

# Pilot Analysis: Building a Logistic Regression Model of Decisions

In this document, we build a logistic regression model of intervention decisions. This model is basically a psychometric function which estimates two parameters: 1. The *point of subjective equality* (PSE): The level of evidence at which users see the team as having equal utility with or without the new player. This parameter reflects bias in decision-making, either toward or away from intervening. 2. The *just-noticeable difference* (JND): The amount of evidence necessary to increase the user's likelihood of intervening from 50% (at the PSE) to 75% (an arbitrary performance threshold). JNDs are inversely proportional to the slope of the linear model. They reflect the sensitivity of the user to evidence such that smaller JNDs suggest that users evaluate prospects more consistently and larger JNDs suggest a higher degree of noise in the perception of utility.

## Load and Prepare Data

We load worker responses from our pilot and do some preprocessing.

```
# read in data
full_df <- read_csv("pilot-anonymous.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   workerId = col_character(),
##   batch = col_integer(),
##   n_trials = col_integer(),
##   n_data_conds = col_integer(),
##   condition = col_character(),
##   start_means = col_character(),
##   cutoff = col_integer(),
##   max_bonus = col_integer(),
##   numeracy = col_integer(),
##   gender = col_character(),
##   age = col_character(),
##   education = col_character(),
##   chart_use = col_character(),
##   difficulties = col_character(),
##   intervene = col_integer(),
##   outcome = col_character(),
##   pSup = col_integer(),
##   sdDiff = col_integer(),
##   trial = col_character(),
##   trialIdx = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```

# preprocessing
responses_df <- full_df %>%
  rename( # rename to convert away from camel case
    worker_id = workerId,
    ground_truth = groundTruth,
    sd_diff = sdDiff,
    p_award_with = pAwardWith,
    p_award_without = pAwardWithout,
    account_value = accountValue,
    p_superiority = pSup,
    start_time = startTime,
    resp_time = respTime,
    trial_dur = trialDur,
    trial_idx = trialIdx
  ) %>%
  # remove practice and mock trials from responses dataframe, leave in full version
  filter(trial_idx != "practice", trial_idx != "mock") %>%
  # mutate rows where intervene == -1 for some reason
  mutate(
    intervene = if_else(intervene == -1,
      # repair
      if_else((payoff == (award_value - 1) | payoff == -1),
        1, # payed for intervention
        0), # didn't pay for intervention
      # don't repair
      as.numeric(intervene) # hack to avoid type error
    )
  ) %>%
  # set up factors for modeling
  mutate(
    # add a variable to note whether the chart they viewed showed means
    means = as.factor((start_means == "True" & as.numeric(trial) <= (n_trials / 2)) | (start_means == "False" & as.numeric(trial) > (n_trials / 2))),
    start_means = as.factor(start_means == "True"),
    sd_diff = as.factor(sd_diff),
    trial_number = as.numeric(trial)
  )
)

head(responses_df)

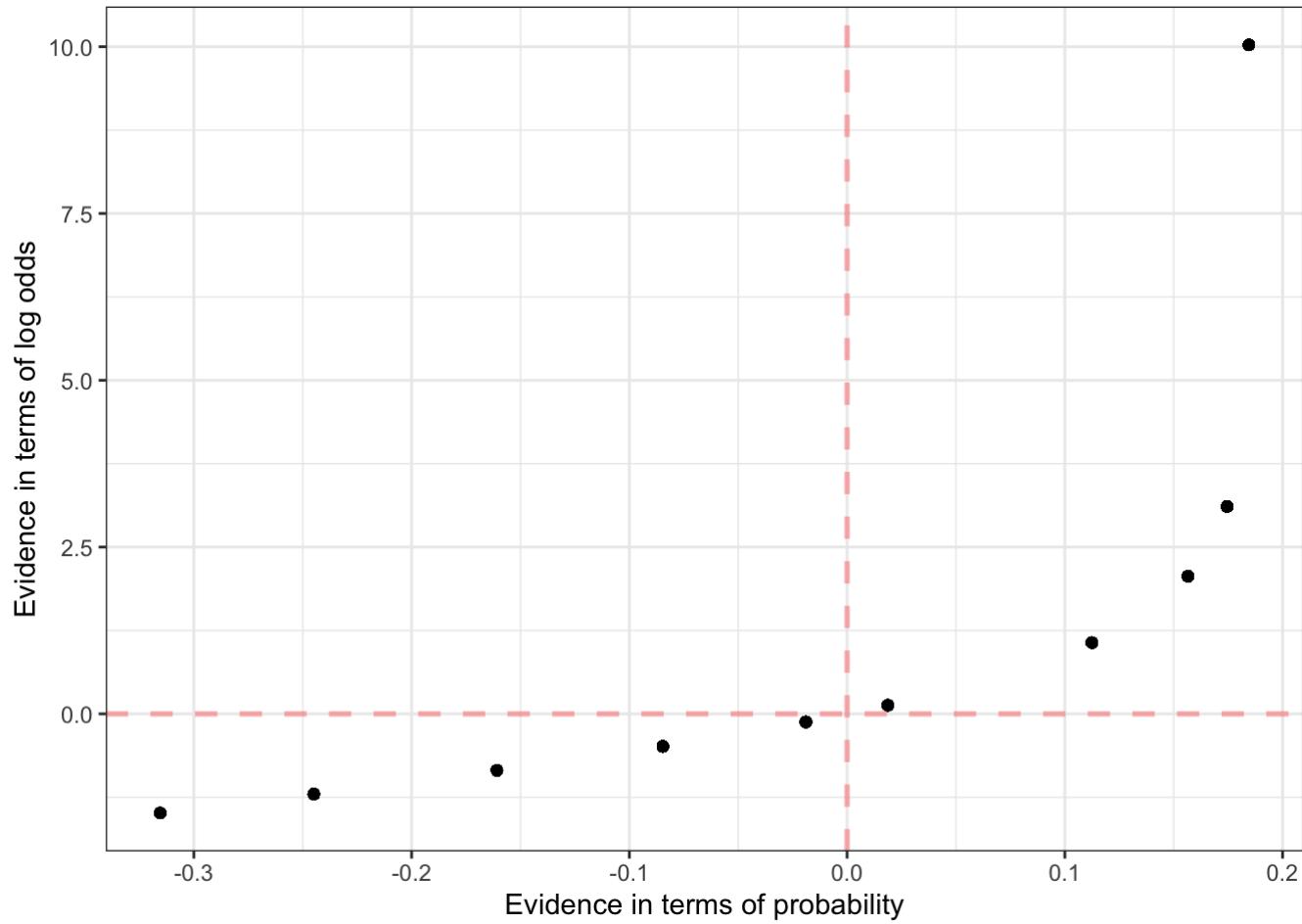
```

```
## # A tibble: 6 x 37
##   worker_id batch n_trials n_data_cons condition baseline es_threshold
##   <chr>      <int>     <int>       <int> <chr>        <dbl>      <dbl>
## 1 290ad208     2        34         18 HOPS        0.5       0.9
## 2 290ad208     2        34         18 HOPS        0.5       0.9
## 3 290ad208     2        34         18 HOPS        0.5       0.9
## 4 290ad208     2        34         18 HOPS        0.5       0.9
## 5 290ad208     2        34         18 HOPS        0.5       0.9
## 6 290ad208     2        34         18 HOPS        0.5       0.9
## # ... with 30 more variables: start_means <fct>, award_value <dbl>,
## #   starting_value <dbl>, exchange <dbl>, cutoff <int>, max_bonus <int>,
## #   total_bonus <dbl>, duration <dbl>, numeracy <int>, gender <chr>,
## #   age <chr>, education <chr>, chart_use <chr>, difficulties <chr>,
## #   account_value <dbl>, ground_truth <dbl>, intervene <dbl>,
## #   outcome <chr>, p_award_with <dbl>, p_award_without <dbl>,
## #   p_superiority <int>, payoff <dbl>, resp_time <dbl>, sd_diff <fct>,
## #   start_time <dbl>, trial <chr>, trial_dur <dbl>, trial_idx <chr>,
## #   means <fct>, trial_number <dbl>
```

We need the data in a format where it is prepared for modeling. This means we want to calculate a scale of evidence in favor of intervention. We calculate this by apply a log odds transform to our utility optimal decision rule, transforming our evidence from differences of probabilities into log odds units consistent with the idea that people perceive probabilities as log odds.

```
model_df <- responses_df %>%
  mutate(
    # evidence scale
    p_diff = p_award_with - (p_award_without + (1 / award_value)),
    evidence = qlogis(p_award_with) - qlogis(p_award_without + (1 / award_value)),
    # scale and center trial order
    trial = (trial_number - as.numeric(n_trials) / 2) / as.numeric(n_trials)
  )

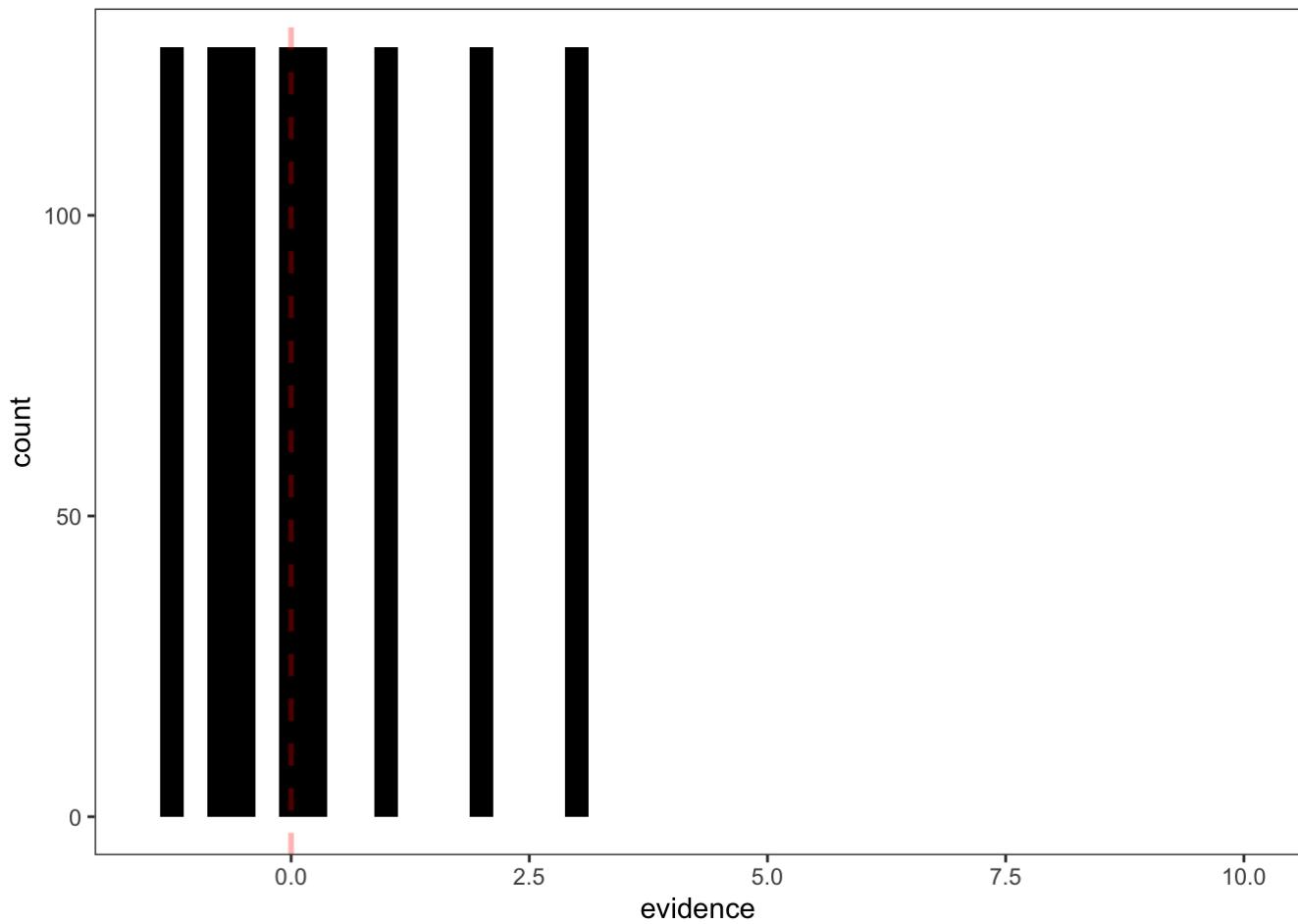
model_df %>%
  ggplot(aes(p_diff, evidence)) +
  geom_point() +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
  # utility optimal decision rule
  geom_hline(yintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
  theme_bw() +
  labs(
    x = "Evidence in terms of probability",
    y = "Evidence in terms of log odds"
  )
```



Let's look at the distribution of levels of evidence sampled on this scale. It should look approximately uniform.

```
model_df %>% ggplot(aes(x = evidence)) +
  geom_histogram(fill = "black", binwidth = 0.25) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
  xlim(quantile(model_df$evidence, c(0, 1))) +
  theme_bw() +
  theme(panel.grid = element_blank())
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Now, lets apply our exclusion criteria.

```

# determine exclusions
exclude_df <- model_df %>%
  # attention check trials where ground truth = c(0.5, 0.999)
  mutate(failed_check = (ground_truth == 0.5 & intervene != 0) | (ground_truth == 0.999
  & intervene != 1)) %>%
  group_by(worker_id) %>%
  summarise(
    failed_attention_checks = sum(failed_check),
    exclude = failed_attention_checks > 0
    # p_sup_less_than_50 = sum(p_superiority < 50) / n(),
    # exclude = (failed_attention_checks > 0 | p_sup_less_than_50 > 0.3)
  ) %>%
  select(worker_id, exclude)

# apply exclusion criteria and remove attention checks to modeling data set
model_df <- model_df %>%
  left_join(exclude_df, by = "worker_id") %>%
  filter(exclude == FALSE) %>%
  filter(ground_truth > 0.5 & ground_truth < 0.999)

# how many remaining workers per condition
model_df %>%
  group_by(condition, start_means) %>% # between subject manipulations
  summarise(
    n_workers = length(unique(worker_id))
  )

```

```

## # A tibble: 4 x 3
## # Groups:   condition [2]
##   condition start_means n_workers
##   <chr>     <fct>          <int>
## 1 HOPs      FALSE           3
## 2 HOPs      TRUE            8
## 3 intervals FALSE           7
## 4 intervals TRUE            6

```

The table above shows the number of included workers in each between-subjects condition.

## Distribution of Decisions

We start as simply as possible by just modeling the distribution of decisions using a logit link function and an intercept model. Here we are just estimating the mean probability of intervening.

Before we fit the model to our data, let's check that our priors seem reasonable. We'll set a pretty wide prior on the intercept parameter and locate it at zero since this reflects a weakly informative expectation of no bias.

```
# get_prior(data = model_df,
#           family = bernoulli(link = "logit"),
#           formula = bf(intervene ~ 1))

# starting as simple as possible: learn the distribution of decisions
prior.logistic_intercept <- brm(data = model_df, family = bernoulli(link = "logit"),
                                  formula = bf(intervene ~ 1),
                                  prior = c(prior(normal(0, 1), class = Intercept)),
                                  sample_prior = "only",
                                  iter = 3000, warmup = 500, chains = 2, cores = 2)
```

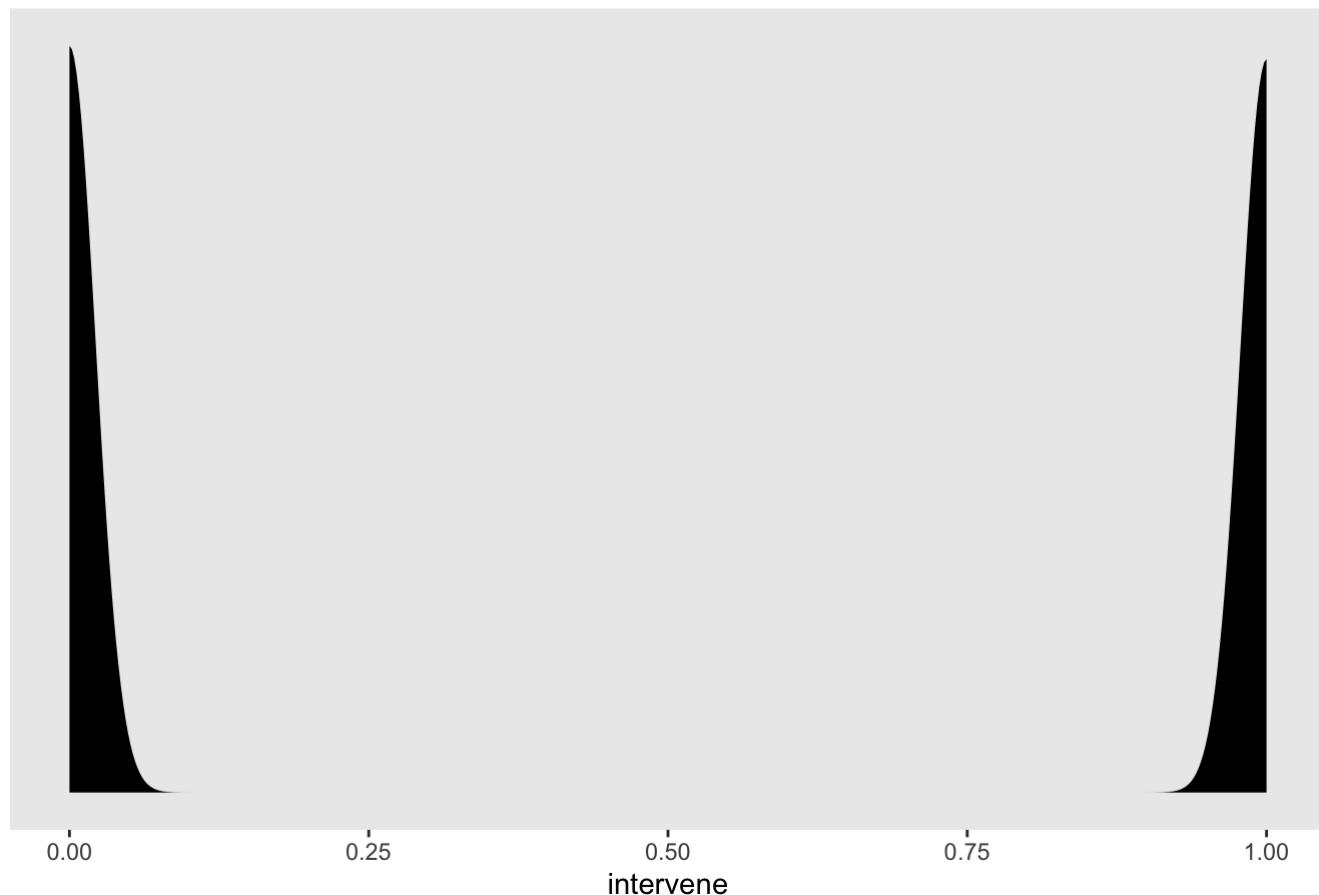
```
## Compiling the C++ model
```

```
## Start sampling
```

Let's look at our prior predictive distribution. This should just be an even split between 0 and 1.

```
# prior predictive check
model_df %>%
  select(-intervene) %>%
  add_predicted_draws(prior.logistic_intercept, prediction = "intervene", seed = 1234) %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Prior predictive distribution for intervention decisions") +
  theme(panel.grid = element_blank())
```

### Prior predictive distribution for intervention decisions



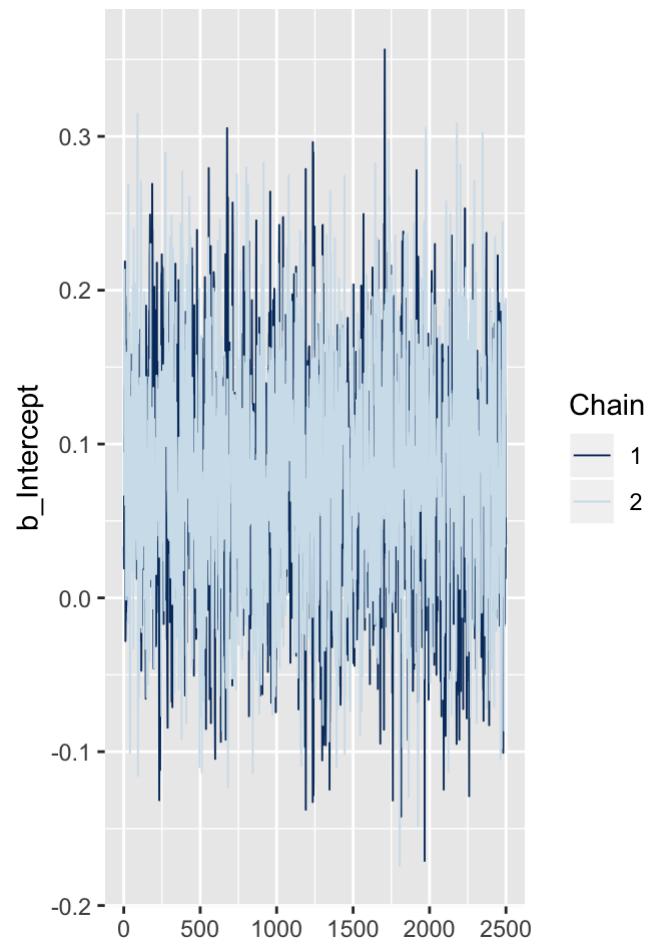
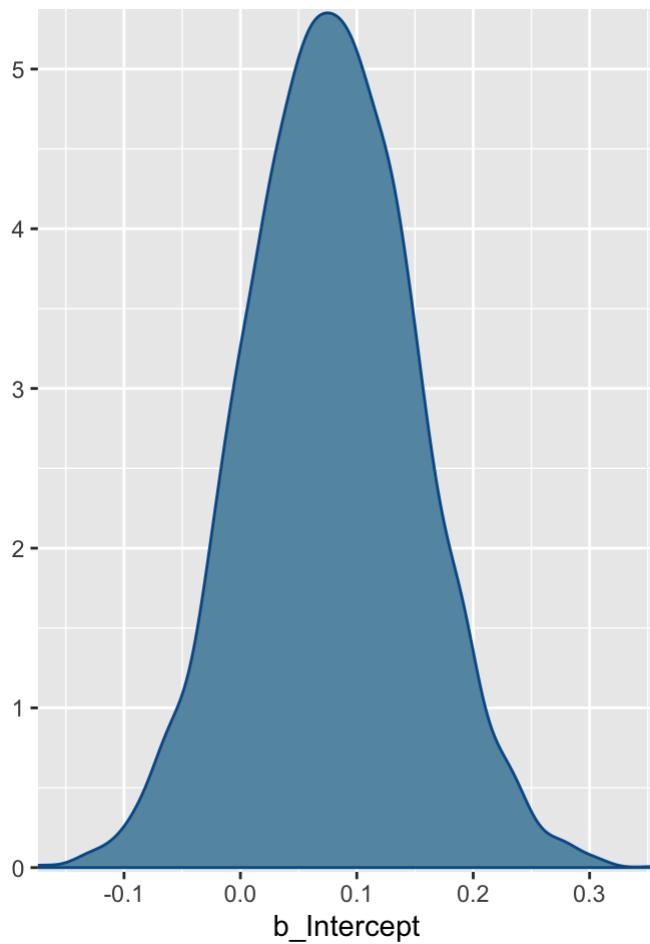
Now, let's fit the model to our data.

```
# starting as simple as possible: learn the distribution of decisions
m.logistic_intercept <- brm(data = model_df, family = bernoulli(link = "logit"),
  formula = bf(intervene ~ 1),
  prior = c(prior(normal(0, 1), class = Intercept)),
  iter = 3000, warmup = 500, chains = 2, cores = 2,
  file = "model-fits/logistic_intercept_mdl")
```

Check diagnostics:

- Trace plots

```
# trace plots
plot(m.logistic_intercept)
```



- Summary

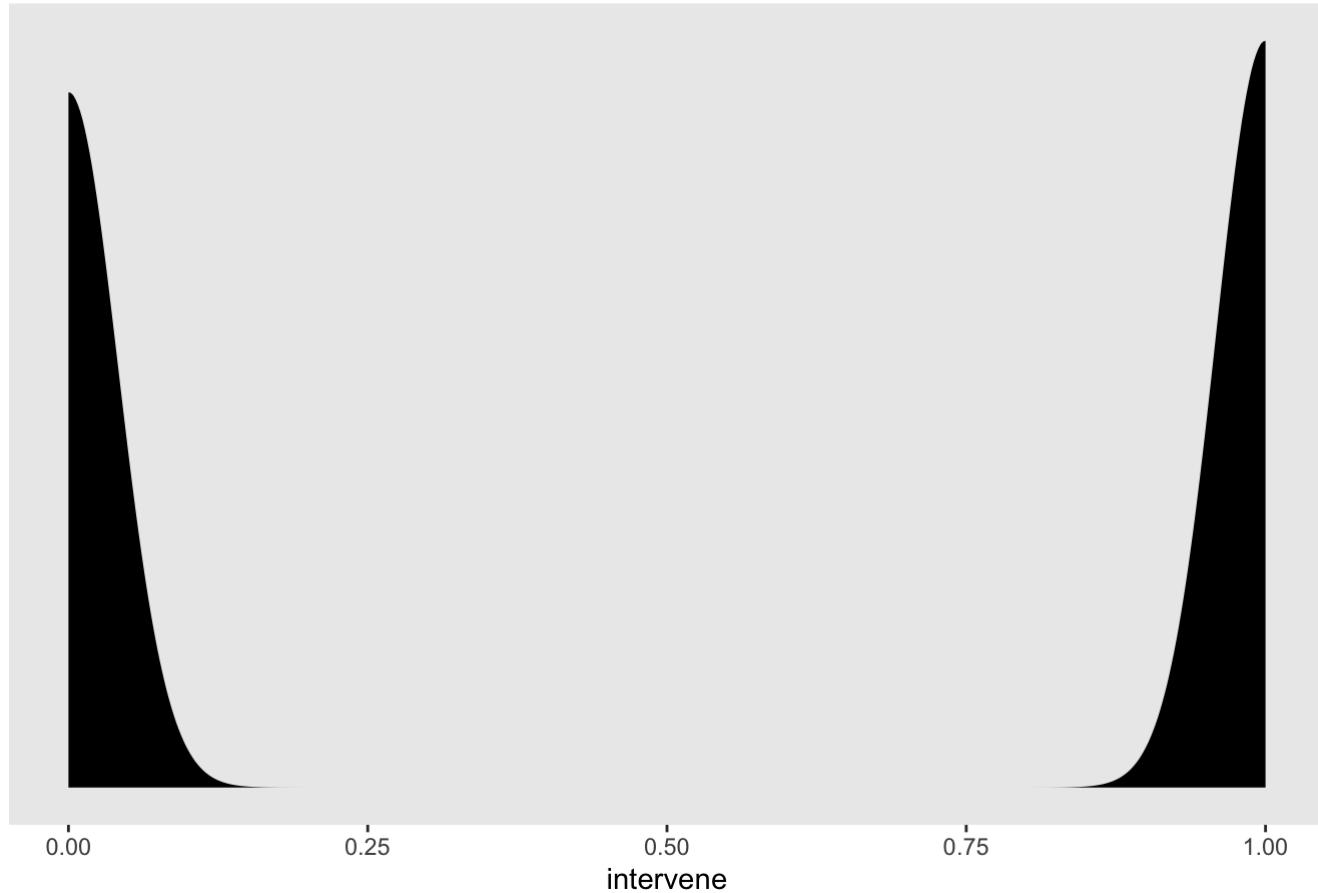
```
# model summary
print(m.logistic_intercept)
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ 1
## Data: model_df (Number of observations: 768)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept     0.08      0.07    -0.06     0.22       1954  1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Let's check out a posterior predictive distribution for intervention decisions.

```
# posterior predictive check
model_df %>%
  select(-intervene) %>% # this model should not be sensitive to evidence
  add_predicted_draws(m.logistic_intercept, prediction = "intervene", seed = 1234, n = 2
00) %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior predictive distribution for intervention") +
  theme(panel.grid = element_blank())
```

Posterior predictive distribution for intervention

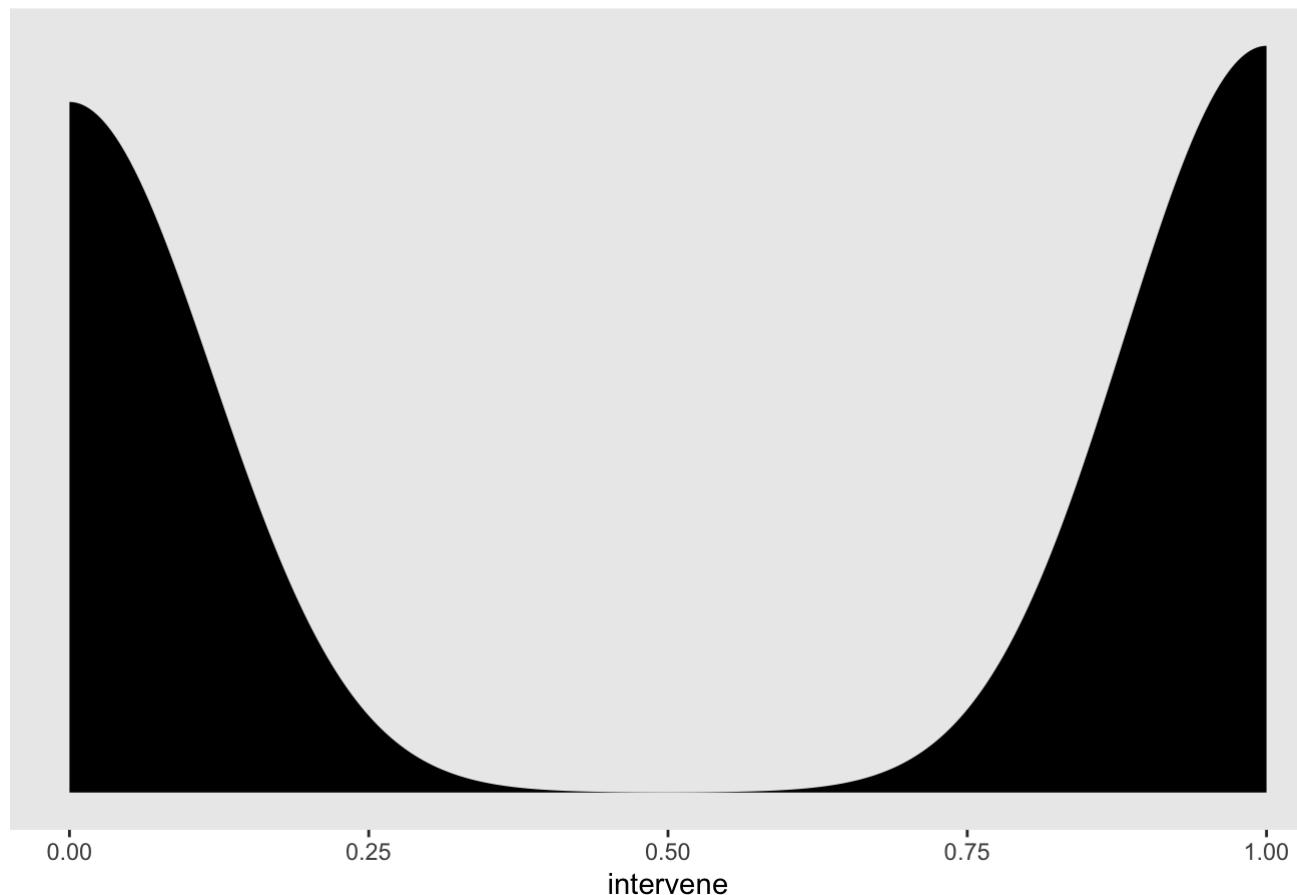


The posterior predictive distribution is about what we'd expect. The slight bias toward intervening is consistent with a positive intercept parameter.

