Estimating Distributional Parameters in Hierarchical Models

Introduction: Variability in Hierarchical Models

Linear Models

$$y_{ij} = \beta_0 + \beta_1 X_{ij} + e_{ij}$$
$$e_{ij} \sim N(0, \sigma^2)$$

- Modelling central tendency
- Response (y_{ij}) is a sum of intercept (β_0), slopes (β_1, β_2, ...), and error (e_{ij})
- Error is assumed to be normally distributed around zero

Linear Models

```
lm(y ~ pred)
```

- Modelling central tendency
- Response (y) is a sum of intercept (implicit), slopes (pred), and error (implicit)
- Error is assumed to be normally distributed around zero

Linear Mixed Effects Models

$$y_{ij} = \beta_0 + \mu_{0i} + (\beta_1 + \mu_{1i})X_{ij} + e_{ij}$$
$$\mu_{0i} \sim N(0, \sigma^2)$$
$$\mu_{1i} \sim N(0, \sigma^2)$$
$$e_{ij} \sim N(0, \sigma^2)$$

- Modelling central tendency
- Response (y_{ij}) is a sum of intercept (β_0), slopes ($\beta_1, \beta_2, \ldots$), random unit intercepts (μ_{0i}), random unit slopes (μ_{1i}), and error (e_{ij})
- Error, random intercepts, and random slopes are assumed to be normally distributed around zero

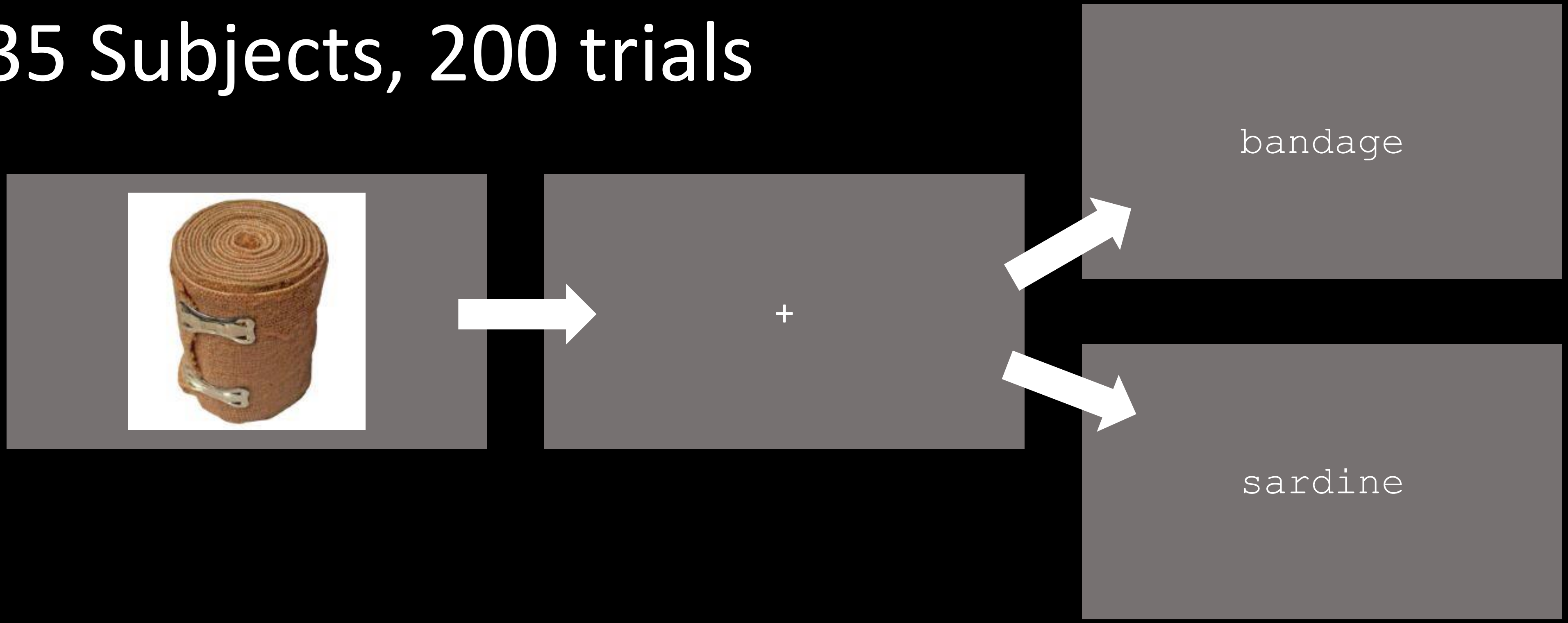
Linear Mixed Effects Models

```
lmer(y ~ pred + (pred | rand_unit))
```

- Modelling central tendency
- Response (y) is a sum of intercept (implicit), slopes (pred), random unit intercepts (pred || rand_unit), random unit slopes (pred | rand_unit), and error (implicit)
- Error, random intercepts, and random slopes are assumed to be normally distributed around zero

Example Non-Gaussian Data: RT

- 2AFC: does the word match the picture?
- Congruency (2) x Predictability (12% – 100%)
- 35 Subjects, 200 trials

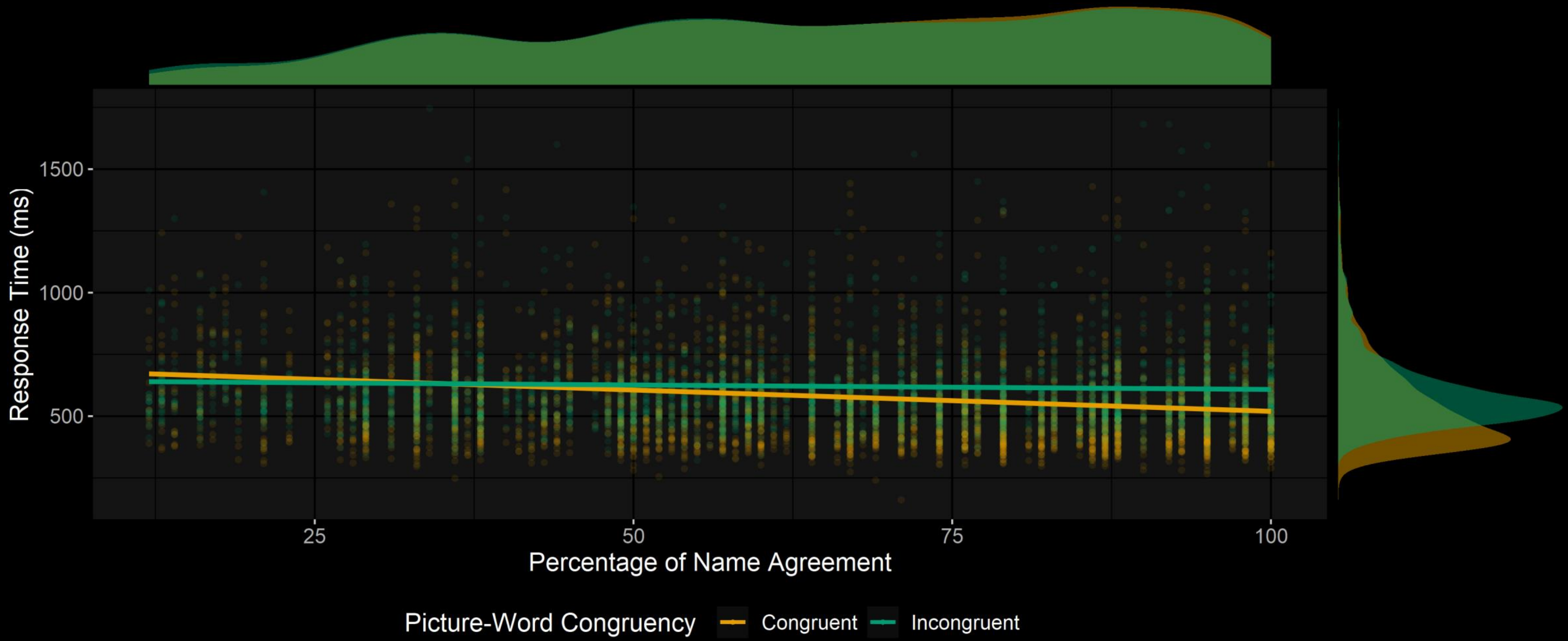


Gamma Family GLMM

```
m_glmer <- glmer(
     rt ~ cong * pred +
          (cong * pred | subj) +
          (cong | image) +
          (1 | word),
     family = Gamma(identity),
     control = glmerControl(
          optimizer = “bobyqa”,
          optCtrl = list(maxfun = 2e5)
     )
)
```

GLMM Results

(Intercept)	667.69541
cong	48.03797
pred	-103.22949
cong:pred	-136.69419



GLMM Results – Random Effects

`summary(m_glmer)`

```
Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 word     (Intercept) 2.337e+02 15.2870
 image    (Intercept) 4.695e+02 21.6688
          cong        1.048e+03 32.3778  0.74
 subj     (Intercept) 2.606e+03 51.0471
          cong        1.940e+03 44.0418   0.47
          pred        1.922e+03 43.8369  -0.60 -0.51
          cong:pred   2.793e+03 52.8472  -0.56 -0.87  0.61
 Residual             4.959e-02  0.2227
Number of obs: 6576, groups:  word, 400; image, 200; subj, 35
```

GLMM Results – Random Effects

```
ranef(m_glmer)
```

ranef(m_glmer)	list [3] (S3: ranef.mer)	List of length 3
word	list [400 x 1] (S3: data.frame)	A data.frame with 400 rows and 1 column
(Intercept)	double [400]	3.1423 6.9309 14.3444 1.4099 -0.0101 -0.6390 ...
image	list [200 x 2] (S3: data.frame)	A data.frame with 200 rows and 2 columns
(Intercept)	double [200]	13.17 18.08 -20.21 45.67 -8.16 3.65 ...
cong	double [200]	14.10 27.02 -27.14 64.37 -21.78 -2.86 ...
subj	list [35 x 4] (S3: data.frame)	A data.frame with 35 rows and 4 columns
(Intercept)	double [35]	286.4 -142.0 -186.4 -12.4 -67.2 69.9 ...
cong	double [35]	26.52 -55.09 -28.47 -16.70 -7.34 20.52 ...
pred	double [35]	37.3 60.0 47.3 1.0 14.2 37.8 ...
cong:pred	double [35]	-63.15 83.25 79.68 14.62 35.09 -3.13 ...

GLMM Results – Random Effects

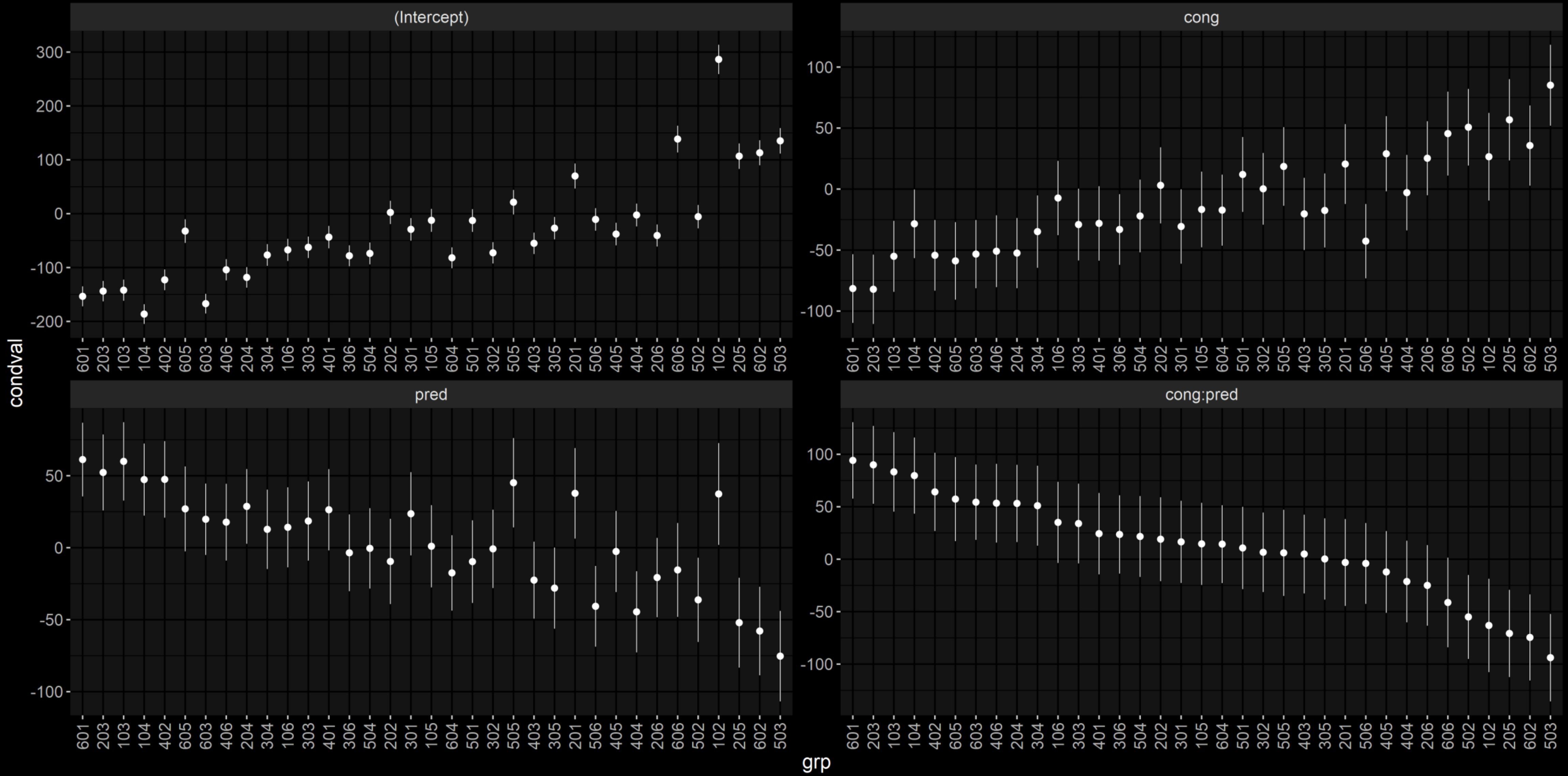
```
m_glmer %>% ranef() %>% as.data.frame()
```

	grpvar	term	grp	condval	condsd
1	word	(Intercept)	accordion	3.14226248	14.48676
2	word	(Intercept)	ammunition	6.93087607	14.04188
3	word	(Intercept)	antenna	14.34436983	14.63559
4	word	(Intercept)	antler	1.40993136	13.89485
5	word	(Intercept)	apricot	-0.01013456	13.88430
6	word	(Intercept)	apron	-0.63898414	14.43003
7	word	(Intercept)	aquarium	-0.19911474	14.62180
8	word	(Intercept)	arrow	4.43943539	14.51931
9	word	(Intercept)	artery	-7.22261749	13.95952
10	word	(Intercept)	[illegible]	[illegible]	[illegible]

