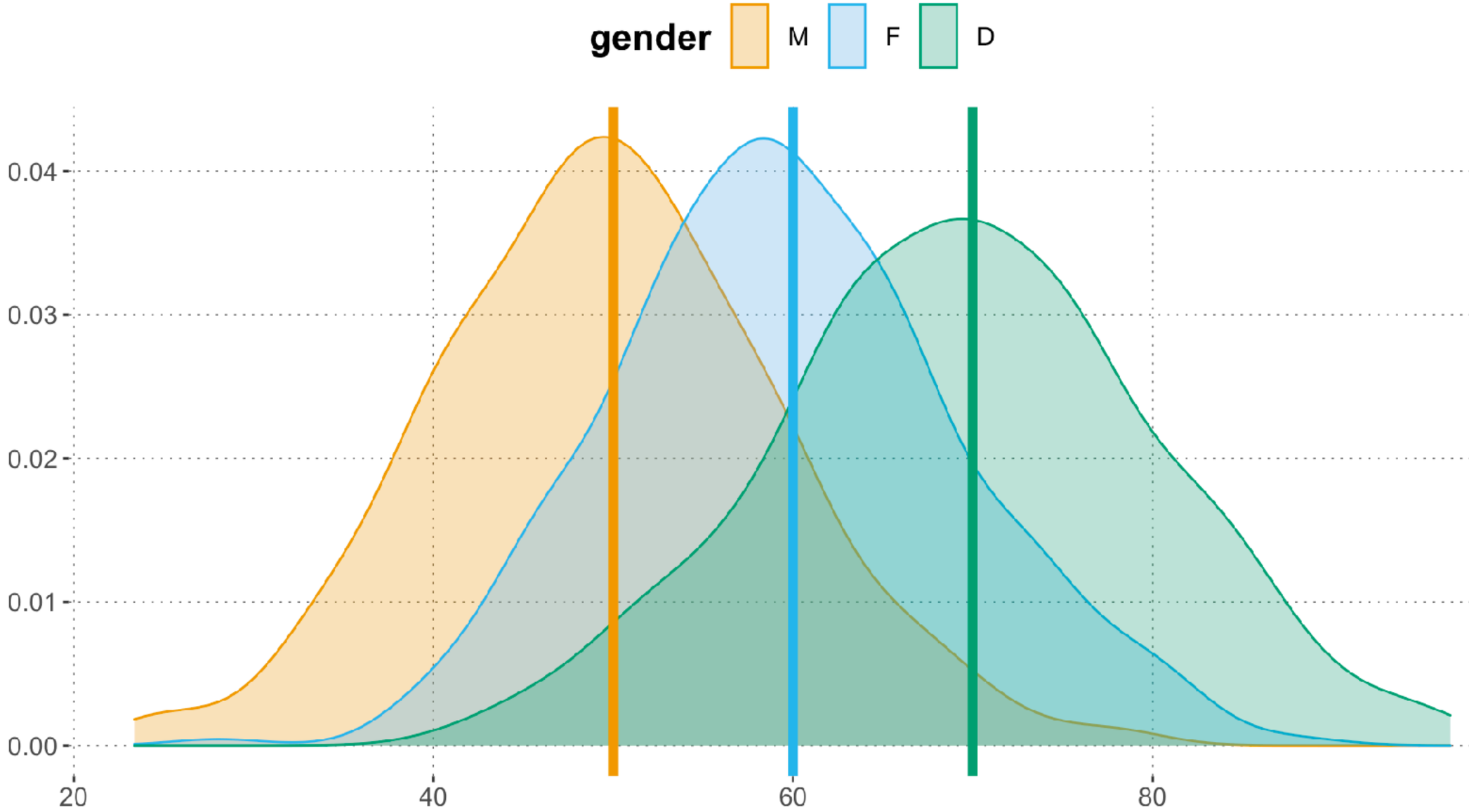


1. multiple regression w/ categorical predictors
    a. contrast coding
    b. derived variables in Bayesian analysis
2. generalized linear models
    a. logistic
    b. multinomial
    c. ordinal
3. excursion: “beyond GLMs”
    a. mixture models
    b. distributional models

contrast coding

Some fake data

three-way categorical variable



Finding numbers for categories

metric predictors

y	x_0	x_1	x_2
42	1	4	163
19	1	7	128
38	1	2	99
⋮	⋮	⋮	⋮

categorical predictor

y	gender
51	M
59	F
73	D
⋮	⋮

Treatment coding

comparing against a reference category

y	gender	x_0	x_1	x_2
51	M	1	0	0
59	F	1	1	0
73	D	1	0	1
⋮	⋮	⋮	⋮	⋮

$$y_i = \sum_{j=0}^{k} \beta_j x_{ij} + \epsilon_i$$

$$\hat{\mu}_M = \beta_0 1 + \beta_1 0 + \beta_2 0$$
$$\hat{\mu}_F = \beta_0 1 + \beta_1 1 + \beta_2 0$$
$$\hat{\mu}_D = \beta_0 1 + \beta_1 0 + \beta_2 1$$

hypotheses

$$\beta_0 = \hat{\mu}_M$$
$$\beta_1 = \hat{\mu}_F - \hat{\mu}_M$$
$$\beta_2 = \hat{\mu}_D - \hat{\mu}_M$$

Treatment coding

comparing against a reference category

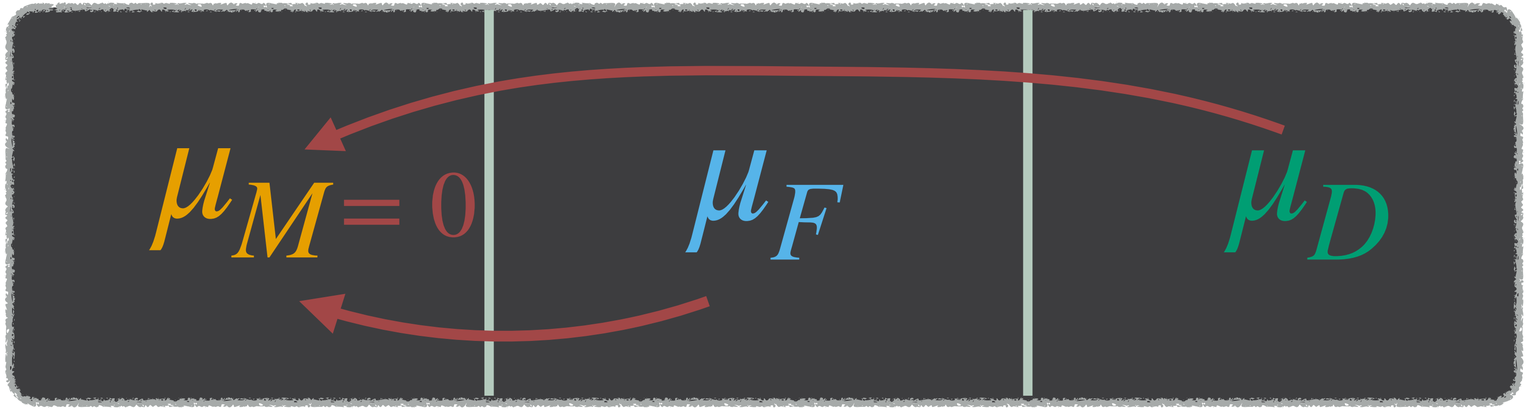
y	gender	x_0	x_1	x_2
51	M	1	0	0
59	F	1	1	0
73	D	1	0	1
⋮	⋮	⋮	⋮	⋮

hypotheses

$$\beta_0 = \hat{\mu}_M$$

$$\beta_1 = \hat{\mu}_F - \hat{\mu}_M$$

$$\beta_2 = \hat{\mu}_D - \hat{\mu}_M$$



Cell means coding

estimating a mean for each cell

y	gender	x_0	x_1	x_2
51	M	1	0	0
59	F	0	1	0
73	D	0	0	1
⋮	⋮	⋮	⋮	⋮

hypotheses

$$\beta_0 = \hat{\mu}_M$$
$$\beta_1 = \hat{\mu}_F$$
$$\beta_2 = \hat{\mu}_D$$

$\mu_M = 0$ | $\mu_F = 0$ | $\mu_D = 0$

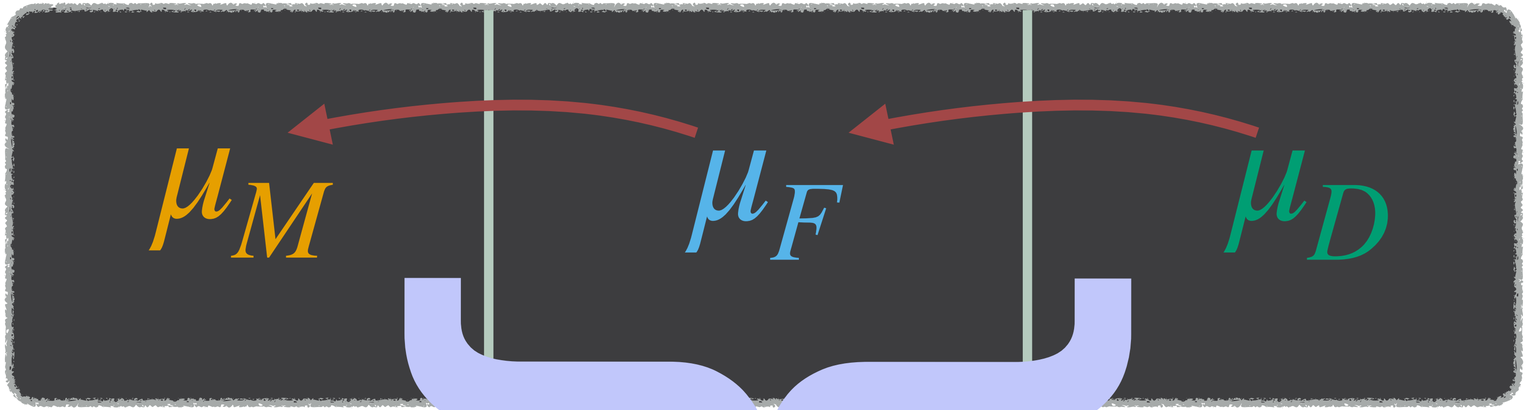
Simple difference coding

estimating a mean for each cell

y	gender	x_0	x_1	x_2
51	M	1	$-2/3$	$-1/3$
59	F	1	$1/3$	$-1/3$
73	D	1	$1/3$	$2/3$
⋮	⋮	⋮	⋮	⋮

hypotheses

$$\beta_0 = \bar{\mu}$$
$$\beta_1 = \hat{\mu}_F - \hat{\mu}_M$$
$$\beta_2 = \hat{\mu}_D - \hat{\mu}_F$$