How do the posterior predictions compare to the observed data?

```
# data density
model_df %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Data distribution for intervention") +
  theme(panel.grid = element_blank())
```

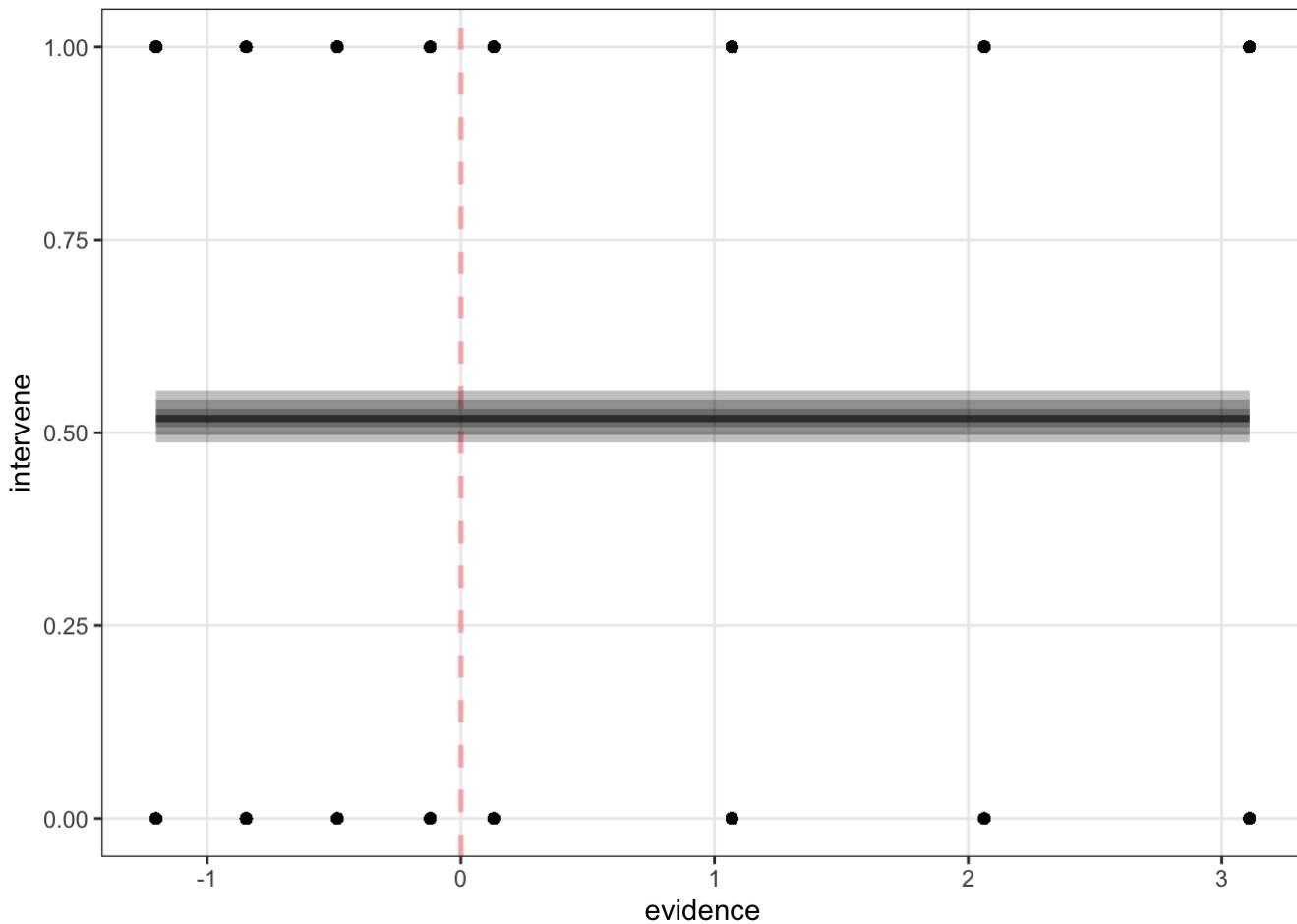
### Data distribution for intervention



Since these look pretty good, even for an intercept model, it seems like other predictive checks will be a better indicator of fit going forward.

Let's take a look at the estimated psychometric function. This should not have any slope. It is just an estimate of what proportion of the time people intervene.

```
model_df %>%
  select(evidence, intervene) %>%
  add_fitted_draws(m.logistic_intercept, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene)) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
#  geom_line(aes(y = pf, group = .draw)) +
  stat_lineribbon(aes(y = pf), .width = c(.95, .80, .50), alpha = .25, fill = "black") +
  geom_point(alpha = .15, color = "black") +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  theme_bw() +
  theme(panel.grid.minor = element_blank())
```



## Linear Model with Logit Link Function

Now we'll add a slope parameter to our model to make it a simple linear model where decisions to intervene are a function of the probability of getting the award with the new player.

Before we fit our model let's check that our priors seem reasonable. We'll keep the same weak prior on our intercept parameter, and we'll set a similarly weak prior on the sensitivity to evidence, centering our prior on a one-to-one correspondence between units of evidence and perceived evidence.

```
# get_prior(data = model_df, family = bernoulli(link = "logit"), formula = bf(intervene ~ 1 + evidence))

# simple logistic regression
prior.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                       formula = bf(intervene ~ 1 + evidence),
                       prior = c(prior(normal(0, 1), class = Intercept),
                                 prior(normal(1, 1), class = b)),
                       sample_prior = "only",
                       iter = 3000, warmup = 500, chains = 2, cores = 2)
```

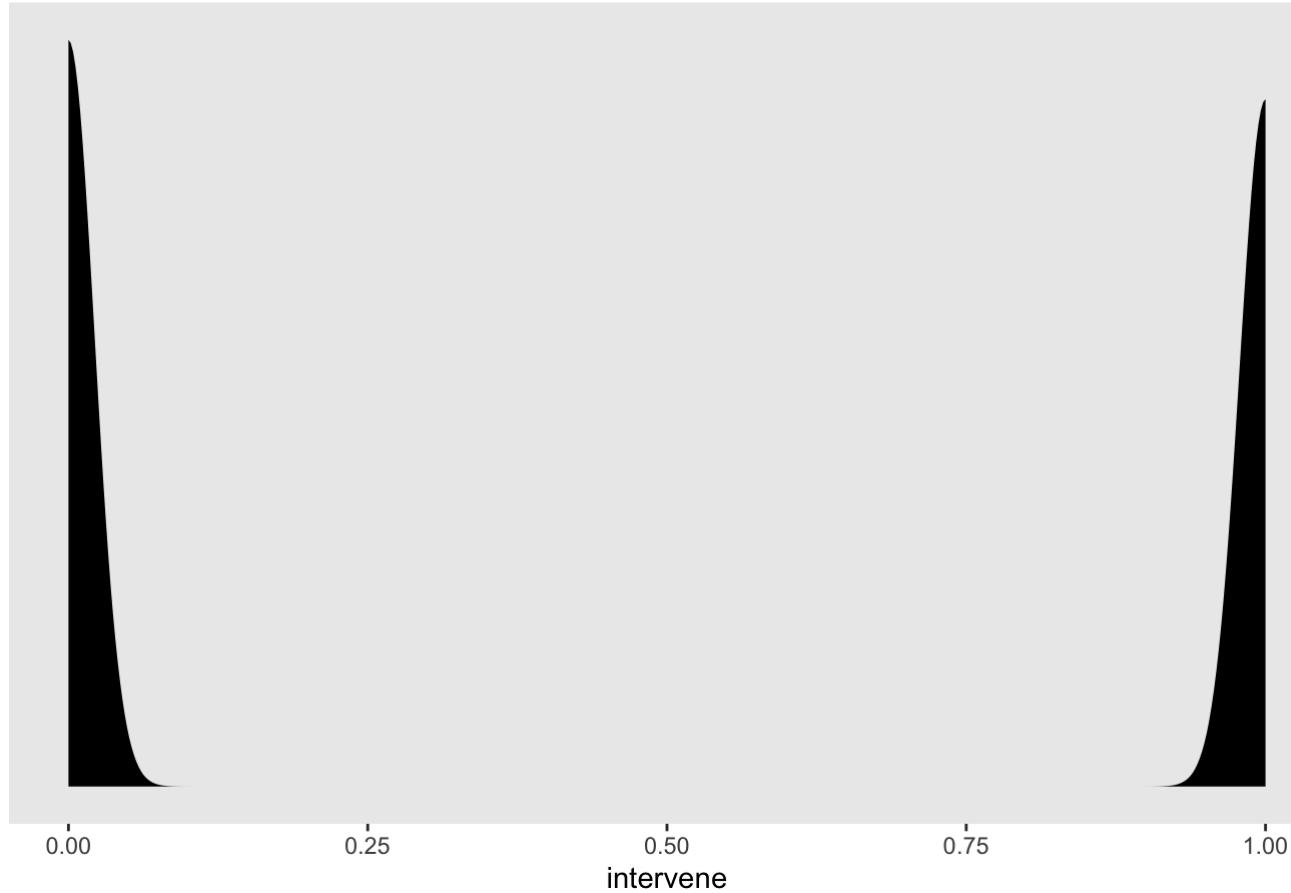
```
## Compiling the C++ model
```

```
## Start sampling
```

Let's look at our prior predictive distribution. This should still be an even split between 0 and 1, although now our model should be sensitive to evidence.

```
# prior predictive check
model_df %>%
  select(evidence) %>%
  add_predicted_draws(prior.logistic, prediction = "intervene", seed = 1234) %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Prior predictive distribution for intervention decisions") +
  theme(panel.grid = element_blank())
```

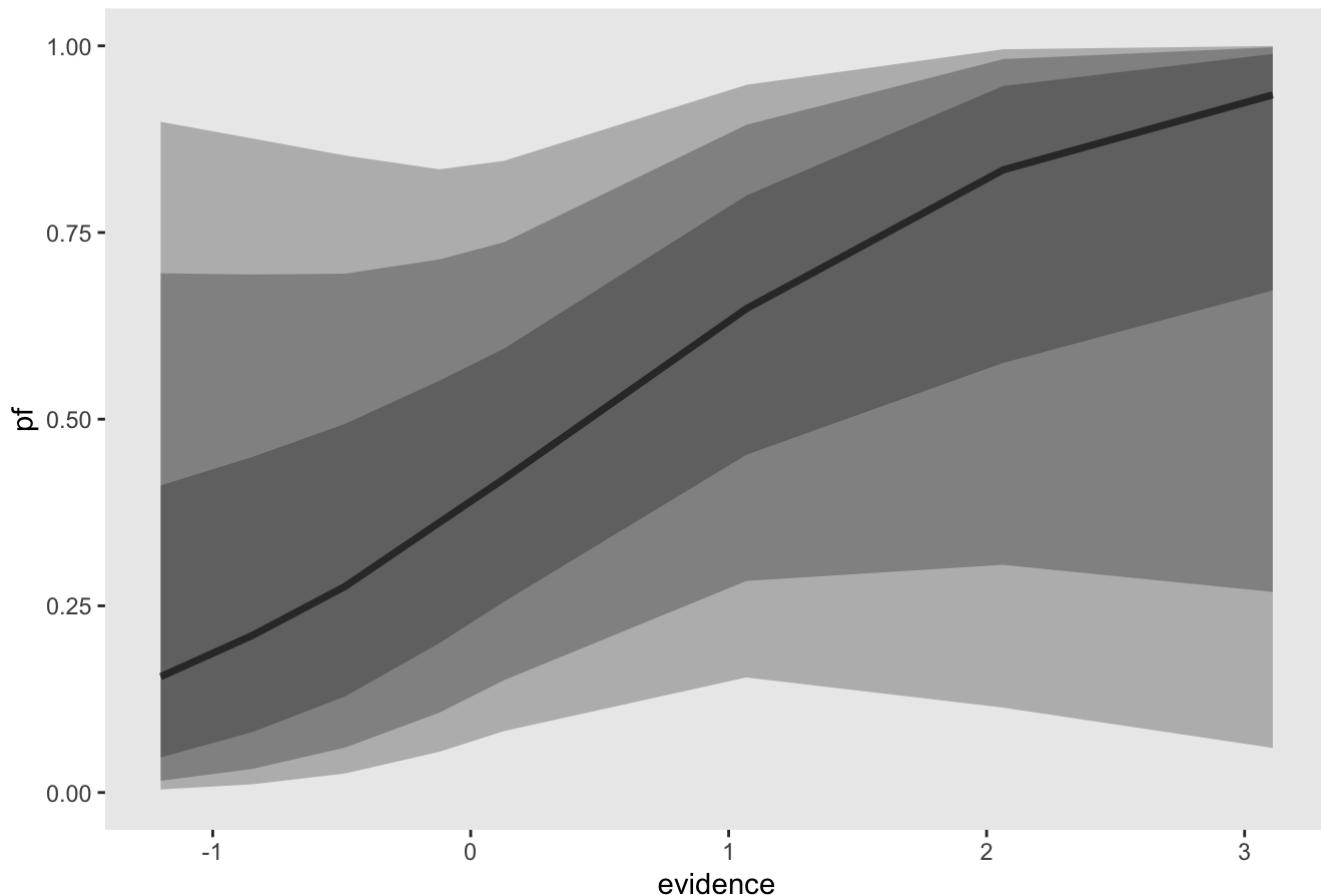
Prior predictive distribution for intervention decisions



Perhaps a better way of seeing this is the predicted fit. We'll look at this instead of prior predictions from here on.

```
# prior predictive check
model_df %>%
  select(evidence) %>%
  add_fitted_draws(prior.logistic, value = "pf", seed = 1234) %>%
  ggplot(aes(x = evidence, y = pf)) +
  stat_lineribbon(.width = c(.95, .80, .50), alpha = .25, fill = "black") +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  labs(subtitle = "Prior predictive PF fit") +
  theme(panel.grid = element_blank())
```

### Prior predictive PF fit



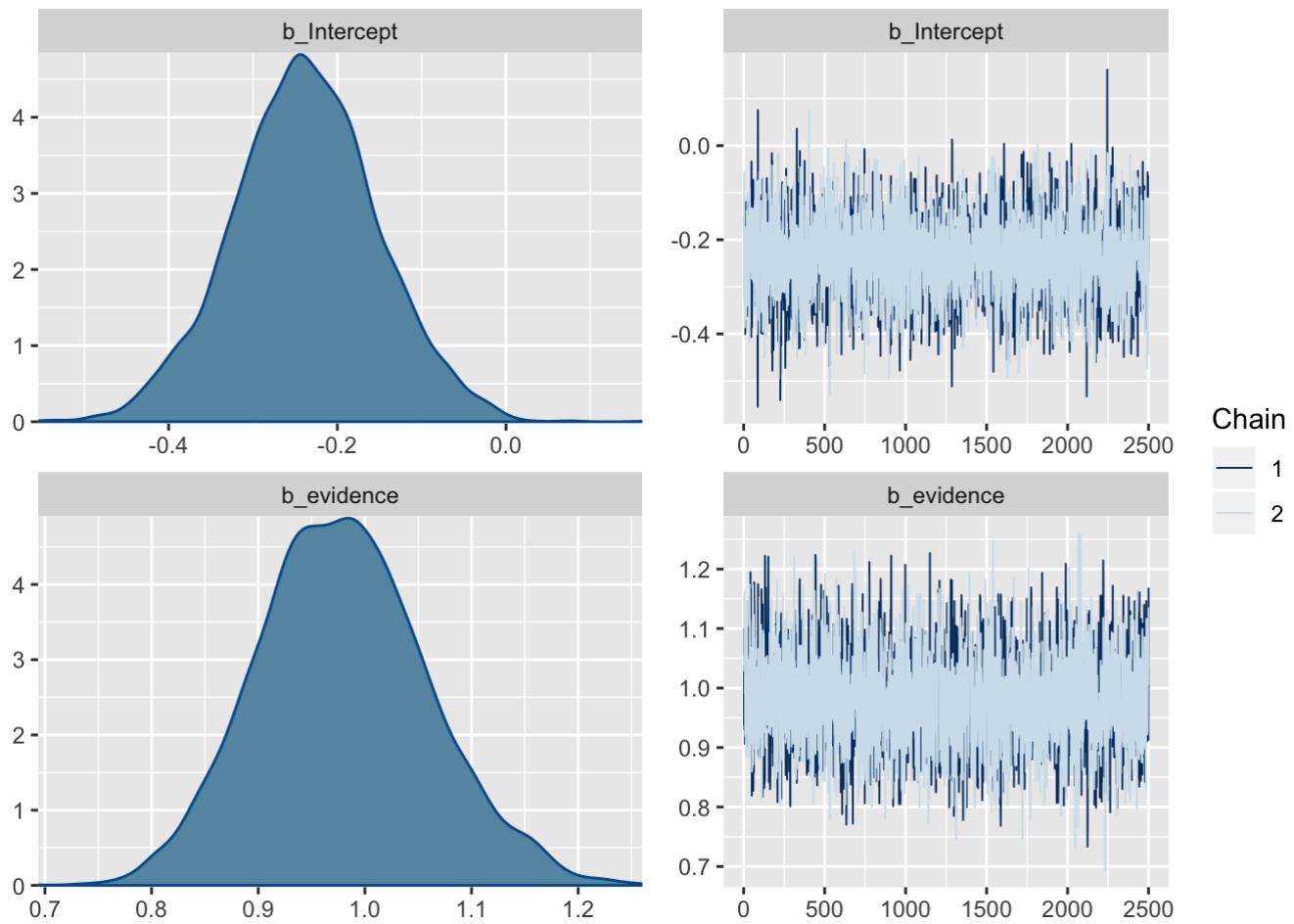
Now, let's fit the model.

```
# logistic regression
m.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                     formula = bf(intervene ~ 1 + evidence),
                     prior = c(prior(normal(0, 1), class = Intercept),
                               prior(normal(1, 1), class = b)),
                     iter = 3000, warmup = 500, chains = 2, cores = 2,
                     file = "model-fits/logistic_mdl")
```

Check diagnostics:

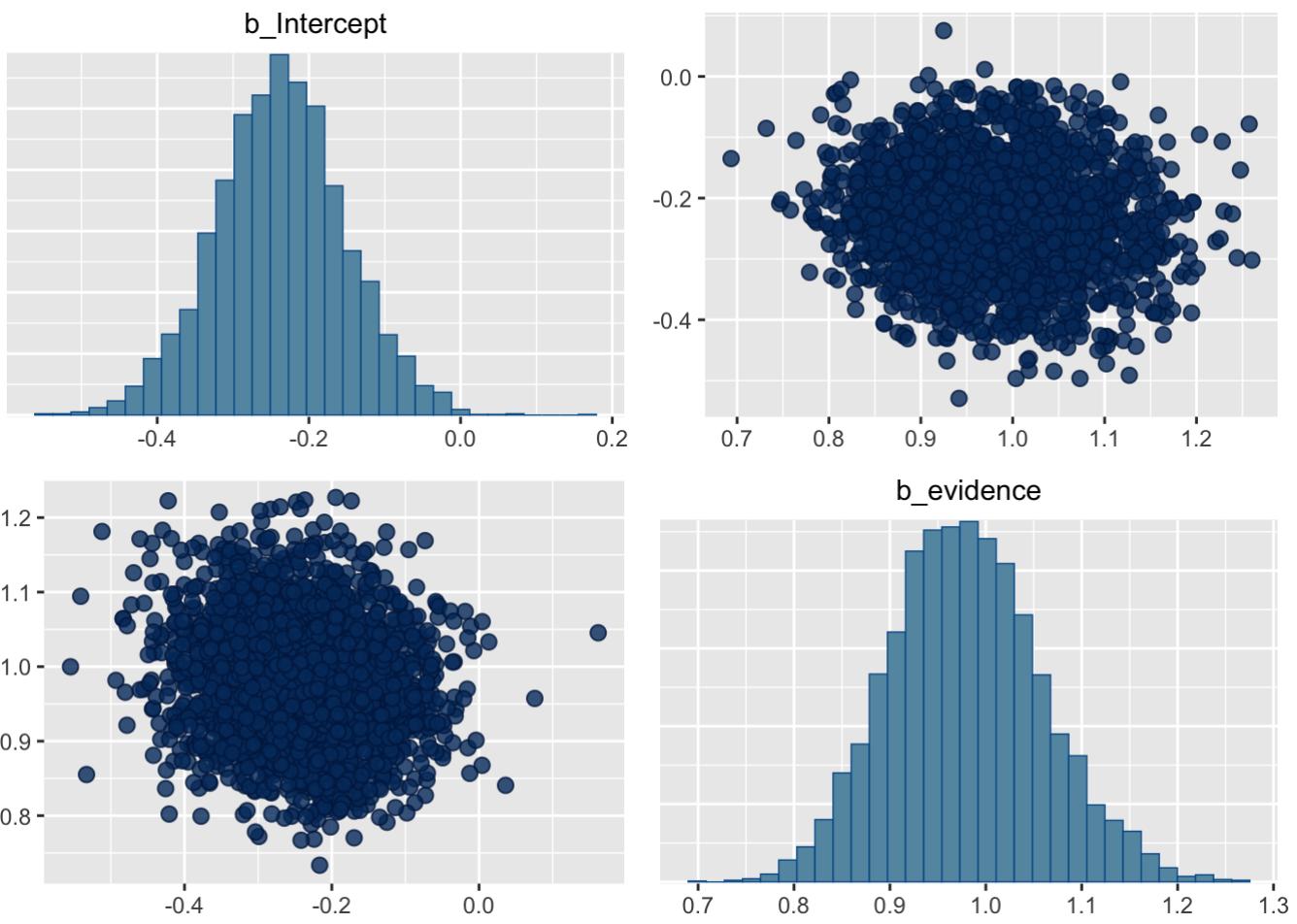
- Trace plots

```
# trace plots
plot(m.logistic)
```



- Pairs plot

```
# pairs plot
pairs(m.logistic)
```



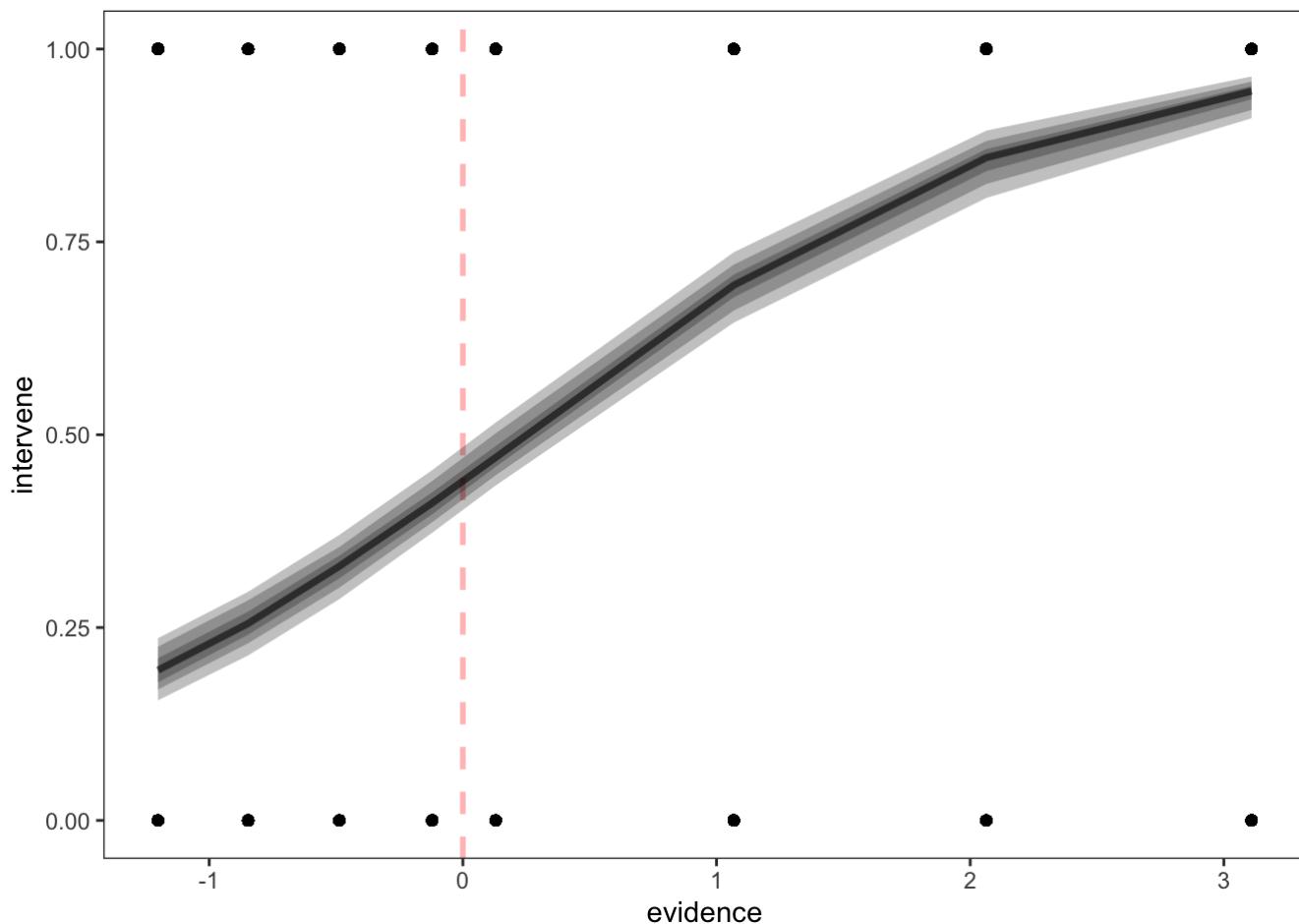
- Summary

```
# model summary
print(m.logistic)
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ 1 + evidence
## Data: model_df (Number of observations: 768)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept     -0.24      0.09    -0.41    -0.07       4605  1.00
## evidence       0.98      0.08     0.83     1.15       2792  1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Let's take a look at the estimated psychometric function.

```
model_df %>%
  add_fitted_draws(m.logistic, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene)) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
# geom_line(aes(y = pf, group = .draw)) +
  stat_lineribbon(aes(y = pf), .width = c(.95, .80, .50), alpha = .25, fill = "black") +
  geom_point(alpha = .15, color = "black") +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  theme_bw() +
  theme(panel.grid = element_blank())
```



## Add Hierarchy for Slopes and Intercepts

The models we've created thus far fail to account for much of the noise in the data. Here, we attempt to parse some heterogeneity in responses by modeling a random effect of worker on slopes and intercepts. This introduces a hierarchical component to our model in order to account for individual differences in the best fitting linear model for each worker's data.

Before we fit our model let's check that our priors seem reasonable. We add priors for the standard deviation of random slopes and intercepts per worker and their correlation. The priors for random effects are moderately weak, and the prior for their correlation is meant to avoid extreme correlations that could mess up the model fit.

```
# get_prior(data = model_df, family = bernoulli(link = "logit"), formula = bf(intervene
~ (1 + evidence/worker_id) + evidence))

# starting as simple as possible: learn the distribution of decisions
prior.wrkr.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                             formula = bf(intervene ~ (1 + evidence|worker_id) + evidenc
e),
                             prior = c(prior(normal(0, 1), class = Intercept),
                                       prior(normal(1, 1), class = b),
                                       prior(normal(0, 0.5), class = sd),
                                       prior(lkj(4), class = cor)),
                             sample_prior = "only",
                             iter = 3000, warmup = 500, chains = 2, cores = 2)
```

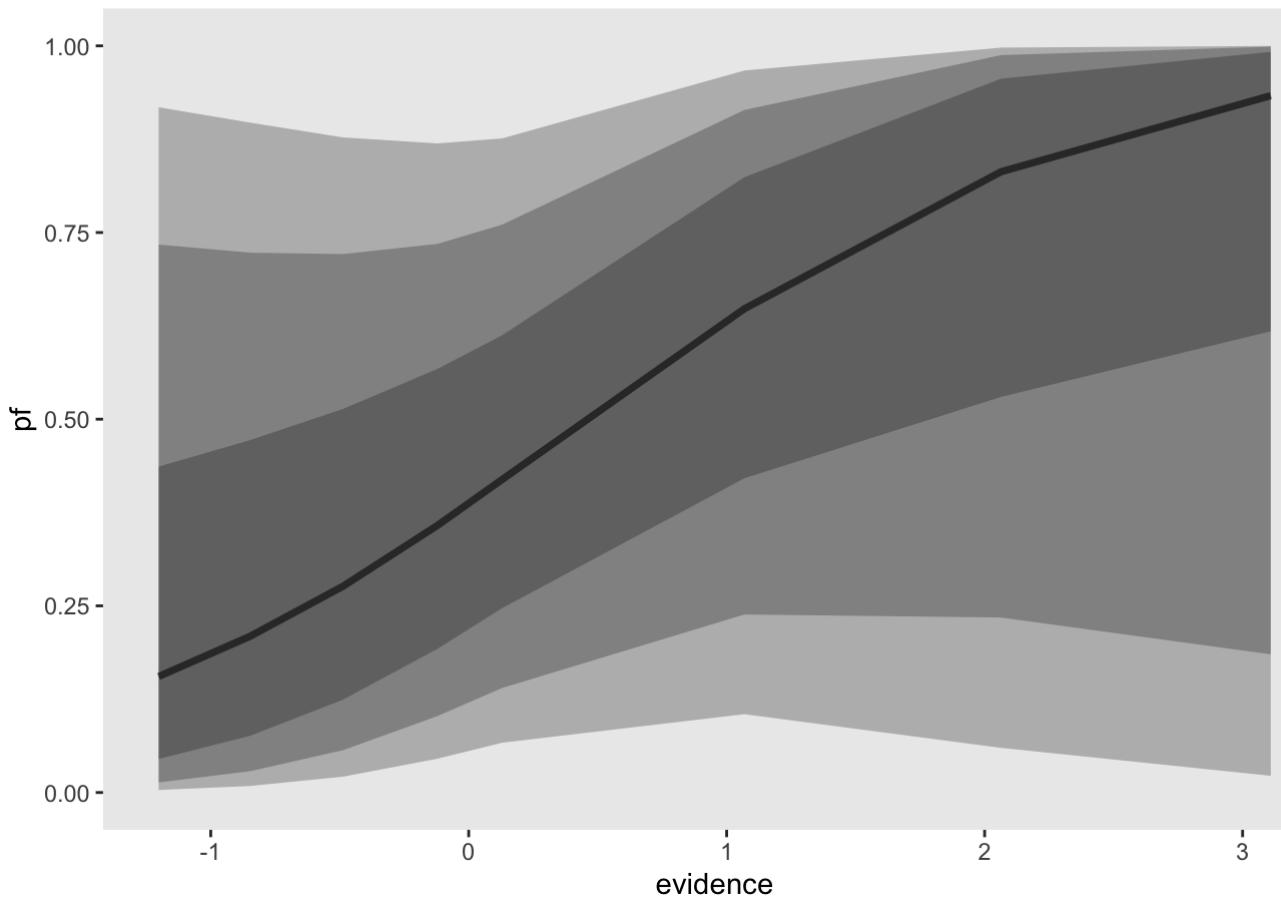
## Compiling the C++ model

## Start sampling

Let's look at predicted model fits to see the space of possible models predicted by our priors.