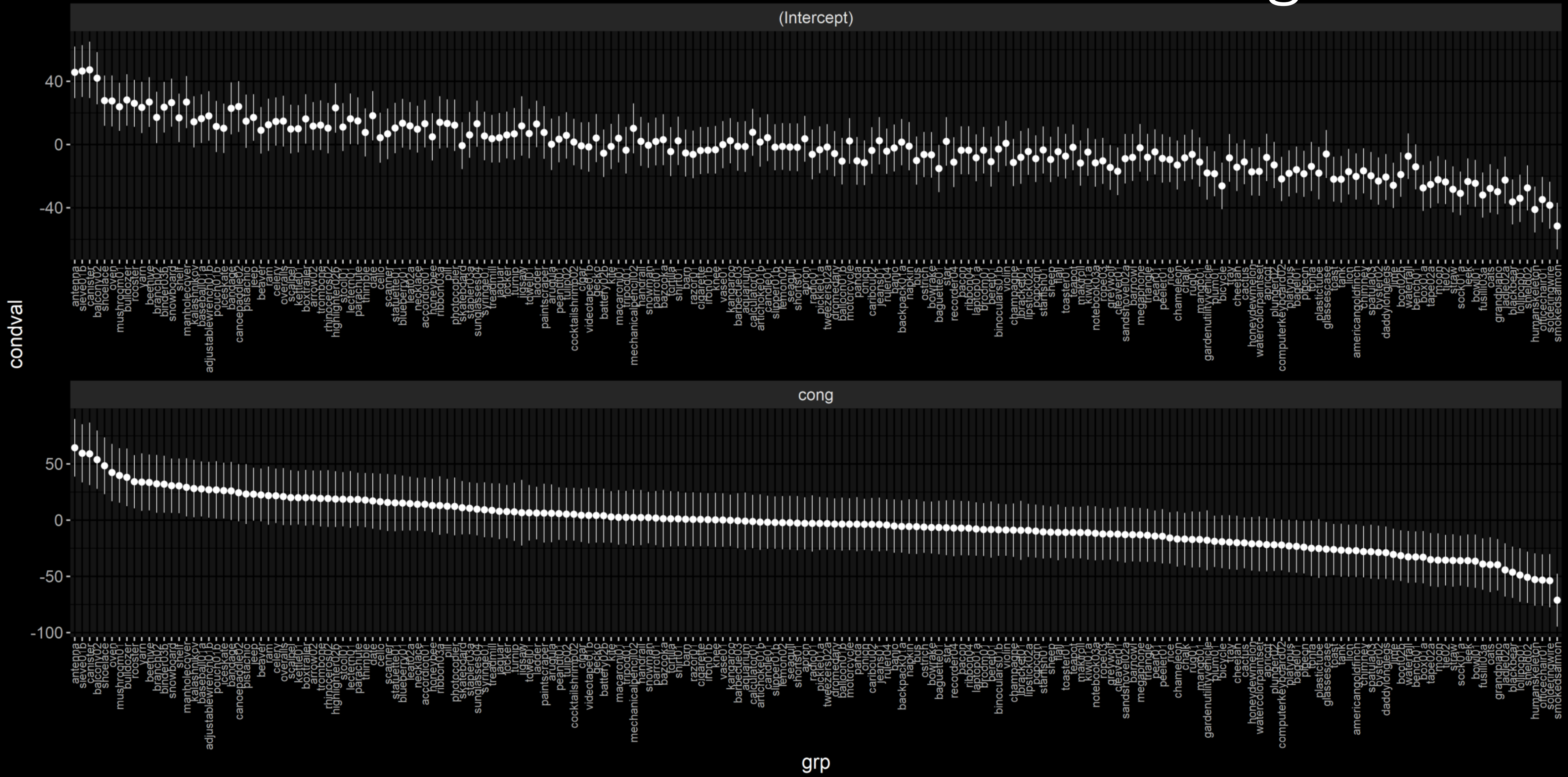
GLMM Results – Random Effects

```
ranef(m_glmer) %>%
  as_tibble() %>%
  filter(grpvar == “subj") %>%
  mutate(grp = fct_reorder2(grp, term, condval)) %>%
  ggplot(aes(
    x = grp, y = condval,
    ymin = condval - condsd,
    ymax = condval + condsd
  )) +
  geom_pointrange(size=0.25) +
  facet_wrap(vars(term), scales="free", nrow=2)
```

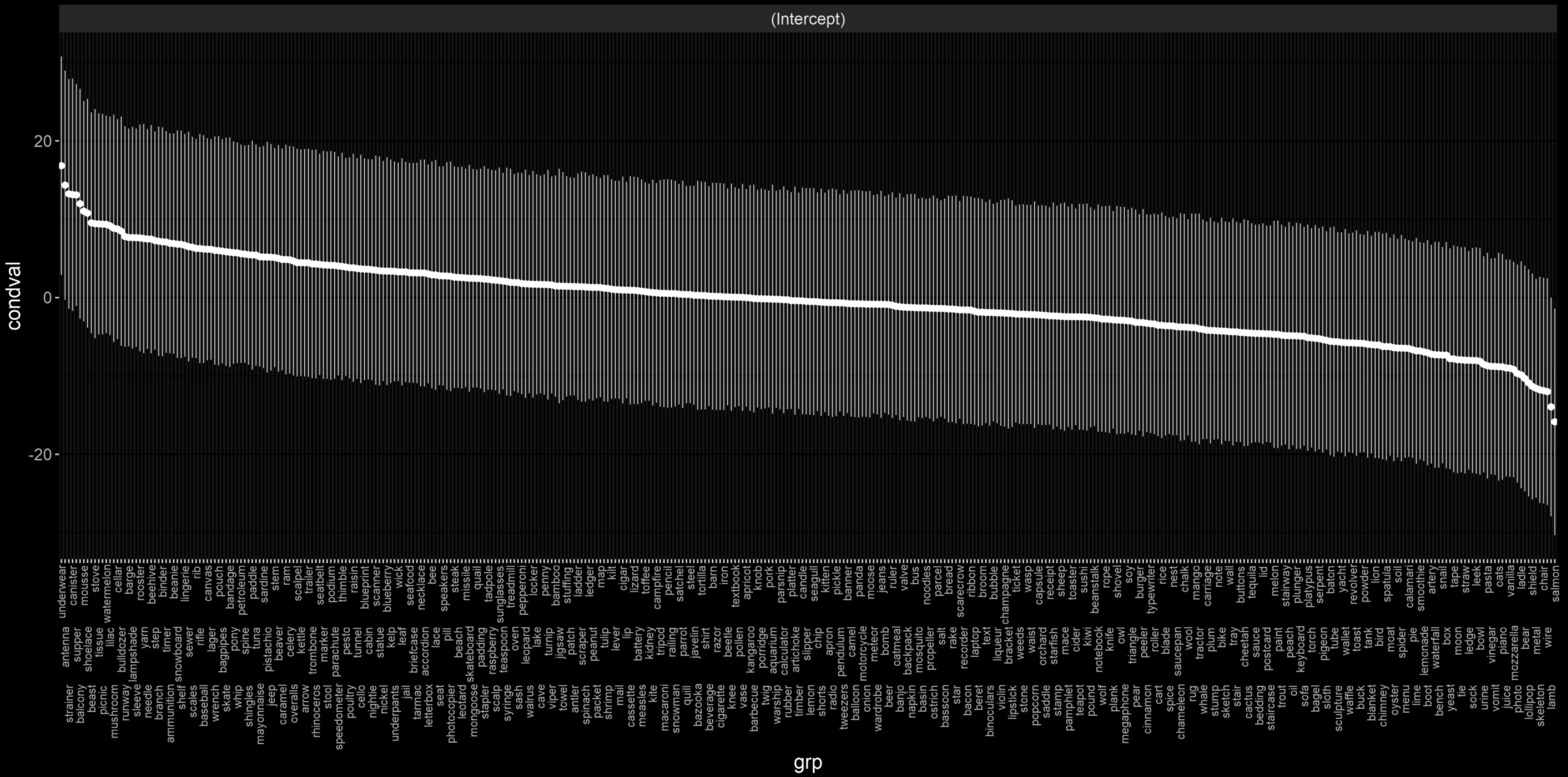
GLMM Results – Random Effects – Subject

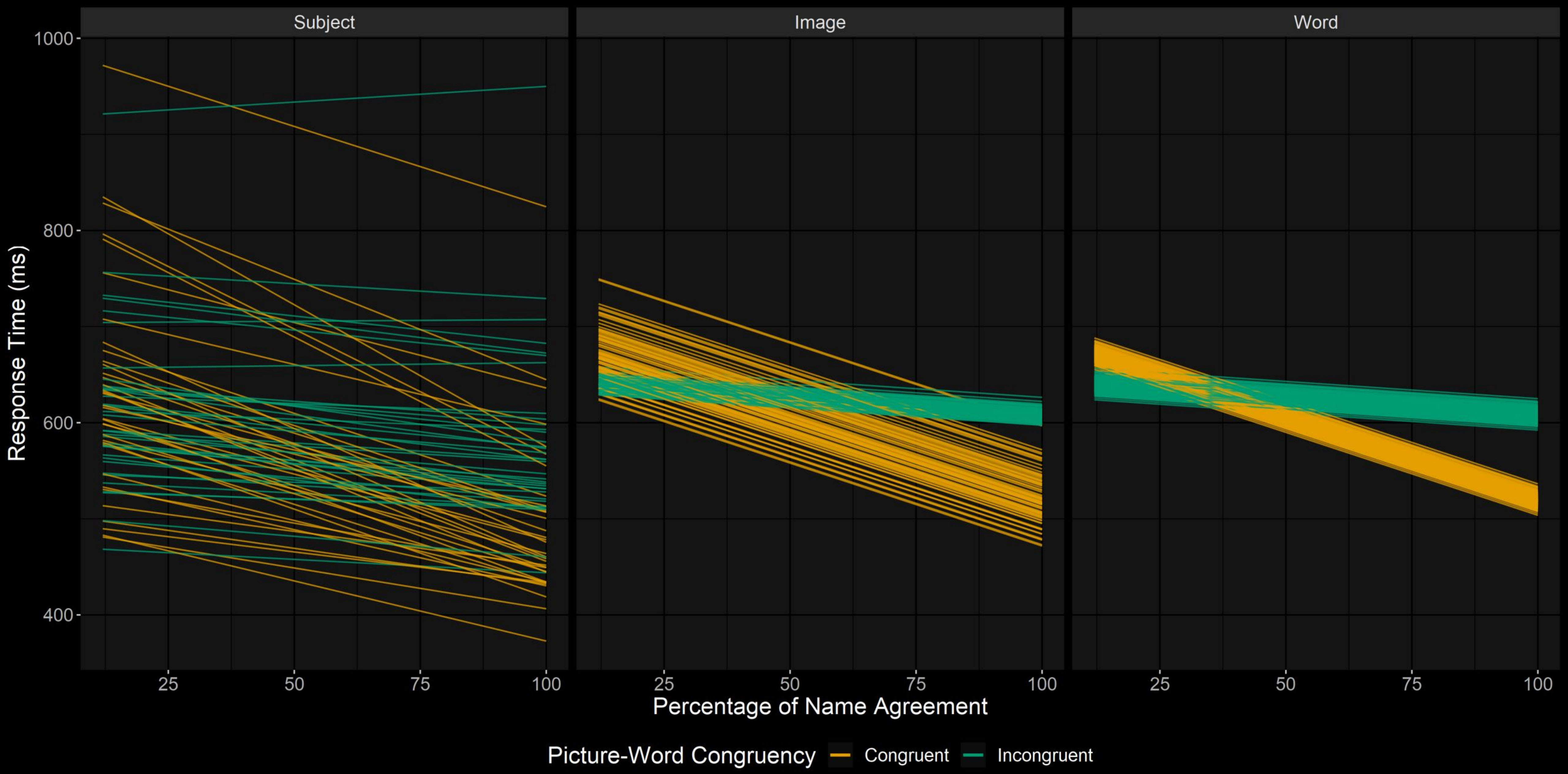


GLMM Results – Random Effects – Image



GLMM Results – Random Effects – Word



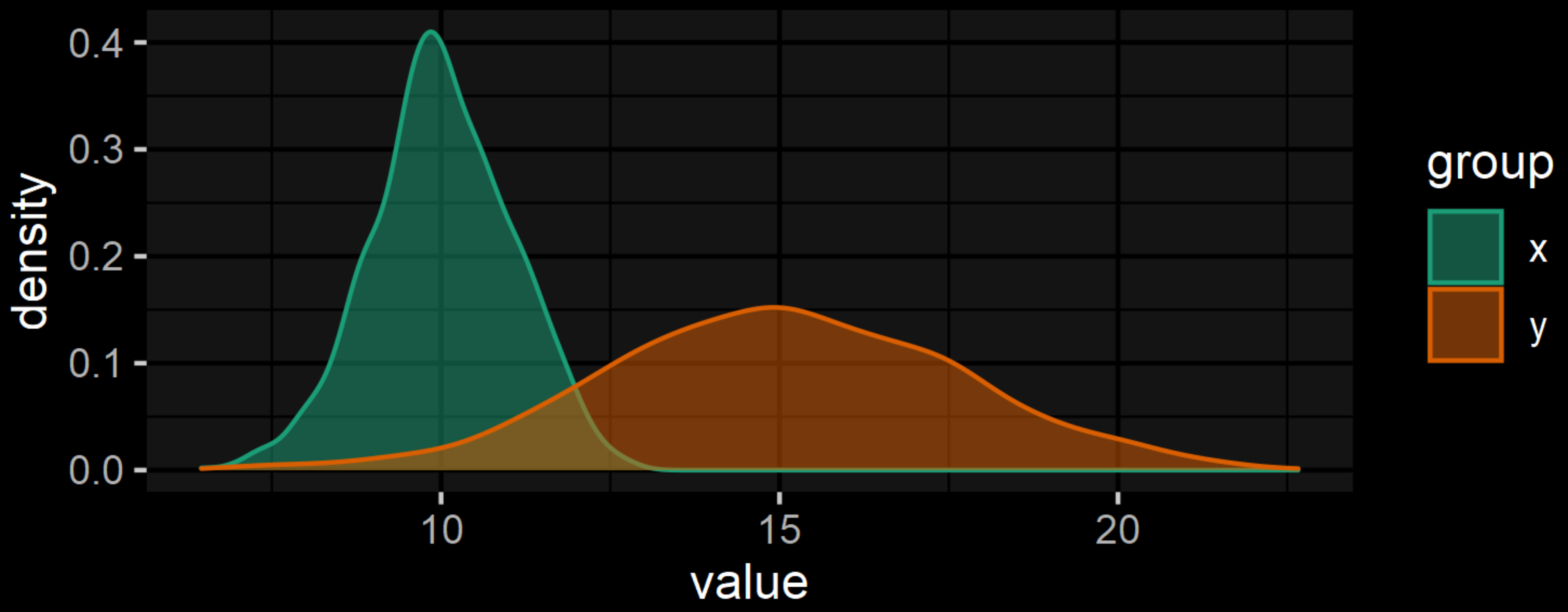


Subject
Image
Word
Response Time (ms)
1000
800
600
400
25
50
75
100
Percentage of Name Agreement
Picture-Word Congruency
Congruent
Incongruent