$$\bar{\mu} = 0$$

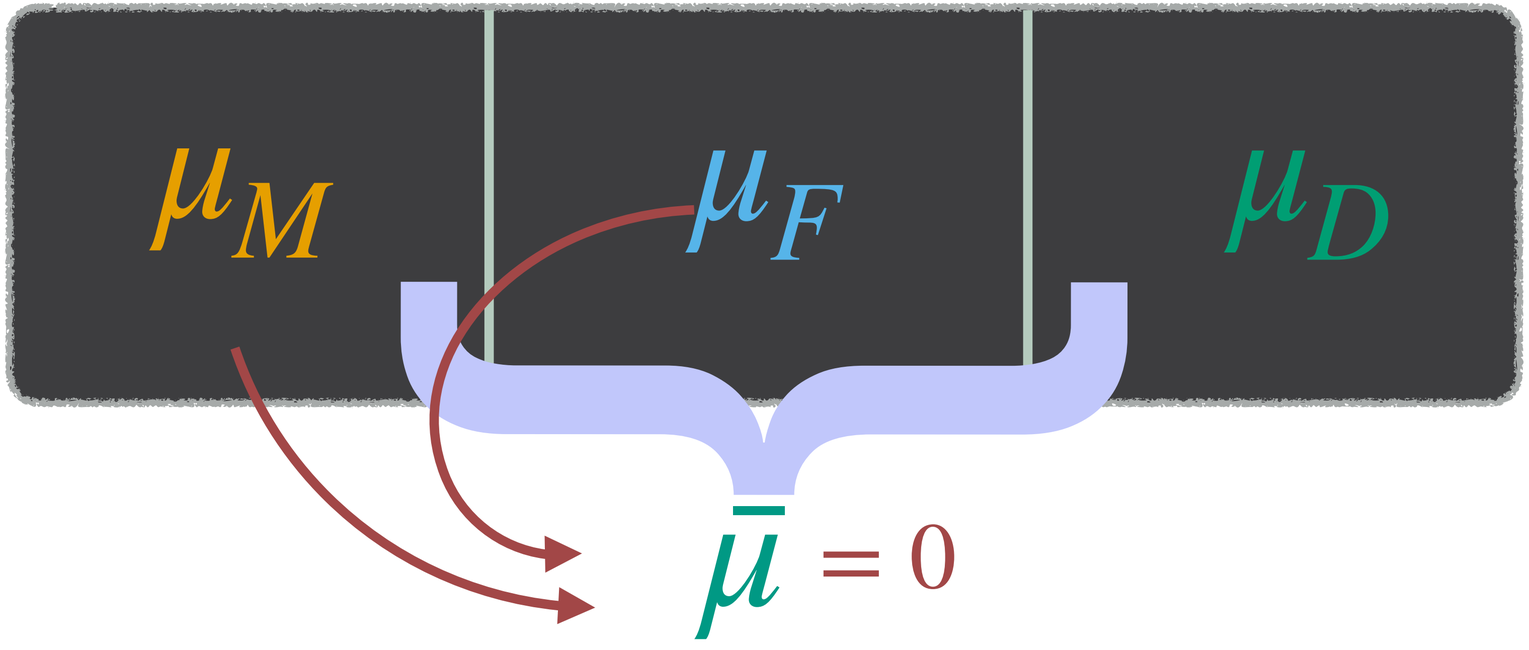
Sum coding

comparing $k - 1$ cells to the grand mean

y	gender	x_0	x_1	x_2
51	M	1	1	0
59	F	1	0	1
73	D	1	-1	-1
⋮	⋮	⋮	⋮	⋮

hypotheses

$$\beta_0 = \bar{\mu}$$
$$\beta_1 = \hat{\mu}_M - \bar{\mu}$$
$$\beta_2 = \hat{\mu}_F - \bar{\mu}$$

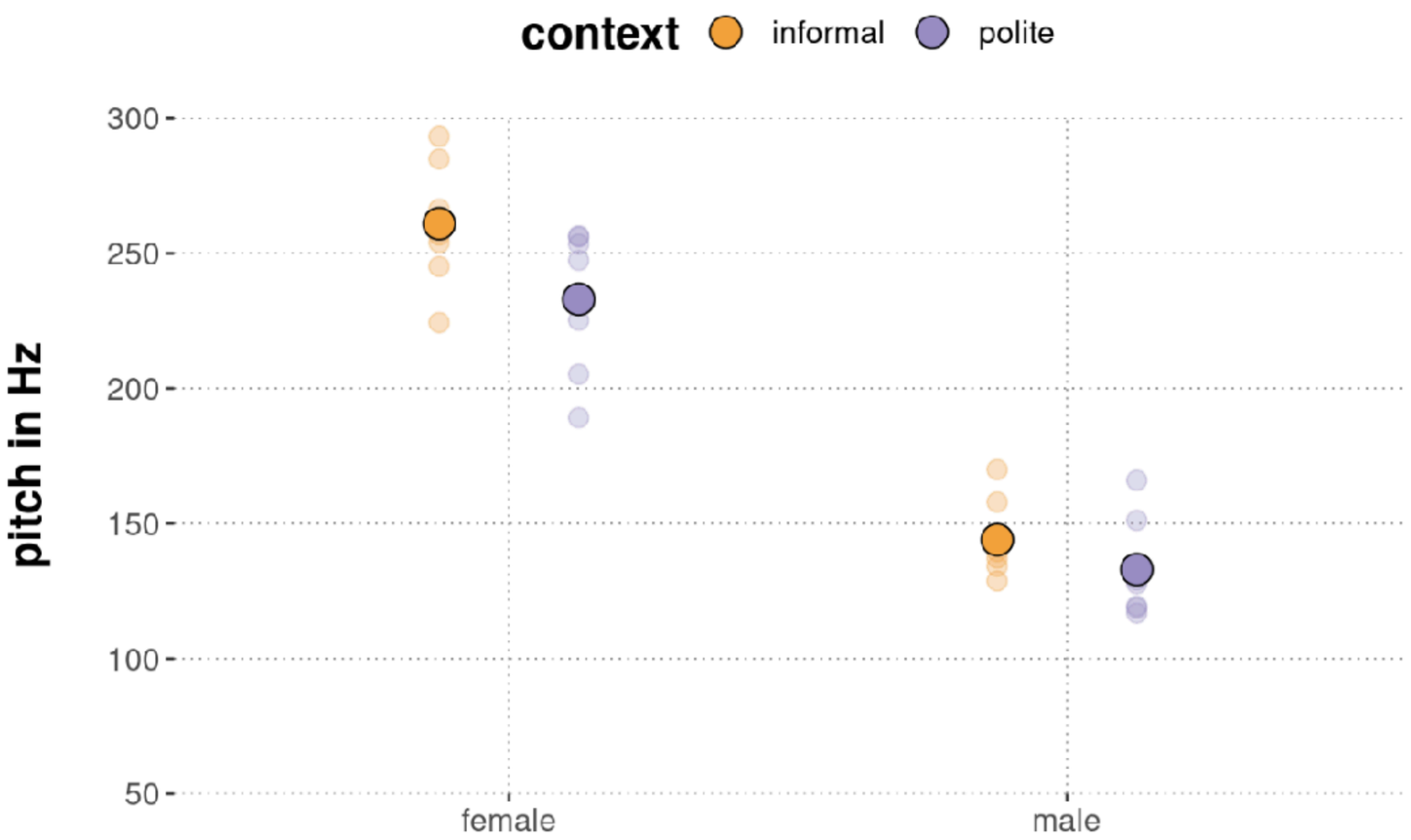


Case study: pitch in context

data from Winter & Grawunder (2012)

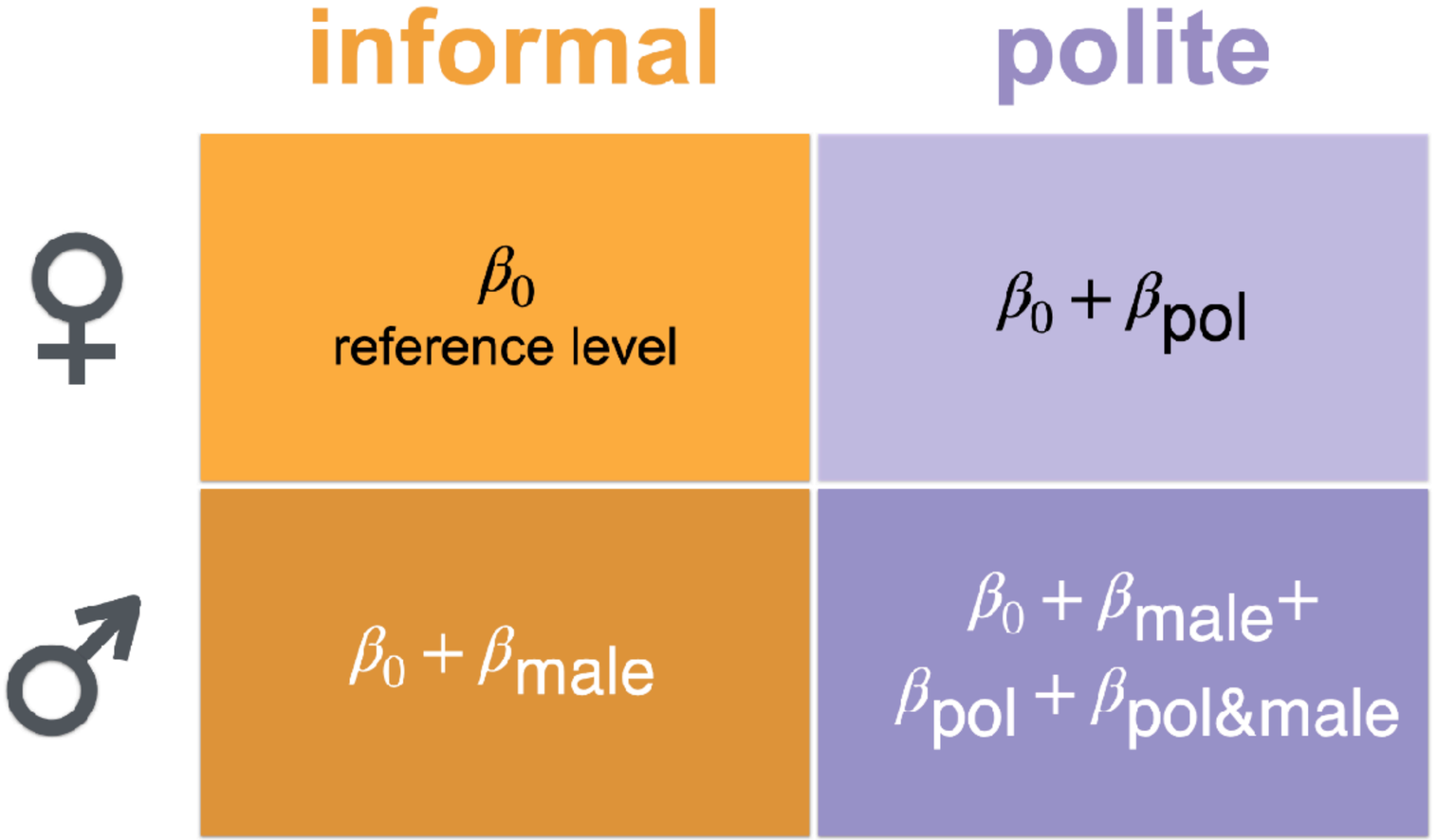
```
politeness_data <- aida::data_polite
politeness_data %>% head(5)
```

```
## # A tibble: 5 × 5
##   subject gender sentence context pitch
##   <chr>   <chr>  <chr>    <chr>   <dbl>
## 1 F1      F      S1       pol      213.
## 2 F1      F      S1       inf      204.
## 3 F1      F      S2       pol      285.
## 4 F1      F      S2       inf      260.
## 5 F1      F      S3       pol      204.
```