```
# prior predictive check
model_df %>%
  select(evidence, worker_id) %>%
  add_fitted_draws(prior.wrkr.logistic, value = "pf", seed = 1234) %>%
  ggplot(aes(x = evidence, y = pf)) +
  stat_lineribbon(.width = c(.95, .80, .50), alpha = .25, fill = "black") +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  labs(subtitle = "Prior predictive PF fit") +
  theme(panel.grid = element_blank())
```

### Prior predictive PF fit

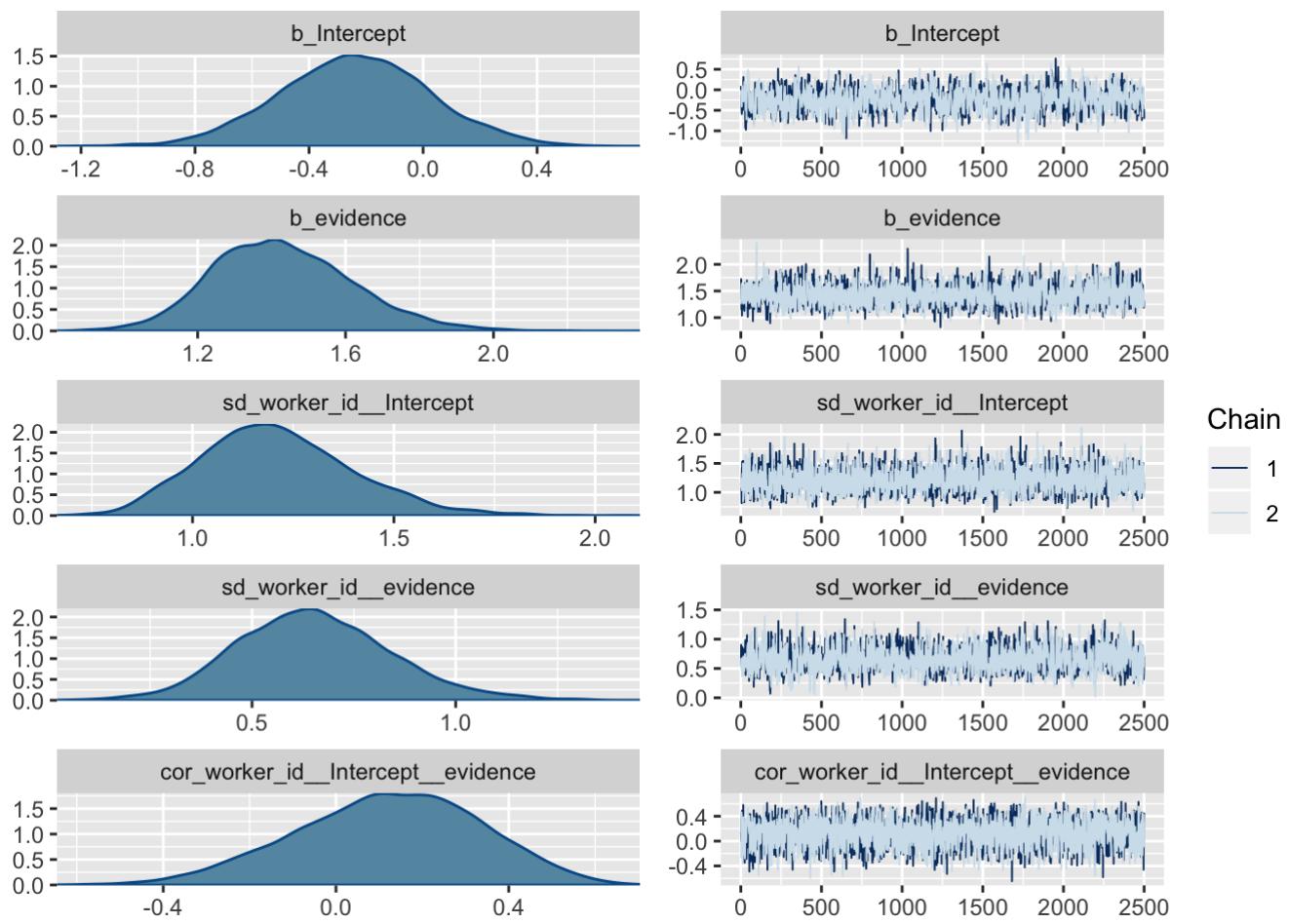


```
# hierarchical linear model with logit link
m.wrkr.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                         formula = bf(intervene ~ (1 + evidence|worker_id) + evidence),
                         prior = c(prior(normal(0, 1), class = Intercept),
                                   prior(normal(1, 1), class = b),
                                   prior(normal(0, 0.5), class = sd),
                                   prior(lkj(4), class = cor)),
                         iter = 3000, warmup = 500, chains = 2, cores = 2,
                         file = "model-fits/logistic_mdl-wrkr")
```

Check diagnostics:

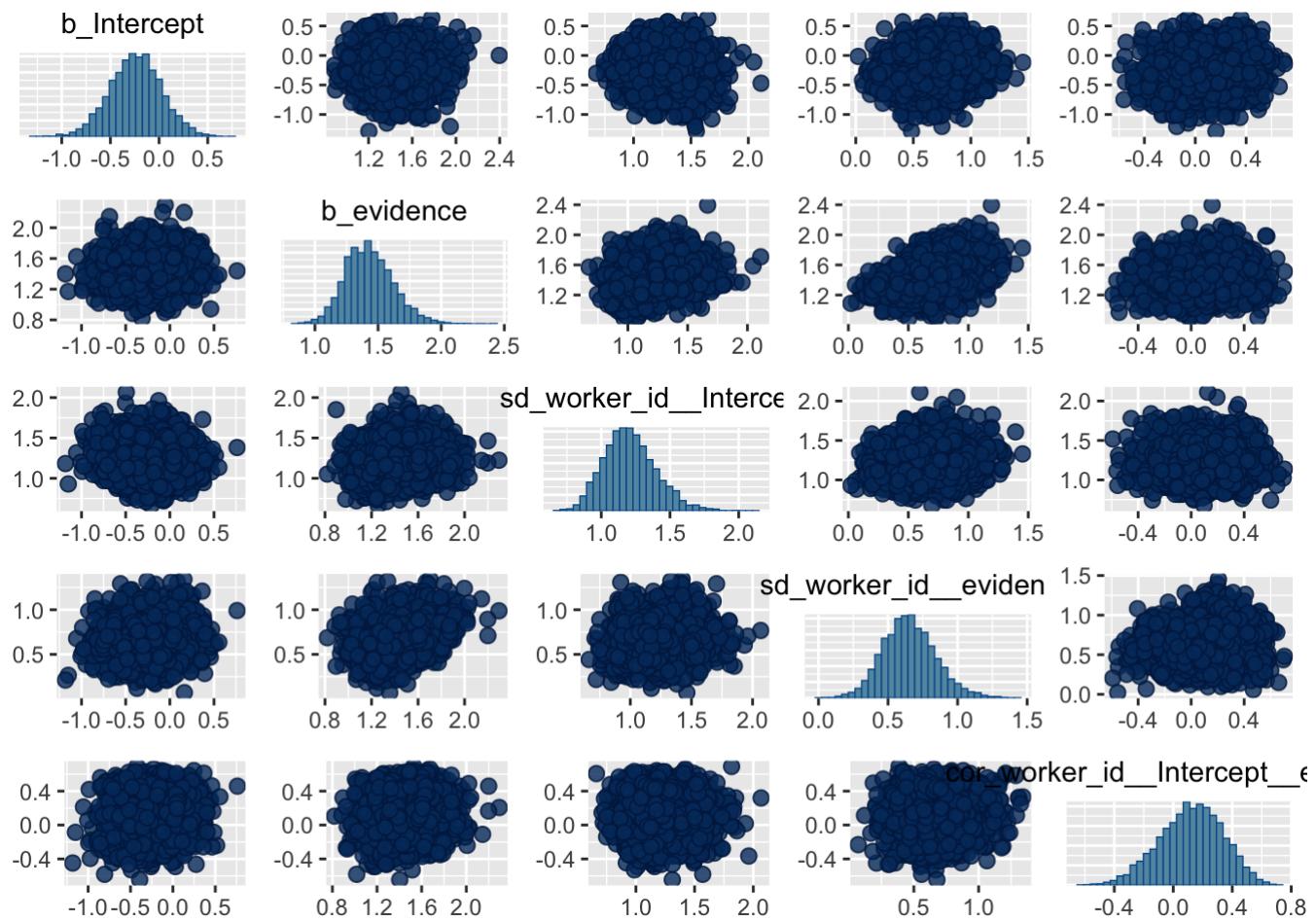
- Trace plots

```
# trace plots
plot(m.wrkr.logistic)
```



- Pairs plot

```
# pairs plot
pairs(m.wrkr.logistic)
```



- Summary

```
# model summary
print(m.wrkr.logistic)
```

```

## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence
## Data: model_df (Number of observations: 768)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 24)
##                               Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)             1.21     0.19    0.88    1.61      2175
## sd(evidence)              0.65     0.19    0.30    1.06      1601
## cor(Intercept,evidence)   0.12     0.21   -0.31    0.51      2561
##                               Rhat
## sd(Intercept)            1.00
## sd(evidence)             1.00
## cor(Intercept,evidence)  1.00
##
## Population-Level Effects:
##                               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept      -0.24      0.26    -0.76     0.28      958 1.00
## evidence       1.43      0.19     1.10     1.83     2212 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

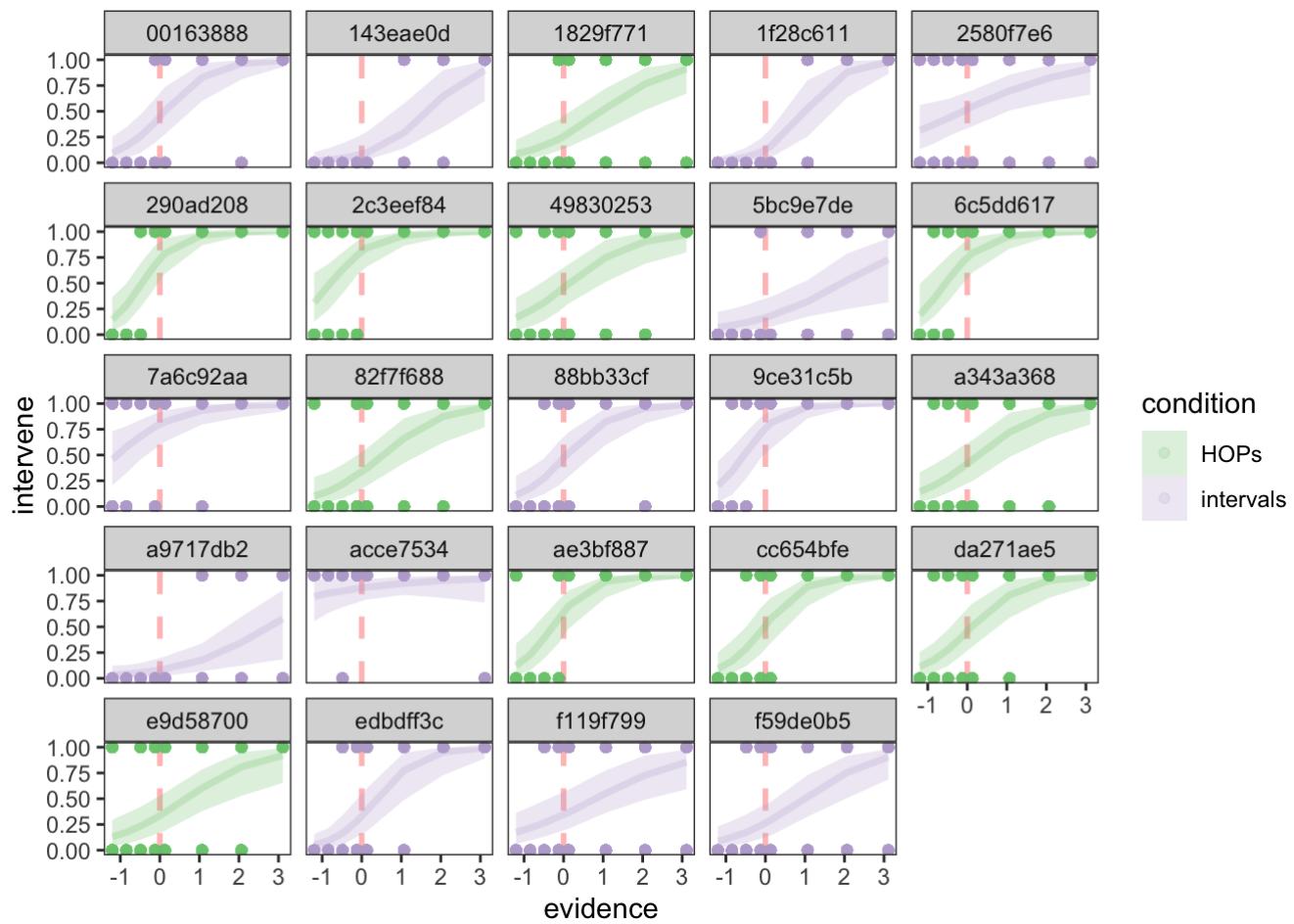
```

Let's take a look at the estimated psychometric functions for each worker.

```

model_df %>%
  group_by(evidence, worker_id) %>%
  add_fitted_draws(m.wrkr.logistic, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = condition, fill = condition)) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
  stat_lineribbon(aes(y = pf), .width = c(.95), alpha = .25) +
  geom_point(alpha = .15) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  theme_bw() +
  theme(panel.grid = element_blank()) +
  facet_wrap(. ~ worker_id)

```



These fits are looking better. Let's try expanding our model to include the manipulations in the experiment that we need to account for to answer our research questions.

## Add Predictors to Answer Research Questions

### Presence/Absence of the Mean

We want to know whether adding means to visualizations biases the point of subjective equality or changes sensitivity to utility on average. We model these effects as a fixed intercept and a fixed slope (i.e., an interaction with the level of evidence).

Before we fit our model let's check that our priors seem reasonable. Now that we are modeling multiple slopes using dummy coding, we need to differentiate between the baseline slope when there are no means (for which we use the same prior as before) and the effect of means on slopes (a difference from baseline which should be located at 0 and have a moderately weak prior).

```
# get_prior(data = model_df, family = bernoulli(link = "logit"), formula = bf(intervene
~ (1 + evidence/worker_id) + evidence*means))

# starting as simple as possible: learn the distribution of decisions
prior.wrkr.means.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                                 formula = bf(intervene ~ (1 + evidence|worker_id) + evi
dence* means),
                                 prior = c(prior(normal(0, 1), class = Intercept),
                                           prior(normal(1, 1), class = b, coef = evidenc
e),
                                           prior(normal(0, 0.5), class = b),
                                           prior(normal(0, 0.5), class = sd),
                                           prior(lkj(4), class = cor)),
                                 sample_prior = "only",
                                 iter = 3000, warmup = 500, chains = 2, cores = 2)
```

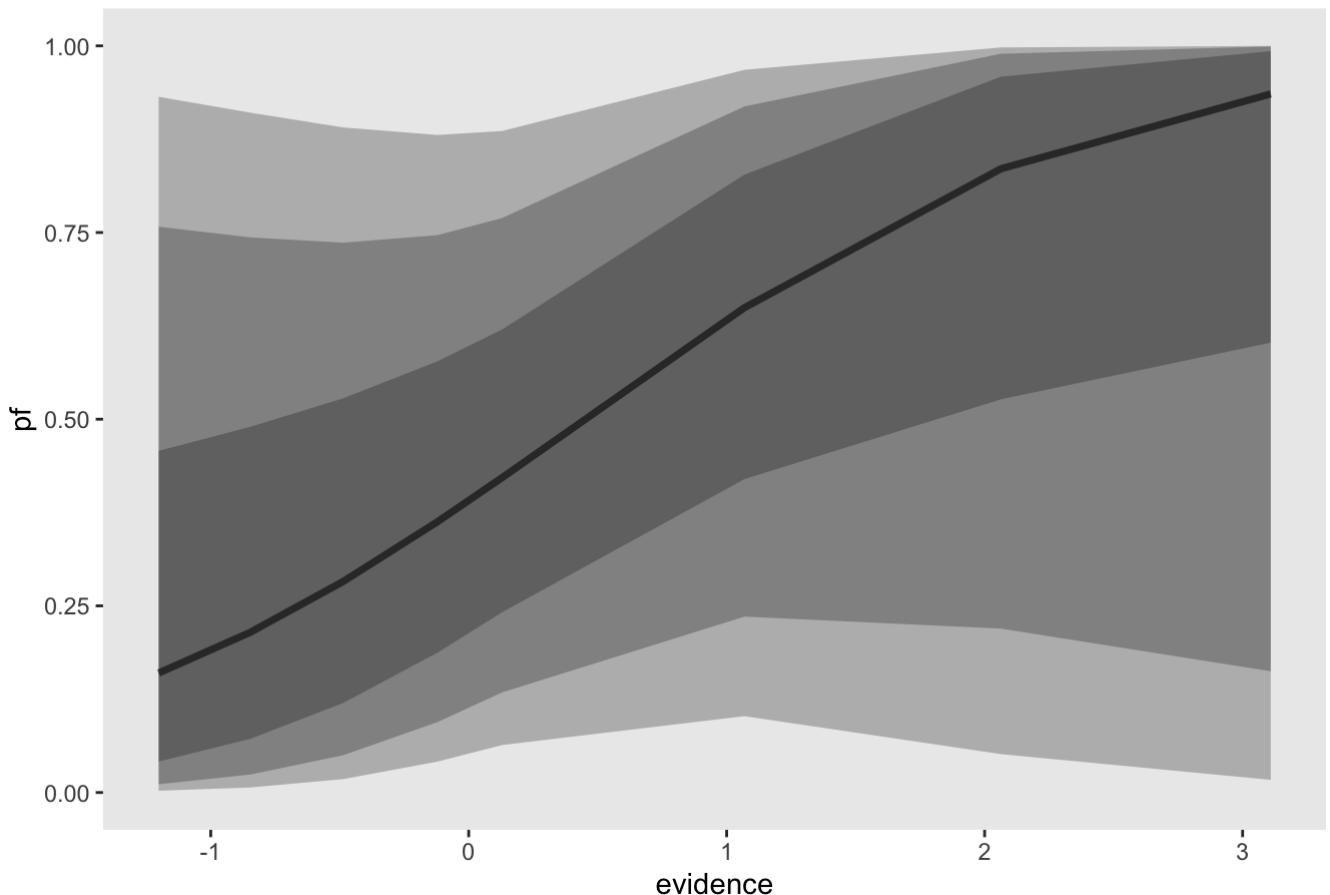
```
## Compiling the C++ model
```

```
## Start sampling
```

Let's look at predicted model fits to see the space of possible models predicted by our priors.

```
# prior predictive check
model_df %>%
  select(evidence, worker_id, means) %>%
  add_fitted_draws(prior.wrkr.means.logistic, value = "pf", seed = 1234) %>%
  ggplot(aes(x = evidence, y = pf)) +
  stat_lineribbon(.width = c(.95, .80, .50), alpha = .25, fill = "black") +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  labs(subtitle = "Prior predictive PF fit") +
  theme(panel.grid = element_blank())
```

### Prior predictive PF fit

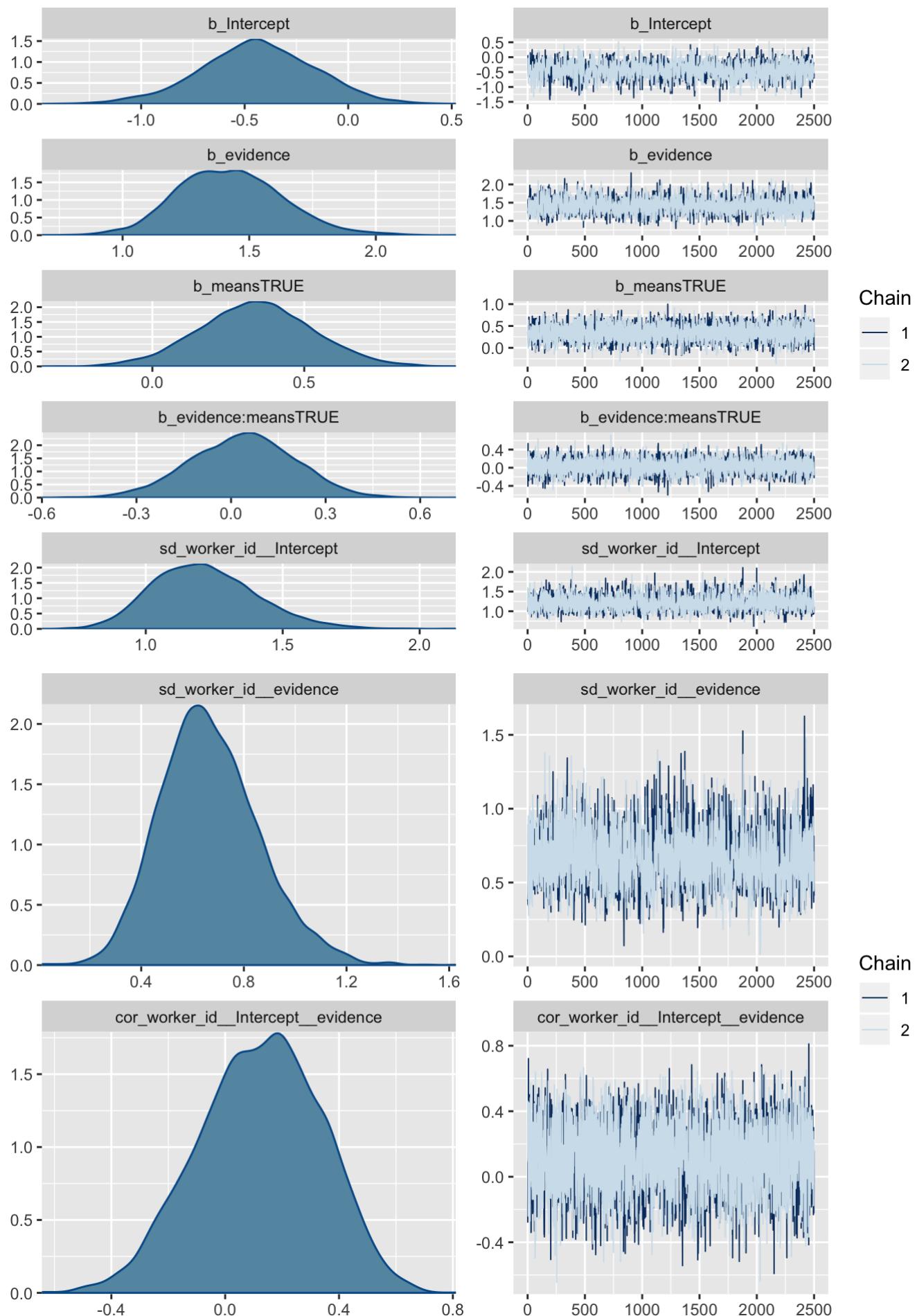


```
# hierarchical linear model with logit link and predictors for the presence/absence of means
m.wrkr.means.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                                formula = bf(intervene ~ (1 + evidence|worker_id) + evidence * means),
                                prior = c(prior(normal(0, 1), class = Intercept),
                                          prior(normal(1, 1), class = b, coef = evidence),
                                          prior(normal(0, 0.5), class = b),
                                          prior(normal(0, 0.5), class = sd),
                                          prior(lkj(4), class = cor)),
                                iter = 3000, warmup = 500, chains = 2, cores = 2,
                                file = "model-fits/logistic_mdl-wrkr_means")
```

Check diagnostics:

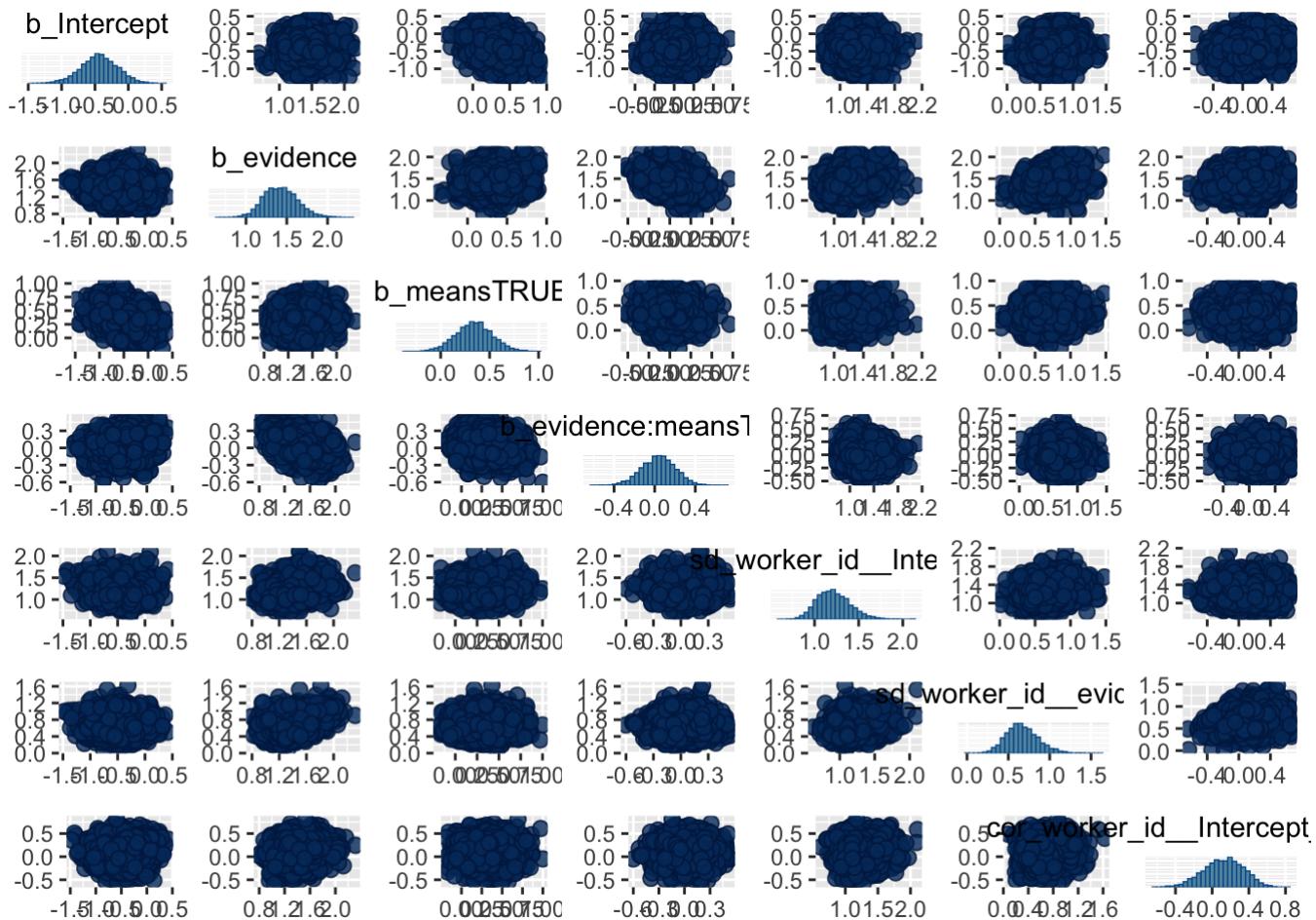
- Trace plots

```
# trace plots
plot(m.wrkr.means.logistic)
```



- Pairs plot

```
# pairs plot
pairs(m.wrkr.means.logistic)
```



- Summary

```
# model summary
print(m.wrkr.means.logistic)
```

```

## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence * means
## Data: model_df (Number of observations: 768)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 24)
##                               Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)             1.22     0.19    0.89    1.63      2442
## sd(evidence)              0.67     0.19    0.33    1.09      1741
## cor(Intercept,evidence)   0.12     0.21   -0.30    0.51      2680
##                               Rhat
## sd(Intercept)            1.00
## sd(evidence)             1.00
## cor(Intercept,evidence)  1.00
##
## Population-Level Effects:
##                               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept                 -0.44     0.28   -1.01    0.10      1280 1.00
## evidence                  1.42     0.21    1.05    1.85      2071 1.00
## meansTRUE                 0.34     0.19   -0.02    0.71      5633 1.00
## evidence:meansTRUE        0.04     0.16   -0.29    0.35      4927 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

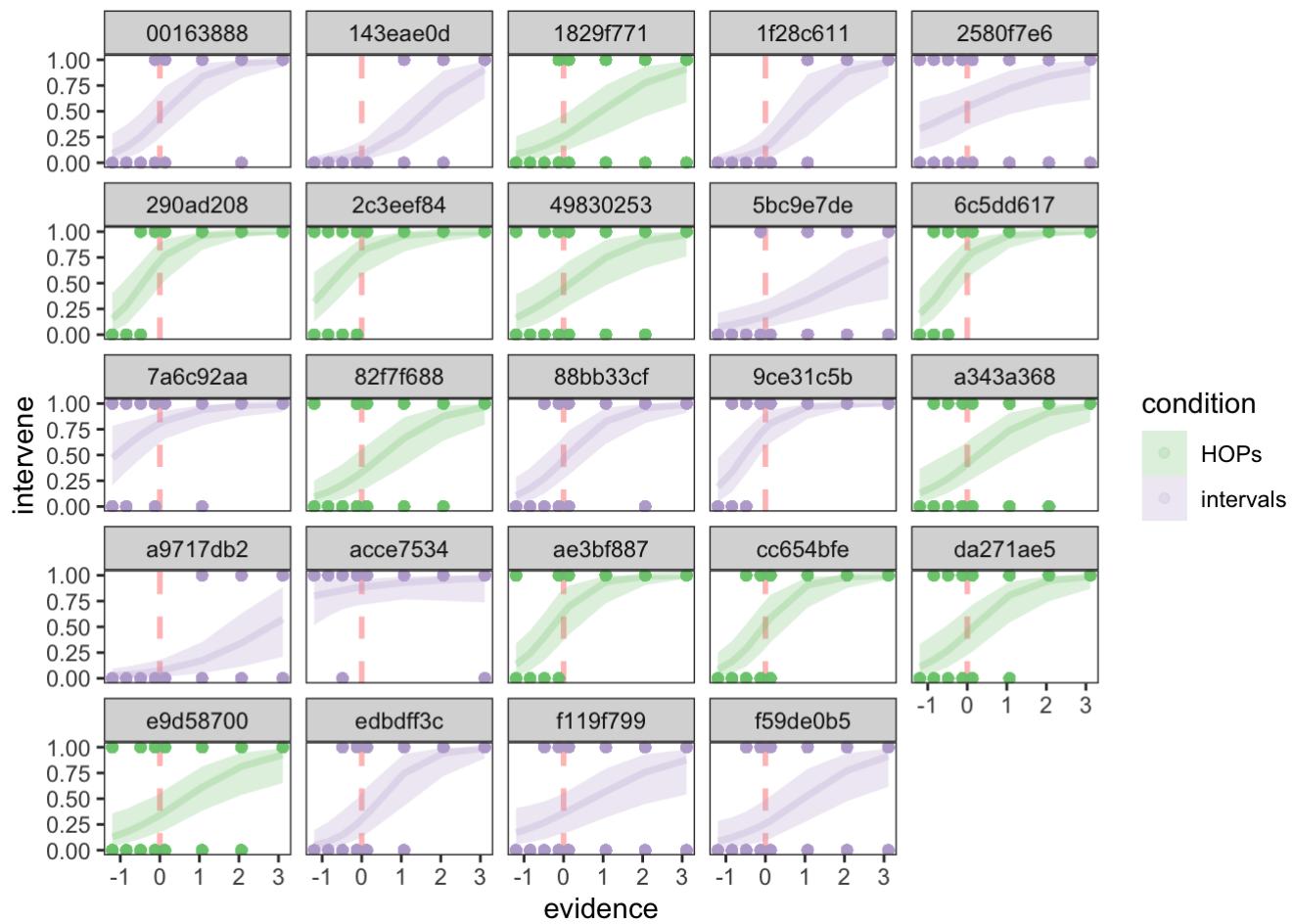
```

Let's take a look at the estimated psychometric functions for each worker.