Estimating Distributional Parameters in Hierarchical Models

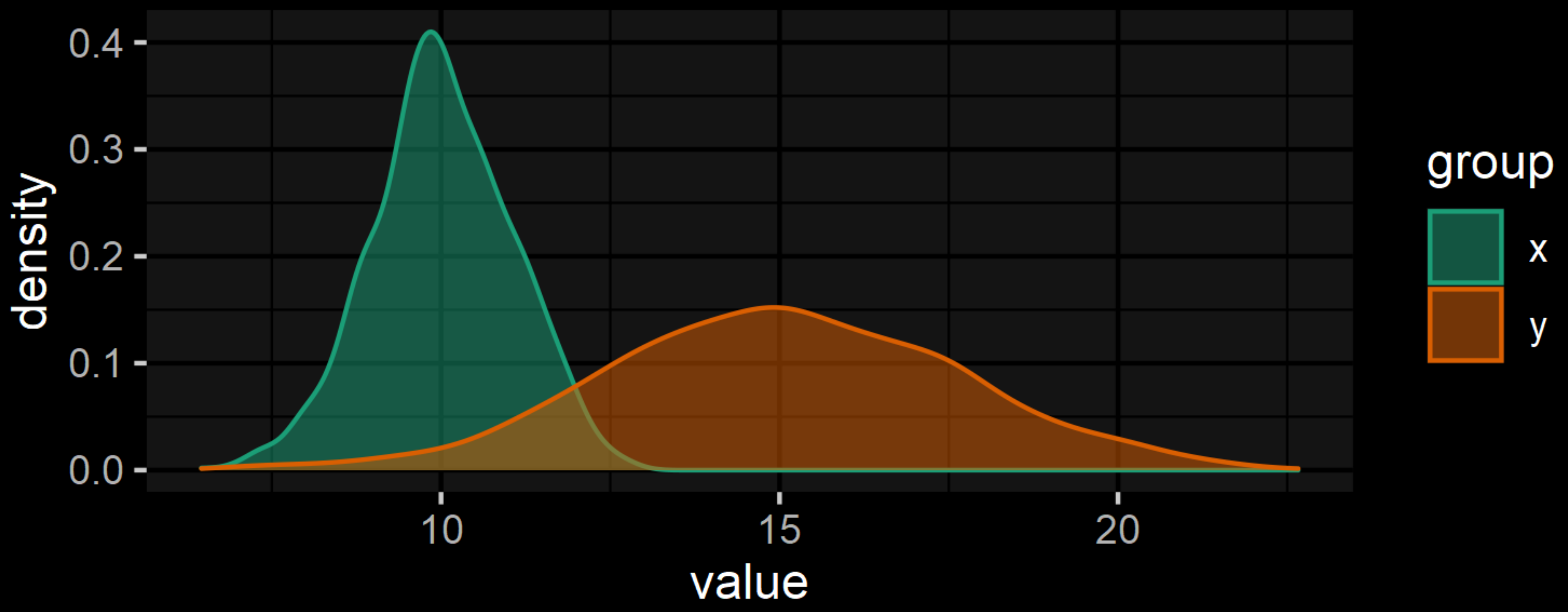
What if Meaningful Effects on Variance?

- All glm variants model single parameters (i.e. central tendency)
- What if your effect looks like this?



What if Meaningful Effects on Variance?

- Mu is higher $F(1, 1998) = 3237$, $p<.001$
- Sigma is higher Levene's $F(1, 1998) = 550$, $p<.001$

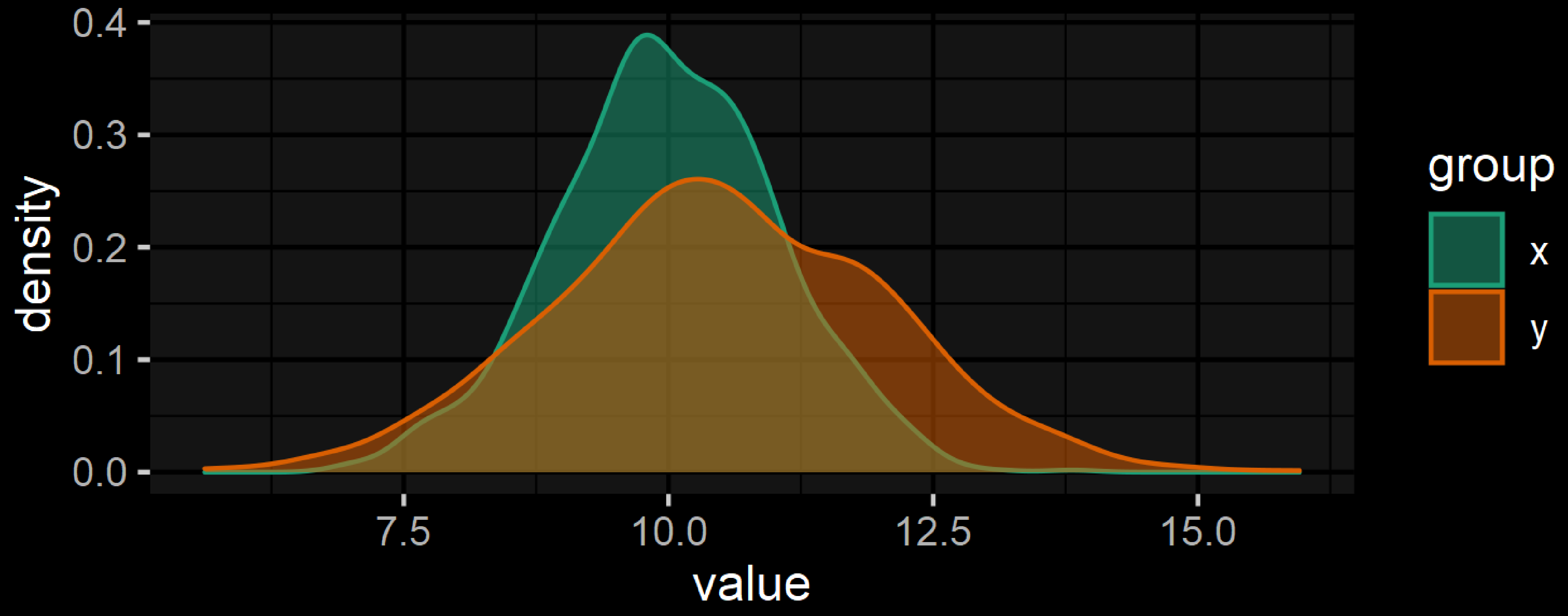


Assumption-free Distribution Comparison

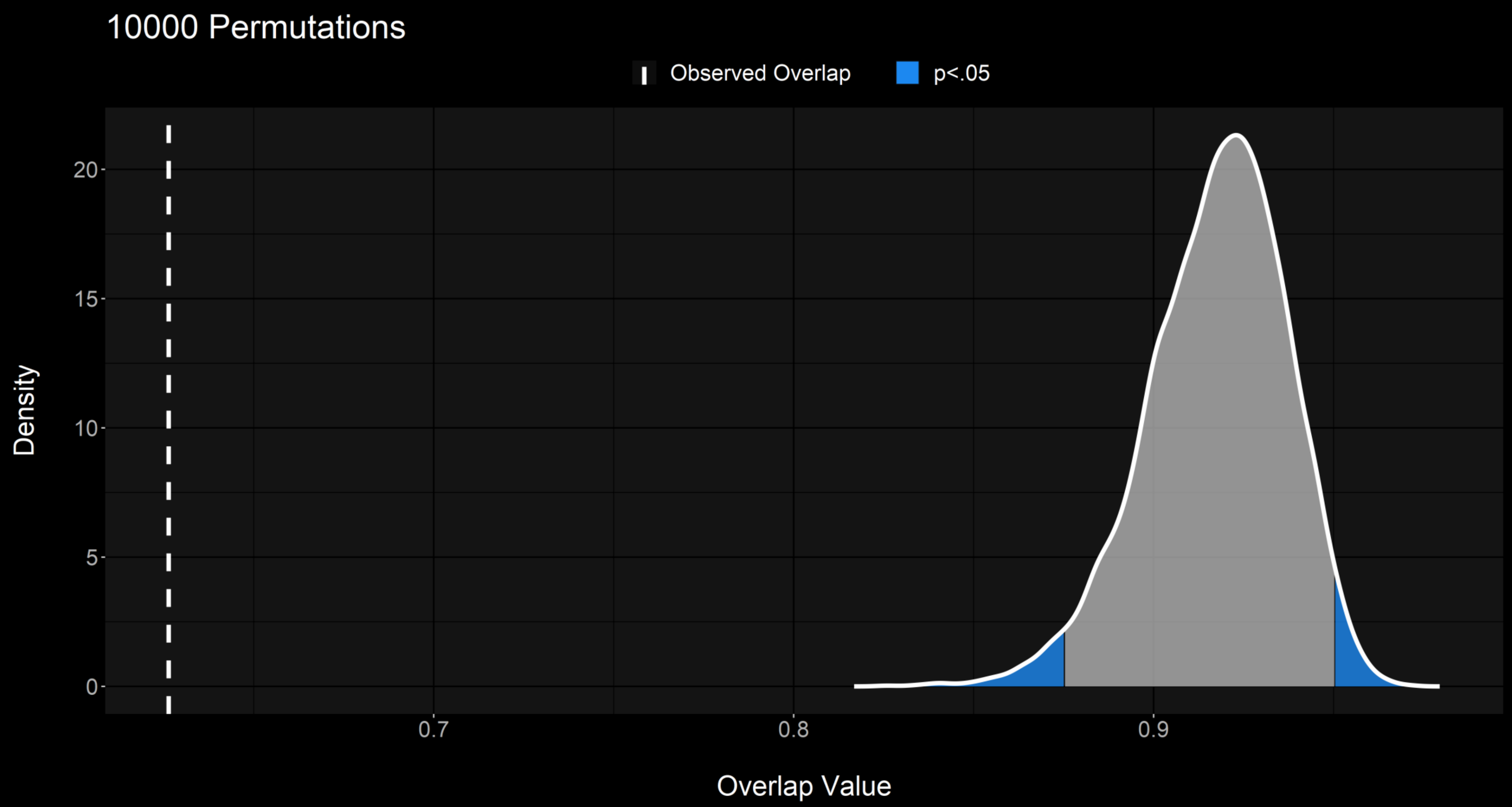
- Within a single model?
- Assumption free distribution comparison (e.g. Kolmogorov–Smirnov) could be one approach!
- Overlapping index (Pastore & Calcagni, 2019) from 0 (no overlap) to 1 (identical distribution)

Assumption-free Distribution Comparison

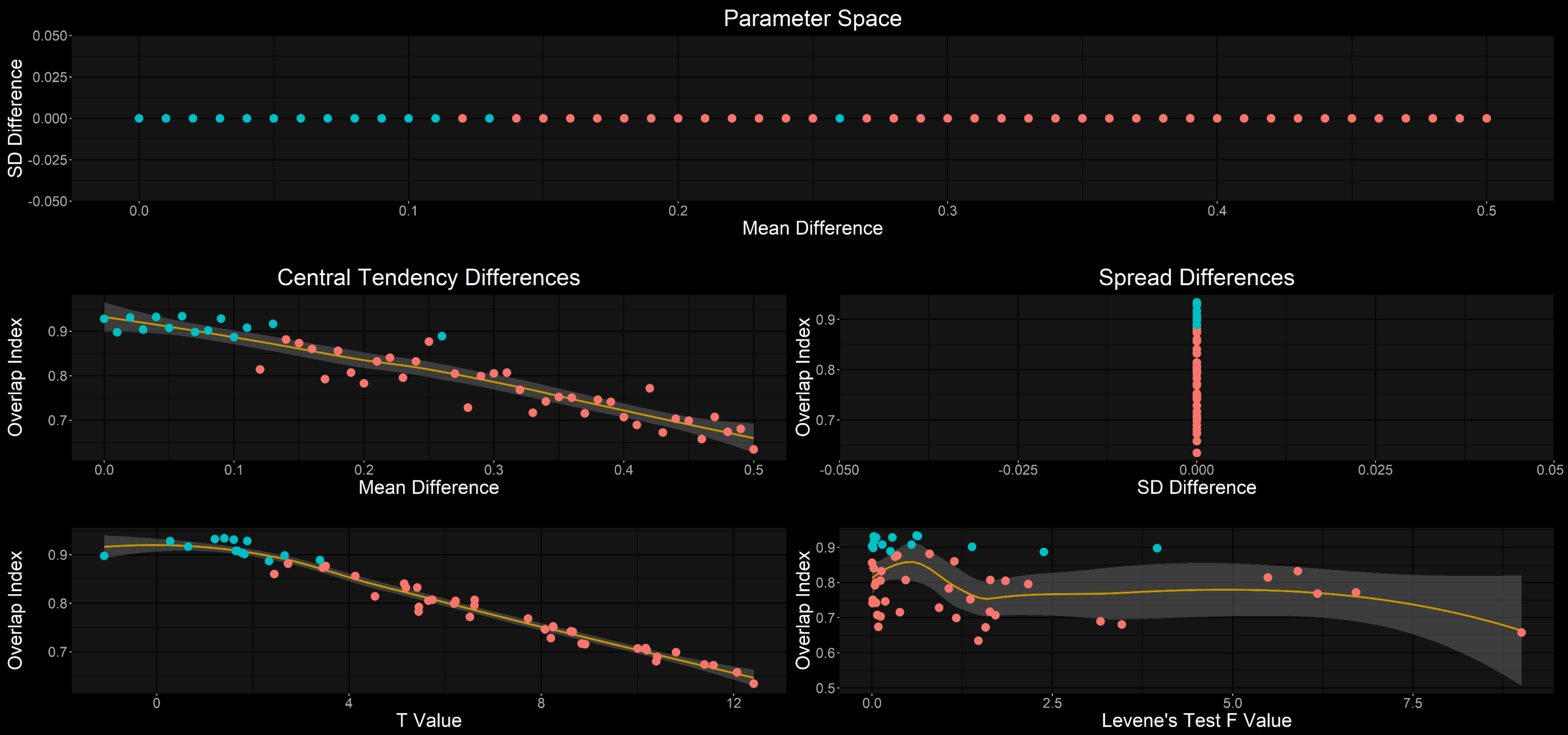
```
x <- rnorm(1000, 10, 1),
y <- rnorm(1000, 10.5, 1.5)
```