read more here

Dummy coding for 2x2 design

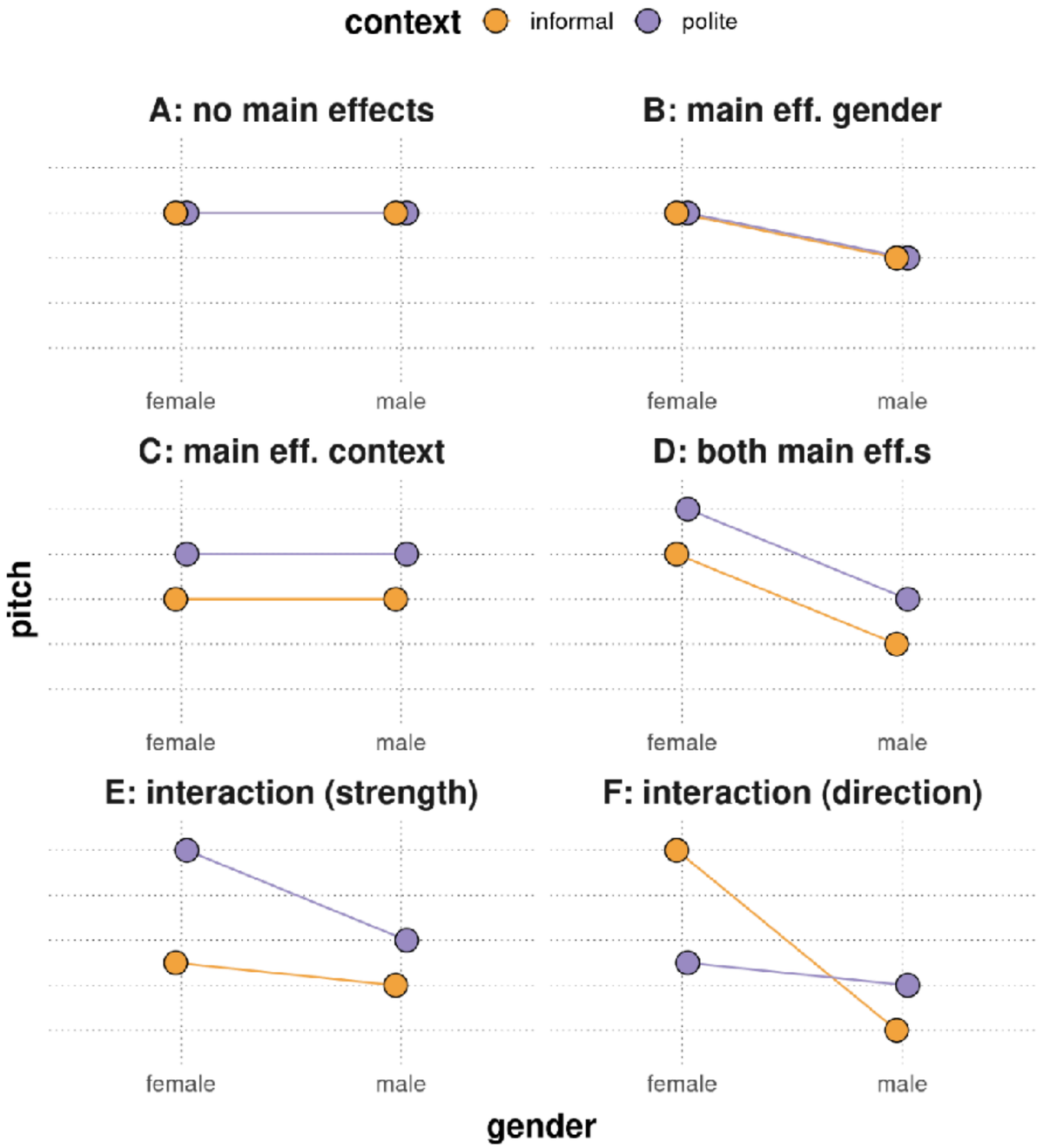


Interaction term $\beta_{pol\&male}$ is a 'difference of differences'

read more here

Main effects & interactions

in 2x2 designs

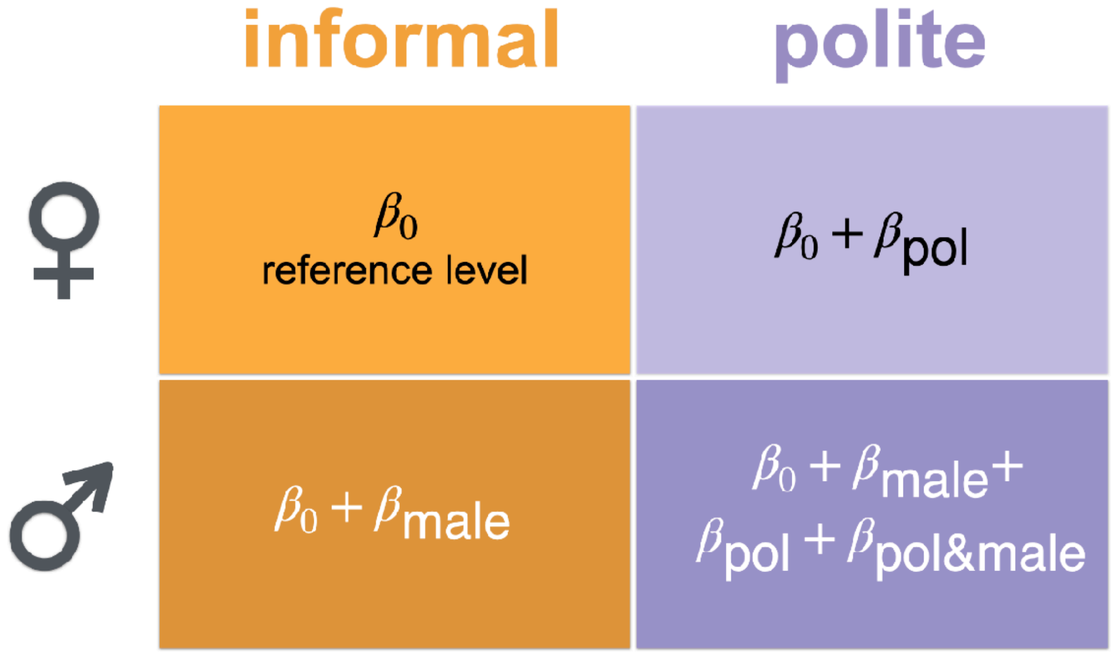


Interaction term $\beta_{\text{pol\&male}}$ is a 'difference of differences'

read more here

Bayesian regression

```
# here, we only use fixed effects
fit_dummy_FE <- brm(
  pitch ~ gender * context,
  data = politeness_df,
  cores = 4,
  iter = 1000
)
```




```
Population-Level Effects:
                  Estimate Est.Error l-95% CI u-95% CI
Intercept           260.56      7.87   244.15   275.21
genderM            -116.16     11.01  -137.31   -94.05
contextpol          -27.23     11.10   -48.38    -5.23
genderM:contextpol   15.77     16.05   -16.54    46.24
```

Which questions about cell mean differences can we address with this information directly?

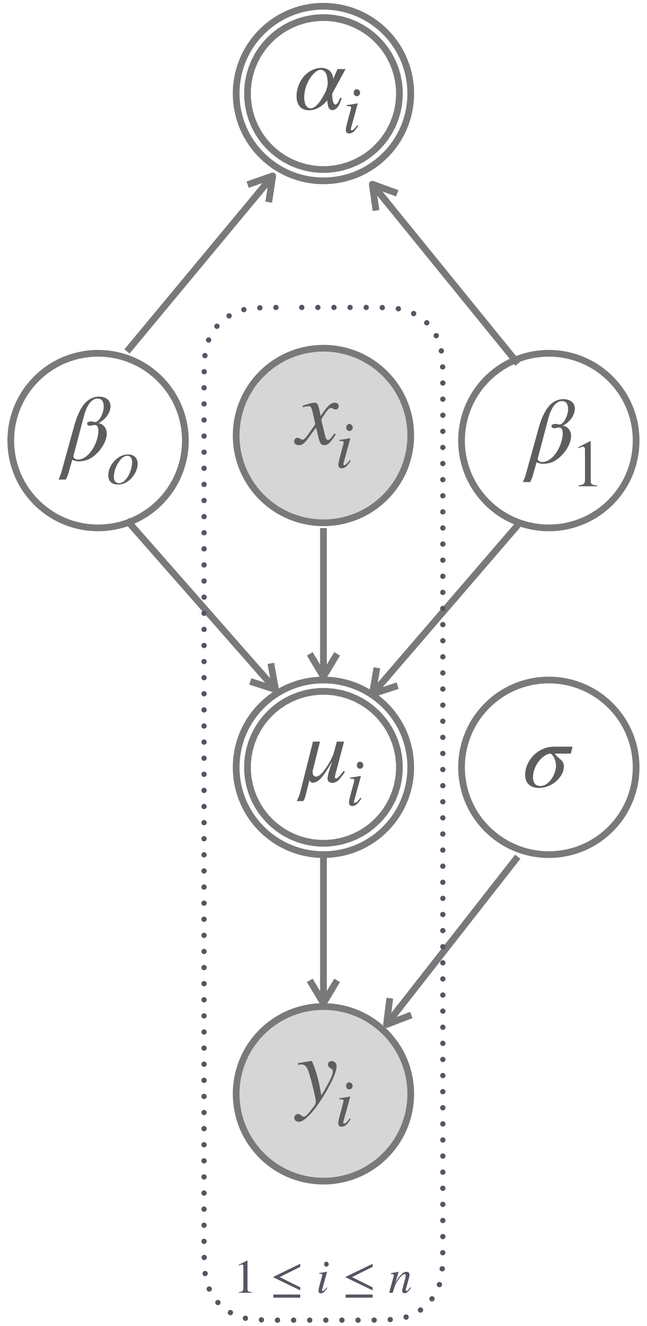
read more here

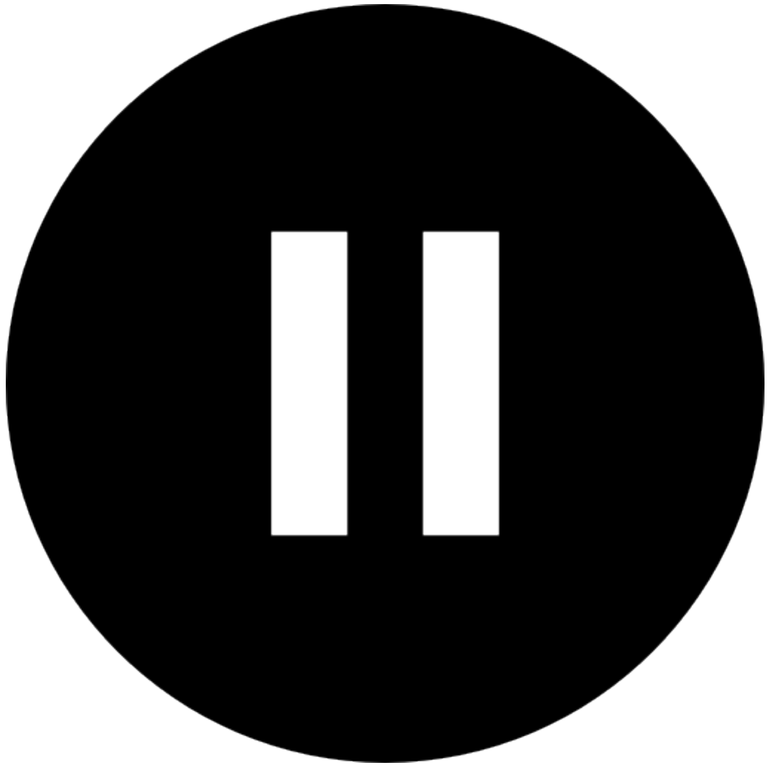
demo

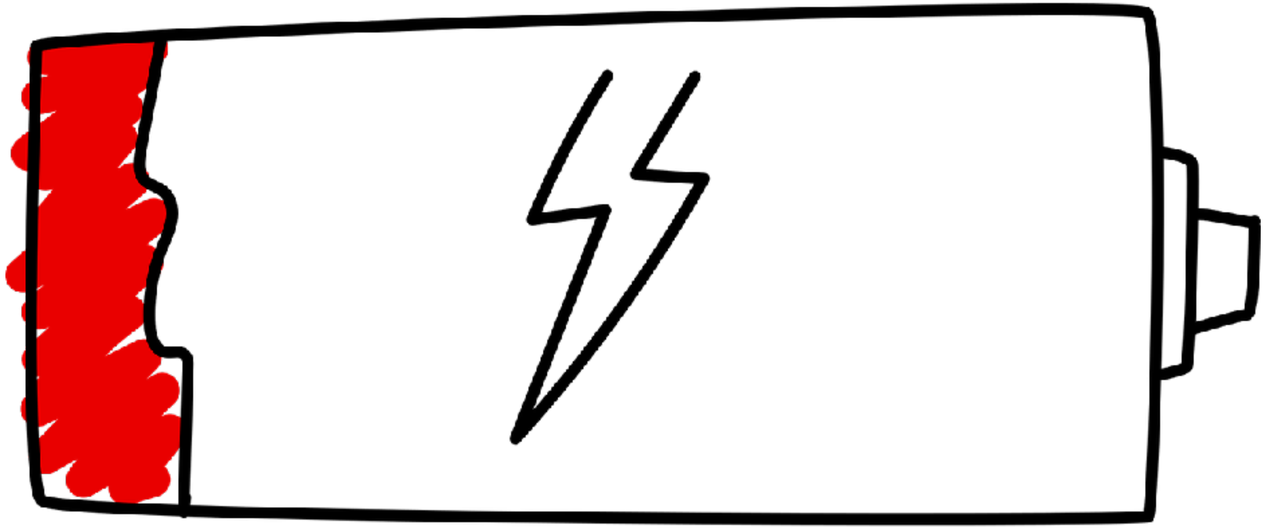
hands-on practice with multiple categorical predictors