```

model_df %>%
  group_by(evidence, worker_id, means) %>%
  add_fitted_draws(m.wrkr.means.logistic, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = condition, fill = condition)) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
  stat_lineribbon(aes(y = pf), .width = c(.95), alpha = .25) +
  geom_point(alpha = .15) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  theme_bw() +
  theme(panel.grid = element_blank()) +
  facet_wrap(. ~ worker_id)

```



It looks like there may be some differences per visualization condition that our model is not capturing.

What do the posteriors for just-noticeable differences (JND) and points of subjective equality (PSE) look like when means are present vs absent? Since we are probing conditional expectations, we'll forego calculating marginal effects by manually combining parameters. Instead we'll use `add_fitted_draws` and `compare_levels` to get slopes and intercepts. Then we'll use these to derive estimates of the JND and PSE for each fitted draw.

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(means) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.means.logistic, re_formula = NA, scale = "linear") %>%
  compare_levels(.value, by = evidence) %>%
  rename(slope = .value)

# get intercepts from linear model
intercepts_df <- model_df %>%
  group_by(means) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.means.logistic, re_formula = NA, scale = "linear") %>%
  rename(intercept = .value)

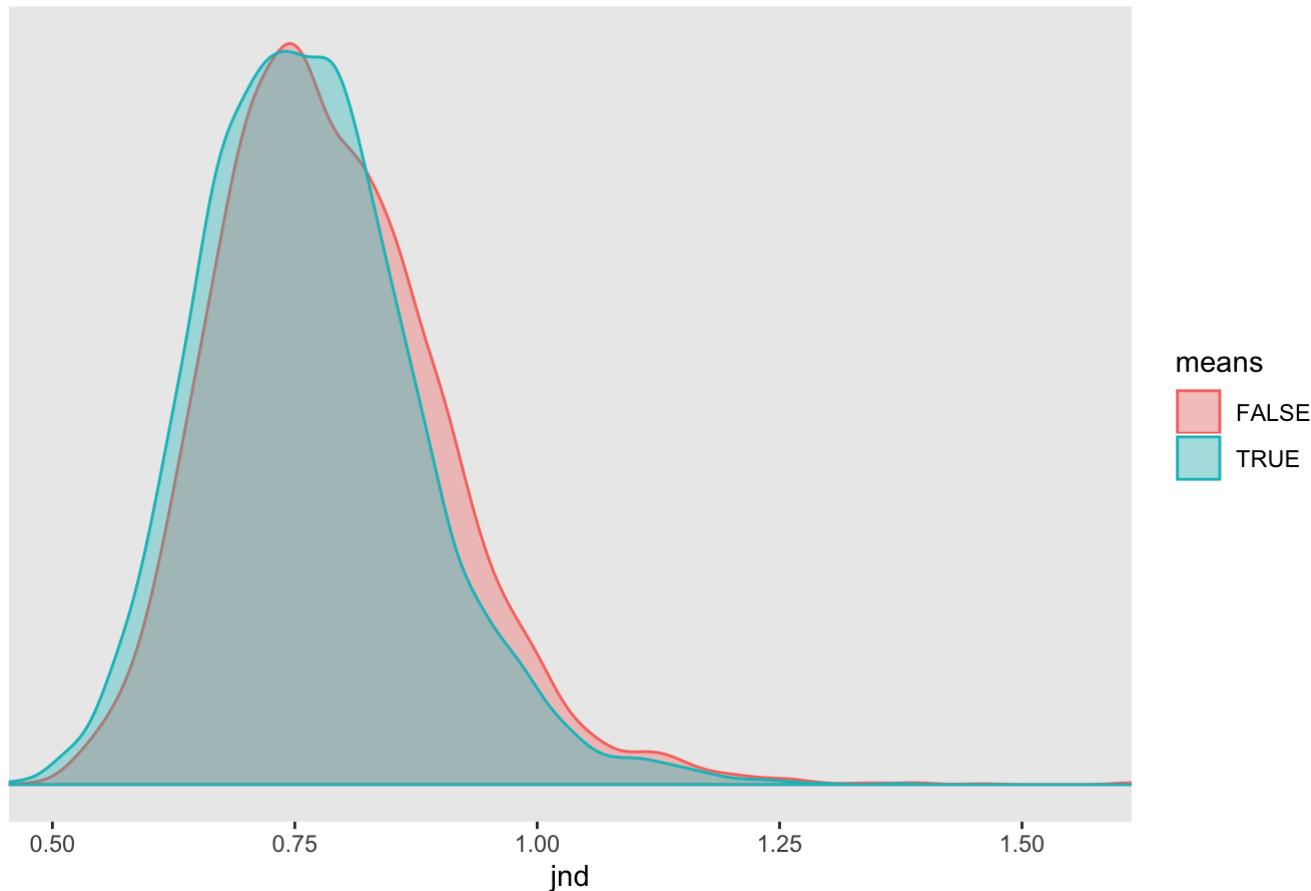
# join dataframes for slopes and intercepts, calculate PSE and JND
stats_df <- slopes_df %>%
  full_join(intercepts_df, by = c("means", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )

```

First, let's look at the estimates of JNDs per condition.

```
stats_df %>%
  # compare_levels(jnd, by = means) %>%
  # mutate(jnd_p_award = exp(jnd) / (1 / unique(model_df$baseline) + 1 / unique(model_df$award_value)) - 1 + exp(jnd)) - unique(model_df$baseline) - 1 / unique(model_df$award_value)) %>%
  ggplot(aes(x = jnd, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  # ggplot(aes(x = jnd_p_award)) +
  # geom_density(alpha = 0.35, fill = "black") +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  # labs(subtitle = "Posterior difference in JNDs with means present vs absent") +
  theme(panel.grid = element_blank())
```

## Posterior JND per condition

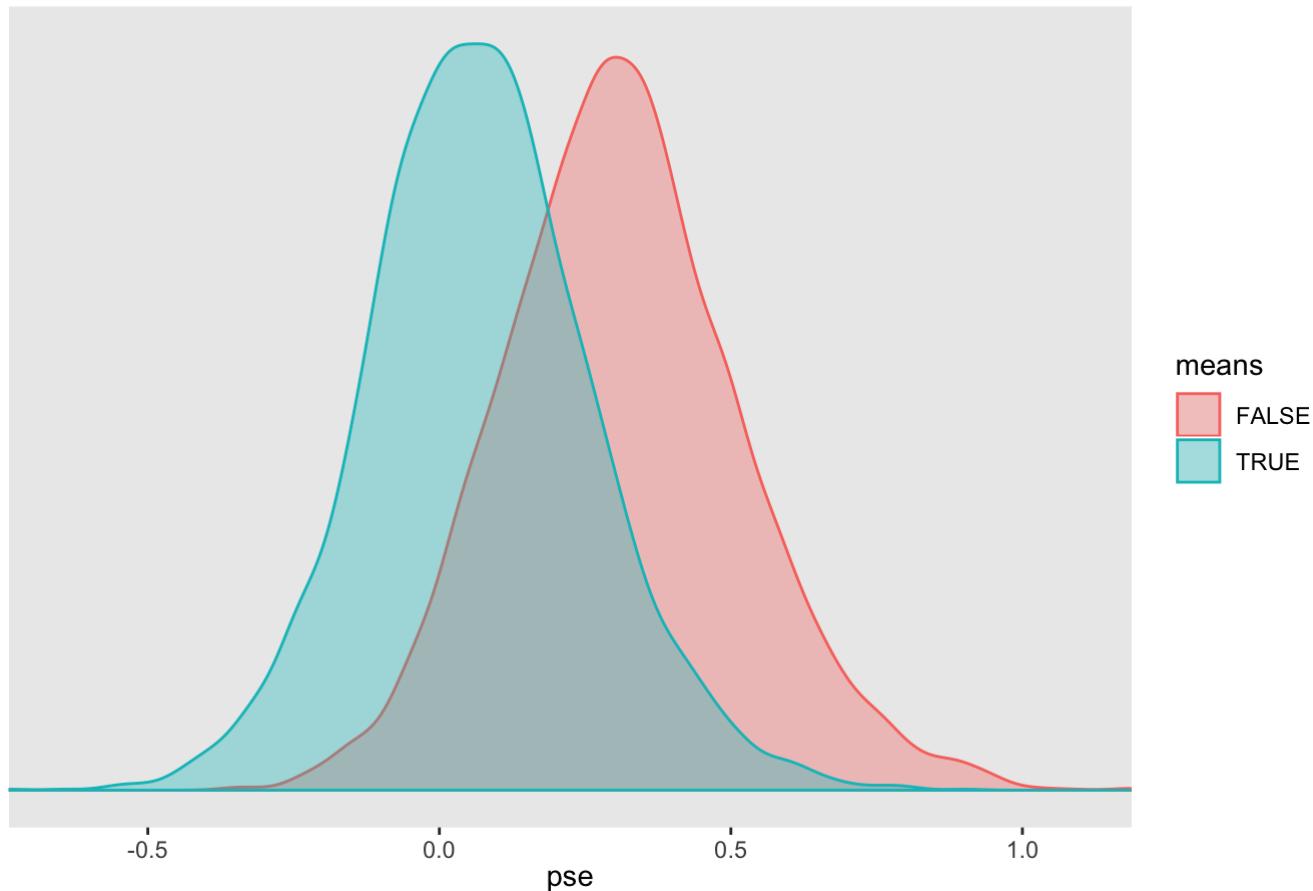


It looks like users are similarly sensitive to evidence whether means are present or absent, ignoring the effect of other manipulations.

Next, we'll look at the point of subjective equality in each condition.

```
stats_df %>%
  ggplot(aes(x = pse, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per condition") +
  theme(panel.grid = element_blank())
```

### Posterior PSE per condition



It looks like the point of subjective equality less biased when means are present, ignoring the impact of other manipulations.

## Uncertainty Shown

We expect that the level of uncertainty shown will impact judgments of effect size. Does it similarly effect the impact of means on decision-making? We test this by adding `sd_diff` to the interaction term in our model specification.

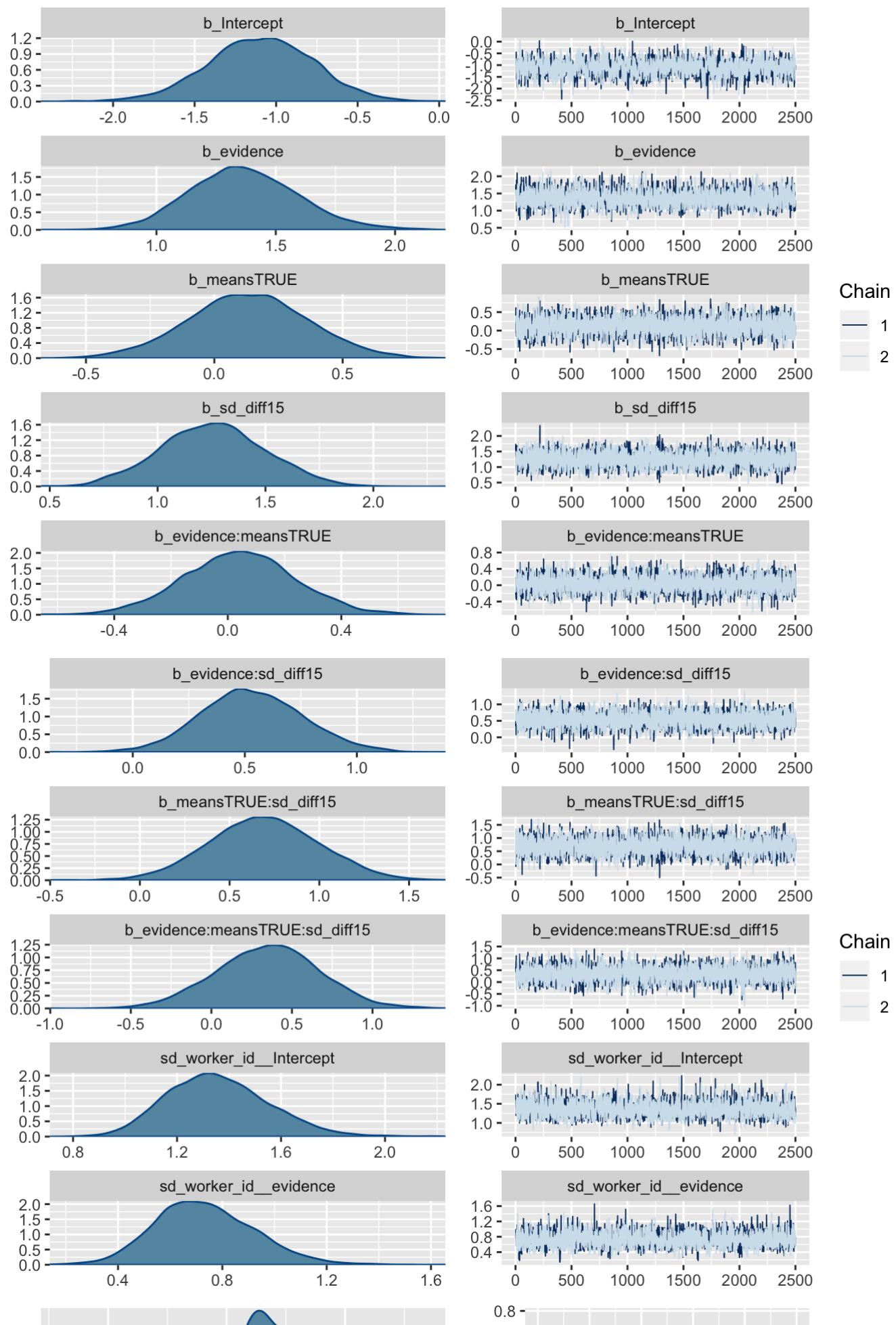
We'll use the same priors as in our previous model, so we'll jump right to fitting the model.

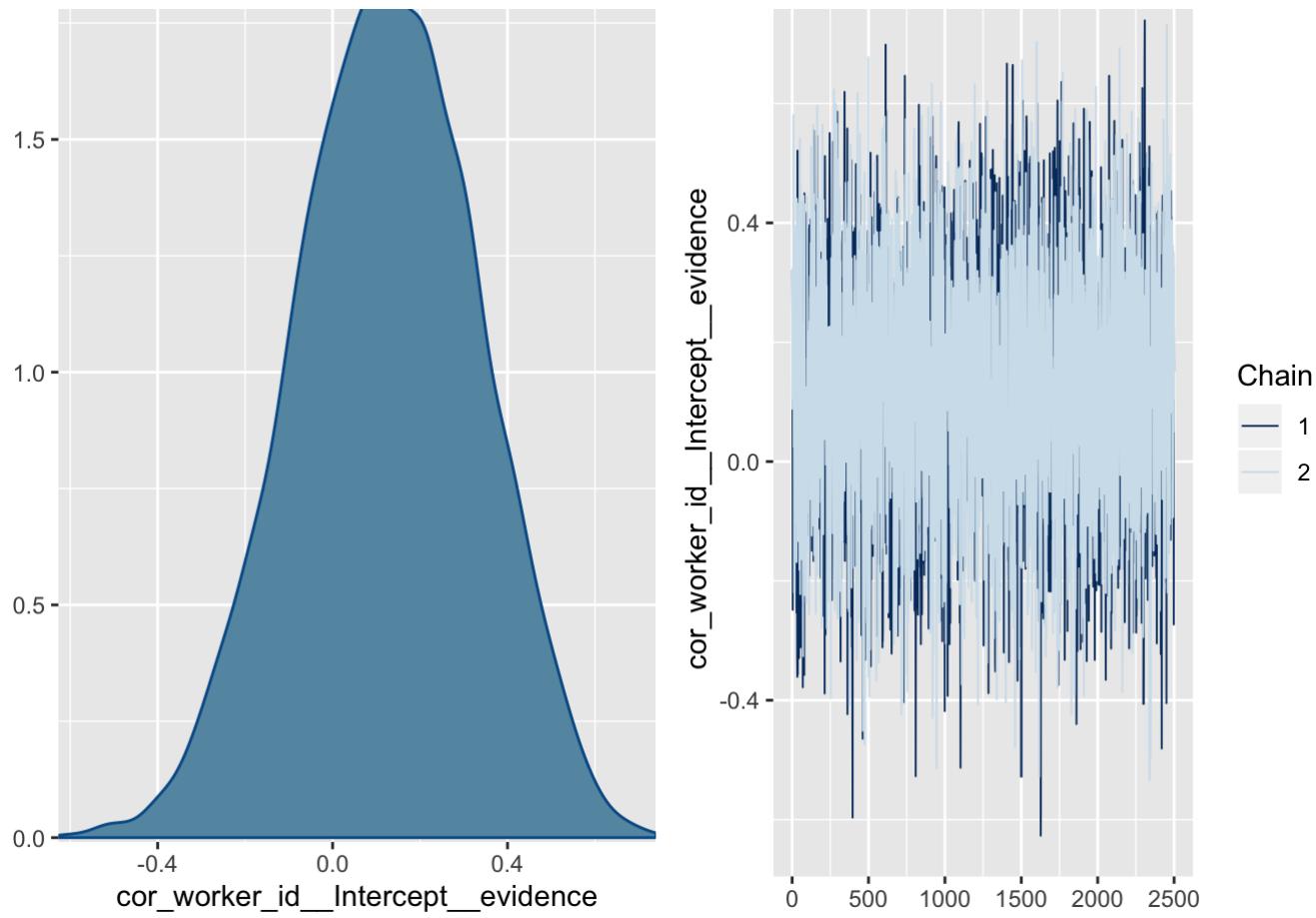
```
# hierarchical linear model with logit link and predictors for the presence/absence of means
m.wrkr.means.sd.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                                    formula = bf(intervene ~ (1 + evidence|worker_id) + evidence*means*sd_diff),
                                    prior = c(prior(normal(0, 1), class = Intercept),
                                              prior(normal(1, 1), class = b, coef = evidence),
                                              prior(normal(0, 0.5), class = b),
                                              prior(normal(0, 0.5), class = sd),
                                              prior(lkj(4), class = cor)),
                                    iter = 3000, warmup = 500, chains = 2, cores = 2,
                                    file = "model-fits/logistic_mdl-wrkr_means_sd")
```

Check diagnostics:

- Trace plots

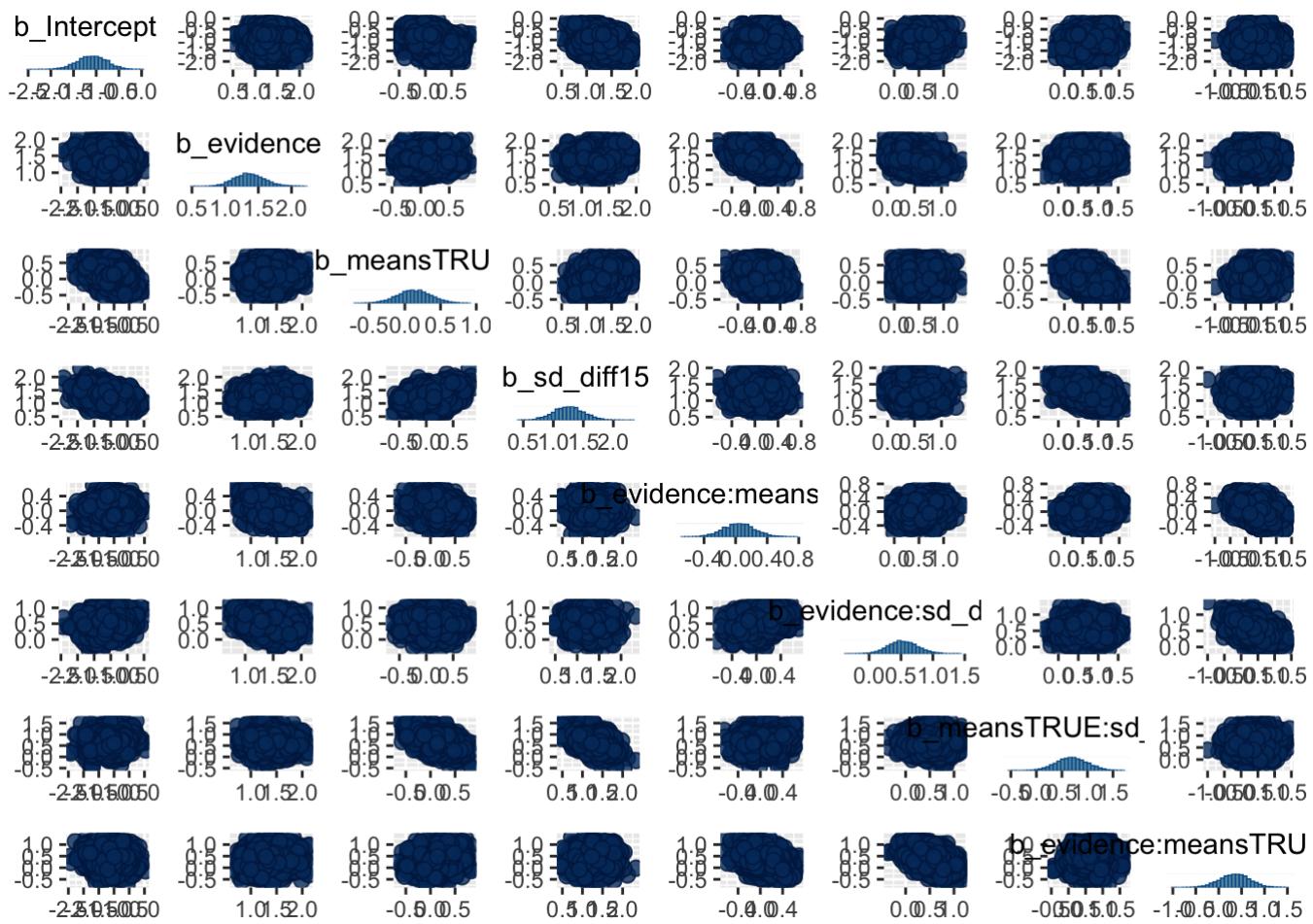
```
# trace plots
plot(m.wrkr.means.sd.logistic)
```





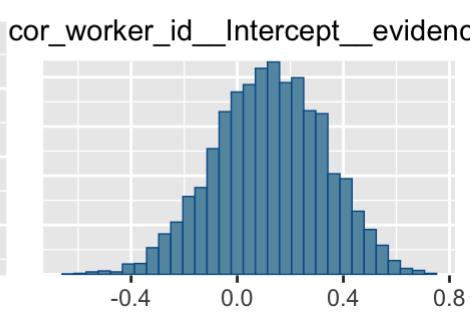
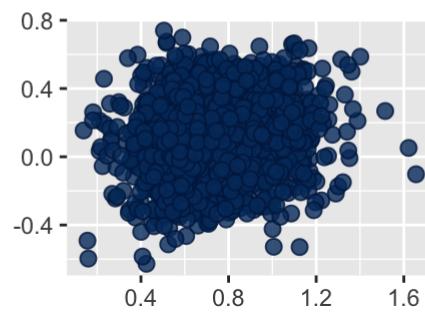
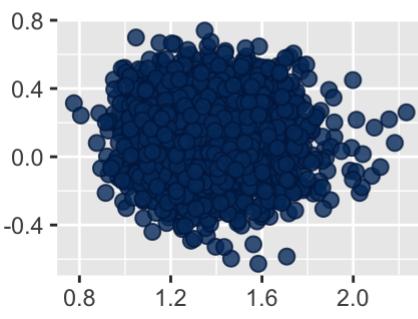
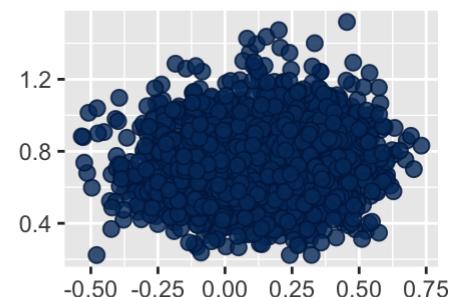
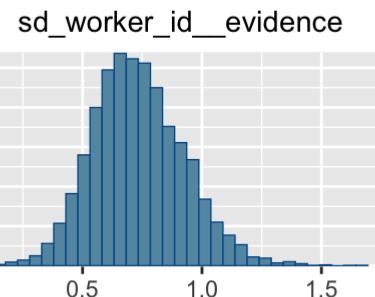
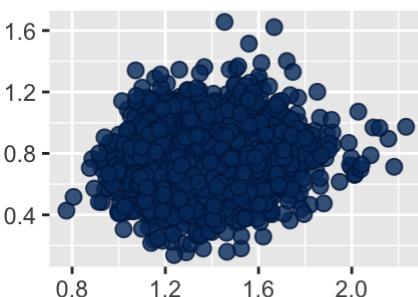
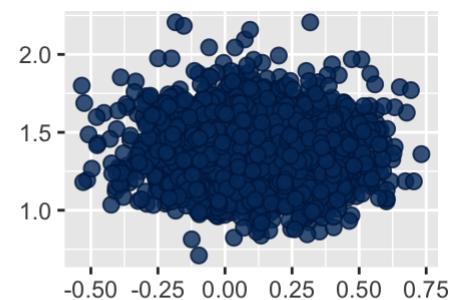
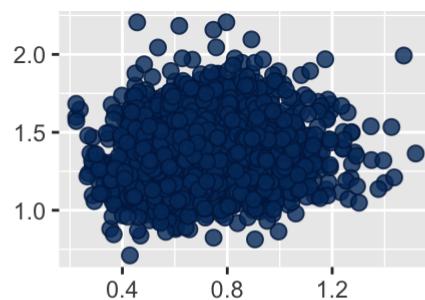
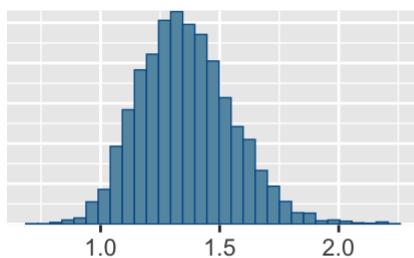
- Pairs plot

```
# pairs plot
pairs(m.wrkr.means.sd.logistic, pars = c("b_"))
```



```
# pairs plot
pairs(m.wrkr.means.sd.logistic, pars = c("sd_worker_id", "cor_worker_id"))
```

sd\_worker\_id\_Intercept



- Summary

```
# model summary
print(m.wrkr.means.sd.logistic)
```

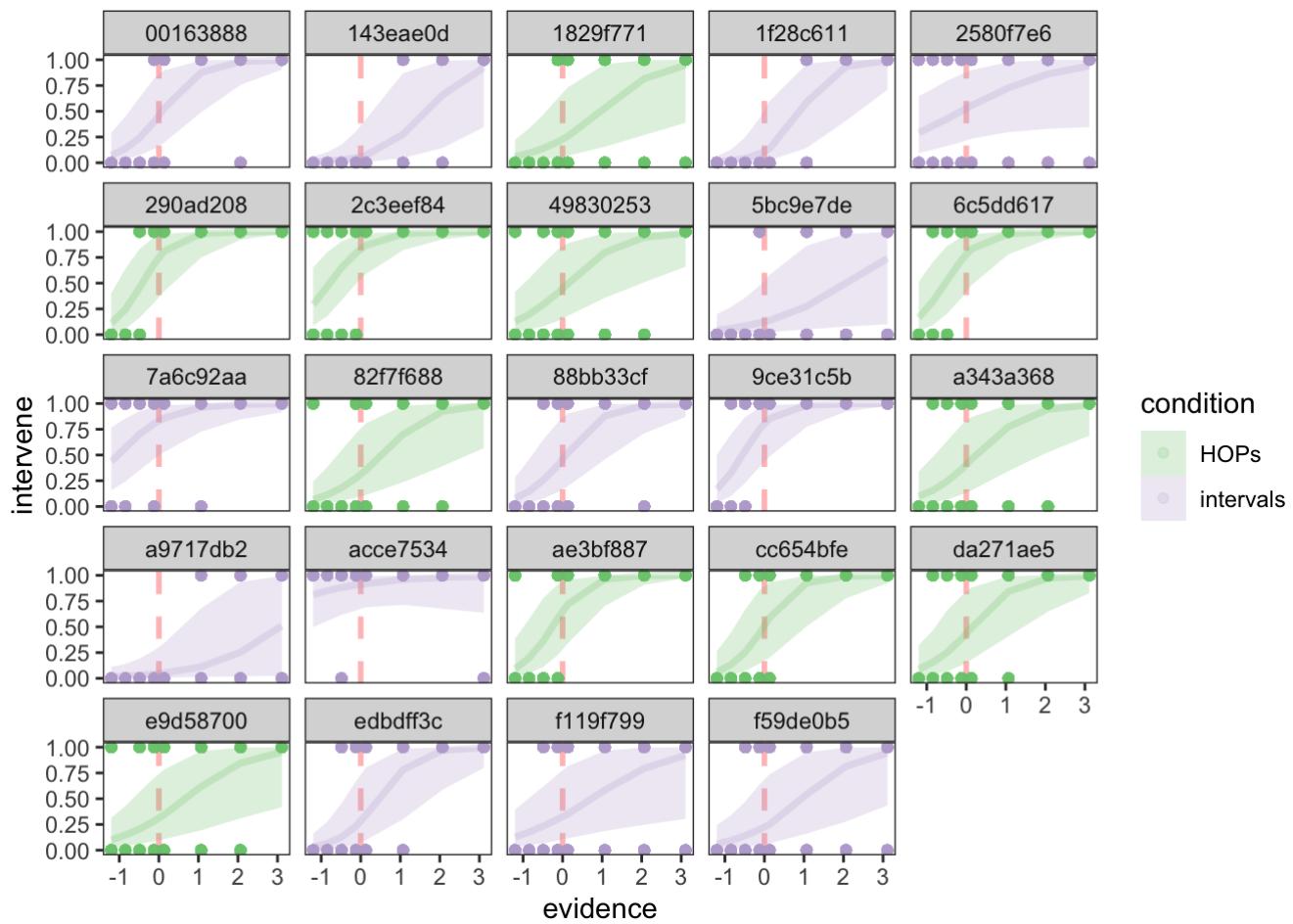
```

## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence * means * sd_diff
## Data: model_df (Number of observations: 768)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 24)
##                                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)                  1.36     0.20    1.02    1.77      2501
## sd(evidence)                  0.73     0.19    0.40    1.13      1626
## cor(Intercept,evidence)      0.12     0.21   -0.29    0.51      2408
##                                         Rhat
## sd(Intercept)                  1.00
## sd(evidence)                  1.00
## cor(Intercept,evidence)      1.00
##
## Population-Level Effects:
##                                         Estimate Est.Error l-95% CI u-95% CI
## Intercept                     -1.10     0.33   -1.76   -0.47
## evidence                      1.37     0.23    0.95    1.84
## meansTRUE                     0.13     0.23   -0.32    0.58
## sd_diff15                     1.26     0.24    0.78    1.74
## evidence:meansTRUE            0.04     0.20   -0.35    0.43
## evidence:sd_diff15            0.53     0.23    0.10    0.99
## meansTRUE:sd_diff15           0.69     0.30    0.11    1.29
## evidence:meansTRUE:sd_diff15  0.35     0.32   -0.28    0.96
##                                         Eff.Sample Rhat
## Intercept                      1301 1.00
## evidence                       2495 1.00
## meansTRUE                      4754 1.00
## sd_diff15                      5068 1.00
## evidence:meansTRUE             4536 1.00
## evidence:sd_diff15             4982 1.00
## meansTRUE:sd_diff15            4340 1.00
## evidence:meansTRUE:sd_diff15  4808 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

```

Let's take a look at the estimated psychometric functions for each worker.

```
model_df %>%
  group_by(evidence, worker_id, means, sd_diff) %>%
  add_fitted_draws(m.wrkr.means.sd.logistic, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = condition, fill = condition)) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
  stat_lineribbon(aes(y = pf), .width = c(.95), alpha = .25) +
  geom_point(alpha = .15) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  theme_bw() +
  theme(panel.grid = element_blank()) +
  facet_wrap(. ~ worker_id)
```



It looks like there may be some differences per visualization condition that our model is not capturing.

What do the posteriors for just-noticeable differences (JND) and points of subjective equality (PSE) look like when means are present vs absent at different levels of uncertainty?