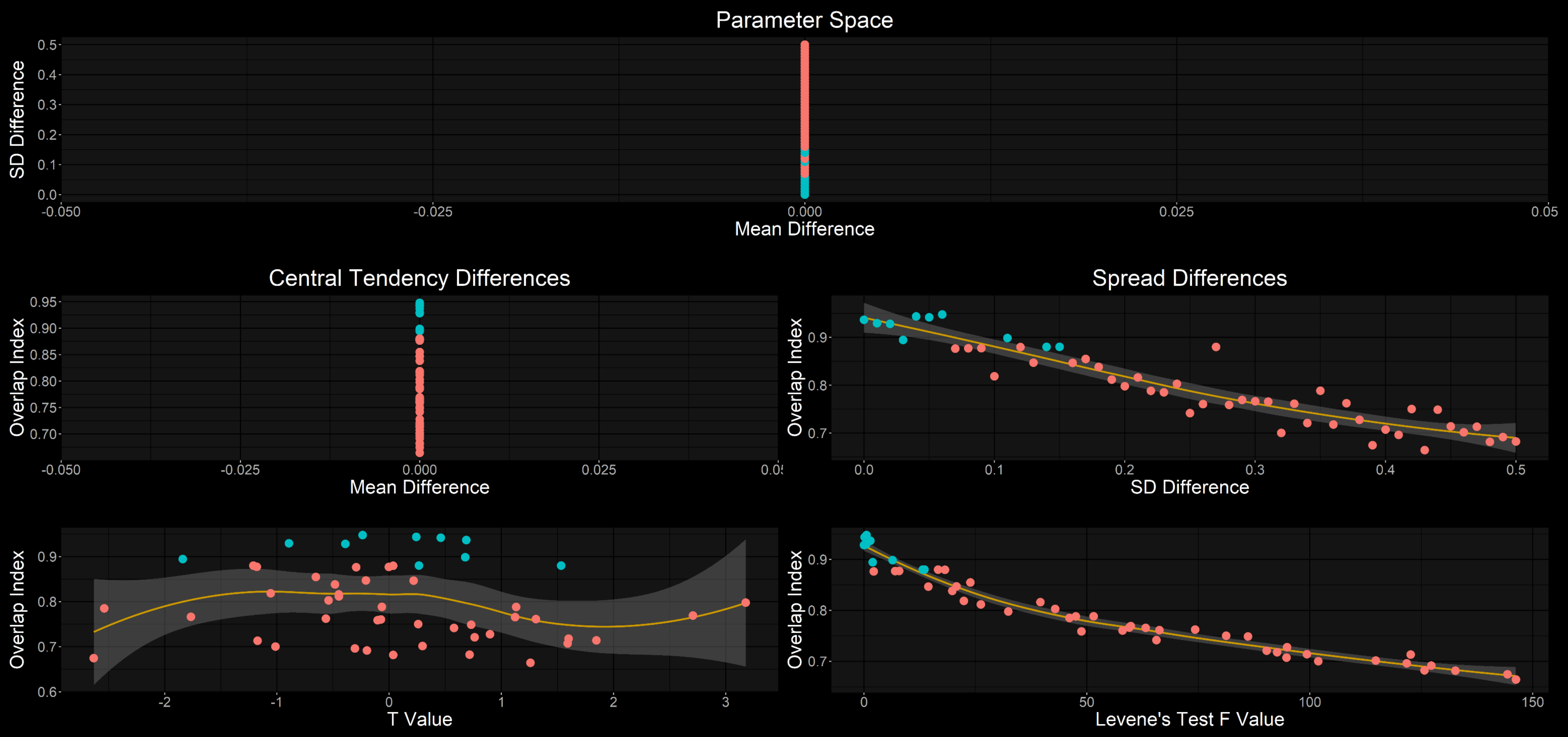
Assumption-free Distribution Comparison



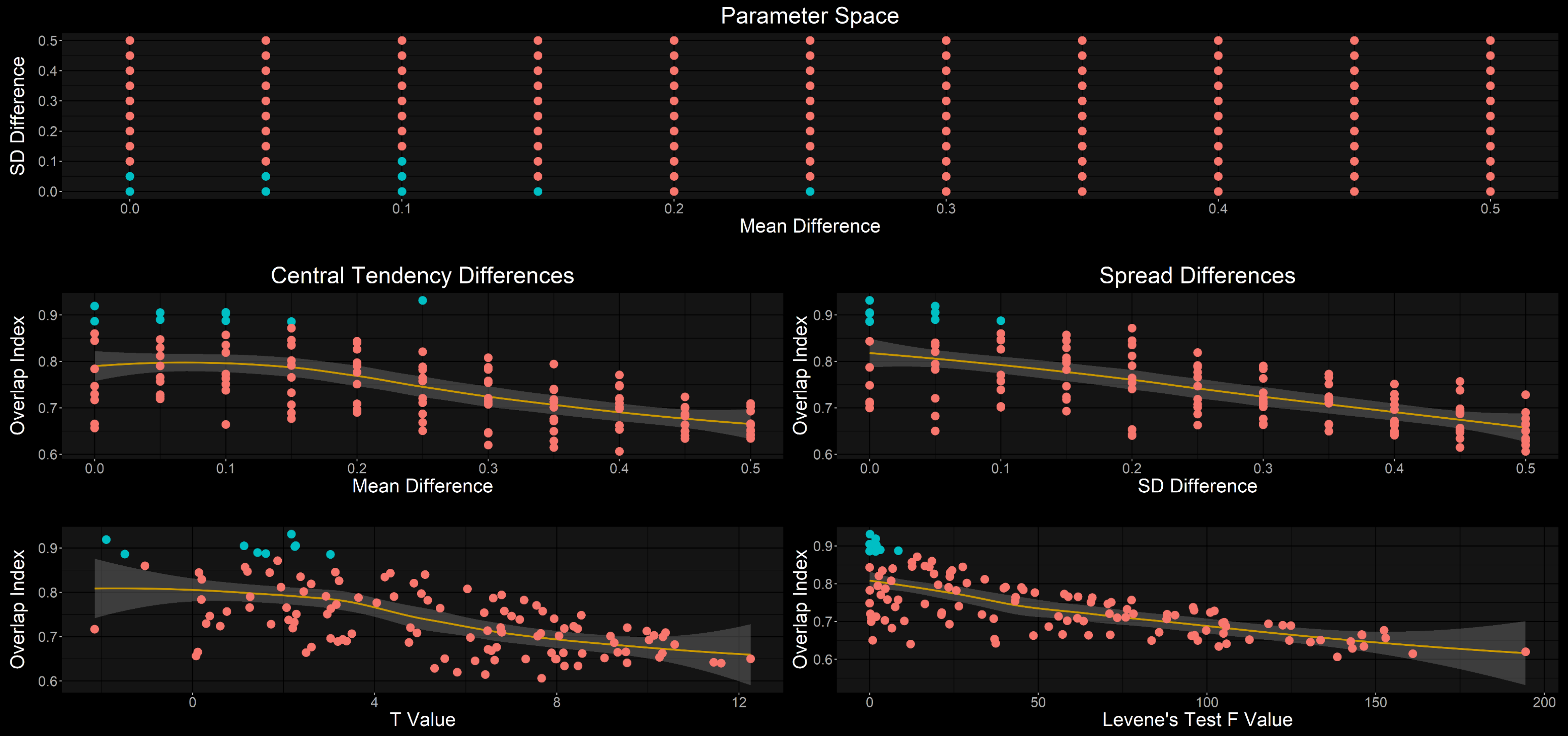
Overlap Index Mu * Sigma Parameter Space



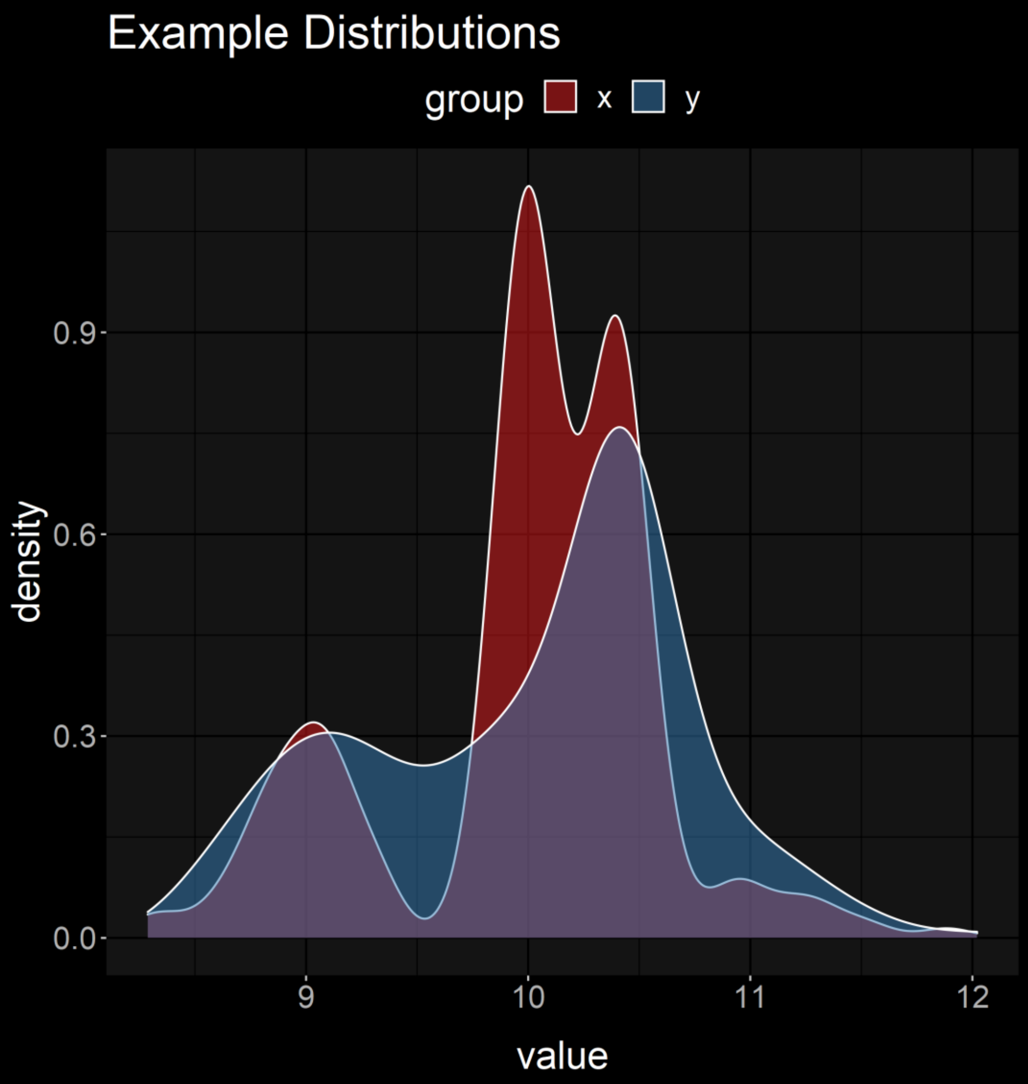
Overlap Index Mu * Sigma Parameter Space