Derived variables

- obtain samples from model parameters
- apply (deterministic) function to each sample
  - to derive (deterministically) a new model variable
- violà: samples from the posterior of a new “derived variable”

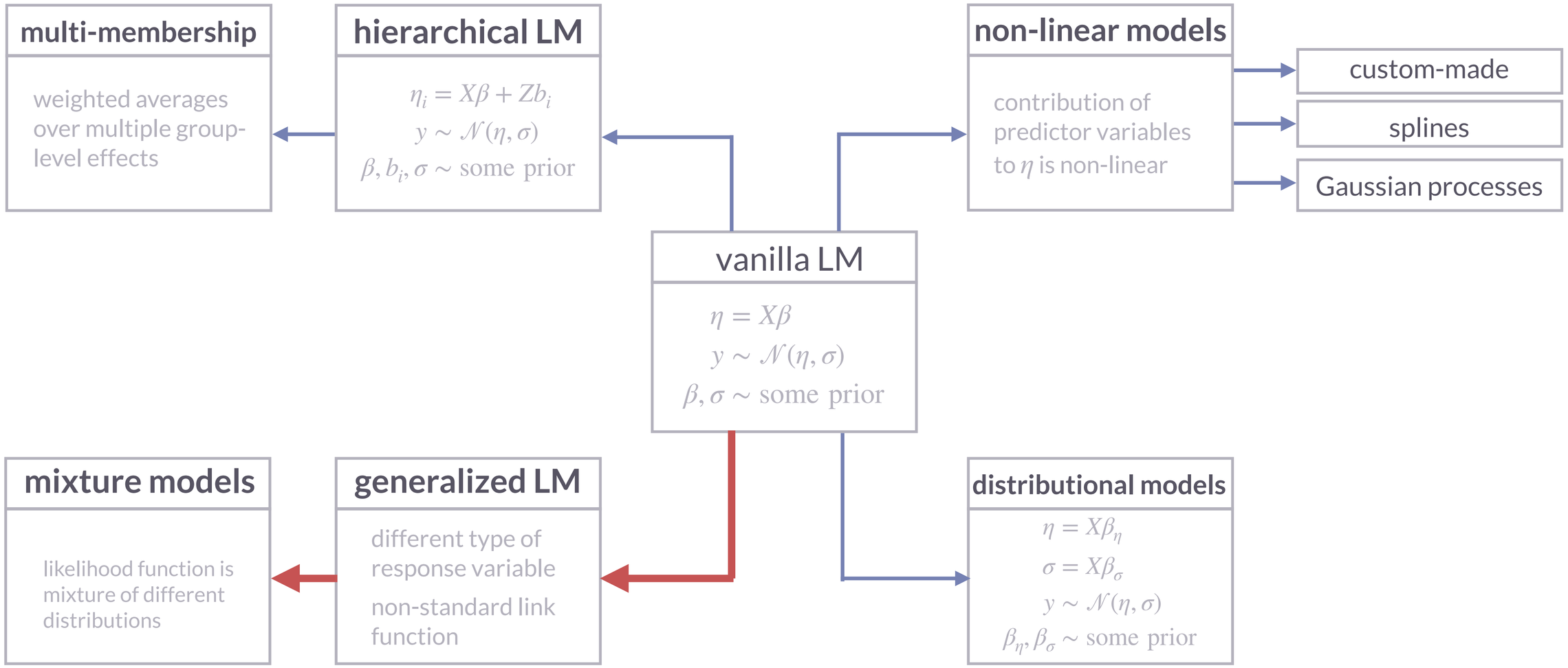






Roadmap “beyond vanilla”

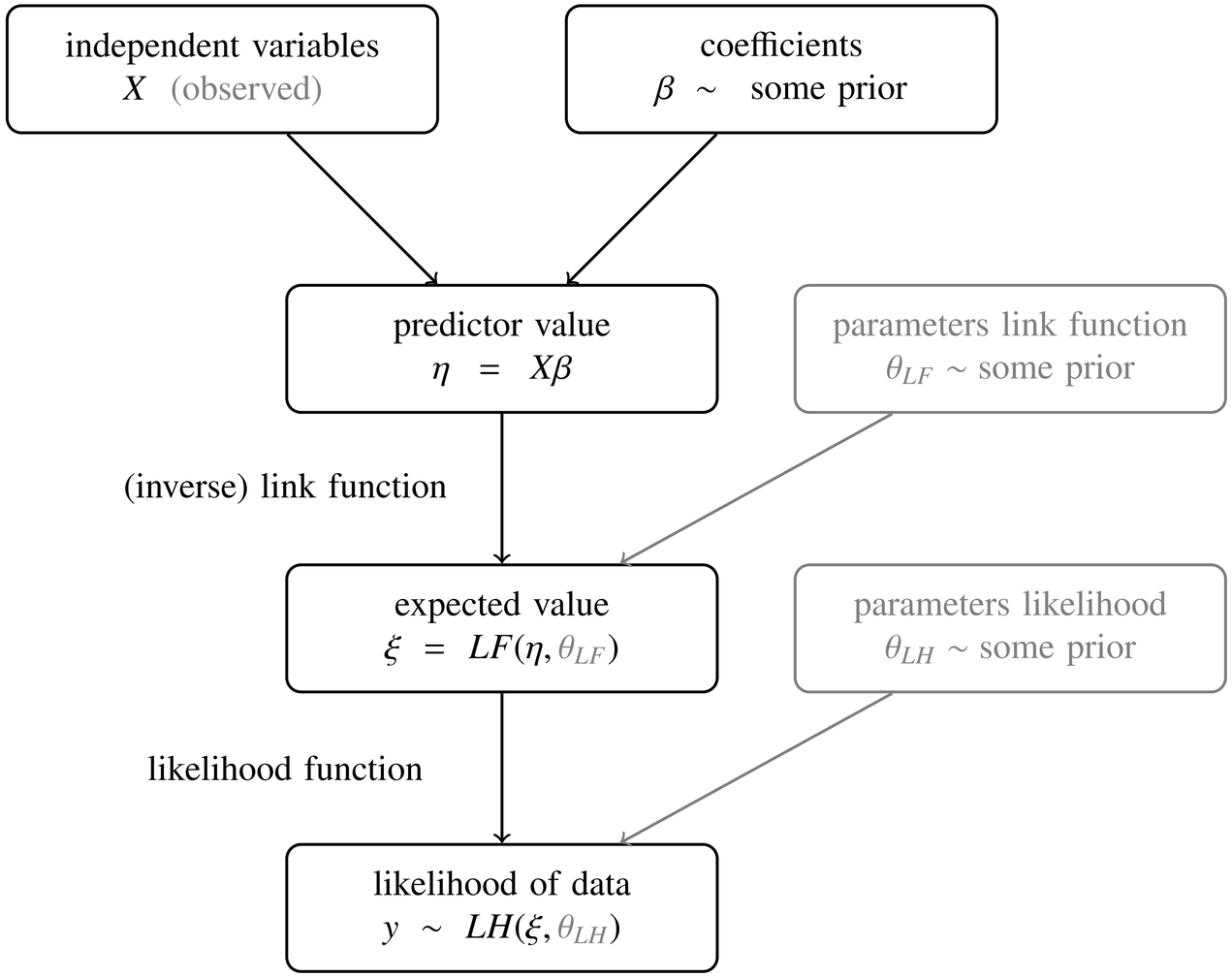
common extensions of linear regression modeling



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]

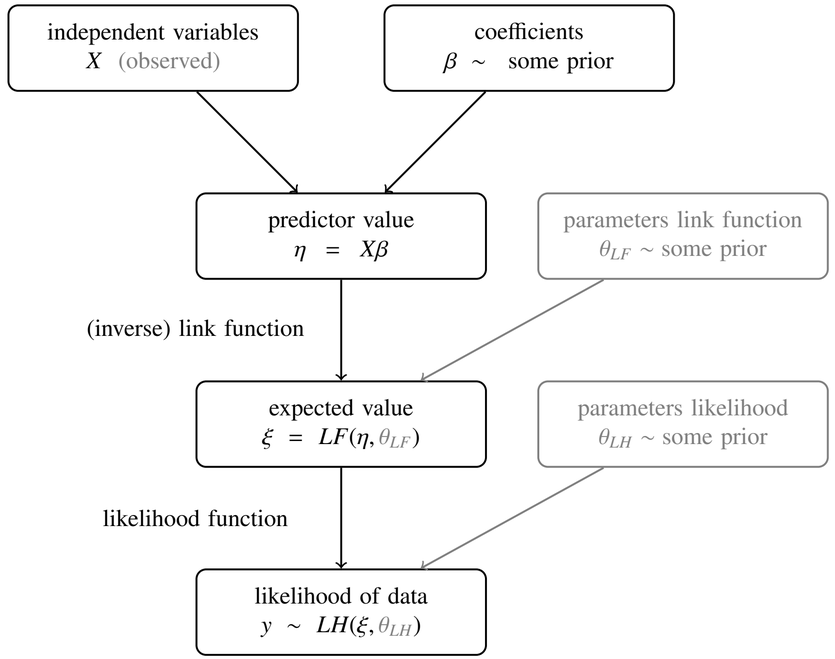
$\xi_i = \exp(\eta_i)$ [predictor of central tendency]