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(means, sd_diff) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.means.sd.logistic, re_formula = NA, scale = "linear") %>%
  compare_levels(.value, by = evidence) %>%
  rename(slope = .value)

# get intercepts from linear model
intercepts_df <- model_df %>%
  group_by(means, sd_diff) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.means.sd.logistic, re_formula = NA, scale = "linear") %>%
  rename(intercept = .value)

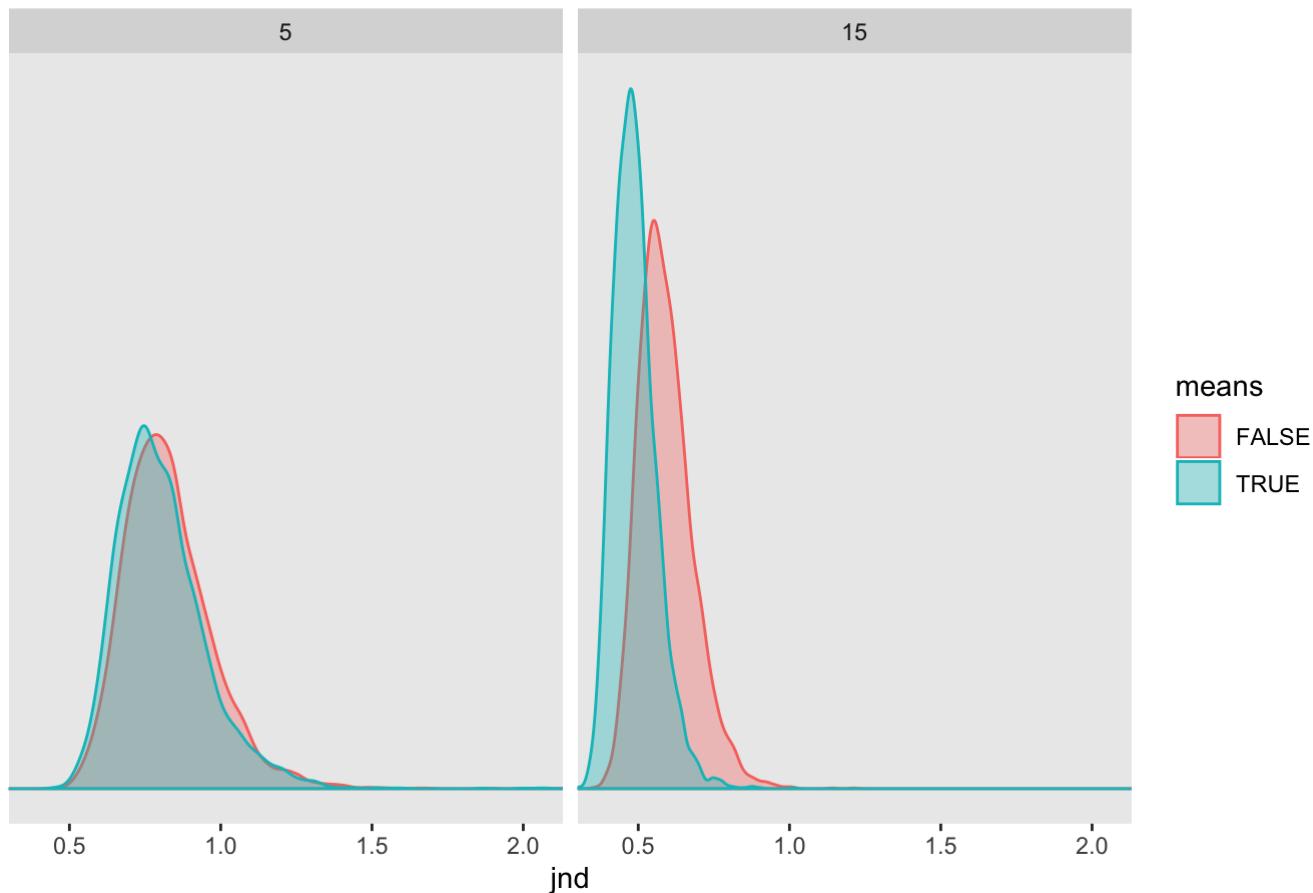
# join dataframes for slopes and intercepts, calculate PSE and JND
stats_df <- slopes_df %>%
  full_join(intercepts_df, by = c("means", "sd_diff", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )

```

First, let's look at the estimates of JNDs per condition.

```
stats_df %>%
  # compare_levels(jnd, by = means) %>%
  # mutate(jnd_p_award = exp(jnd) / (1 / unique(model_df$baseline) + 1 / unique(model_df$award_value)) - 1 + exp(jnd)) - unique(model_df$baseline) - 1 / unique(model_df$award_value)) %>%
  ggplot(aes(x = jnd, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  # ggplot(aes(x = jnd_p_award)) +
  # geom_density(alpha = 0.35, fill = "black") +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  # labs(subtitle = "Posterior difference in JNDs with means present vs absent") +
  theme(panel.grid = element_blank()) +
  facet_grid(. ~ sd_diff)
```

## Posterior JND per condition

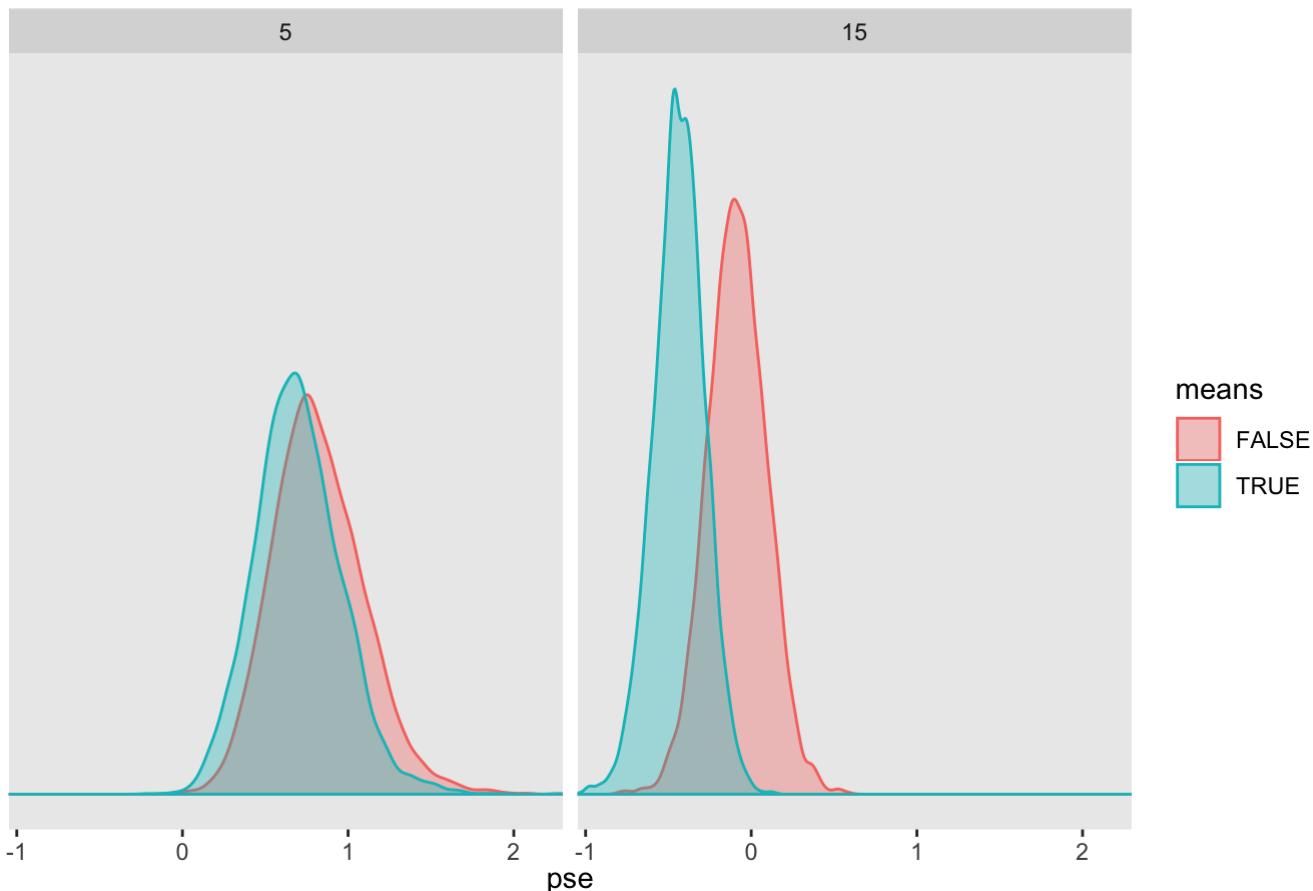


It looks like means improve sensitivity to evidence when mean are present and uncertainty is high, ignoring the effect of other manipulations. This is consistent with the debiasing effect of the mean on effect size judgments when uncertainty is high.

Next, we'll look at the point of subjective equality in each condition.

```
stats_df %>%
  ggplot(aes(x = pse, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per condition") +
  theme(panel.grid = element_blank()) +
  facet_grid(. ~ sd_diff)
```

## Posterior PSE per condition



It looks like the point of subjective equality slightly biased when means are present and uncertainty is low. However, when uncertainty is high, users are less biased without means and tend to lean away from intervention when means are present. This is ignoring the impact of other manipulations.

## Visualization Conditions

We also want to know whether different visualizations conditions bias the point of subjective equality or change sensitivity to utility on average. Just like we did for the effect of extrinsic means, we model these effects as a fixed intercept and a fixed slope (i.e., an interaction with the level of evidence).

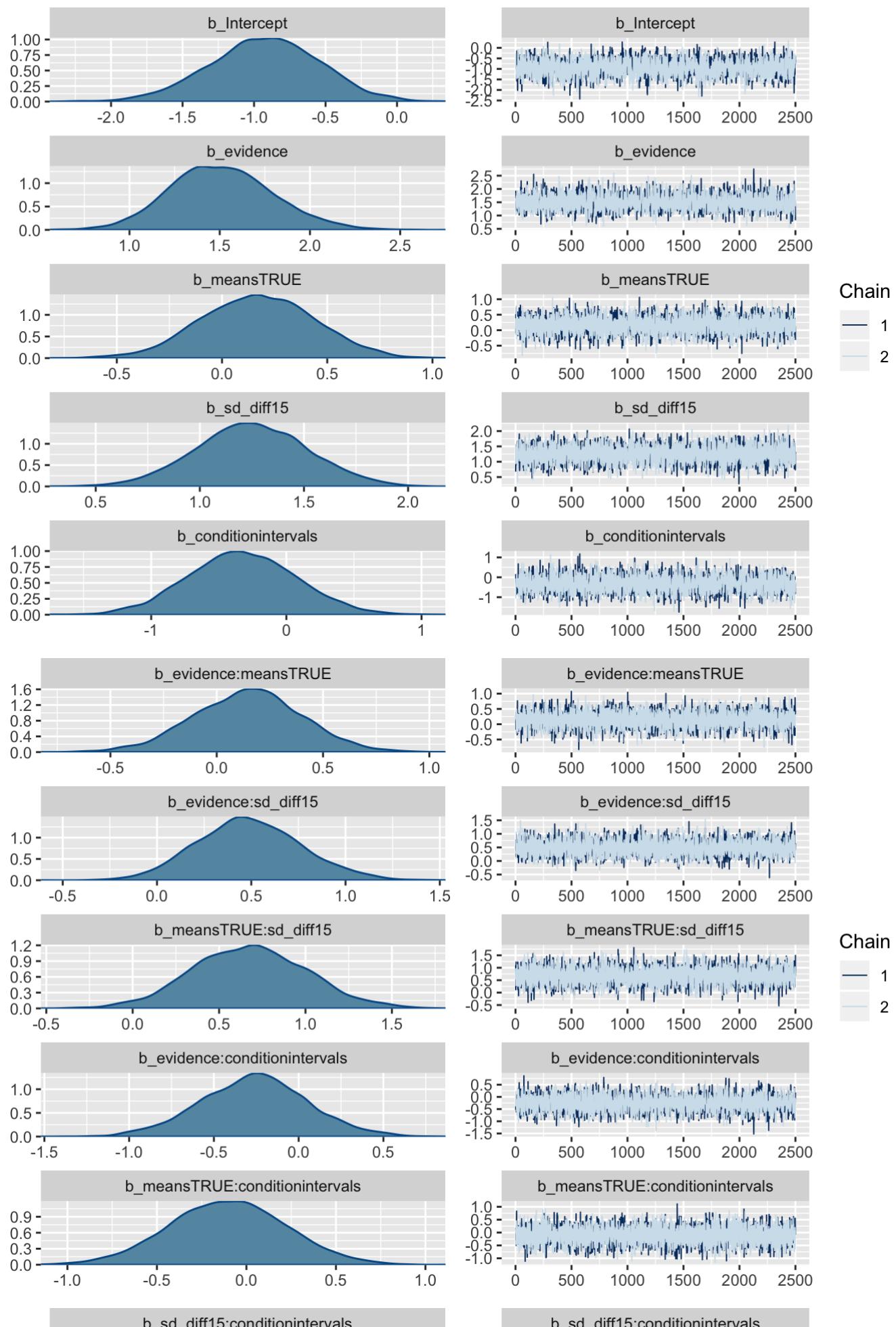
We'll use the same priors as we did for our previous model. Now, let's fit our model.

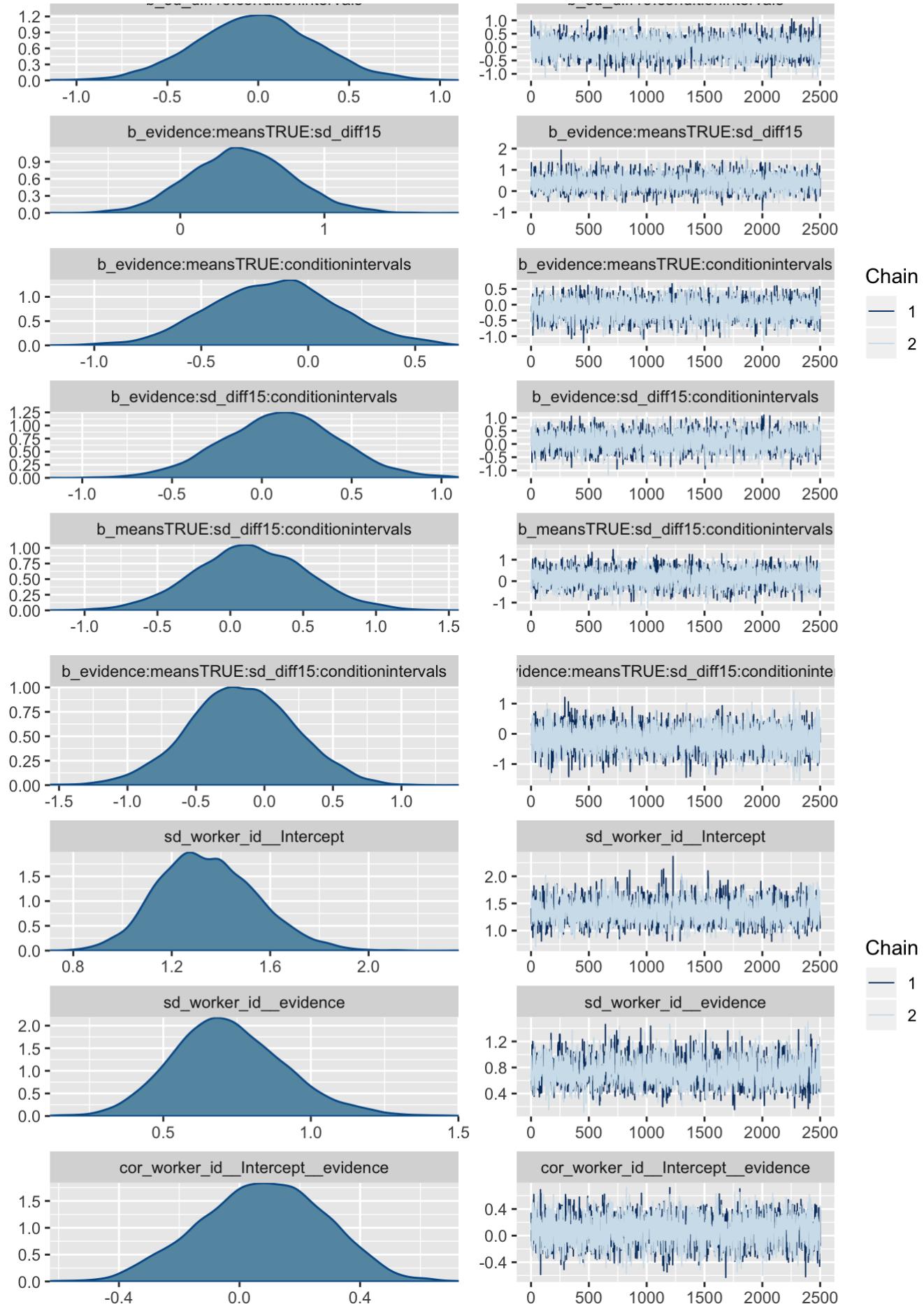
```
# hierarchical linear model with logit link and predictors per visualization condition
m.wrkr.means.sd.vis.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                                         formula = bf(intervene ~ (1 + evidence|worker_id) +
                                         evidence*means*sd_diff*condition),
                                         prior = c(prior(normal(0, 1), class = Intercept),
                                                   prior(normal(1, 1), class = b, coef = evidence),
                                                   prior(normal(0, 0.5), class = b),
                                                   prior(normal(0, 0.5), class = sd),
                                                   prior(lkj(4), class = cor)),
                                         iter = 3000, warmup = 500, chains = 2, cores = 2,
                                         file = "model-fits/logistic_mdl-wrkr_means_sd_vis")
```

Check diagnostics:

- Trace plots

```
# trace plots
plot(m.wrkr.means.sd.vis.logistic)
```





- Summary

```
# model summary  
print(m.wrkr.means.sd.vis.logistic)
```

```

## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence * means * sd_diff * condition
## Data: model_df (Number of observations: 768)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 24)
##                               Estimate Est.Error 1-95% CI u-95% CI Eff.Sample
## sd(Intercept)             1.35     0.20    0.99    1.76      2553
## sd(evidence)              0.72     0.19    0.38    1.14      1853
## cor(Intercept,evidence)   0.08     0.21   -0.33    0.46      3553
##                               Rhat
## sd(Intercept)             1.00
## sd(evidence)              1.00
## cor(Intercept,evidence)   1.00
##
## Population-Level Effects:
##                               Estimate Est.Error
## Intercept                  -0.92     0.38
## evidence                   1.51     0.29
## meansTRUE                  0.17     0.27
## sd_diff15                 1.25     0.27
## conditionintervals         -0.33     0.40
## evidence:meansTRUE         0.15     0.25
## evidence:sd_diff15        0.49     0.27
## meansTRUE:sd_diff15       0.68     0.33
## evidence:conditionintervals -0.26     0.31
## meansTRUE:conditionintervals -0.10     0.32
## sd_diff15:conditionintervals -0.01     0.33
## evidence:meansTRUE:sd_diff15 0.42     0.35
## evidence:meansTRUE:conditionintervals -0.16     0.30
## evidence:sd_diff15:conditionintervals 0.11     0.32
## meansTRUE:sd_diff15:conditionintervals 0.14     0.38
## evidence:meansTRUE:sd_diff15:conditionintervals -0.17     0.39
##                               1-95% CI u-95% CI
## Intercept                  -1.70    -0.18
## evidence                   0.98     2.10
## meansTRUE                  -0.35     0.69
## sd_diff15                 0.73     1.77
## conditionintervals         -1.12     0.45
## evidence:meansTRUE         -0.37     0.63
## evidence:sd_diff15        -0.03     1.05
## meansTRUE:sd_diff15       0.03     1.34
## evidence:conditionintervals -0.89     0.36
## meansTRUE:conditionintervals -0.74     0.53
## sd_diff15:conditionintervals -0.66     0.64
## evidence:meansTRUE:sd_diff15 -0.25     1.14
## evidence:meansTRUE:conditionintervals -0.73     0.43
## evidence:sd_diff15:conditionintervals -0.53     0.75
## meansTRUE:sd_diff15:conditionintervals -0.61     0.92

```

```

## evidence:meansTRUE:sd_diff15:conditionintervals      -0.94      0.59
##                                         Eff.Sample Rhat
## Intercept                                         2186 1.00
## evidence                                         2783 1.00
## meansTRUE                                         3995 1.00
## sd_diff15                                         4561 1.00
## conditionintervals                           3269 1.00
## evidence:meansTRUE                           4561 1.00
## evidence:sd_diff15                           4862 1.00
## meansTRUE:sd_diff15                           4495 1.00
## evidence:conditionintervals                 3323 1.00
## meansTRUE:conditionintervals                 4977 1.00
## sd_diff15:conditionintervals                 4794 1.00
## evidence:meansTRUE:sd_diff15                 4271 1.00
## evidence:meansTRUE:conditionintervals       4756 1.00
## evidence:sd_diff15:conditionintervals       4669 1.00
## meansTRUE:sd_diff15:conditionintervals       4786 1.00
## evidence:meansTRUE:sd_diff15:conditionintervals 5327 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

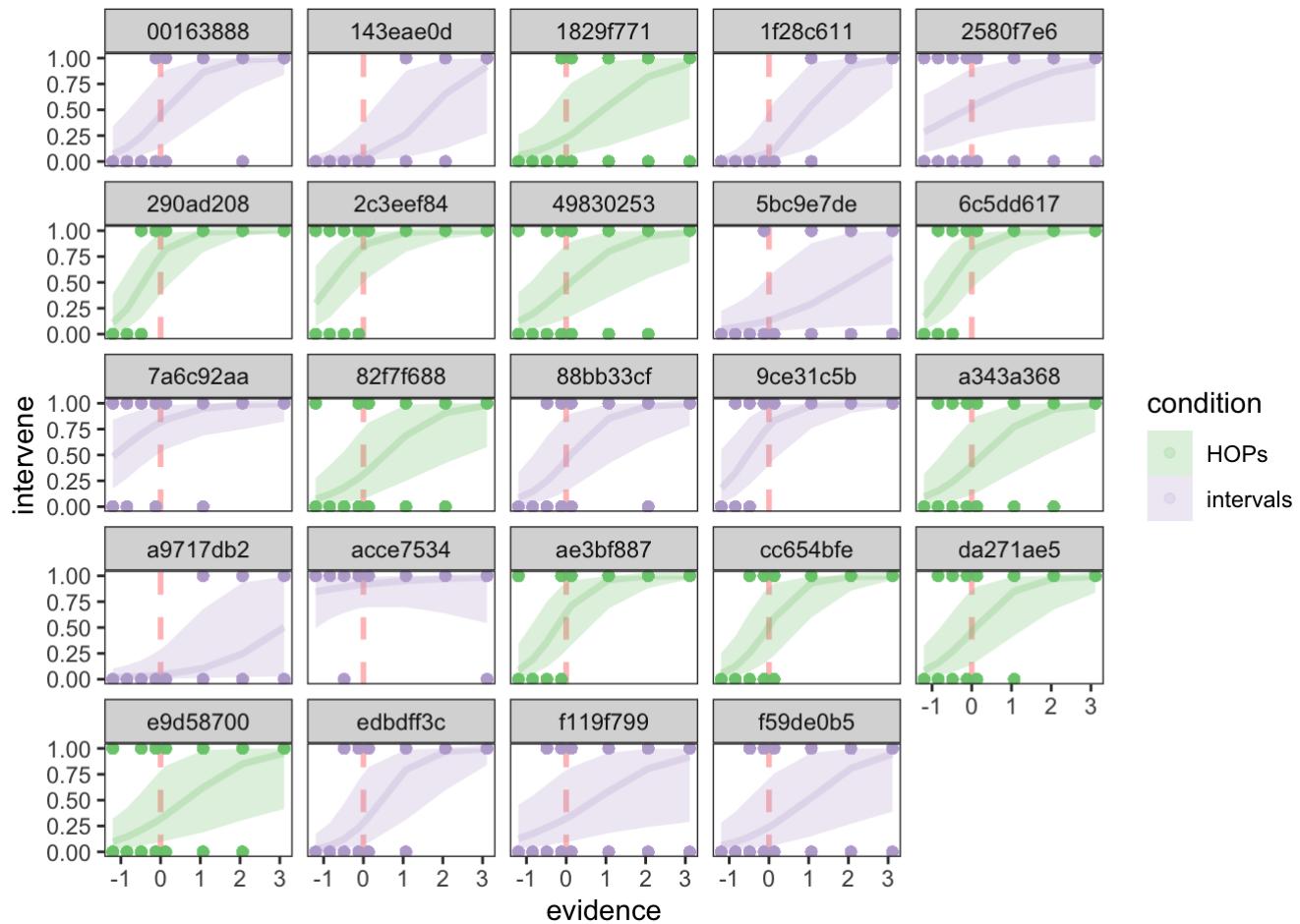
```

Let's take a look at the estimated psychometric functions for each worker.

```

model_df %>%
  group_by(evidence, worker_id, means, sd_diff, condition) %>%
  add_fitted_draws(m.wrkr.means.sd.vis.logistic, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = condition, fill = condition)) +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
  stat_lineribbon(aes(y = pf), .width = c(.95), alpha = .25) +
  geom_point(alpha = .15) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),
                  ylim = quantile(model_df$intervene, c(0, 1))) +
  theme_bw() +
  theme(panel.grid = element_blank()) +
  facet_wrap(. ~ worker_id)

```



What do the posteriors for just-noticeable differences (JND) and points of subjective equality (PSE) look like? We'll start by getting the slopes and intercepts of the linear model and using these to derive estimates of the JND and PSE for each fitted draw.

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(means, sd_diff, condition) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.means.sd.vis.logistic, re_formula = NA, scale = "linear") %>%
  compare_levels(.value, by = evidence) %>%
  rename(slope = .value)

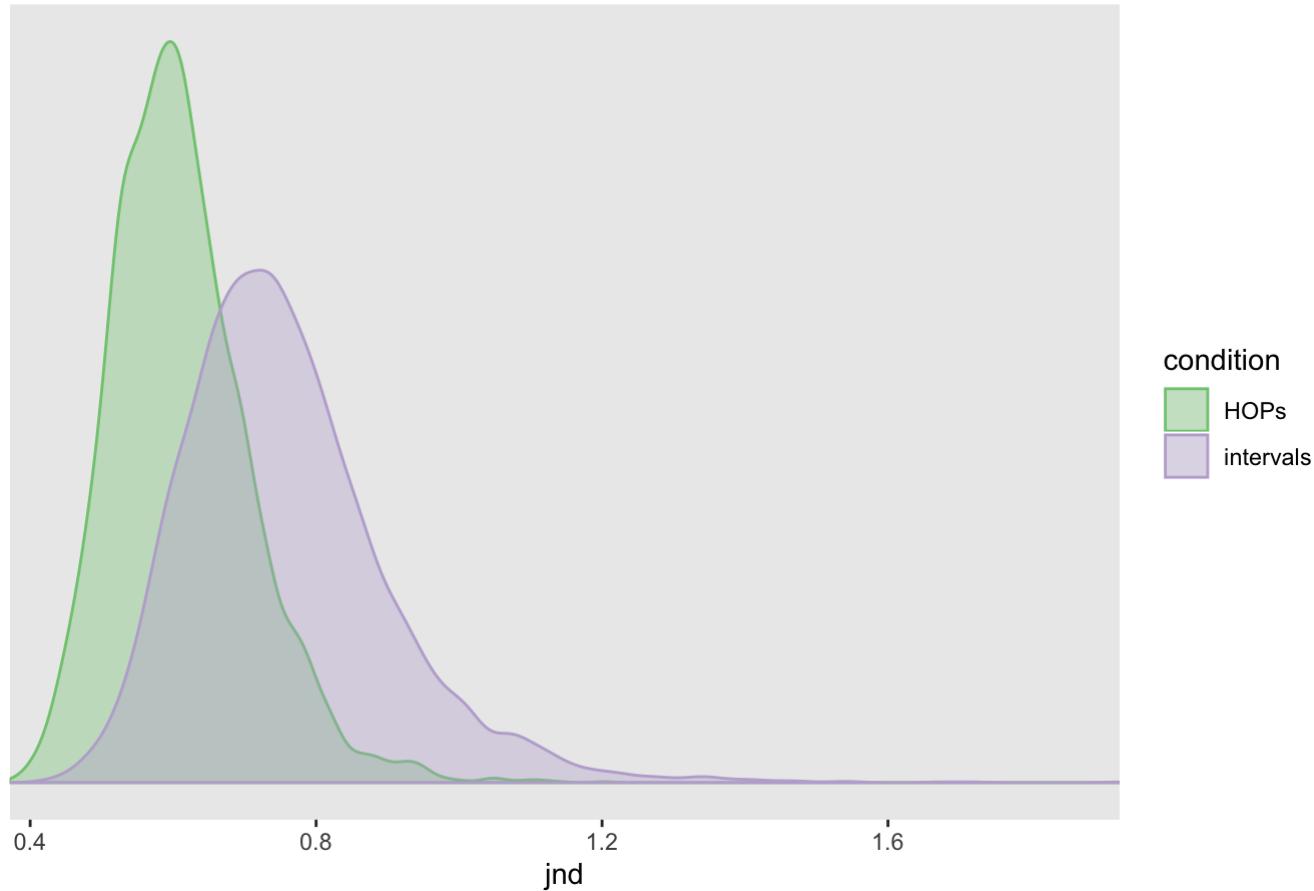
# get intercepts from linear model
intercepts_df <- model_df %>%
  group_by(means, sd_diff, condition) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.means.sd.vis.logistic, re_formula = NA, scale = "linear") %>%
  rename(intercept = .value)

# join dataframes for slopes and intercepts, calculate PSE and JND
stats_df <- slopes_df %>%
  full_join(intercepts_df, by = c("means", "sd_diff", "condition", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )
```

First, let's look at the estimates of JNDs per visualization, marginalizing across other manipulations.

```
stats_df %>%
  group_by(condition, .draw) %>%          # marginalize out other manipulations
  summarise(jnd = weighted.mean(jnd)) %>%
  ggplot(aes(x = jnd, group = condition, color = condition, fill = condition)) +
  geom_density(alpha = 0.35) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per visualization") +
  theme(panel.grid = element_blank())
```

Posterior JND per visualization

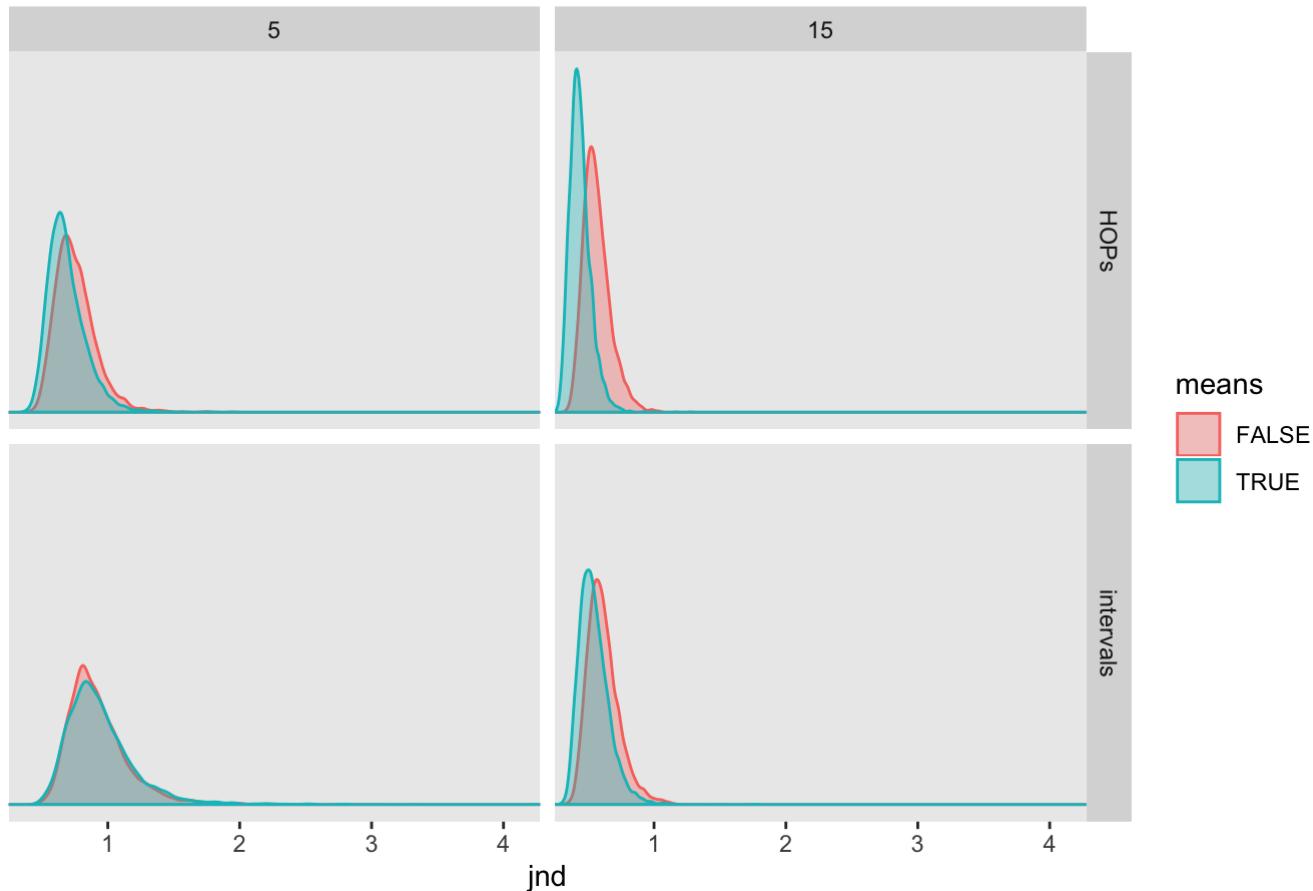


It looks like users are slightly more sensitive to evidence (i.e., JNDs are slightly smaller) in the HOPs condition.