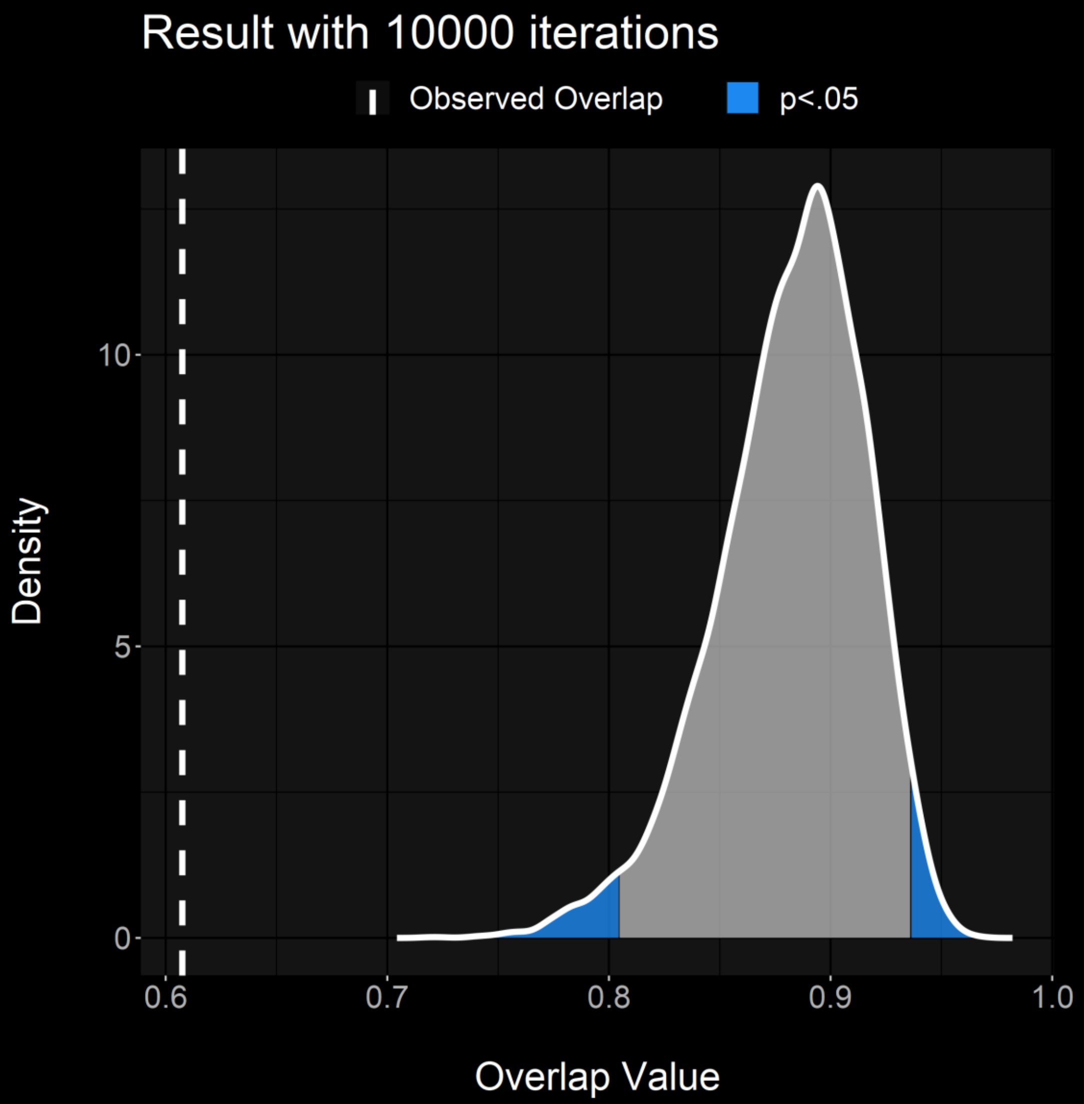


Overlap Index Mu * Sigma Parameter Space

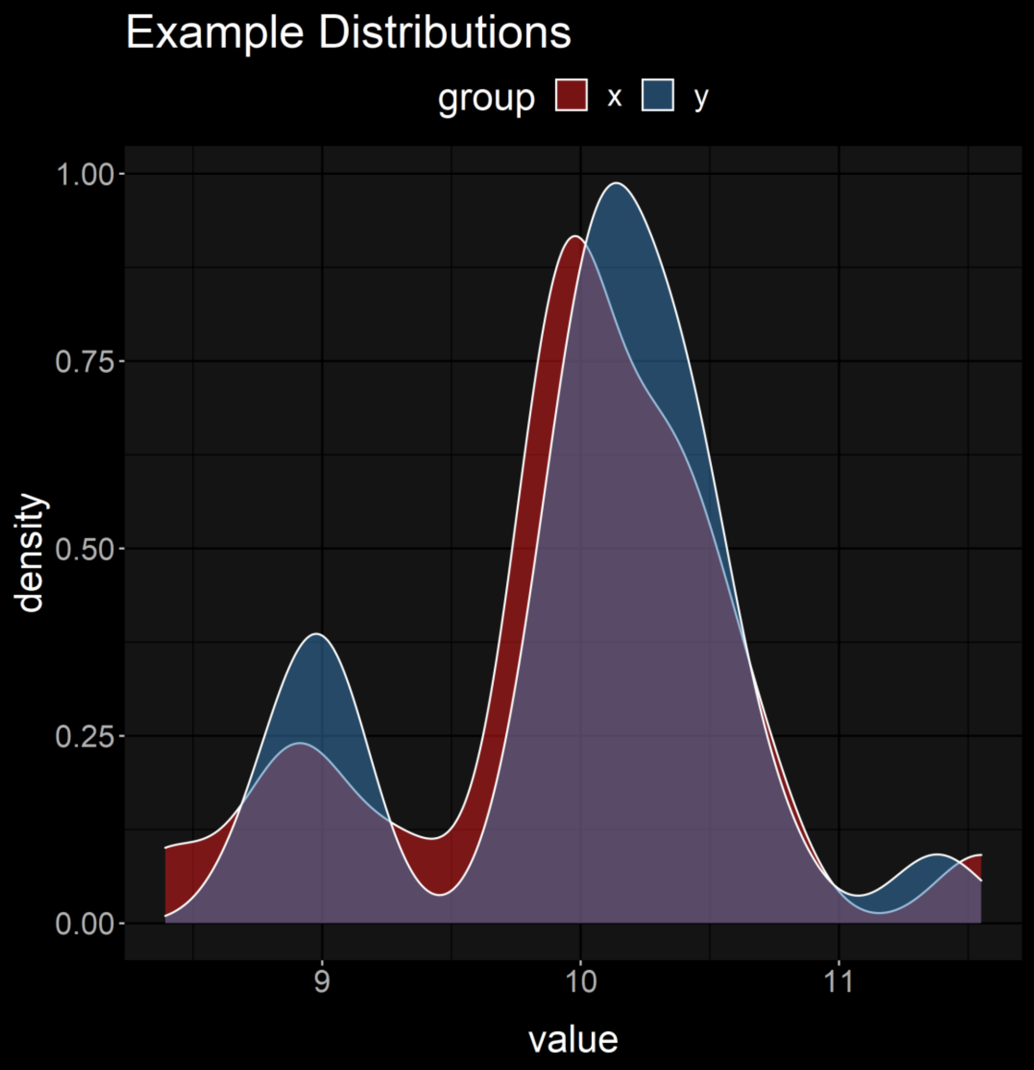


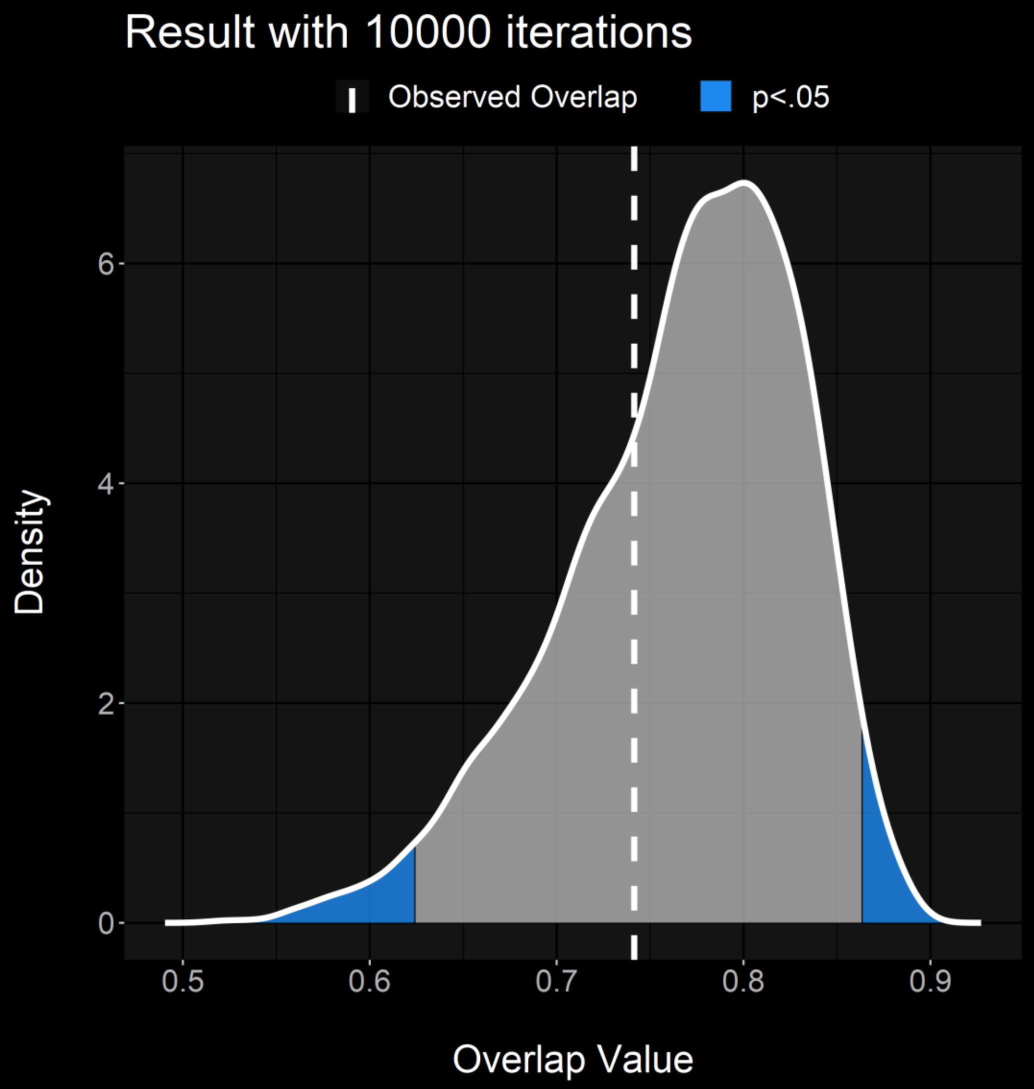
Weirder Distribution Example





Weirder Distribution Example





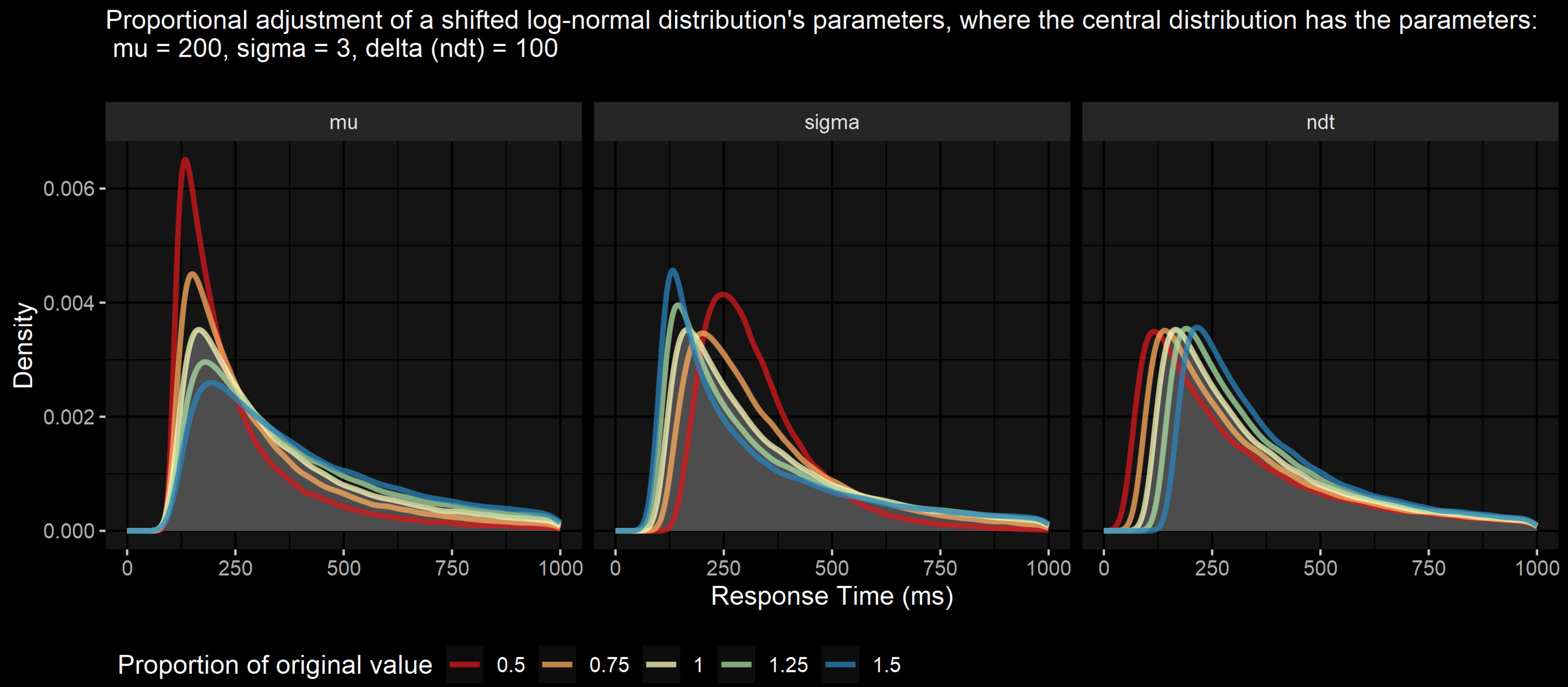
Summary so far

- Assumption-free approaches are flexible but don’t allow us to test/make any specific predictions
- Equivalent of shrugging and saying “yeah idk probs something going on there” (though useful for very weird distributions)
- Explicitly modelling multiple parameters of an assumed distribution can give us more meaningful info

Distributional Parameters in brms

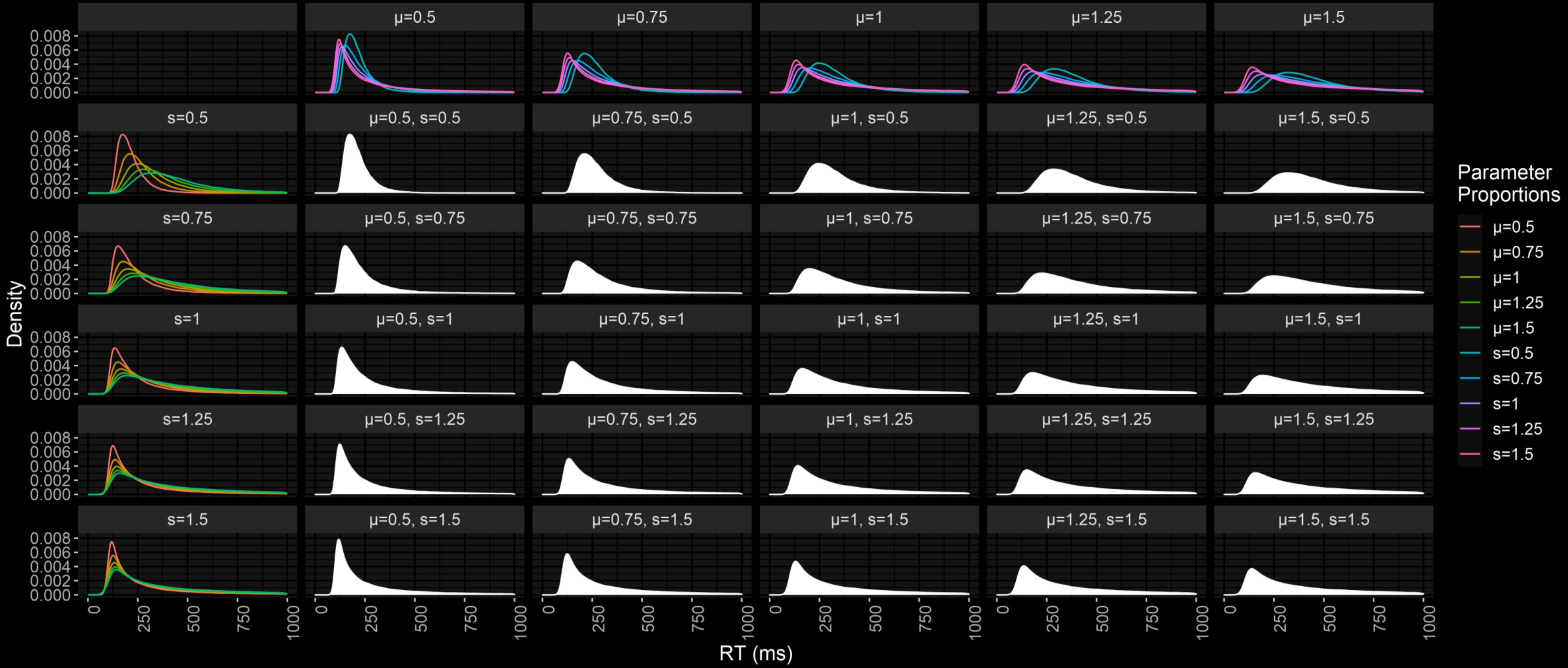
```
brm(
     bf(
          dv ~ Intercept + iv + (iv | rand_unit),
          sigma ~ Intercept + iv + (iv | rand_unit)
     ),
     control = list(
          adapt_delta = 0.999,
          max_treedepth = 12
     ),
     sample_all_pars = TRUE
)
```