$y_i \sim \text{Poisson}(\xi_i)$ [likelihood]

read more here

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	$\xi = \exp(\eta)$	$y \sim \text{Poisson}(\xi)$

read more here

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_prepped2,
  family = cumulative()
)
```

Logistic regression

Logistic regression

Definition

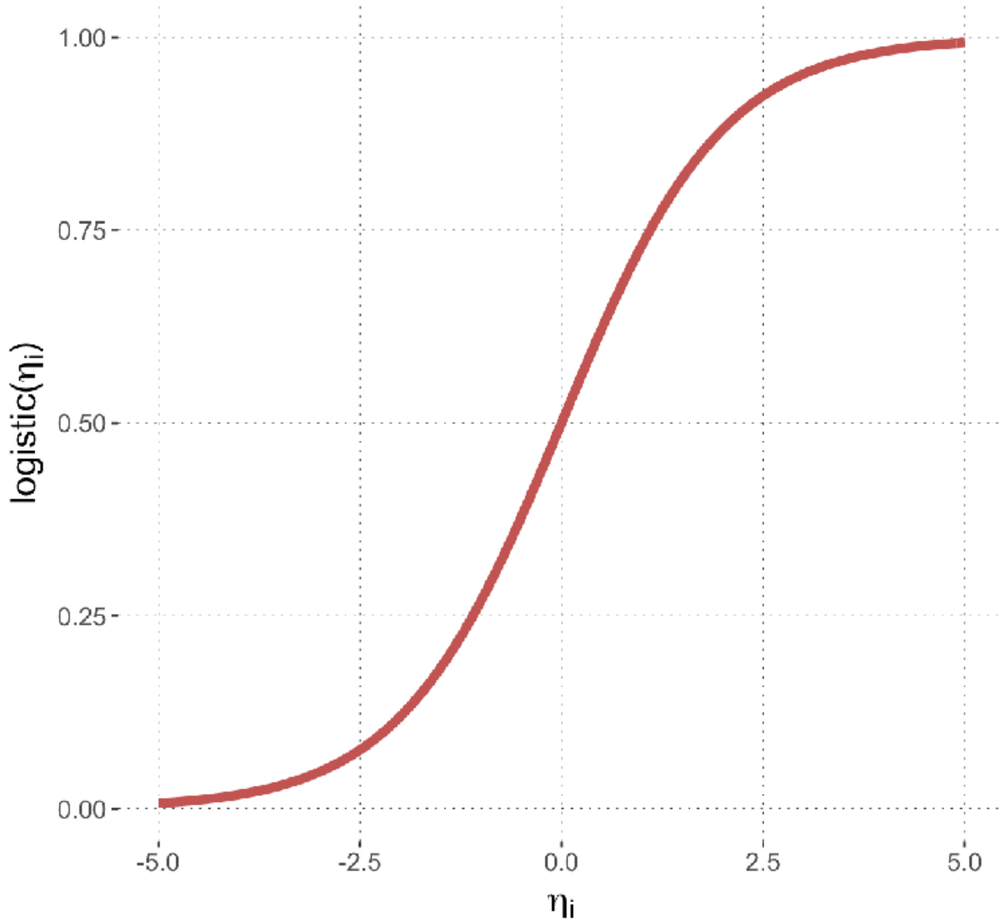
$\eta_i = \mathbf{x}_i \cdot \beta$ [linear predictor]

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

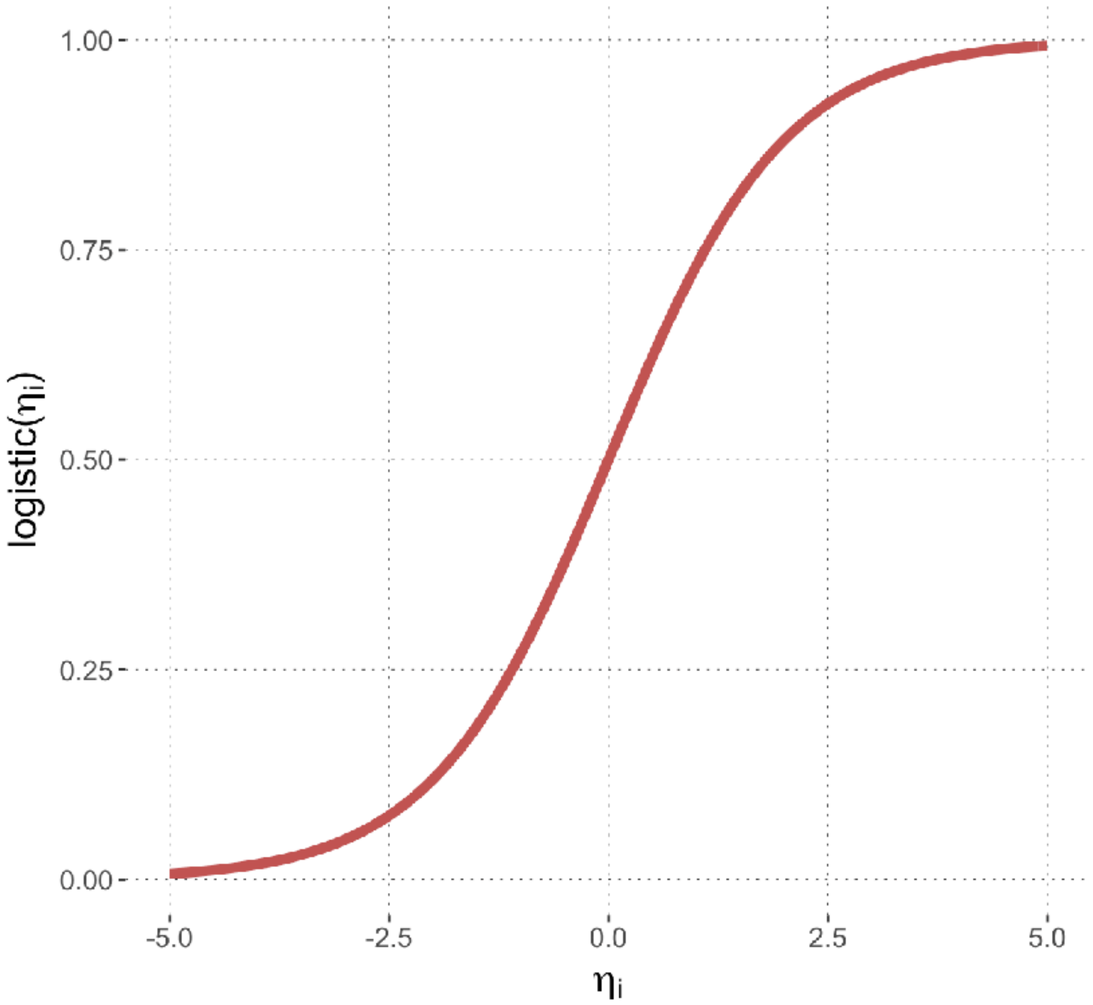


read more here

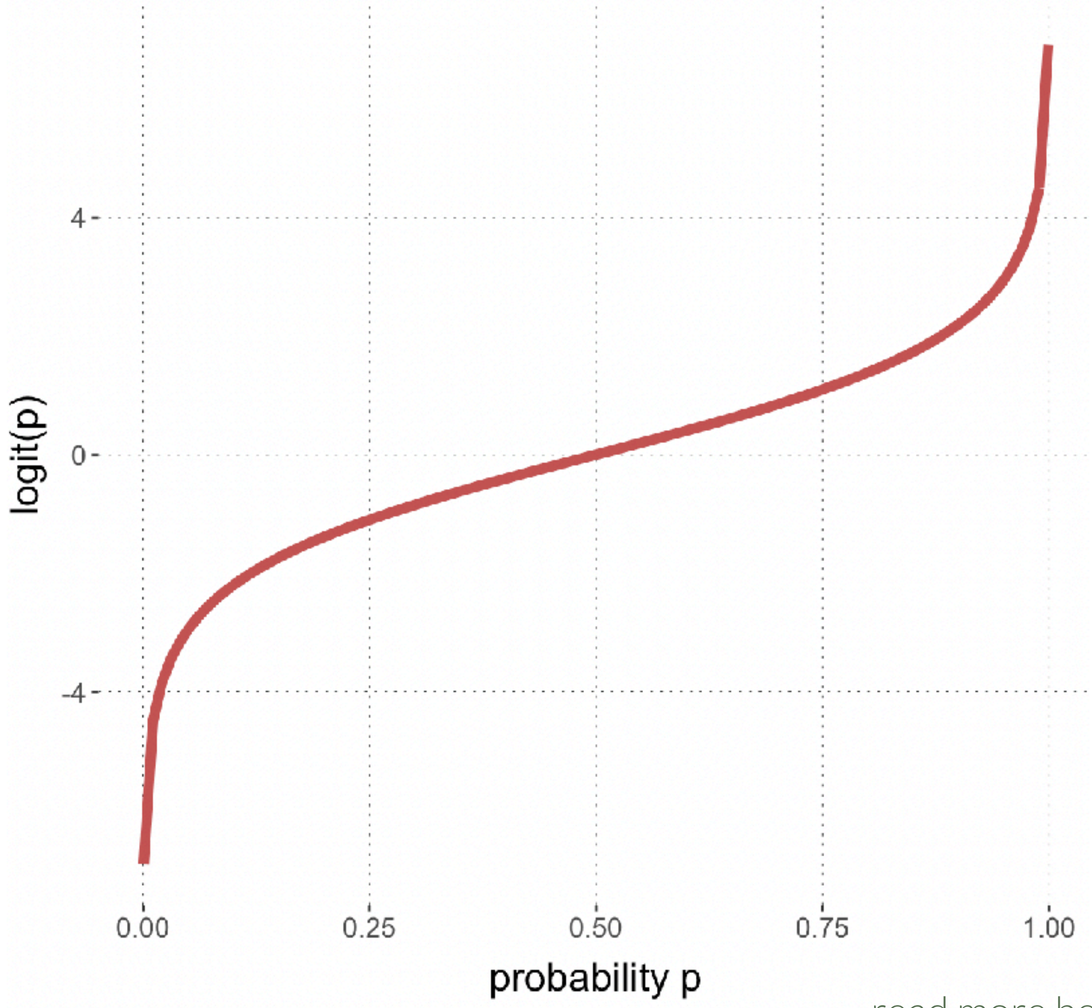
Logistic regression

Link & inverse link function

Link function: logistic



"Inverse" link function: logit



read more here

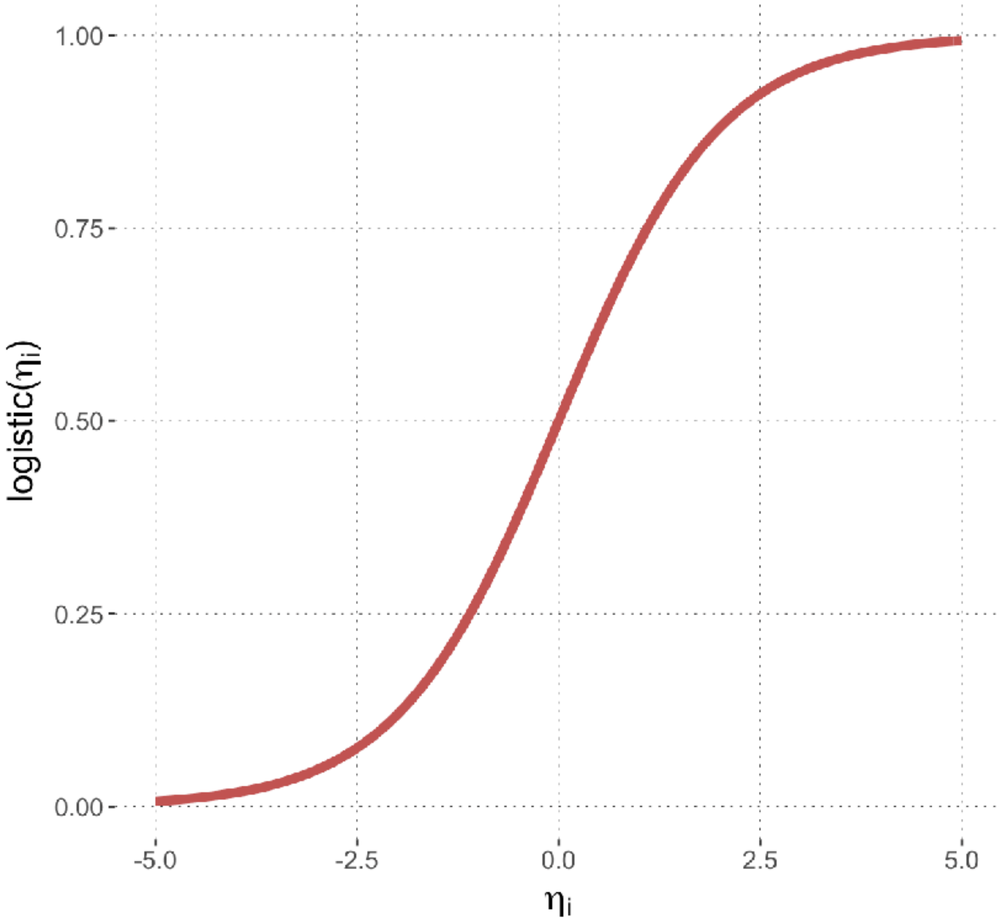
Logistic regression
interpretation

- 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

$$\eta_1 - \eta_2 = \log \frac{\xi_1}{1-\xi_1} - \log \frac{\xi_2}{1-\xi_2} = \log \left(\frac{\xi_1}{1-\xi_1} \frac{1-\xi_2}{\xi_2} \right)$$

$$\Leftrightarrow \frac{\xi_1}{1-\xi_1} = \exp(\eta_1 - \eta_2) \frac{\xi_2}{1-\xi_2}$$

Link function: logistic



read more here

demo

predictive samples from a logistic regression model

Multinomial regression

Multinomial regression

- we want to predict probabilities $\mathbf{p} = \langle p_1, \ldots, p_k \rangle$
  - 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:

$$s_j = \mathbf{x}_i \cdot \beta^j$$

- probabilities from soft-max:

$$p_j = \frac{\exp s_j}{\sum_{j'=1}^{k} \exp s_j'}$$

Interpretation

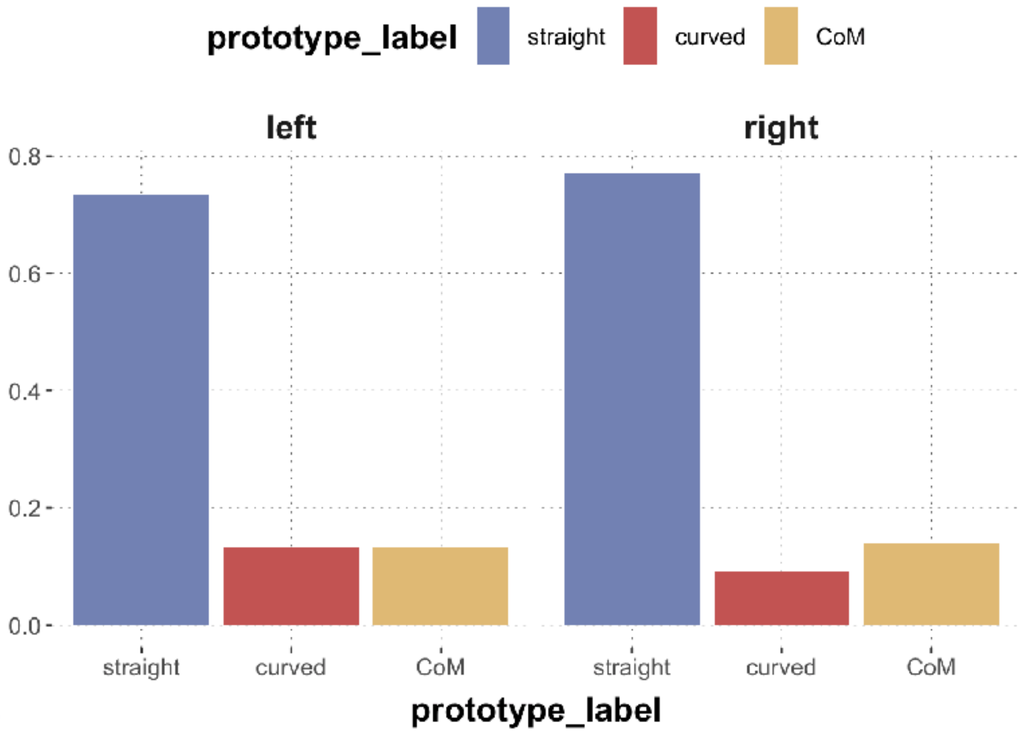
- linear predictor predictor s_j represents $\log(p_j/p_1)$
- Think of this as ...

$k - 1$ **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()
)
```