What if we look at the full interaction effect on JNDs?

```
stats_df %>%
  ggplot(aes(x = jnd, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  theme(panel.grid = element_blank()) +
  facet_grid(condition ~ sd_diff)
```

## Posterior JND per condition

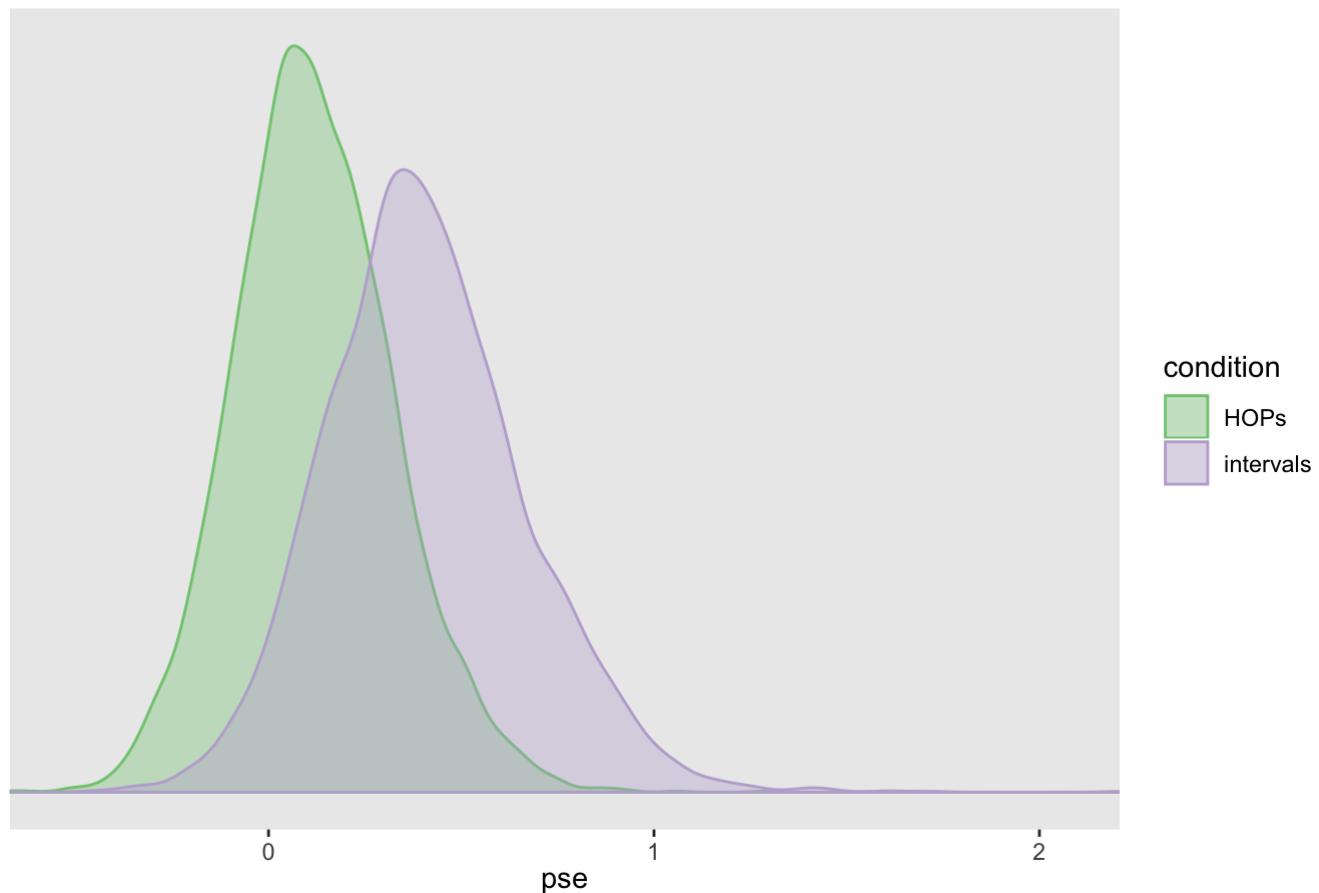


It seems like means help improve sensitivity to evidence a little bit, especially with HOPs when uncertainty is high.

Next, we'll look at the point of subjective equality for each visualization condition, marginalizing out other manipulations.

```
stats_df %>%
  group_by(condition, .draw) %>%          # marginalize out other manipulations
  summarise(pse = weighted.mean(pse)) %>%
  ggplot(aes(x = pse, group = condition, color = condition, fill = condition)) +
  geom_density(alpha = 0.35) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per visualization") +
  theme(panel.grid = element_blank())
```

### Posterior PSE per visualization

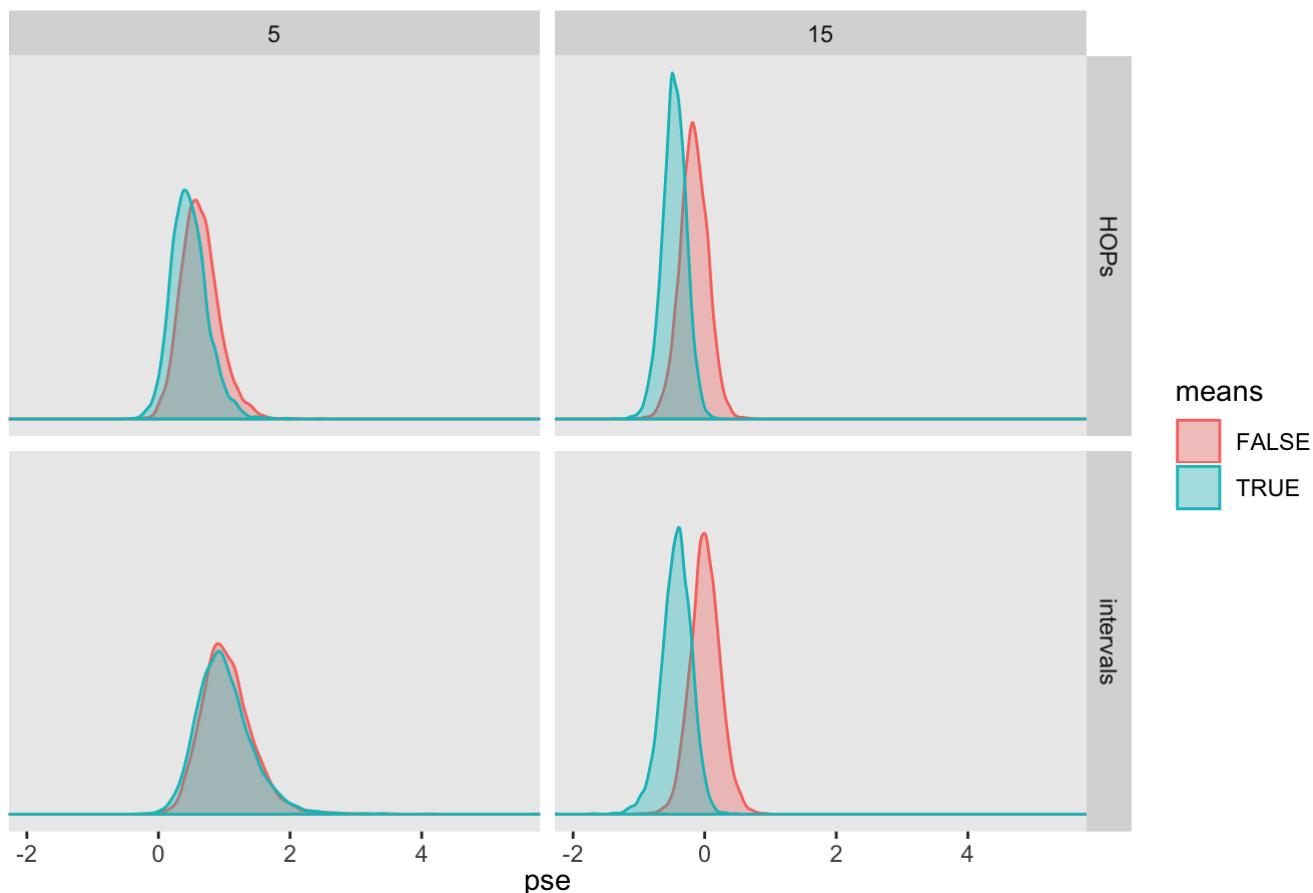


It looks like the point of subjective equality is least biased with HOPs.

What if we look at the full interaction effect on PSE?

```
stats_df %>%
  ggplot(aes(x = pse, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  theme(panel.grid = element_blank()) +
  facet_grid(condition ~ sd_diff)
```

## Posterior JND per condition



In both conditions, adding means seems to bias people against intervention relative to the utility optimal strategy when uncertainty is high.

## Building Up Random Effects for Within-Subjects Manipulations

In the minimal model to answer our research questions above, estimates for the effect of means are noisier than we would like. We'll try to better account for heterogeneity across subjects by adding more random effects to our model for each within subjects manipulation.

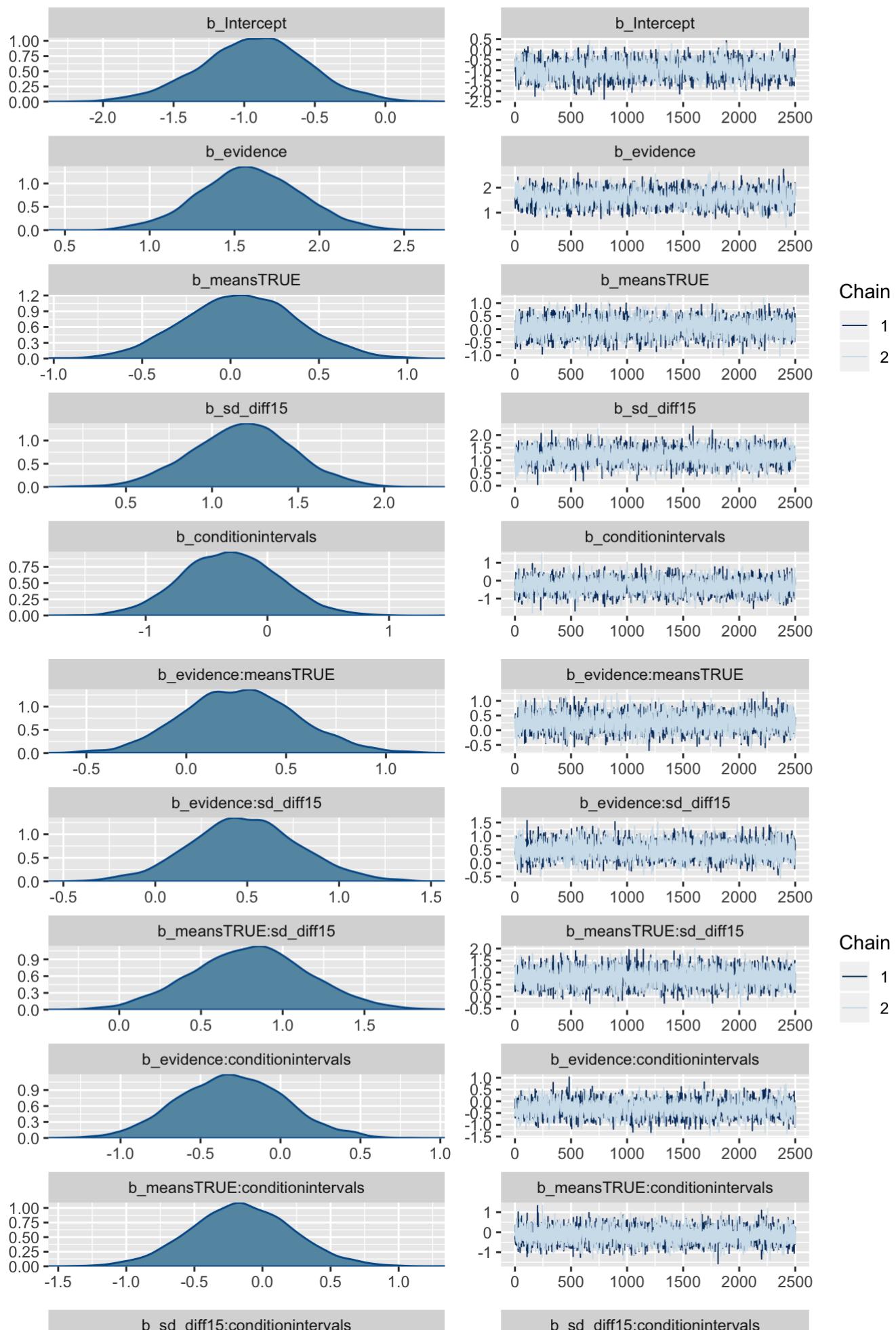
Following a principle of model expansion, we will make this changes cumulatively. We include a series of model specifications that capture plausible structure in the data and fit without any sampling issues.

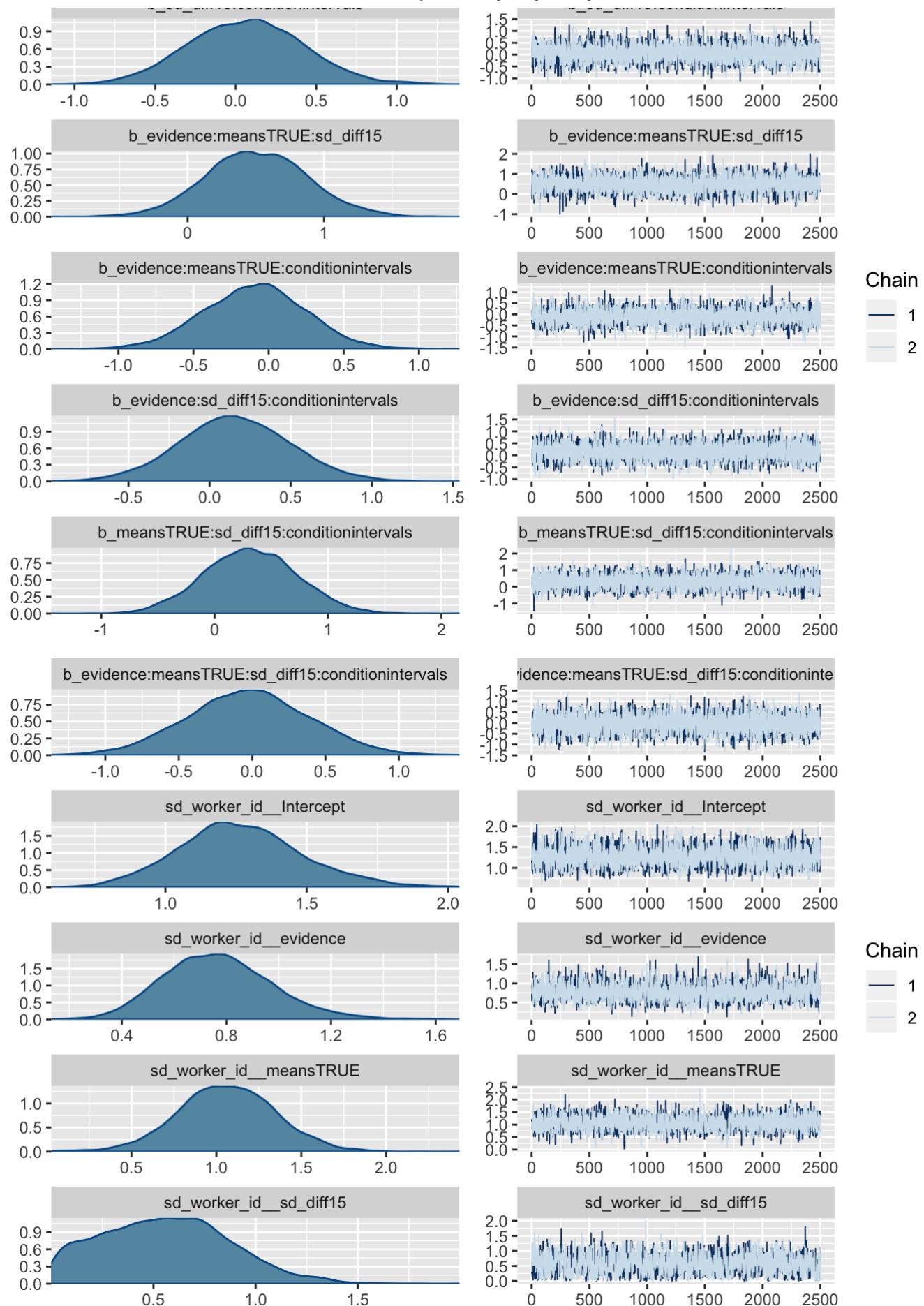
## Random Effects for the Interaction of Means and Uncertainty Shown

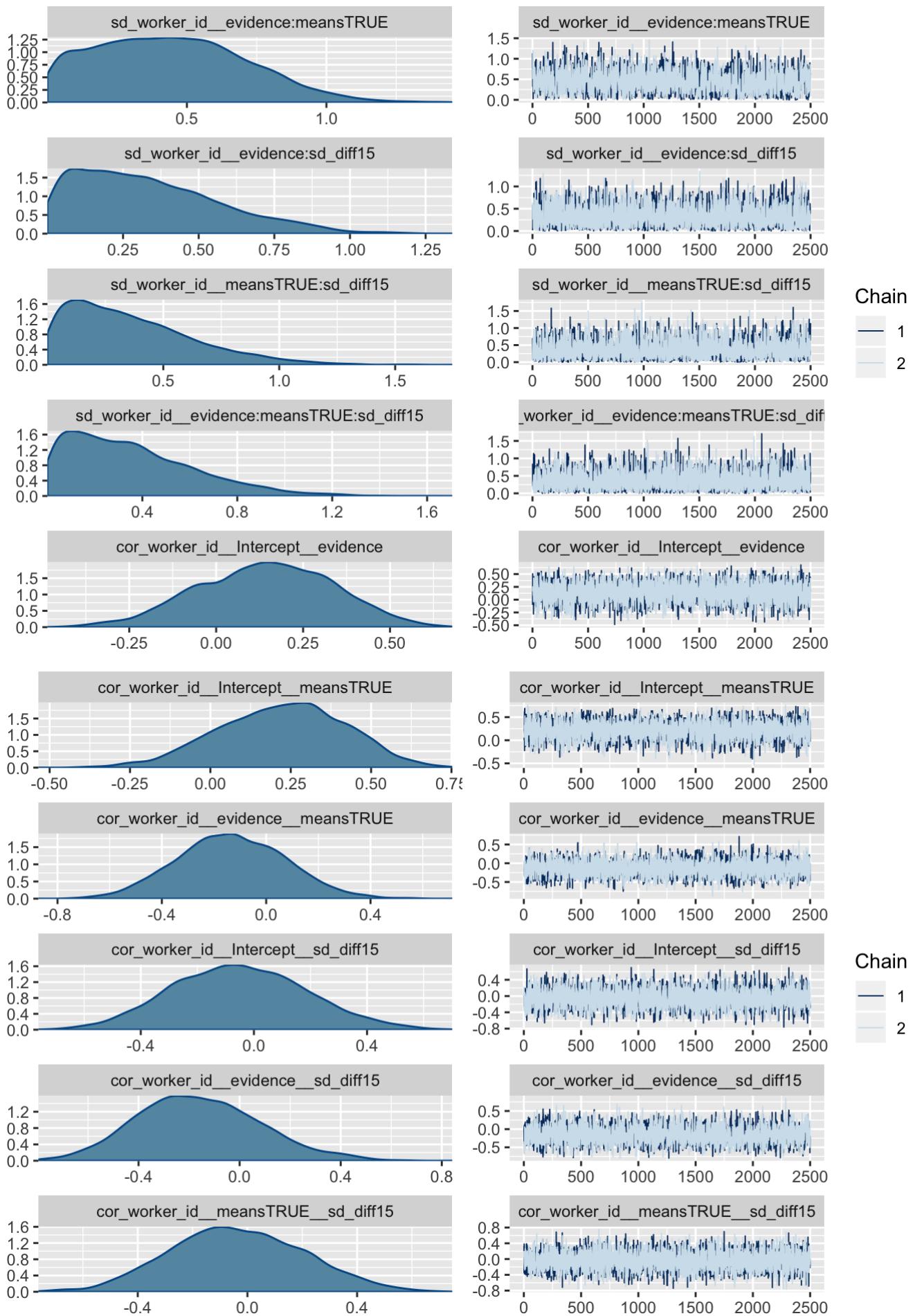
This model adds random effects for the presence/absence of the mean and the level of uncertainty to mirror the structure of the fixed effects for these variables. This tells our model that each individual can respond differently to these factors, which may help to parse heterogeneity in responses.

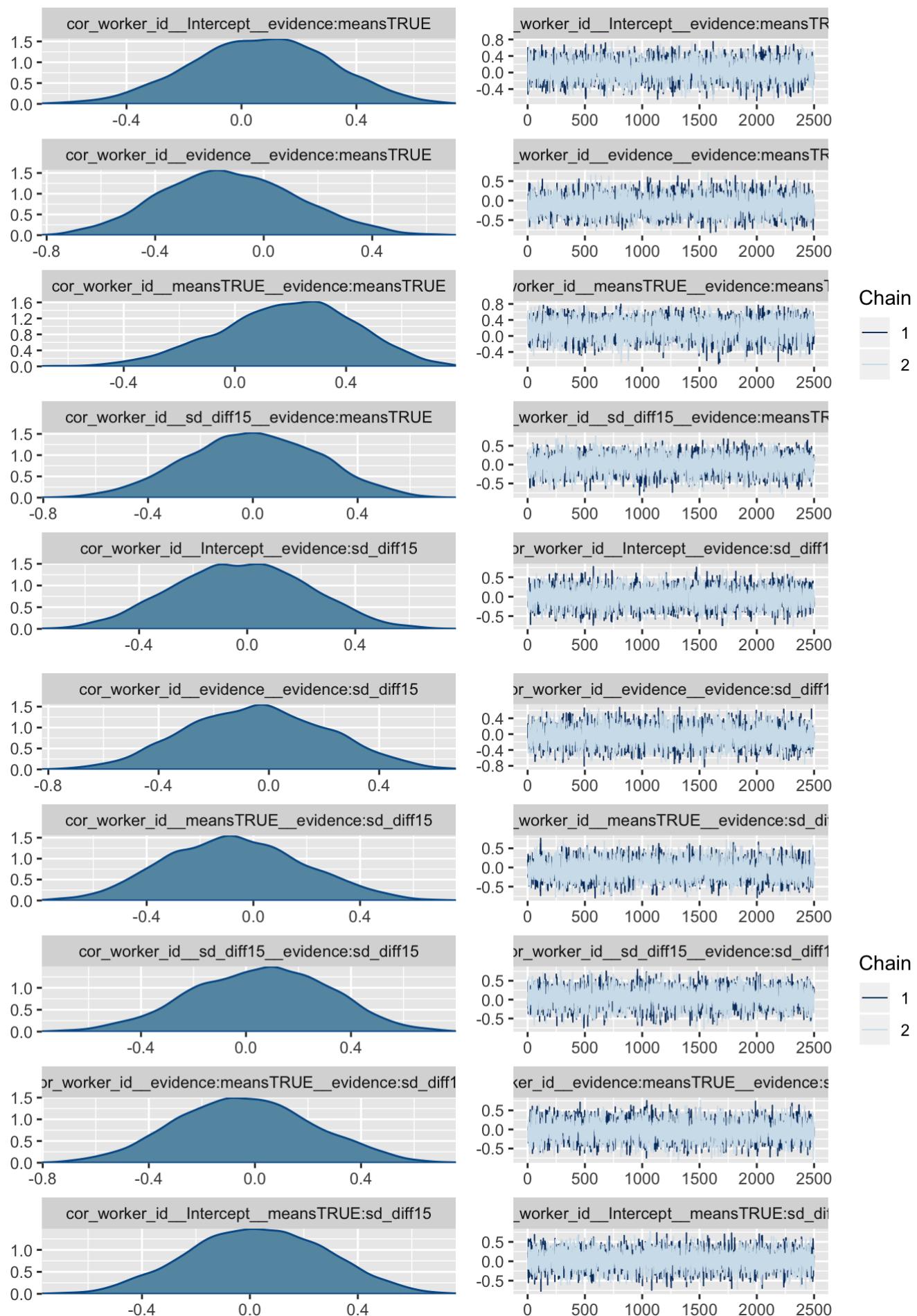
```
# hierarchical logistic regression
m.wrkr.means.sd.vis.logistic.r.means.sd <- brm(
  data = model_df, family = bernoulli(link = "logit"),
  formula = bf(intervene ~ (1 + evidence*means*sd_diff|worker_id) + evidence*means*sd_di
ff*condition),
  prior = c(prior(normal(0, 1), class = Intercept),
            prior(normal(1, 1), class = b, coef = evidence),
            prior(normal(0, 0.5), class = b),
            prior(normal(0, 0.5), class = sd),
            prior(lkj(4), class = cor)),
  iter = 3000, warmup = 500, chains = 2, cores = 2,
  file = "model-fits/logistic_mdl-wrkr_means_sd_vis-r_means_sd")
```

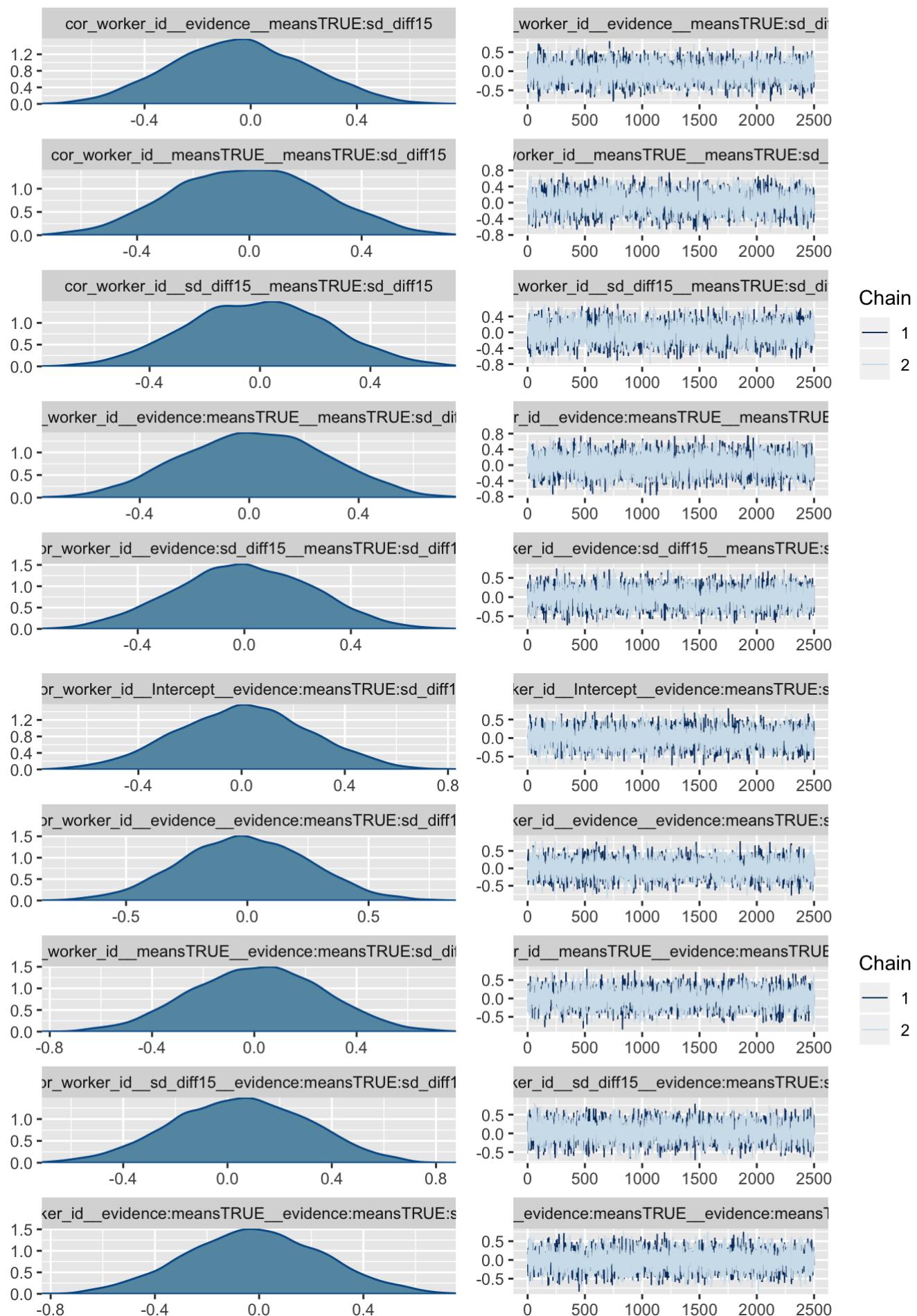
```
# trace plots
plot(m.wrkr.means.sd.vis.logistic.r.means.sd)
```