Shifted Log-Normal Distribution



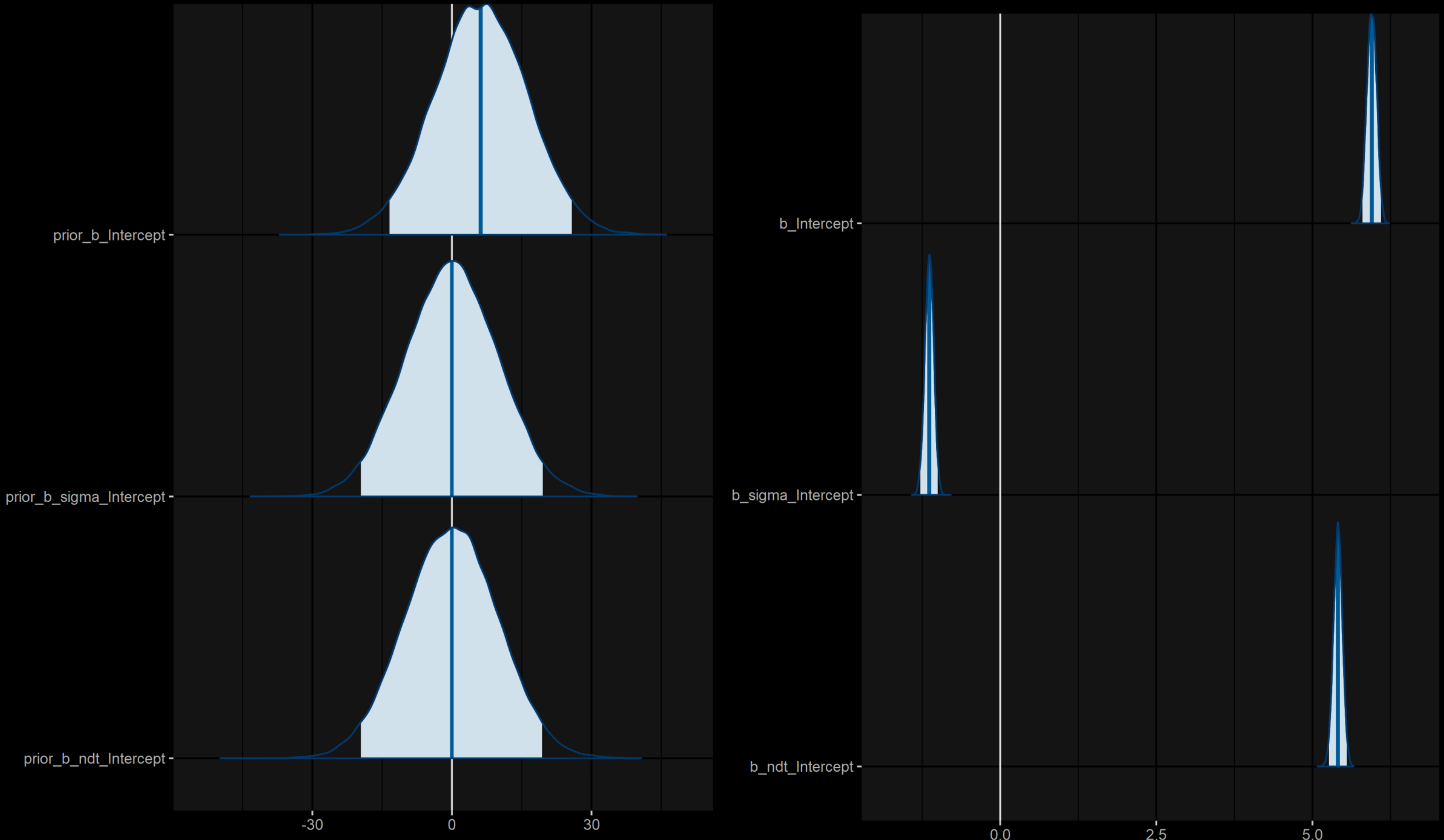
Shifted Log-Normal Distribution

Proportional adjustment of mu and sigma in a shifted log-normal distribution, where the central distribution has the parameters:
mu = 200, sigma = 3

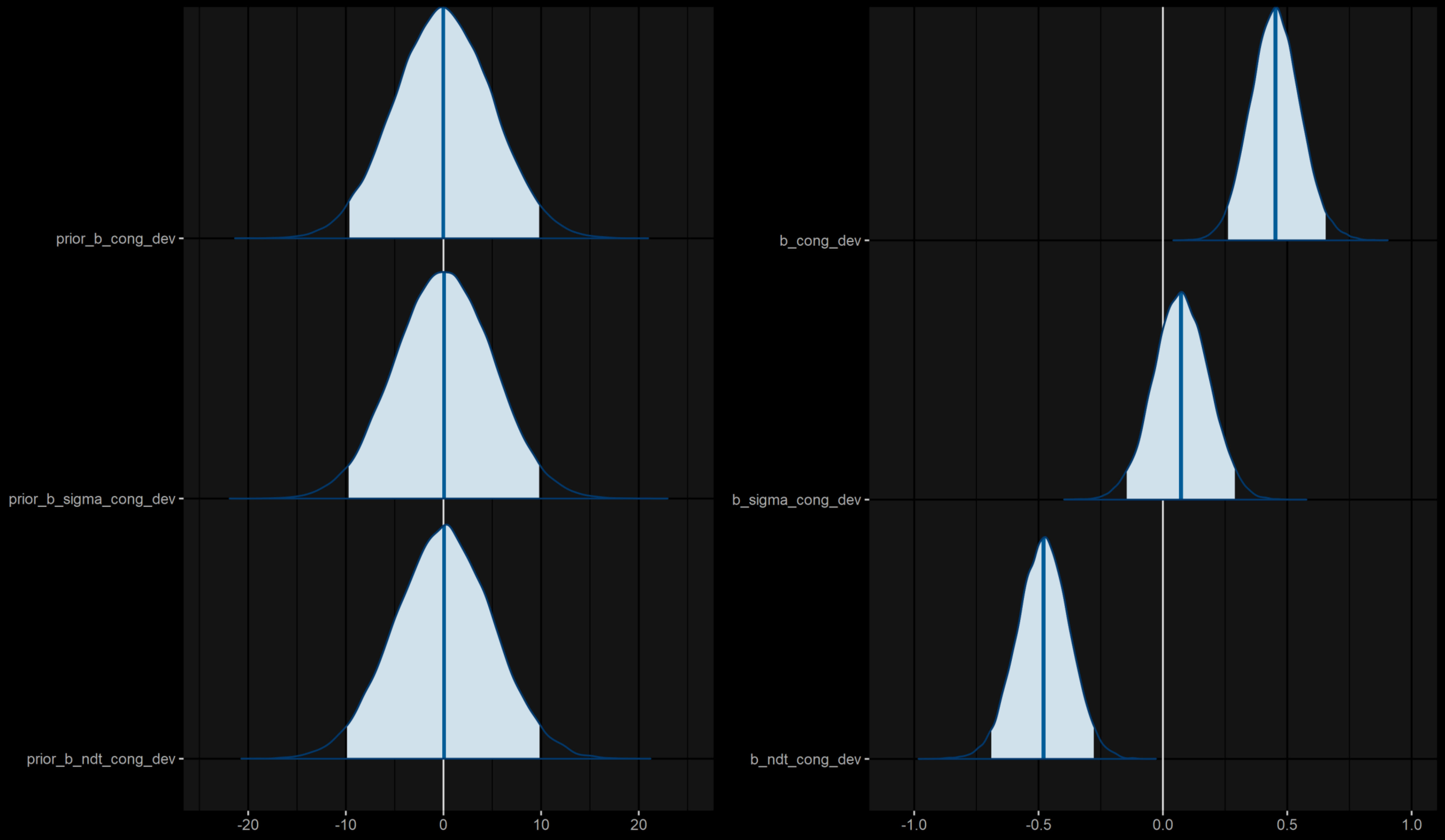


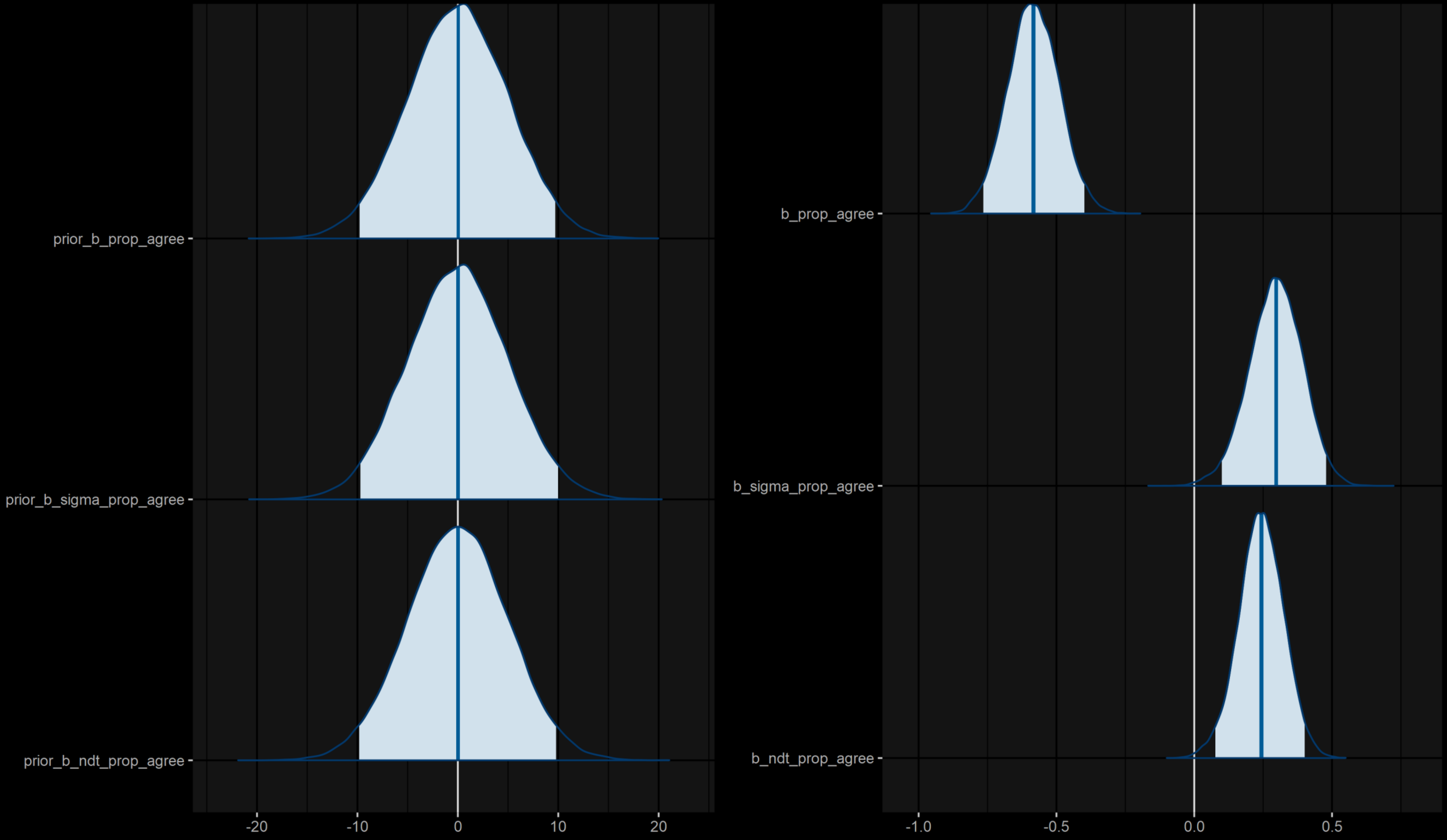
Bayesian Shifted Log-Normal Mixed Effects Model with Distributional Parameters

```
brms::bf(
      rt ~ Intercept + cong * pred +
            (cong * pred | subj) +
            (cong | image) +
            (1 | word),
      sigma ~ rt ~ Intercept + cong * pred +
            (cong * pred | subj) +
            (cong | image) +
            (1 | word),
      ndt ~ rt ~ Intercept + cong * pred +
            (cong * pred | subj) +
            (cong | image) +
            (1 | word)
)
```