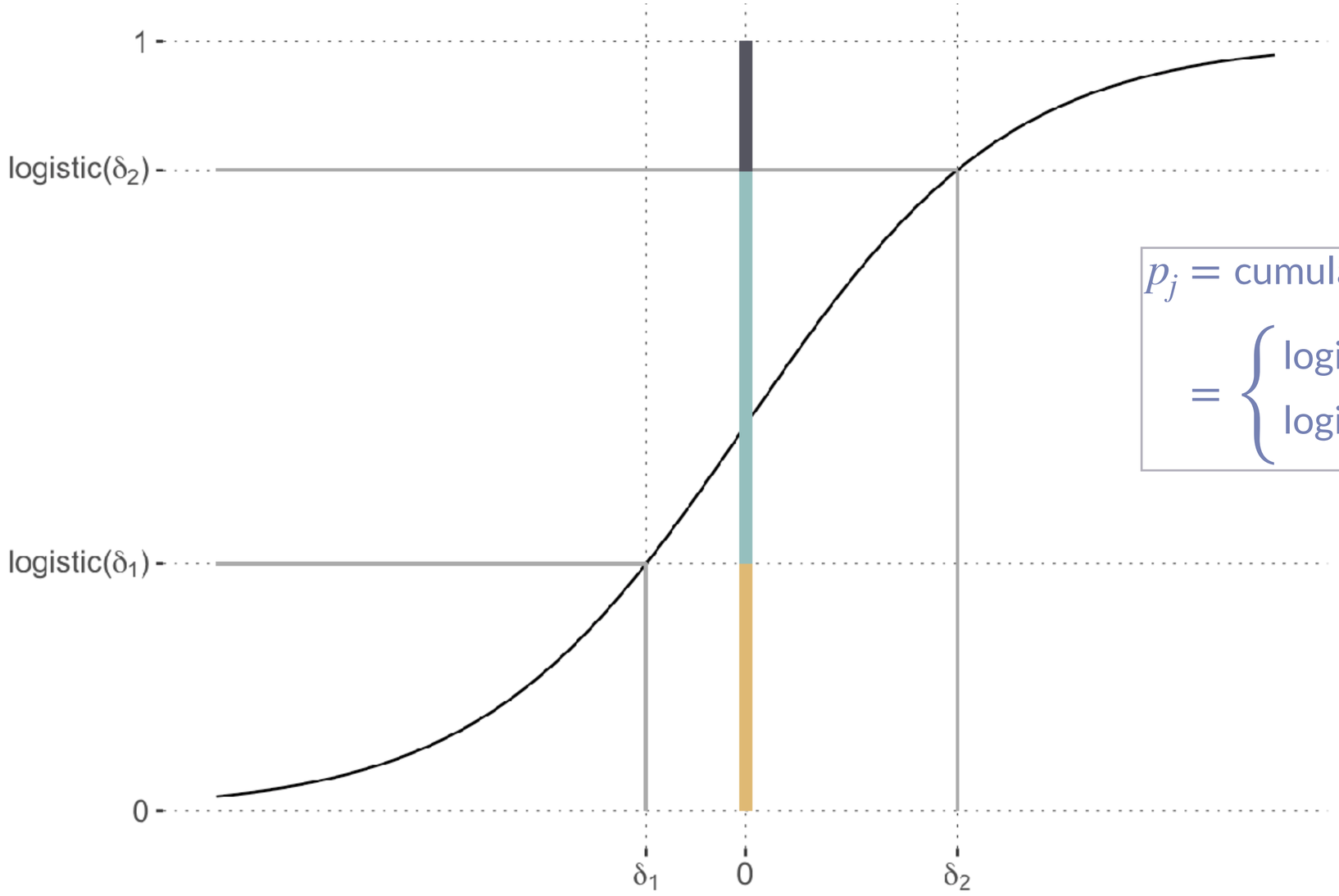



```
 Family: categorical
  Links: mucurved = logit; muCoM = logit
Formula: prototype_label ~ target_position
   Data: data_MT_prepped (Number of observations: 2052)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

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

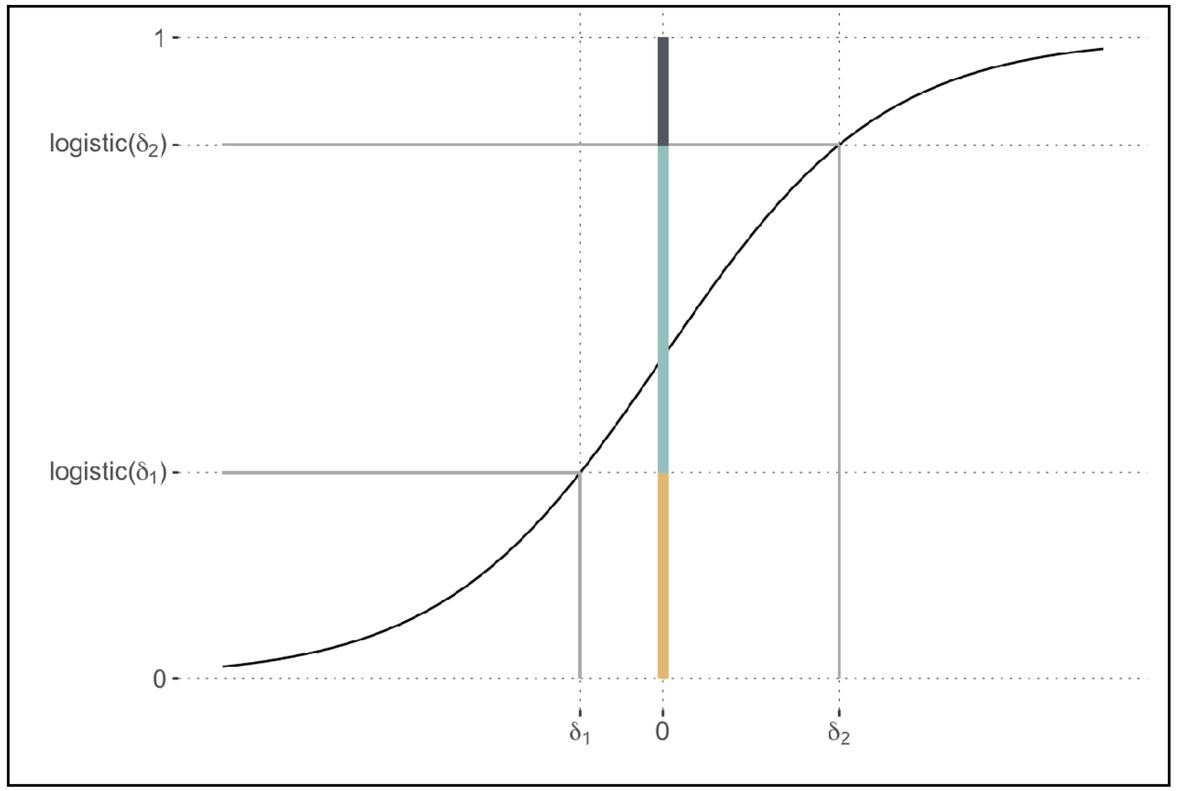


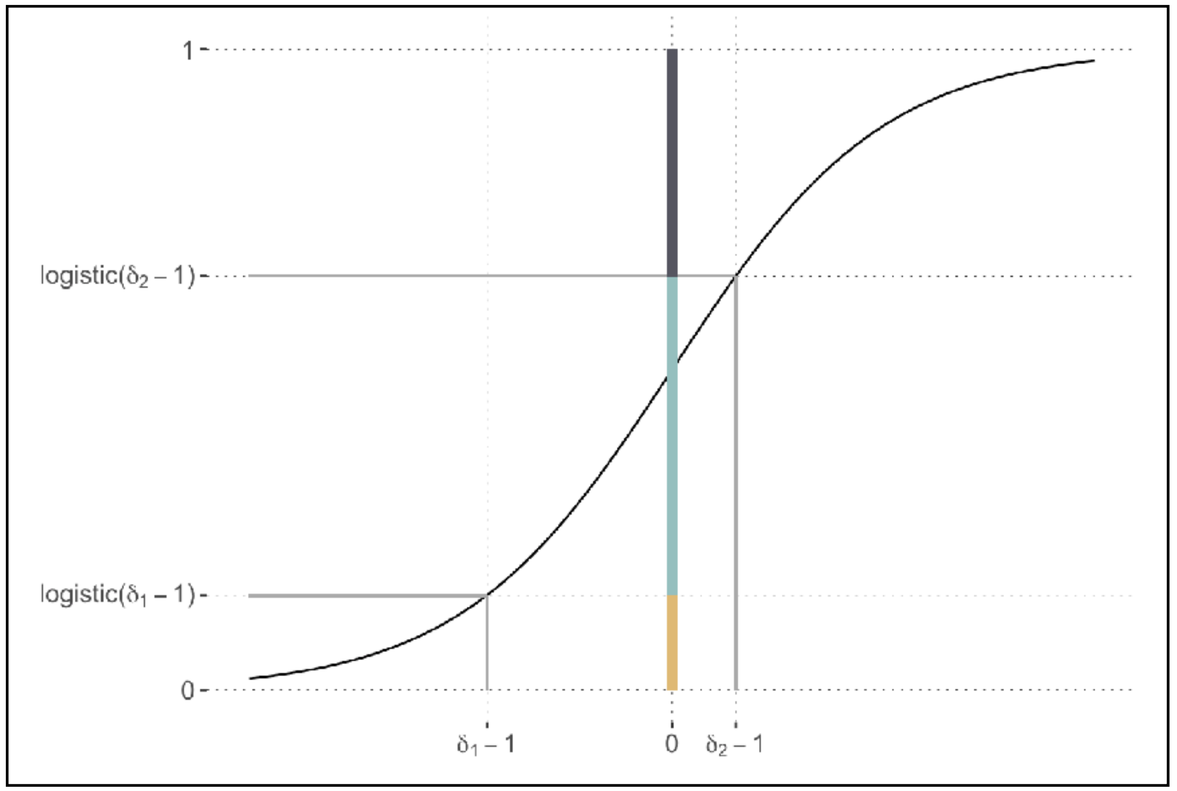
$$p_j = \text{cumulative-logit}(\eta_1; \delta)$$

$$= \begin{cases} \text{logistic}(\delta_1 - \eta_1) & \text{if } j = 1 \\ \text{logistic}(\delta_i - \eta_1) - p_{j-1} & \text{if } j > 1 \end{cases}$$

Ordinal regression

cumulative logit





$\eta_i = \mathbf{x}_i \cdot \beta$ [linear predictor]

$\xi_i = \text{cumulative-logit}(\eta_i; \delta)$ [predictor of central tendency]

$y_i \sim \text{Categorical}(\xi_i)$ [likelihood]

Ordinal regression

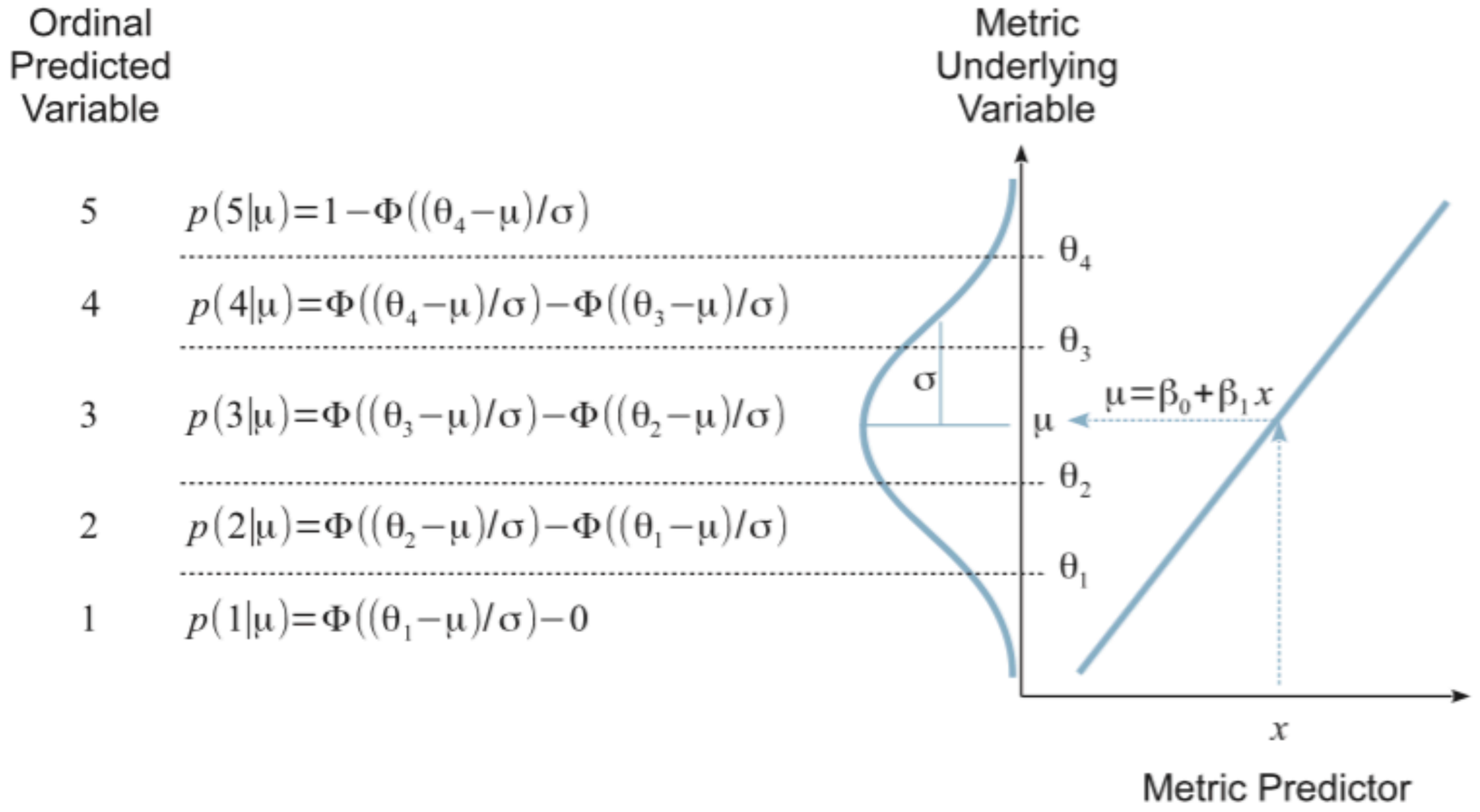
cumulative logit