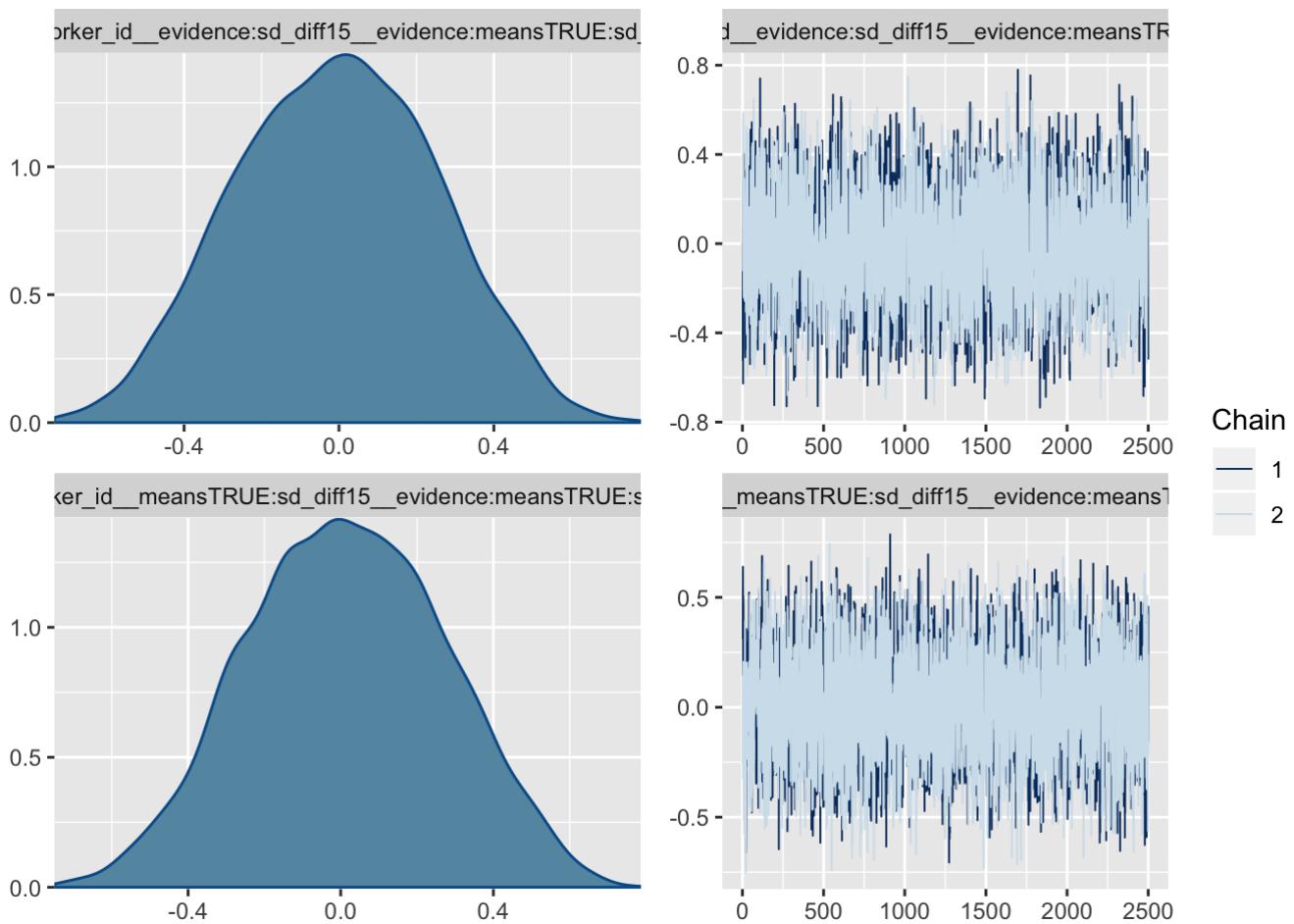












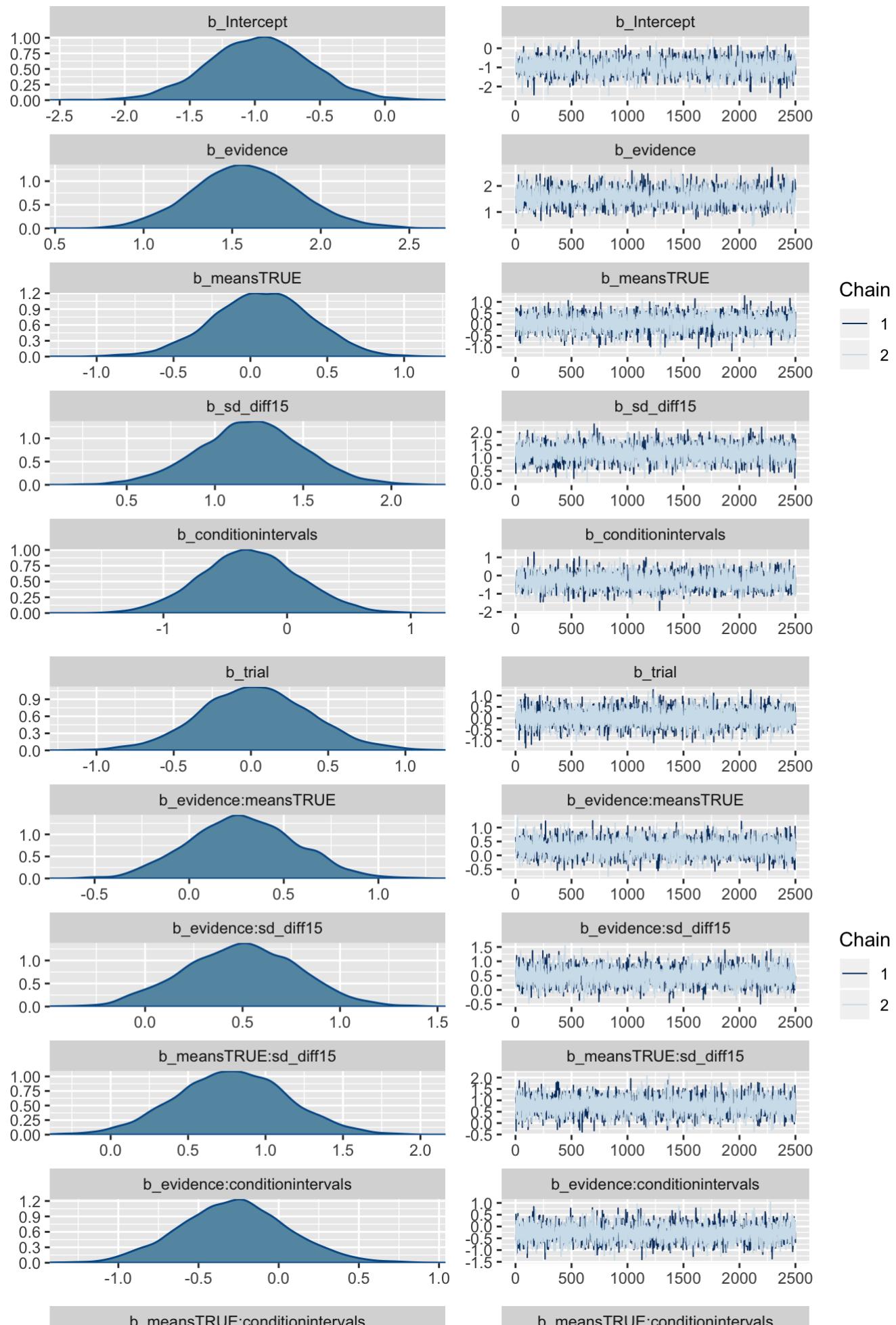
## Mixed Effects of Trial Order on JND and PSE

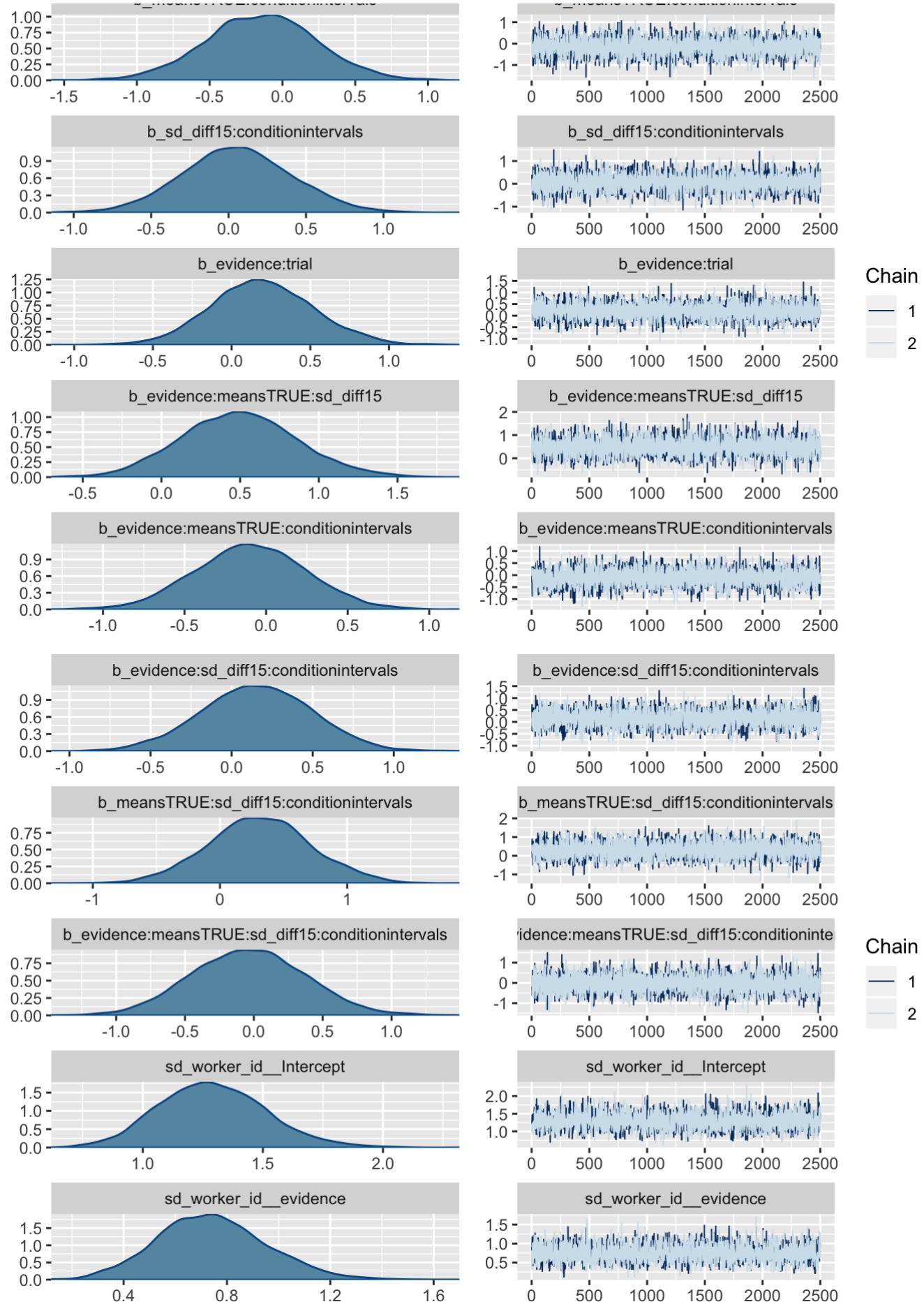
This model adds mixed effects of trial order on JNDs and PSEs. This functionally adds a learning component to our model of the latent sense of utility on which decisions are based.

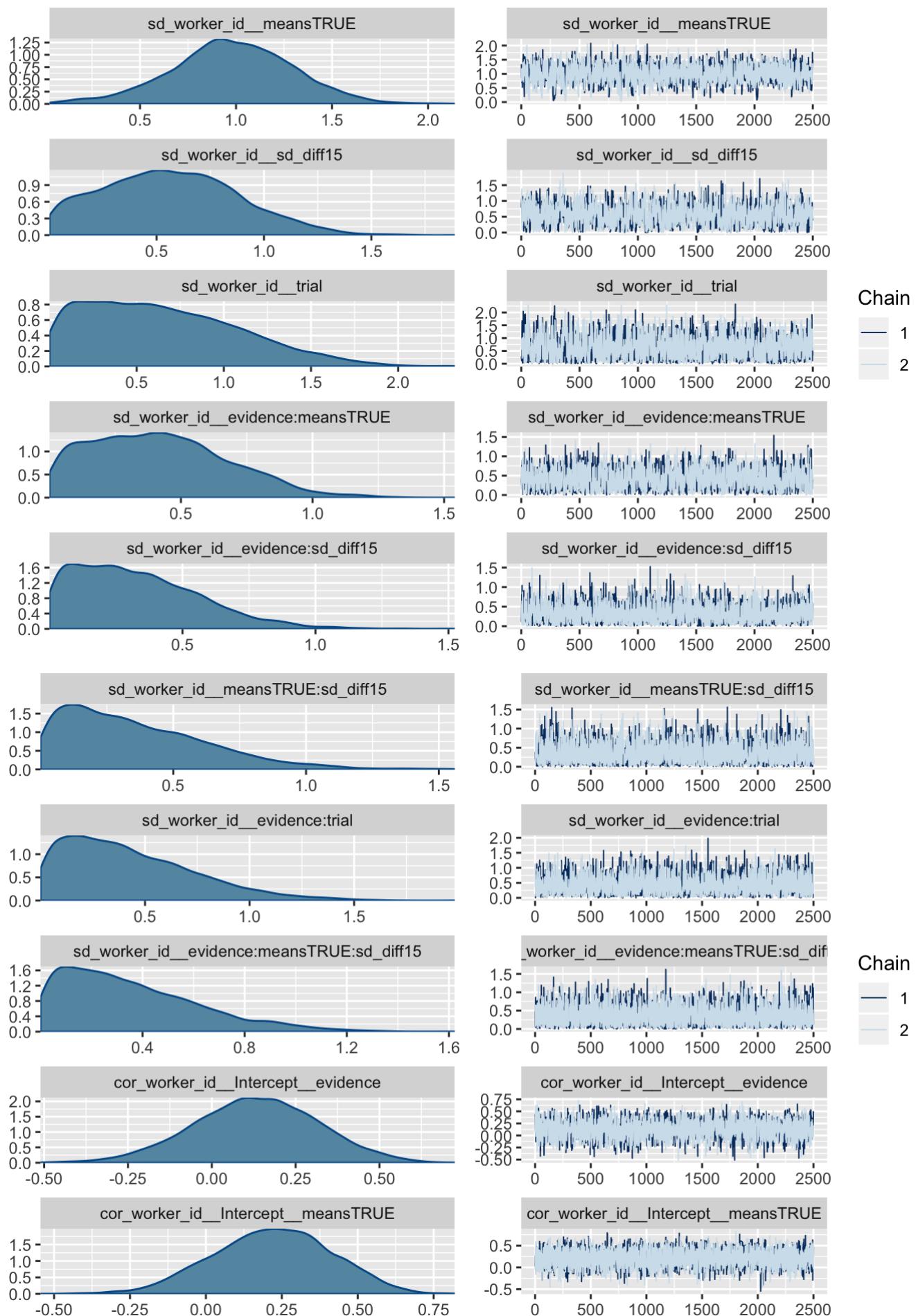
```
# hierarchical logistic regression
m.wrkr.means.sd.vis.logistic.r.means.sd.trial <- brm(
  data = model_df, family = bernoulli(link = "logit"),
  formula = bf(intervene ~ (1 + evidence*means*sd_diff + evidence*trial|worker_id) + evi
  dence*means*sd_diff*condition + evidence*trial),
  prior = c(prior(normal(0, 1), class = Intercept),
            prior(normal(1, 1), class = b, coef = evidence),
            prior(normal(0, 0.5), class = b),
            prior(normal(0, 0.5), class = sd),
            prior(lkj(4), class = cor)),
  iter = 3000, warmup = 500, chains = 2, cores = 2,
  file = "model-fits/logistic_mdl-wrkr_means_sd_vis-r_means_sd_trial")
```

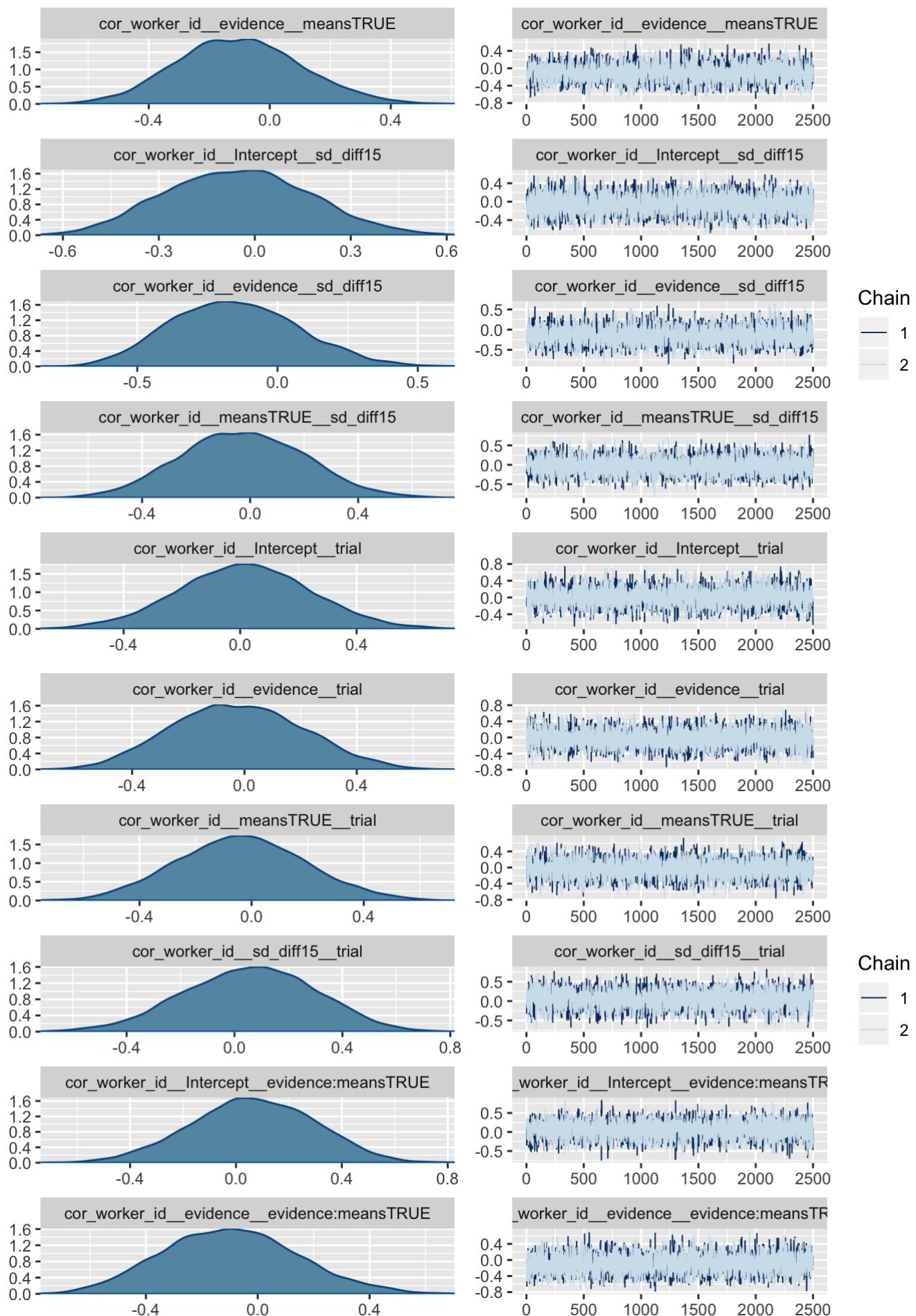
```
# trace plots
plot(m.wrkr.means.sd.vis.logistic.r.means.sd.trial)
```

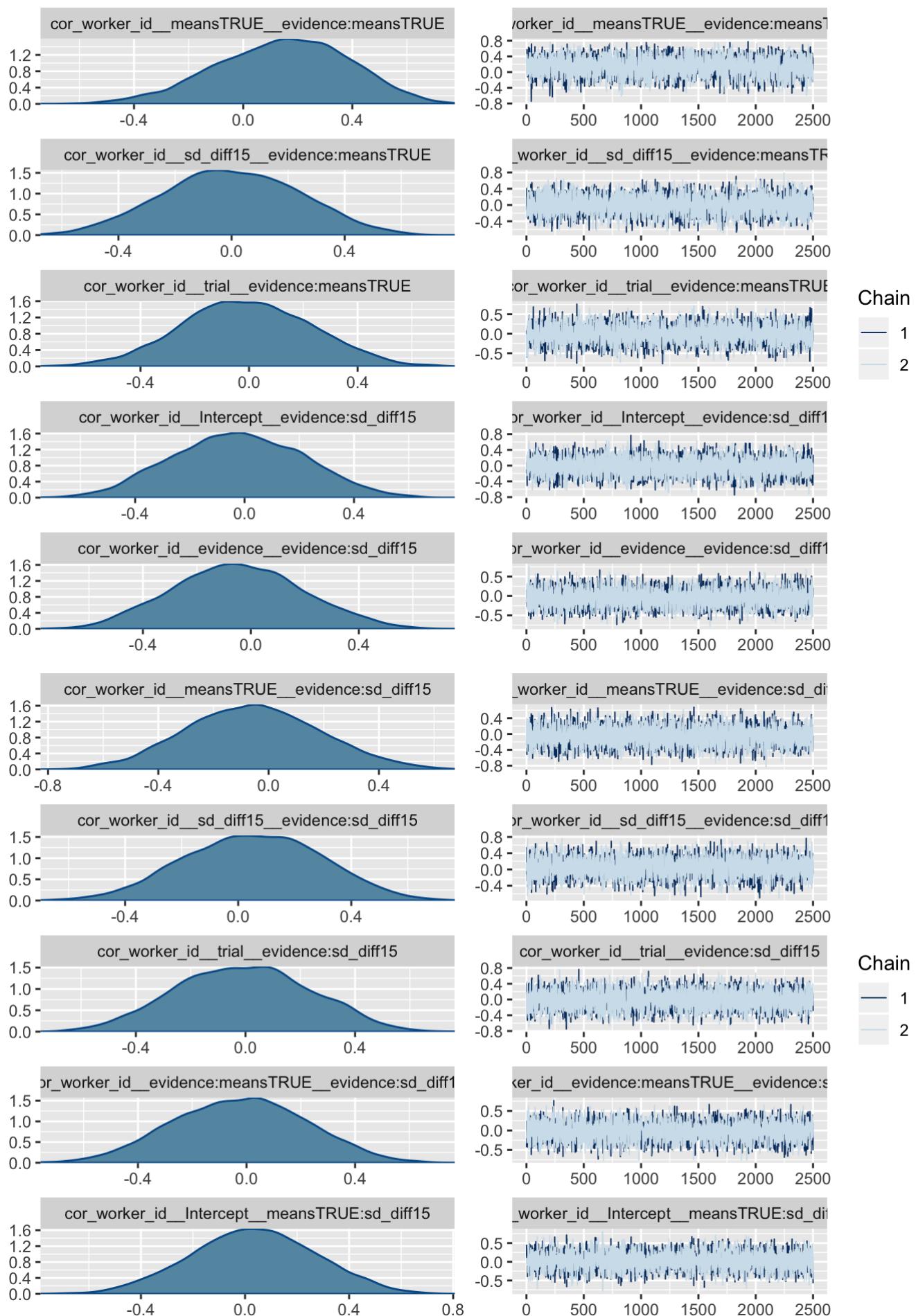


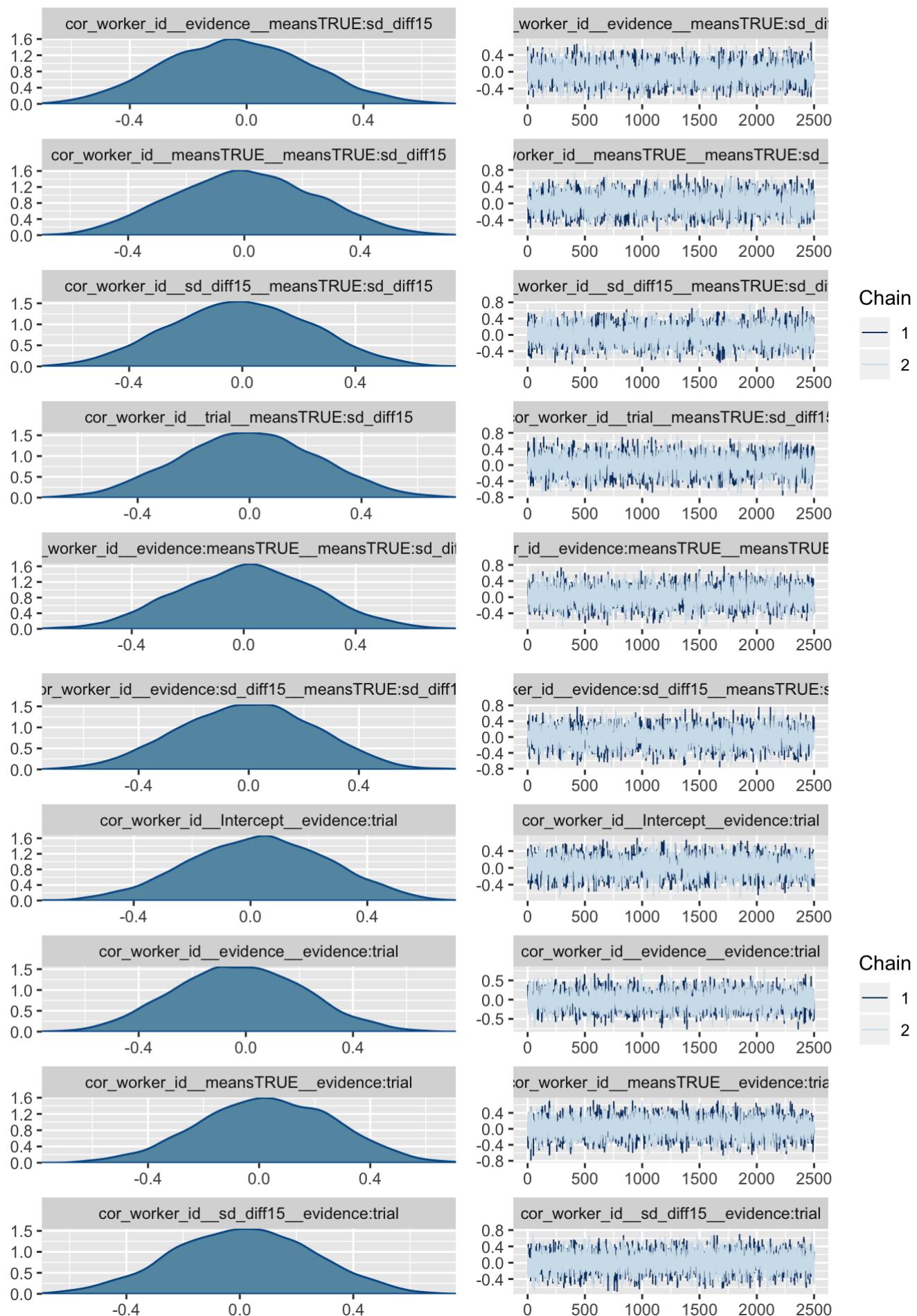


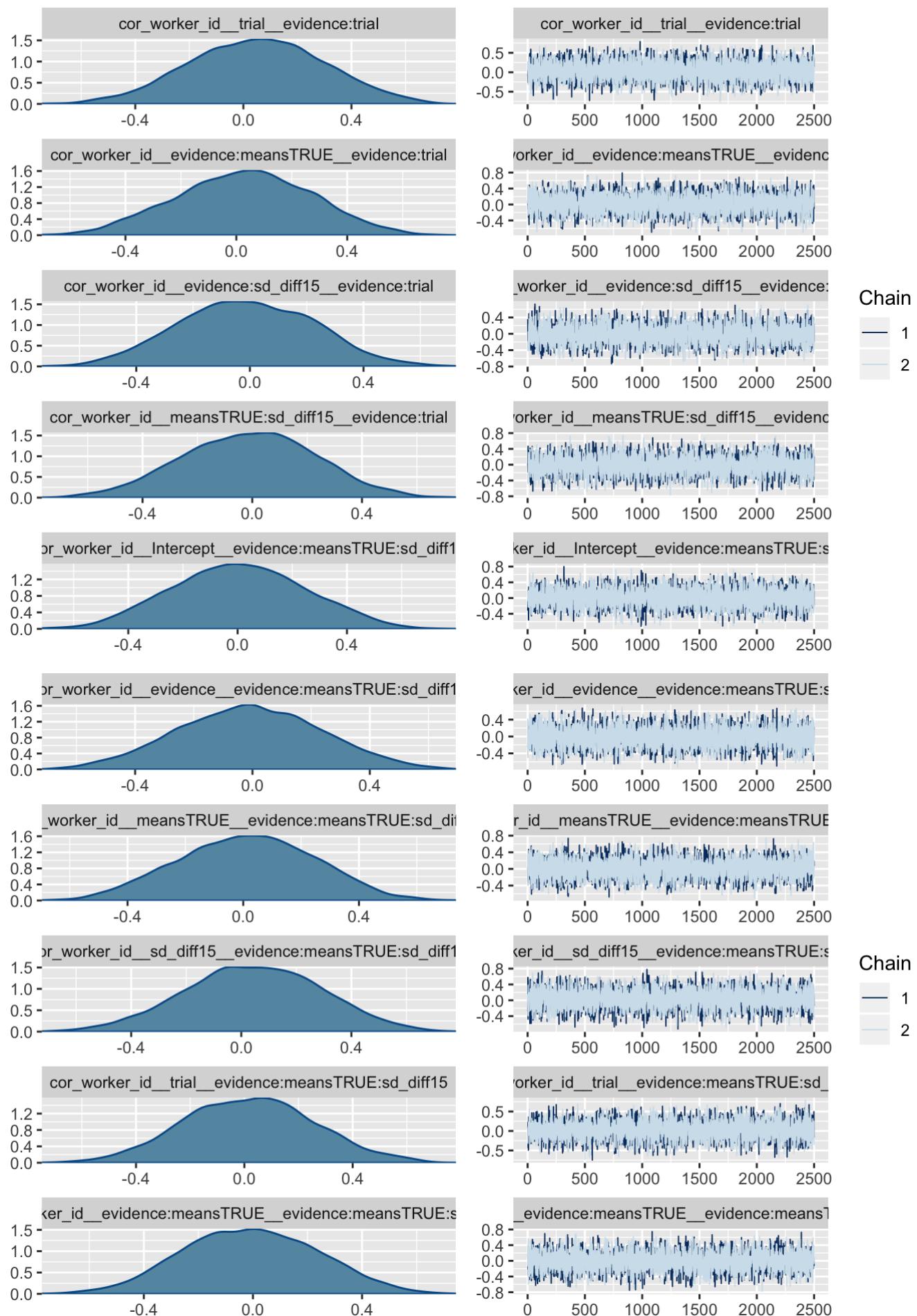


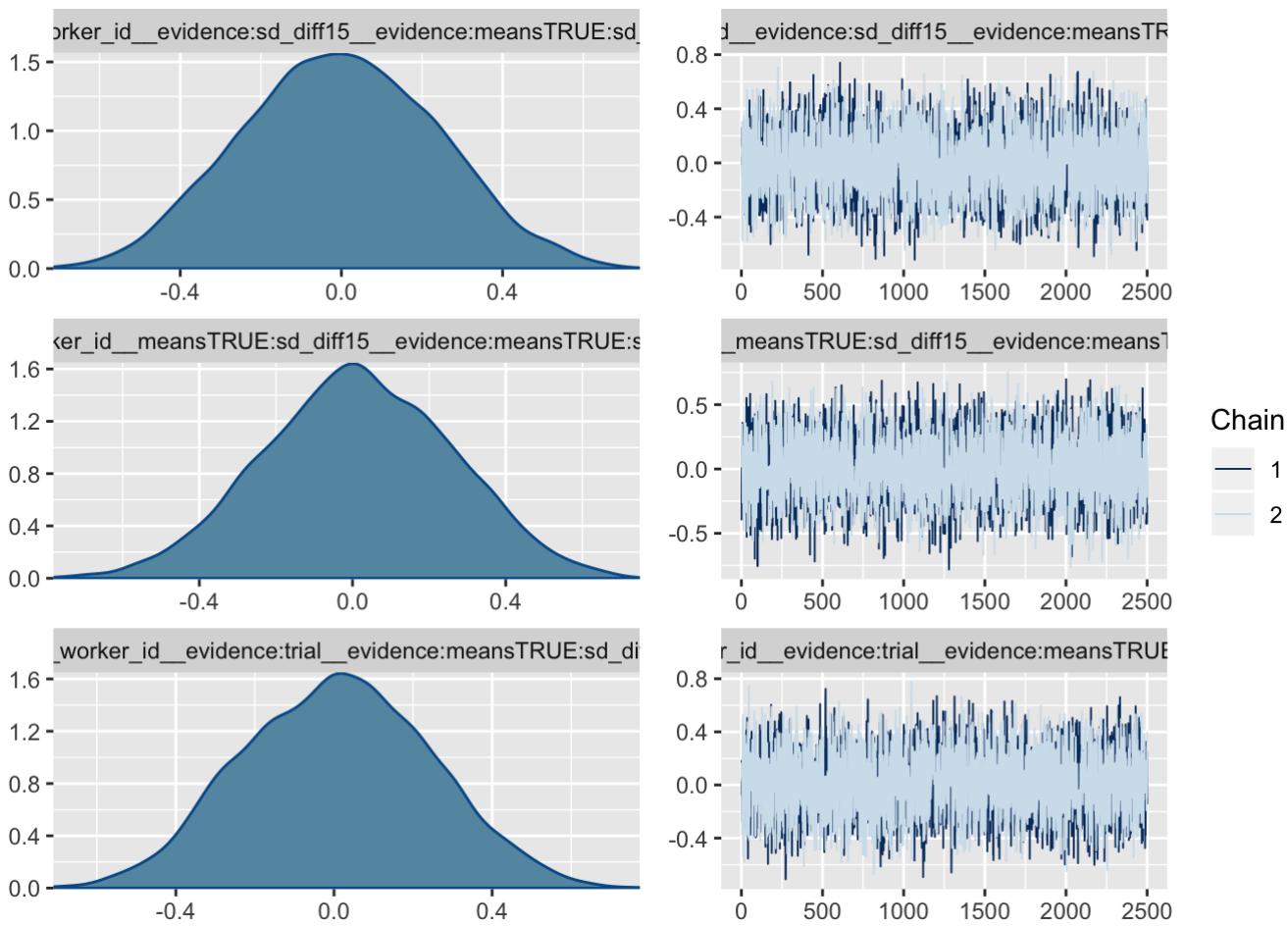












## Model Comparison

Each time we add a random effect, the number of parameters multiplies. We want to make sure these parameters are contributing to the predictive validity of the model more than they risk overfitting. We'll evaluate this by using WAIC to compare models. Whichever model has the smallest value of WAIC is the one that has the best predictive validity for the fewest parameters.

```
waic(m.wrkr.means.sd.vis.logistic, m.wrkr.means.sd.vis.logistic.r.means.sd, m.wrkr.mean
s.sd.vis.logistic.r.means.sd.trial)
```

```

##  

WAIC  

## m.wrkr.means.sd.vis.logistic  

593.03  

## m.wrkr.means.sd.vis.logistic.r.means.sd  

560.67  

## m.wrkr.means.sd.vis.logistic.r.means.sd.trial  

560.77  

## m.wrkr.means.sd.vis.logistic - m.wrkr.means.sd.vis.logistic.r.means.sd  

32.36  

## m.wrkr.means.sd.vis.logistic - m.wrkr.means.sd.vis.logistic.r.means.sd.trial  

32.27  

## m.wrkr.means.sd.vis.logistic.r.means.sd - m.wrkr.means.sd.vis.logistic.r.means.sd.tri  

al -0.09  

##  

SE  

## m.wrkr.means.sd.vis.logistic  

30.38  

## m.wrkr.means.sd.vis.logistic.r.means.sd  

28.73  

## m.wrkr.means.sd.vis.logistic.r.means.sd.trial  

28.59  

## m.wrkr.means.sd.vis.logistic - m.wrkr.means.sd.vis.logistic.r.means.sd  

7.29  

## m.wrkr.means.sd.vis.logistic - m.wrkr.means.sd.vis.logistic.r.means.sd.trial  

7.58  

## m.wrkr.means.sd.vis.logistic.r.means.sd - m.wrkr.means.sd.vis.logistic.r.means.sd.tri  

al 1.66

```

It looks like predictors for trial order don't make much of a difference in WAIC. We'll leave that effect in our models going forward in the spirit of expanding our model to represent a maximal amount of structure in the data.