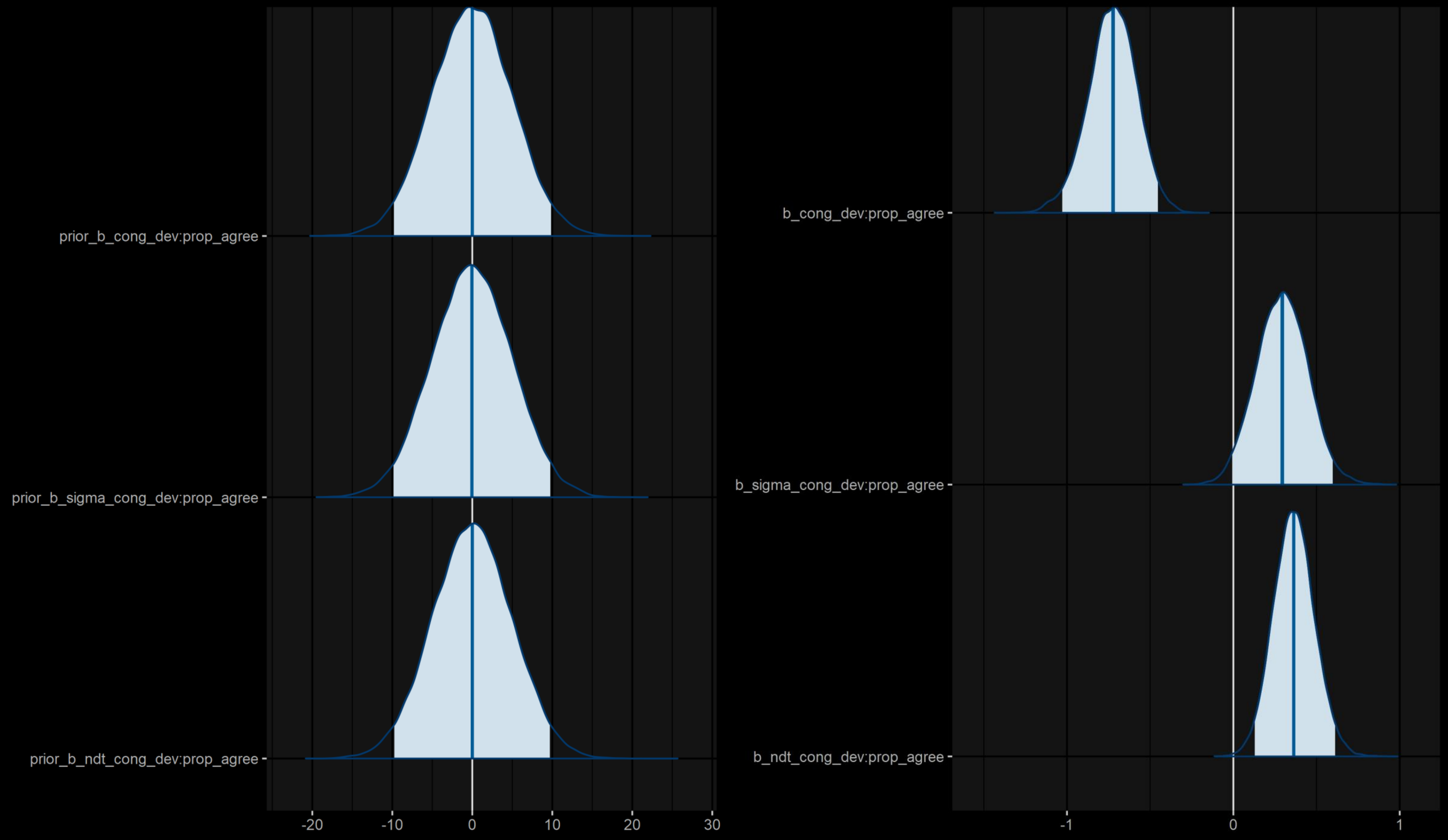


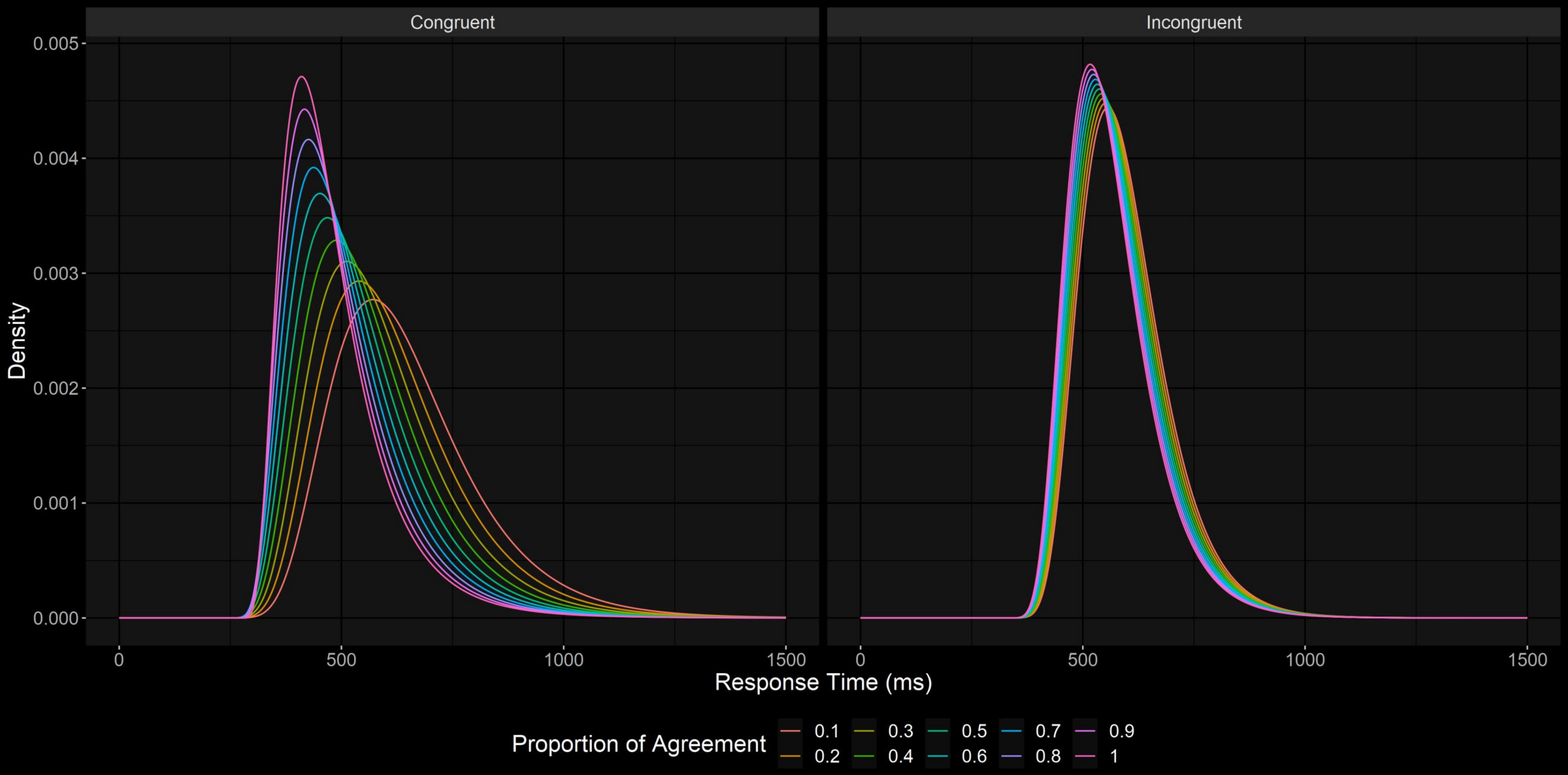
prior_b_Intercept
prior_b_sigma_Intercept
prior_b_ndt_Intercept
-30
0
30
b_Intercept
b_sigma_Intercept
b_ndt_Intercept
0.0
2.5
5.0

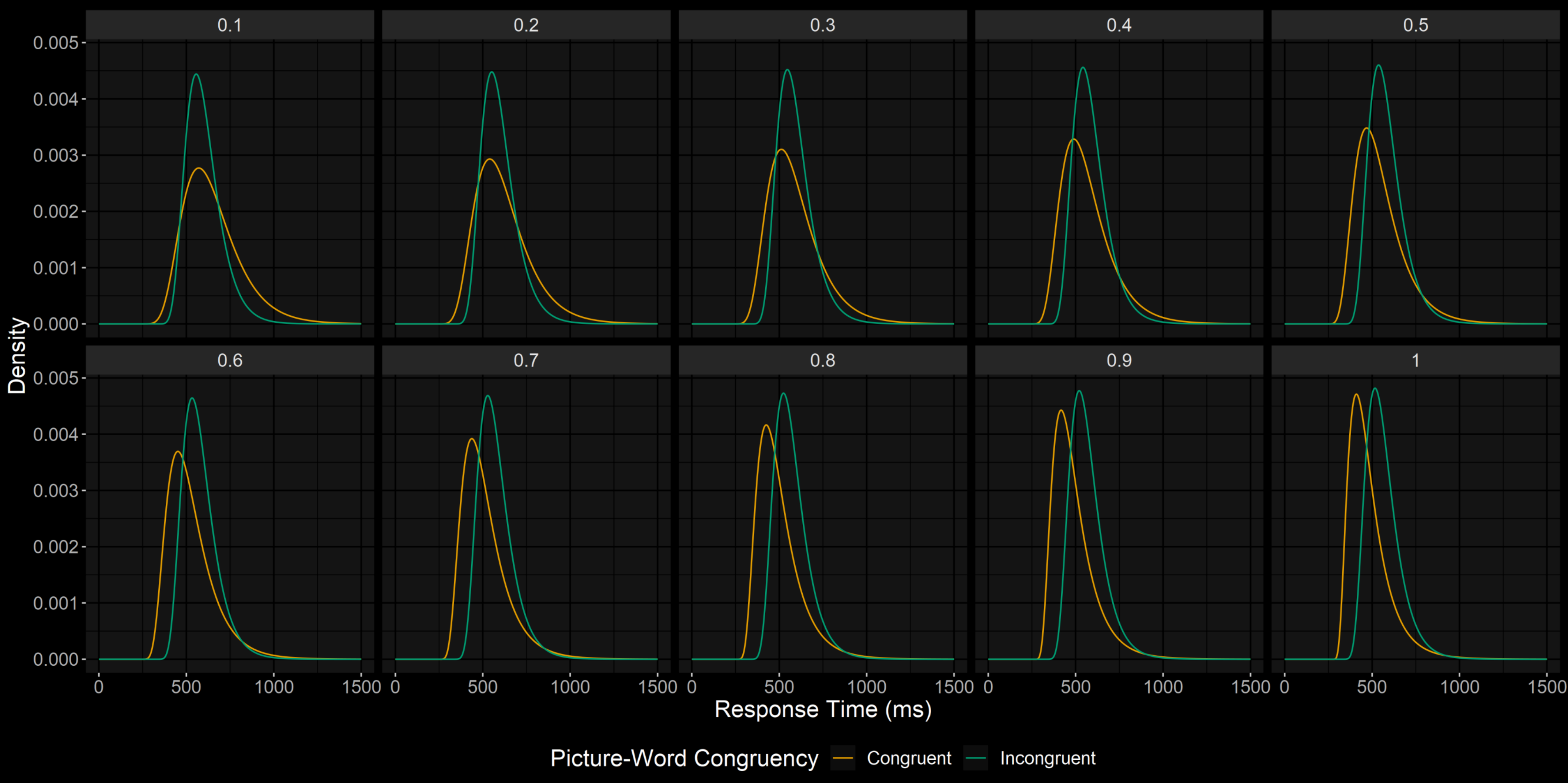

prior_b_cong_dev
prior_b_sigma_cong_dev
prior_b_ndt_cong_dev
-20
-10
0
10
20
b_cong_dev
b_sigma_cong_dev
b_ndt_cong_dev
-1.0
-0.5
0.0
0.5
1.0



prior_b_prop_agree
prior_b_sigma_prop_agree
prior_b_ndt_prop_agree
-20
-10
0
10
20
b_prop_agree
b_sigma_prop_agree
b_ndt_prop_agree
-1.0
-0.5
0.0
0.5


prior_b_cong_dev:prop_agree
prior_b_sigma_cong_dev:prop_agree
prior_b_ndt_cong_dev:prop_agree
-20
-10
0
10
20
30
b_cong_dev:prop_agree
b_sigma_cong_dev:prop_agree
b_ndt_cong_dev:prop_agree
-1
0
1


Congruent
Incongruent
0.005
0.004
0.003
0.002
0.001
0.000
Density
0
500
1000
1500
Response Time (ms)
Proportion of Agreement
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



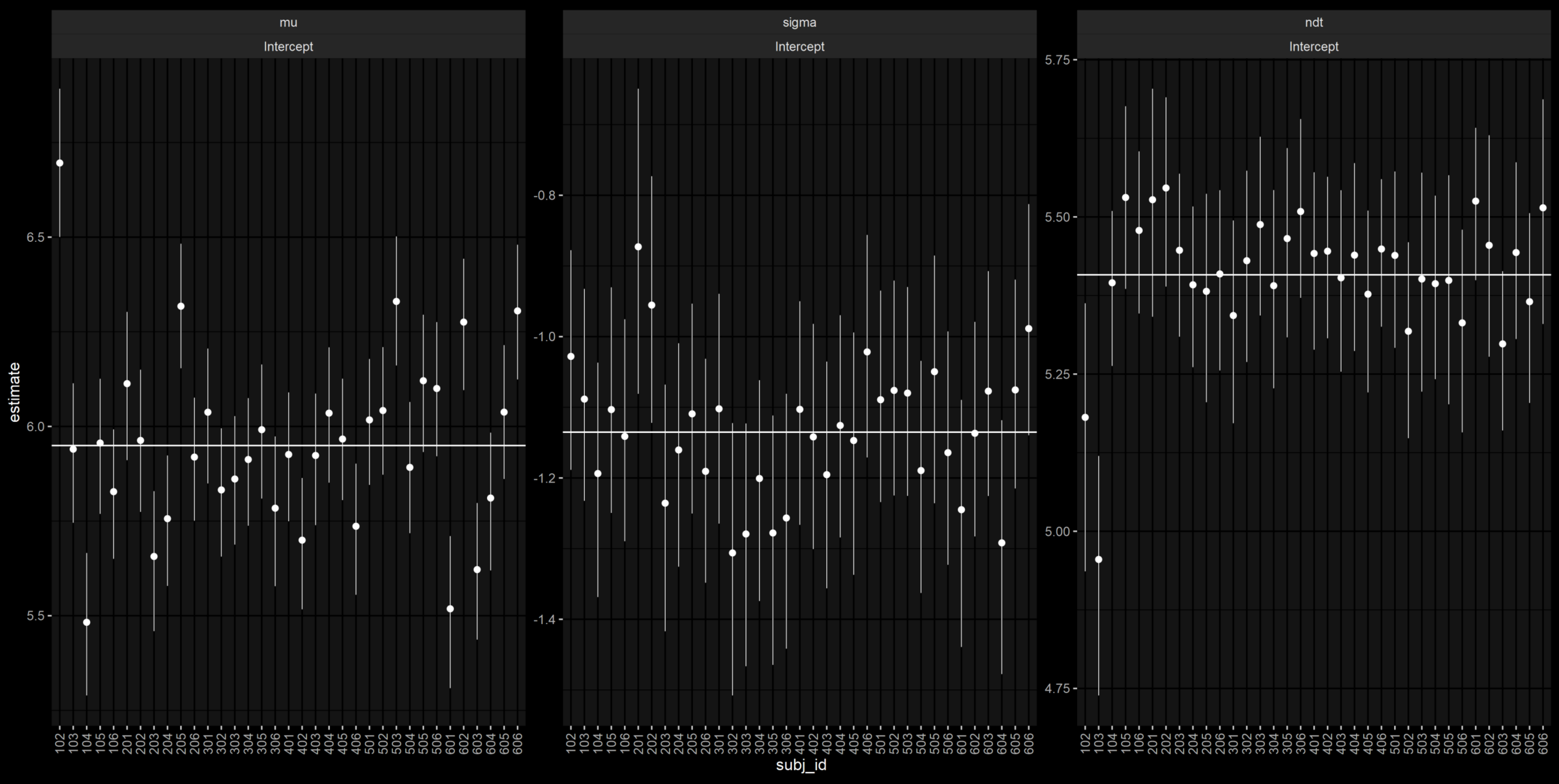
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Density
0.000
0.001
0.002
0.003
0.004
0.005
0
500
1000
1500
Response Time (ms)
Picture-Word Congruency
Congruent
Incongruent