```
fit_ordinal <- brm(
  formula = prototype_label ~ MAD,
  data = data_MT_prepped2,
  family = cumulative()
)
```

```
 Family: cumulative
  Links: mu = logit; disc = identity
Formula: prototype_label ~ MAD
   Data: data_MT_prepped2 (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 probit



one happy families

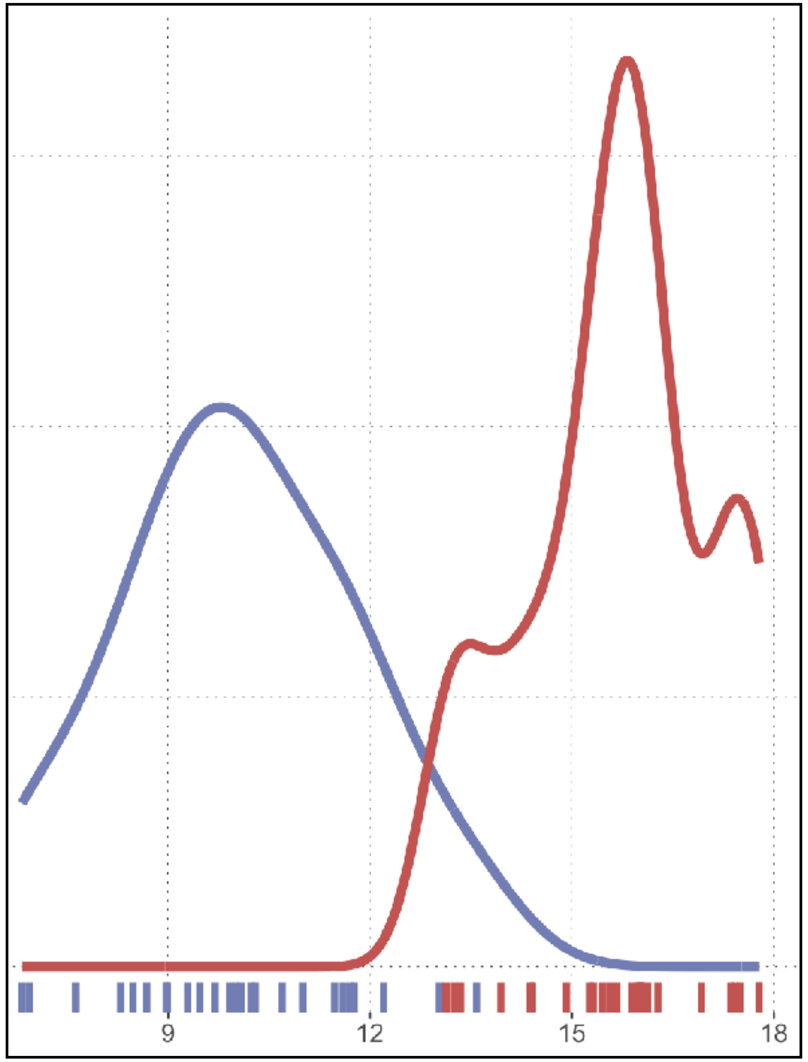
Families in BRMS

- list of available families: link
- explanation of available families: link
- how to write you own: link