Bayesian Results – Random Effects

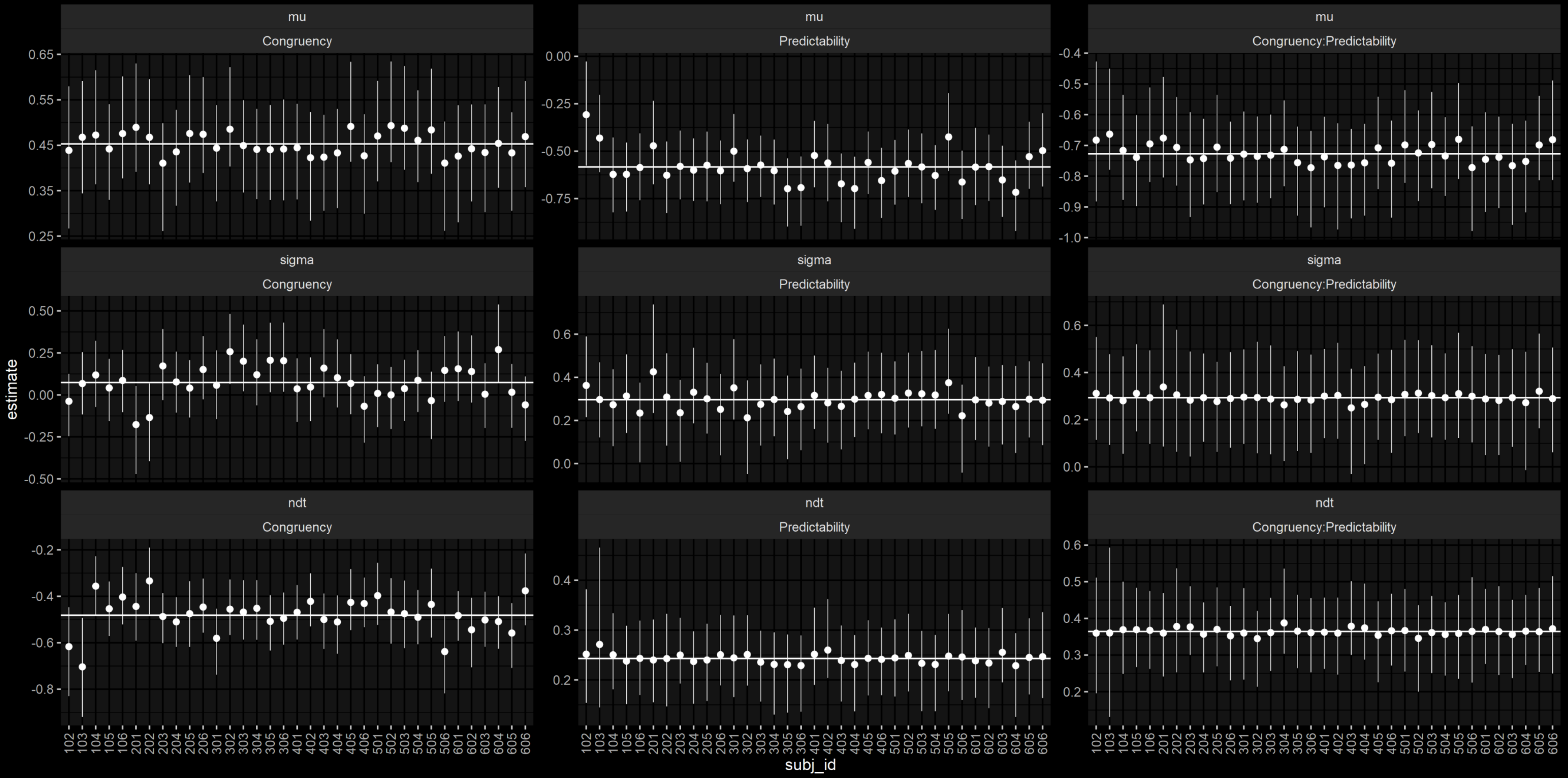
```
ranef(m_bme)
```

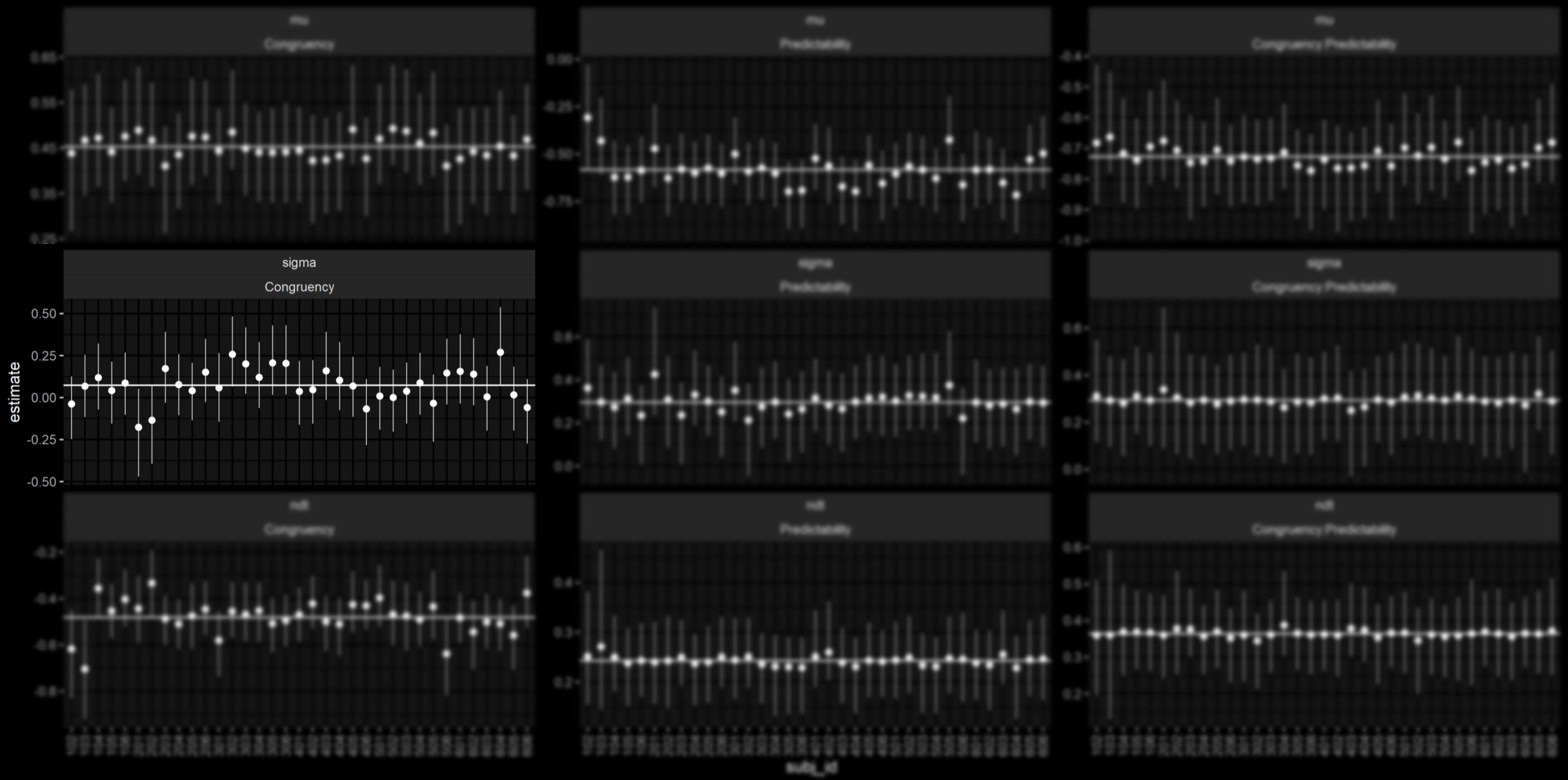
ranef(m_bme)	list [3]	List of length 3
image	double [200 x 4 x 6]	0.068645 0.105955 -0.043636 0.206602 -0.007710 0.023579 0.060256 0.059649 ...
string	double [400 x 4 x 3]	0.003315 0.011355 0.017058 0.006073 0.004984 -0.004099 0.030242 0.030648 ...
subj_id	double [35 x 4 x 12]	0.746325 -0.009589 -0.466826 0.006702 -0.121838 0.163541 0.099576 0.095293 ...

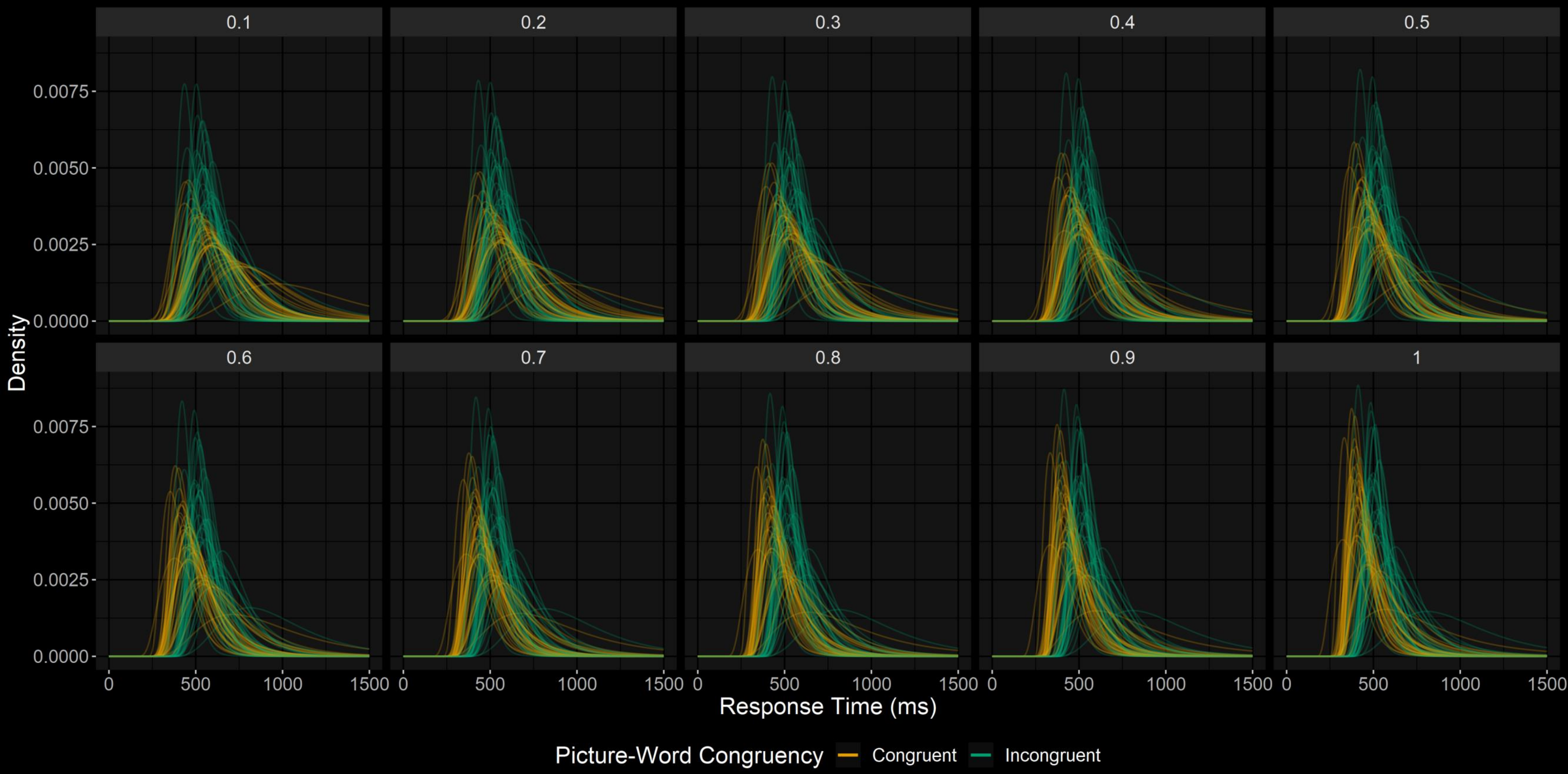
ID (e.g. subj_01, subj_02...) * value (est, err, Q2.5, Q97.5) * fixed parameter



mu
sigma
ndt
Intercept
estimate
subj_id
6.5
6.0
5.5
-0.8
-1.0
-1.2
-1.4
5.75
5.50
5.25
5.00
4.75
102 103 104 105 106 201 202 203 204 205 206 301 302 303 304 305 306 401 402 403 404 405 406 501 502 503 504 505 506 601 602 603 604 605 606


mu
Congruency
Predictability
Congruency:Predictability
sigma
ndt
estimate
subj_id


sigma
Congruency
0.50
0.25
0.00
-0.25
-0.50
estimate



0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.0075
0.0050
0.0025
0.0000
Density
0
500
1000
1500
Response Time (ms)
Picture-Word Congruency
Congruent
Incongruent

Caveats

- Computationally intensive if using non-informative priors for complex hierarchical formulae
- Have to avoid temptation to try over-infer about mechanisms unless using more cognitively informed models (e.g. drift diffusion)

Summary

Hierarchical models with maximal structures for distributional parameters are a robust and appropriate way of looking at or accounting for subject/item/etc variability in fixed effects when you’re interested in more than central tendency.

But, if you *can* assume no systematic differences in distributional parameters, GLMMs will suffice (and save you a lot of time and effort)!