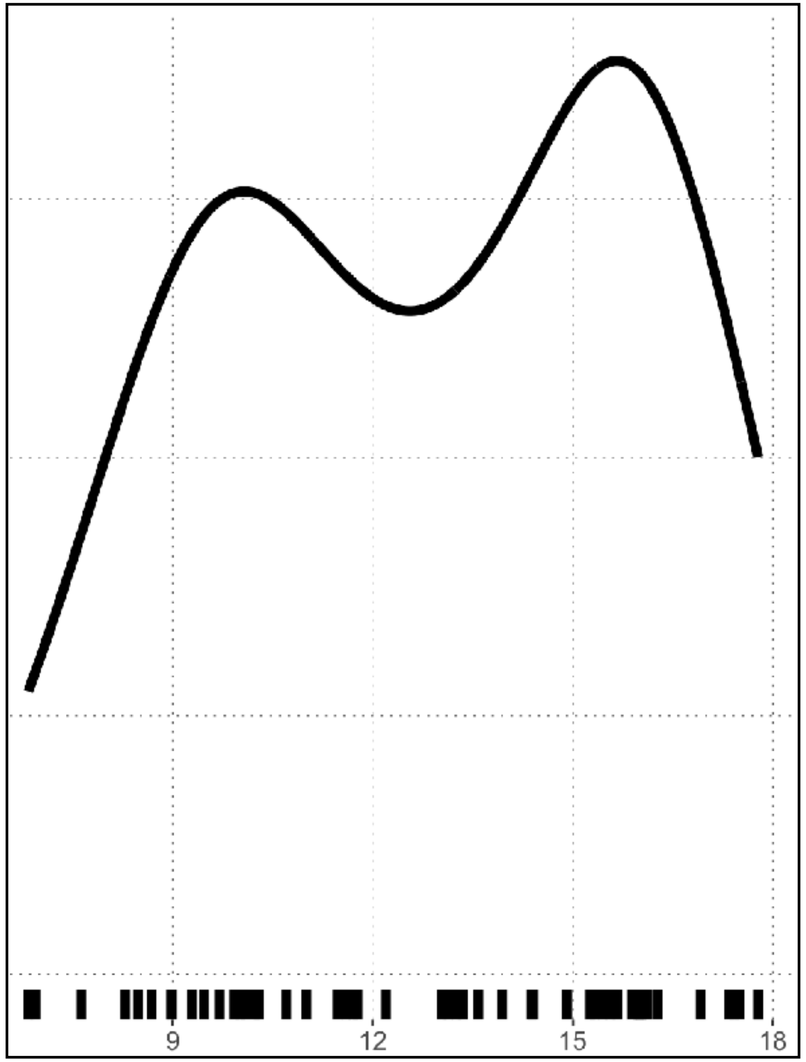
finite
mixture models

Multi-modal response distributions

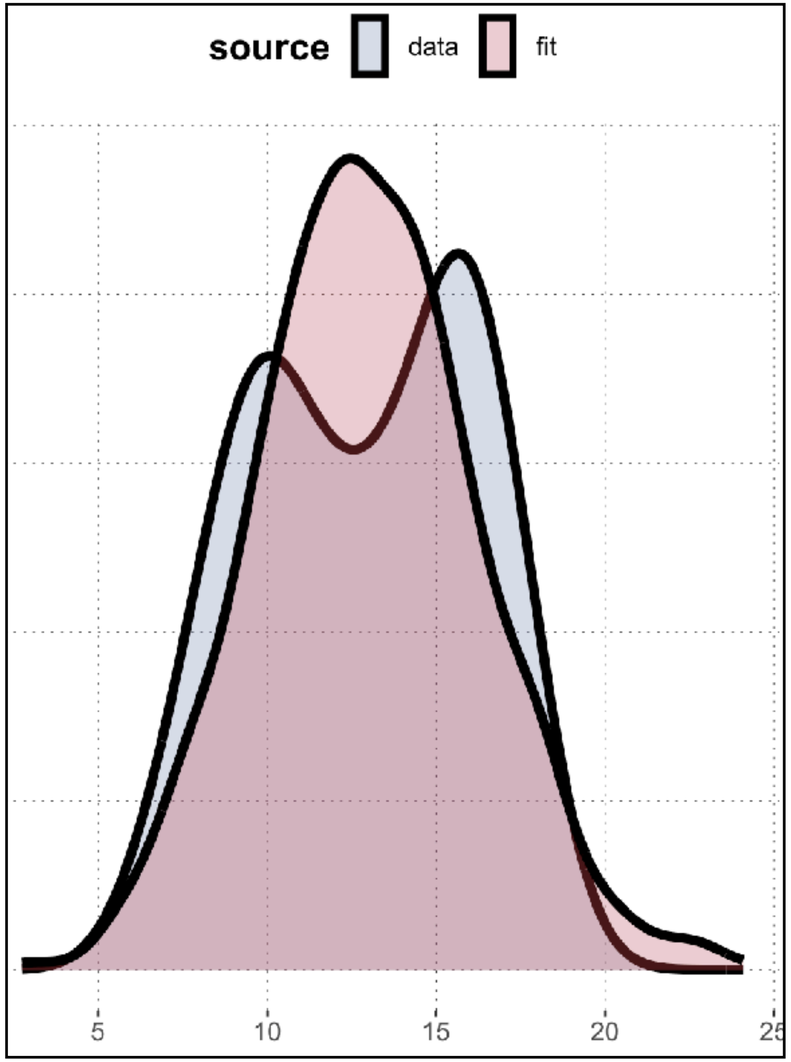
Two-component data



Treated as one for analysis



Posterior predictive … failure



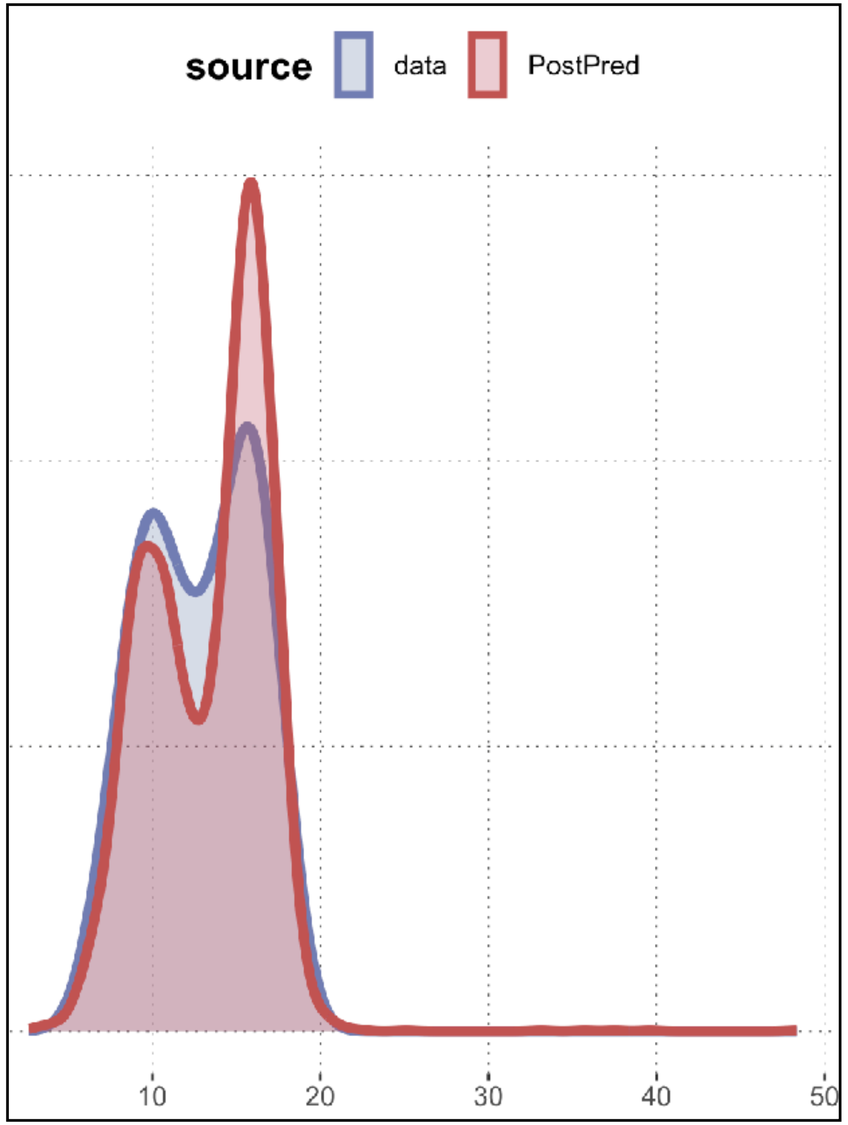
Mixture models

mixing multiple components in the LH function

Let $\langle f_1, \ldots, f_k \rangle$ 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

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

theta1     0.54      0.16     0.18     0.88
theta2     0.46      0.16     0.12     0.82
```

means estimated for each component

SDs estimated for each component

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:

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

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:

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

Hurdle and inflation models in BRMS

as of March 2025

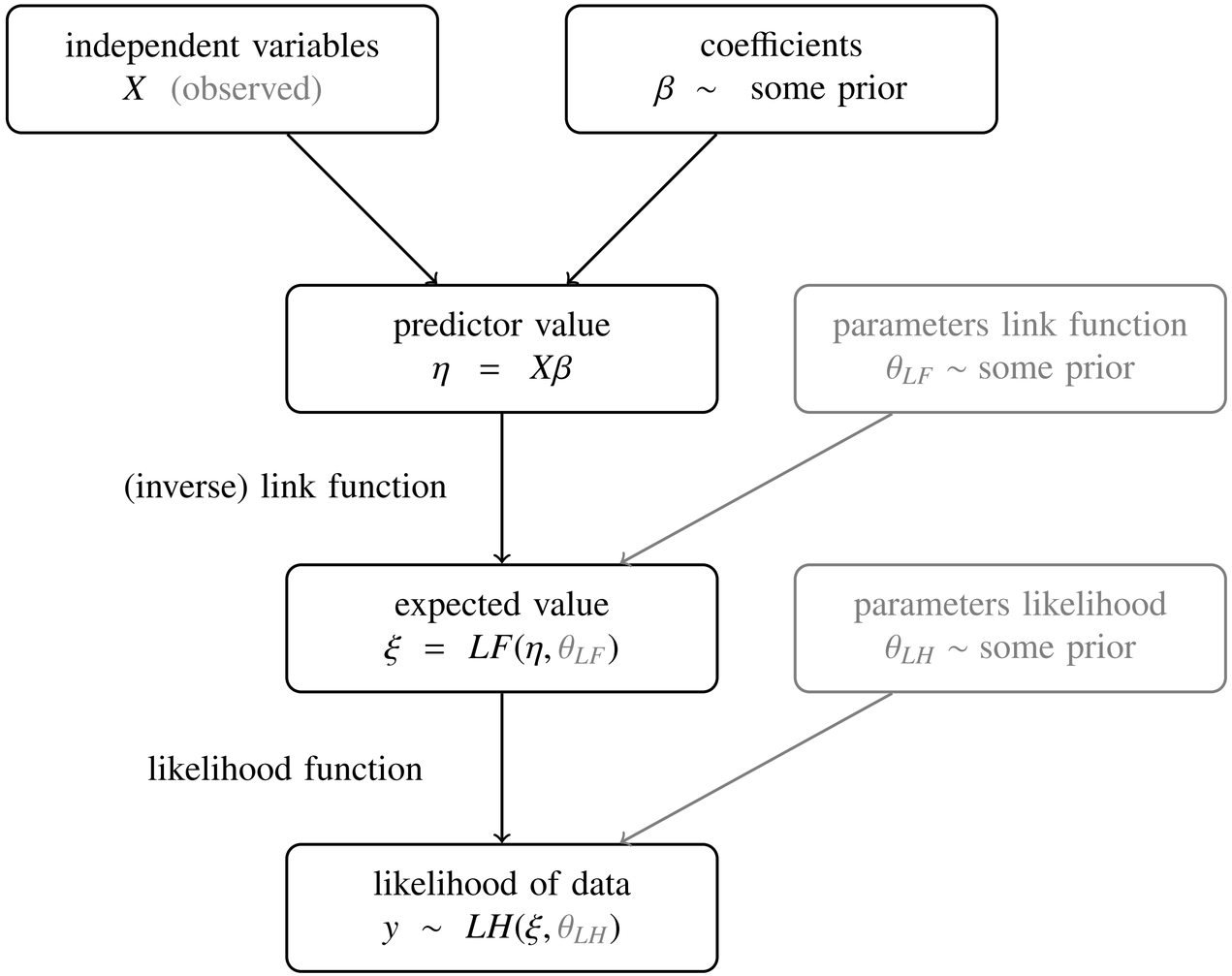
```
hurdle_poisson(link = "log", link_hu = "logit")

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")

hurdle_cumulative(
  link = "logit",
  link_hu = "logit",
  link_disc = "log",
  threshold = "flexible"
)

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")

zero_inflated_beta_binomial(
  link = "logit",
  link_phi = "log",
  link_zi = "logit"
)
```


full family documentation here

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:

$$\eta = X\beta$$

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

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

demo

section “distributional models”

recap & preparation

Recap & preparation

- **recap**
  - multiple regression & categorical predictors
  - generalized linear regression (& beyond)

- **preparation**
  - multi-level models
  - model comparison
  - model criticism