## Add Predictor for Block Order

Does it matter whether participants start the study using means? Here we add fixed effects to our model to look into this possibility.

Again, we'll use the same priors as we did for our previous models. Now, let's fit our model.

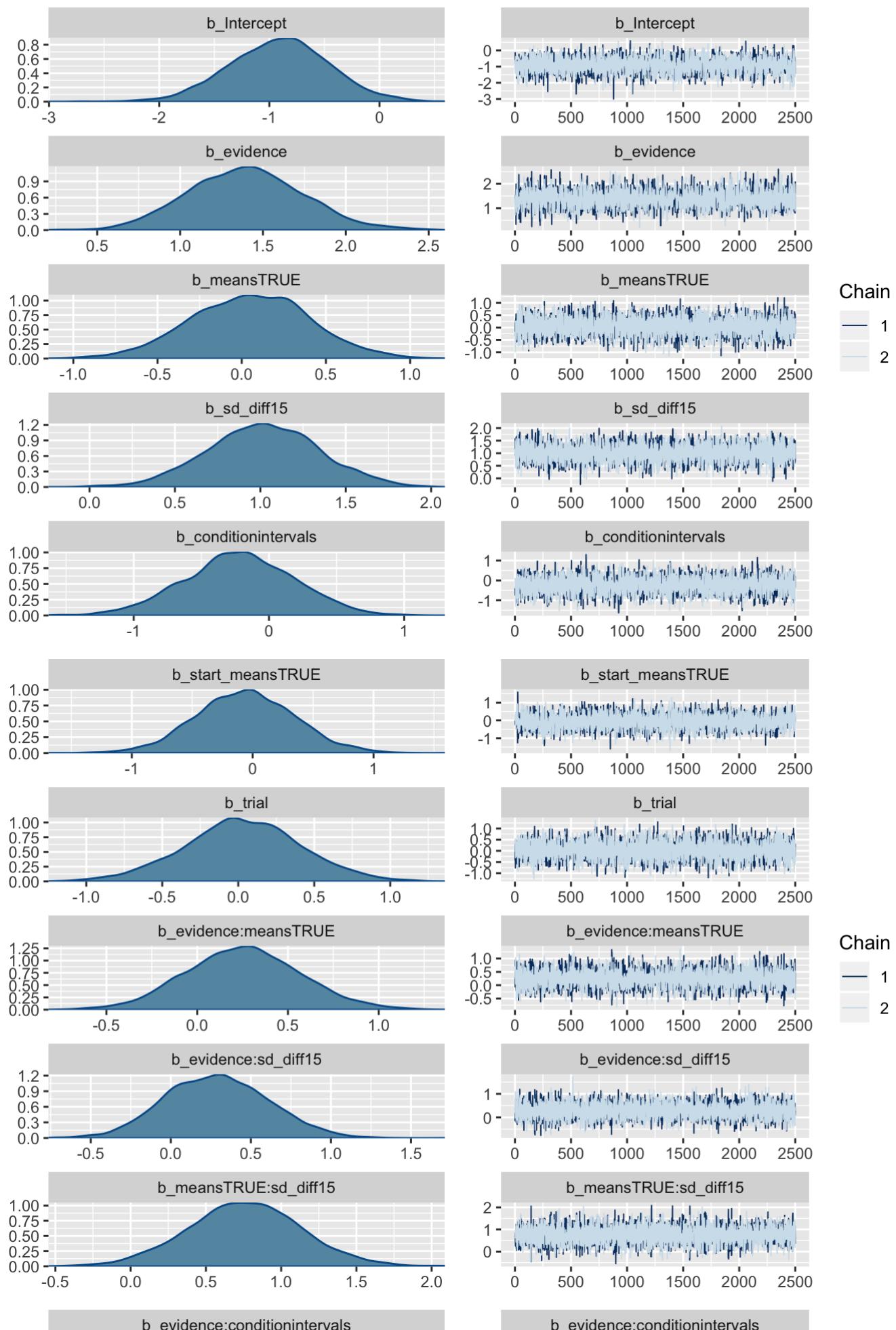
```

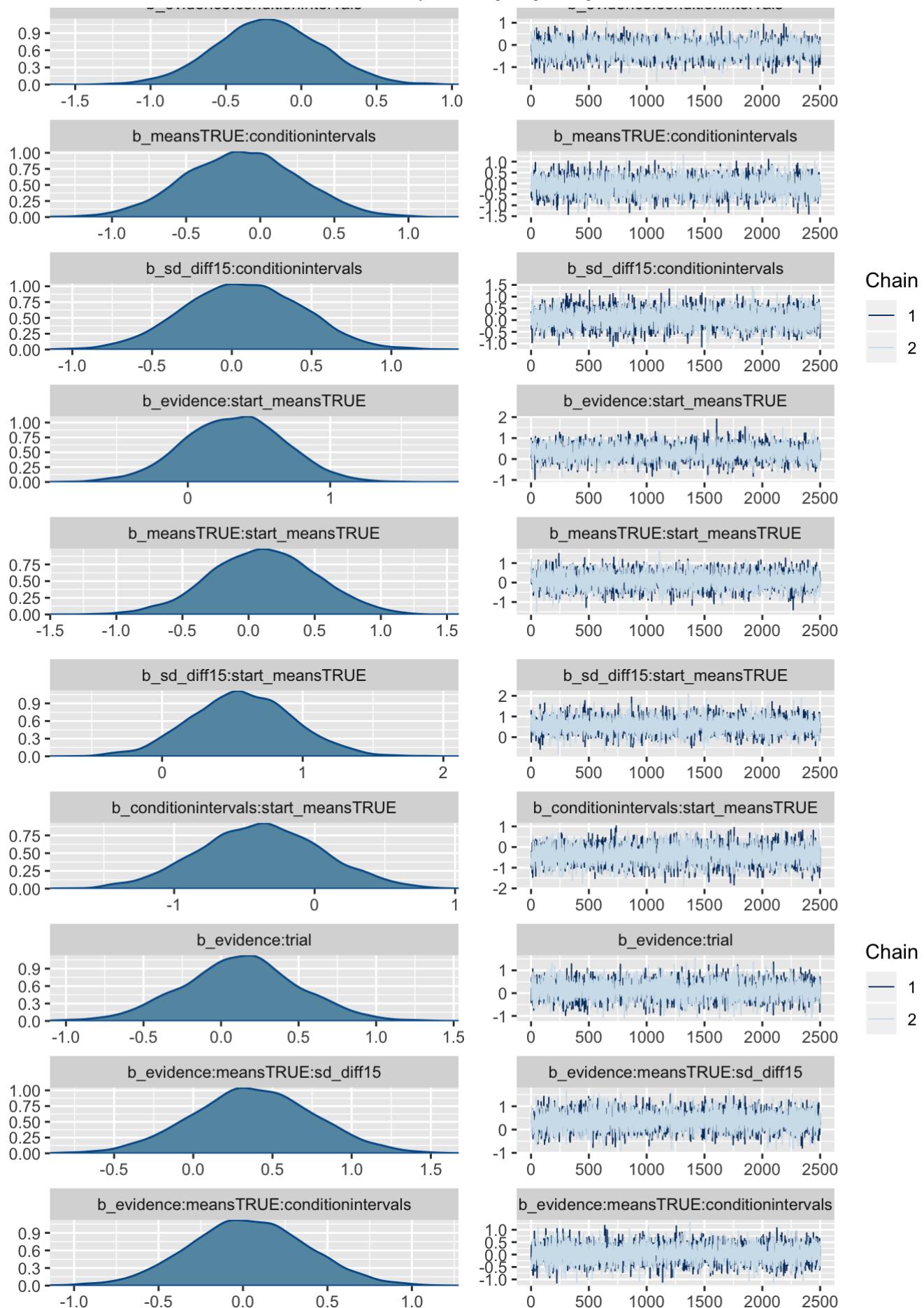
# hierarchical logistic regression
m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial <- brm(
  data = model_df, family = bernoulli(link = "logit"),
  formula = bf(intervene ~ (1 + evidence*means*sd_diff + evidence*trial|worker_id) + evi  

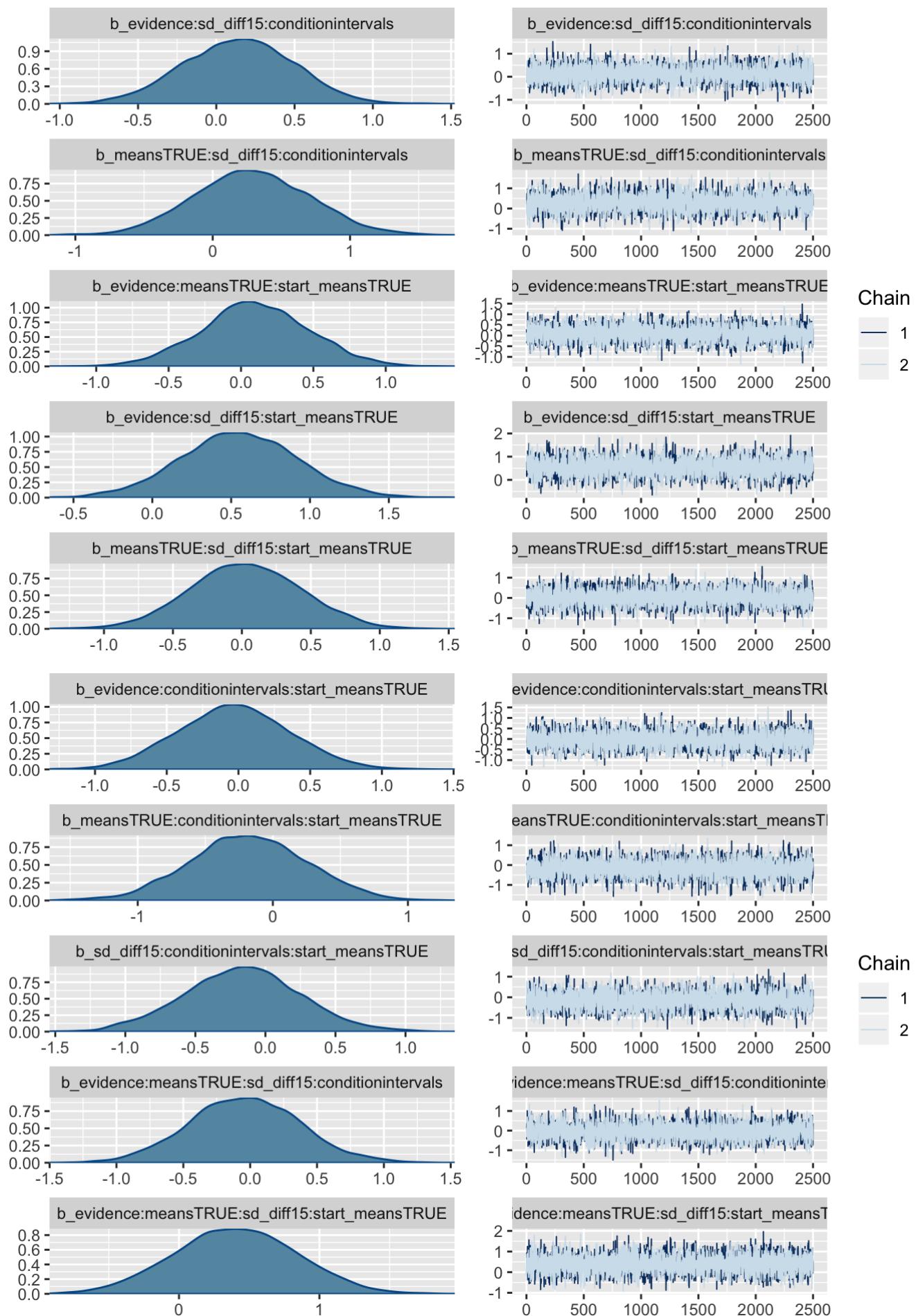
dence*means*sd_diff*condition*start_means + evidence*trial),
  prior = c(prior(normal(0, 1), class = Intercept),
            prior(normal(1, 1), class = b, coef = evidence),
            prior(normal(0, 0.5), class = b),
            prior(normal(0, 0.5), class = sd),
            prior(lkj(4), class = cor)),
  iter = 3000, warmup = 500, chains = 2, cores = 2,
  file = "model-fits/logistic_mdl-wrkr_means_sd_vis_order-r_means_sd_trial")

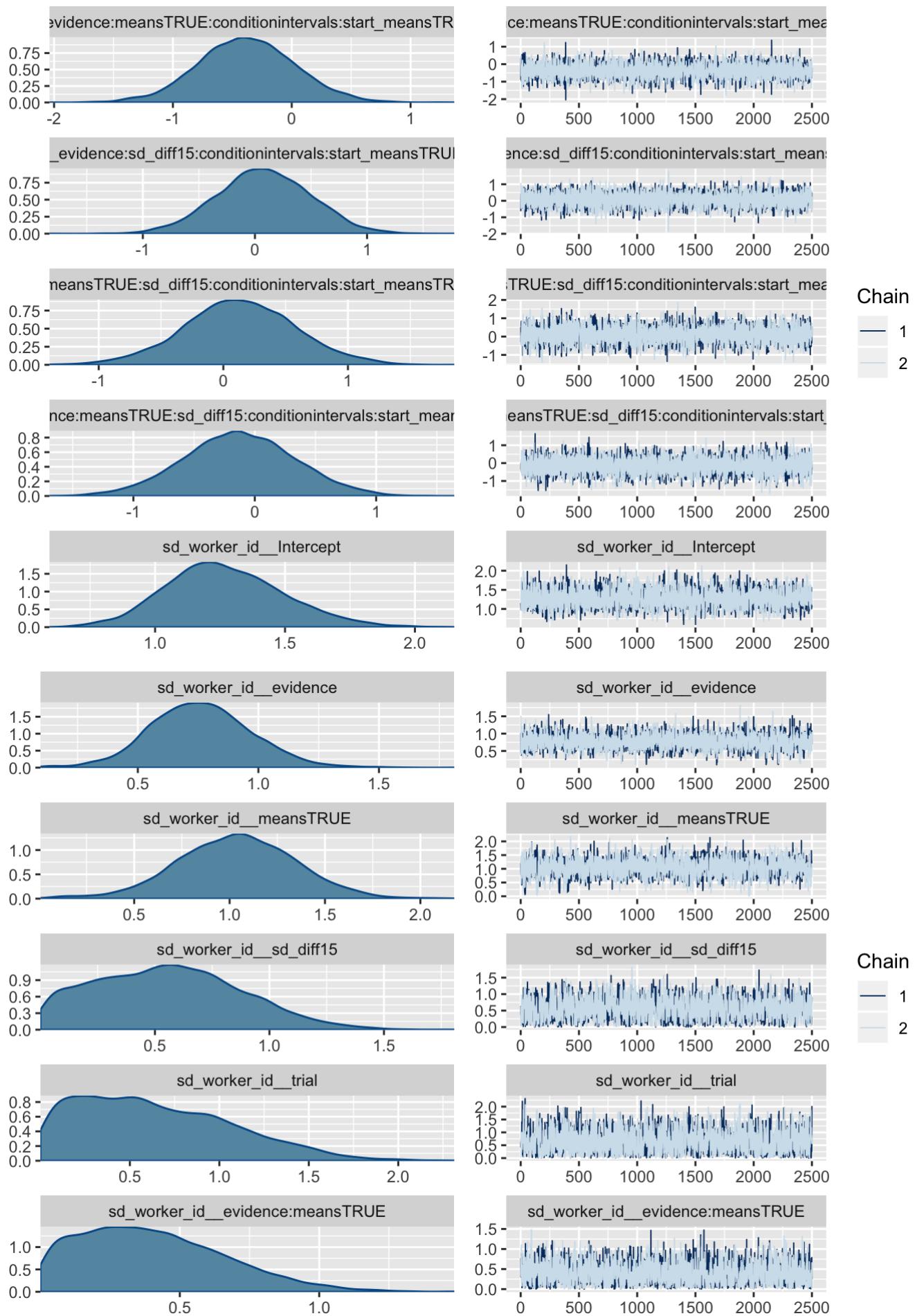
```

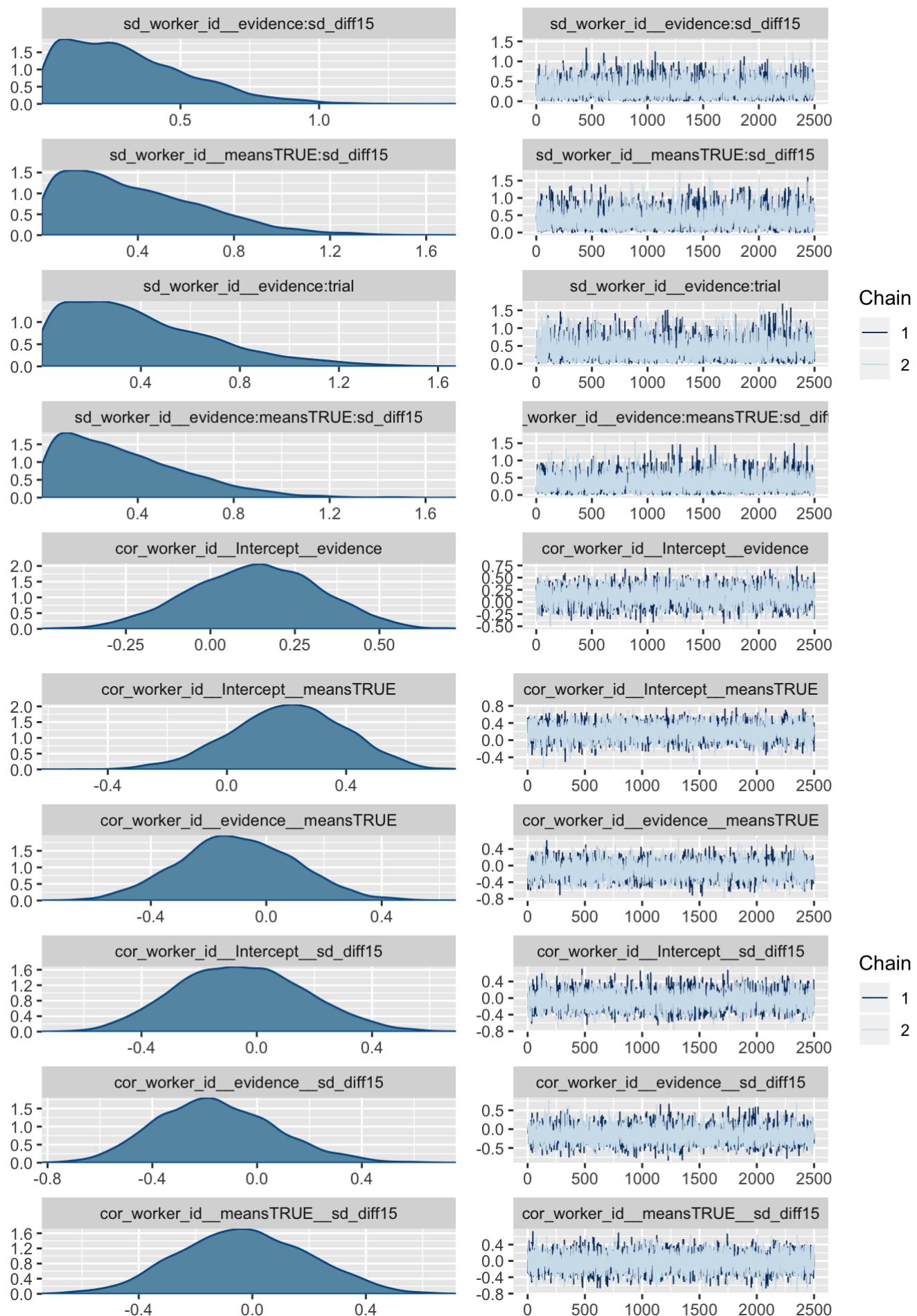
```
# trace plots  
plot(m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial)
```

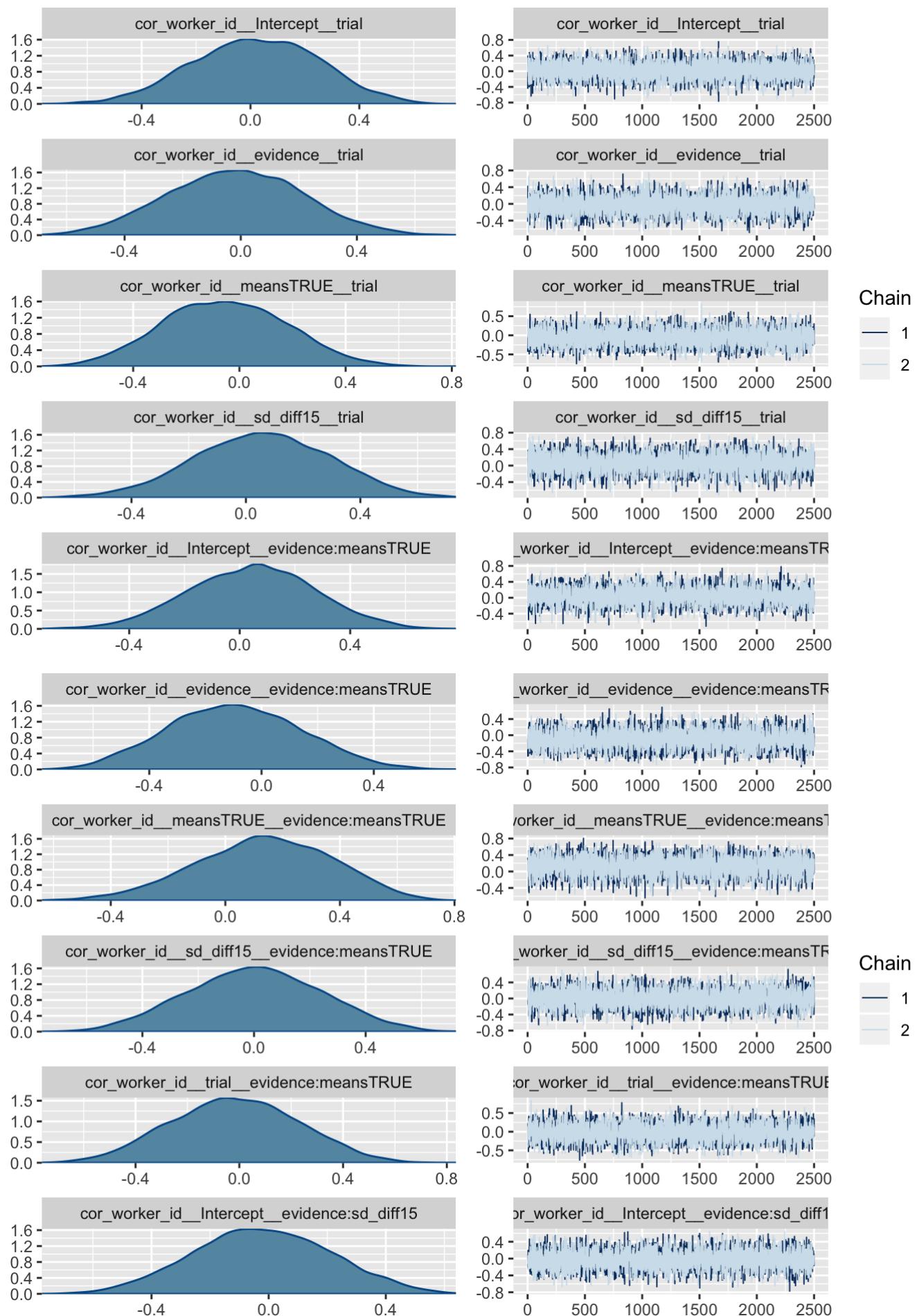


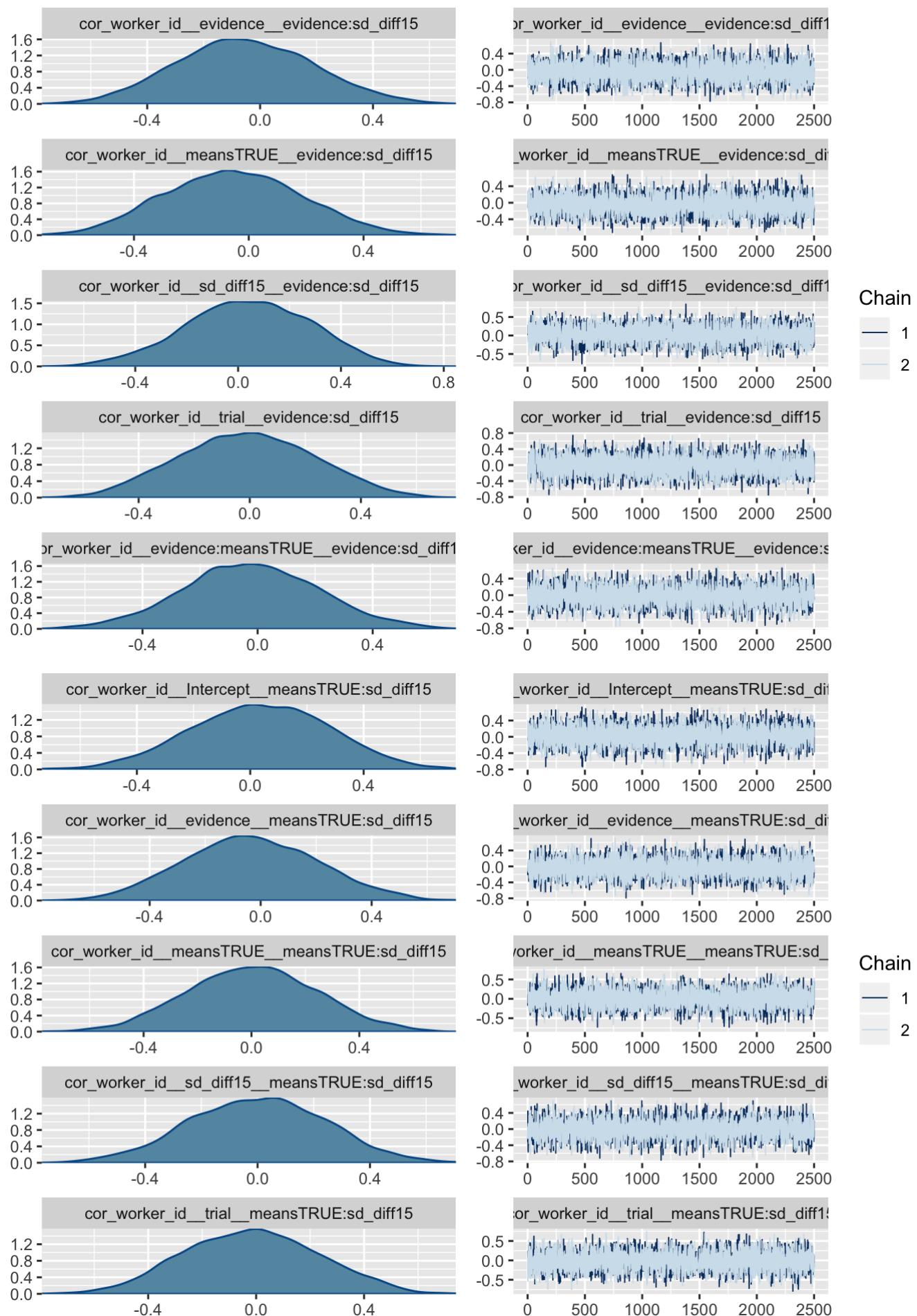


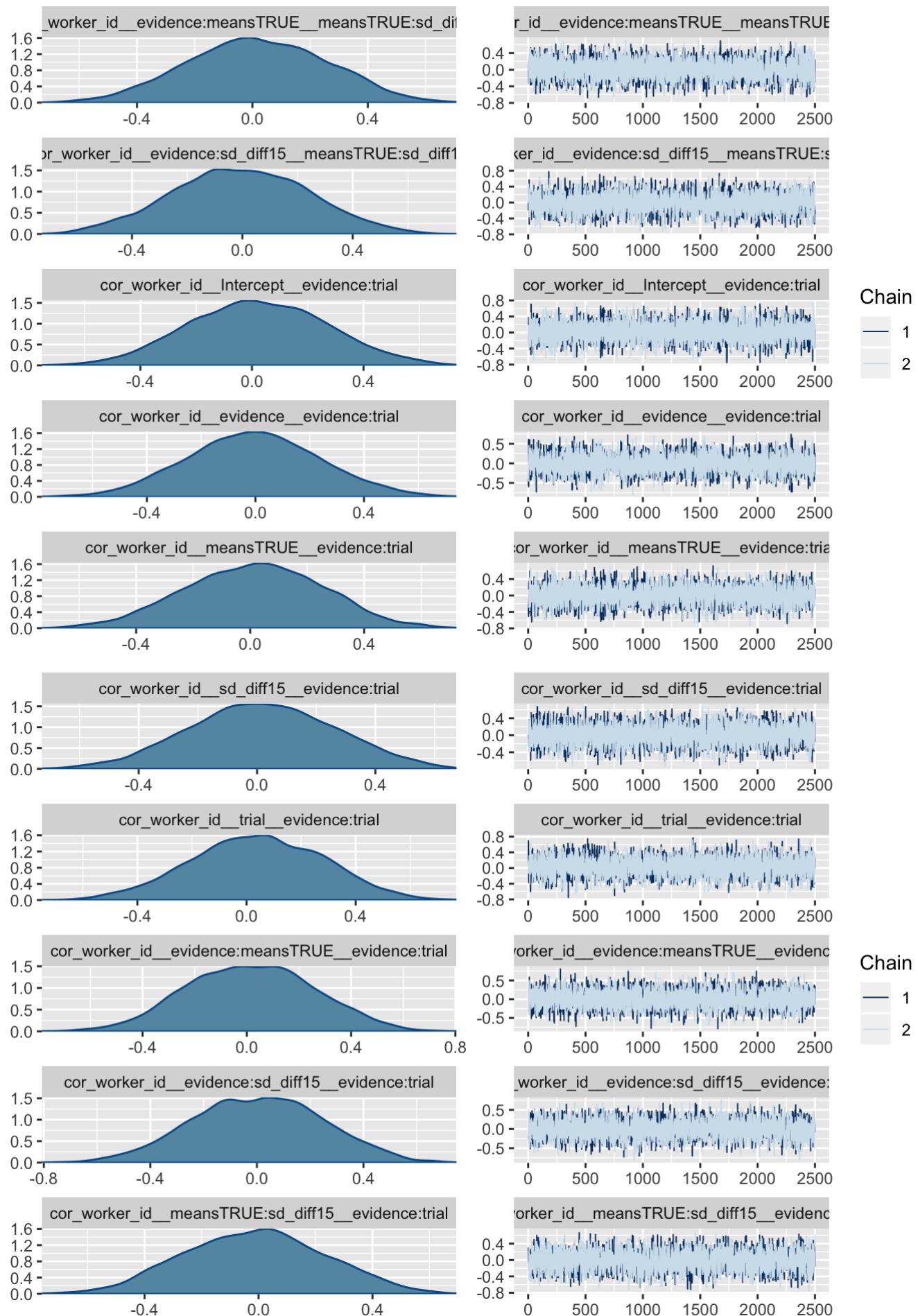


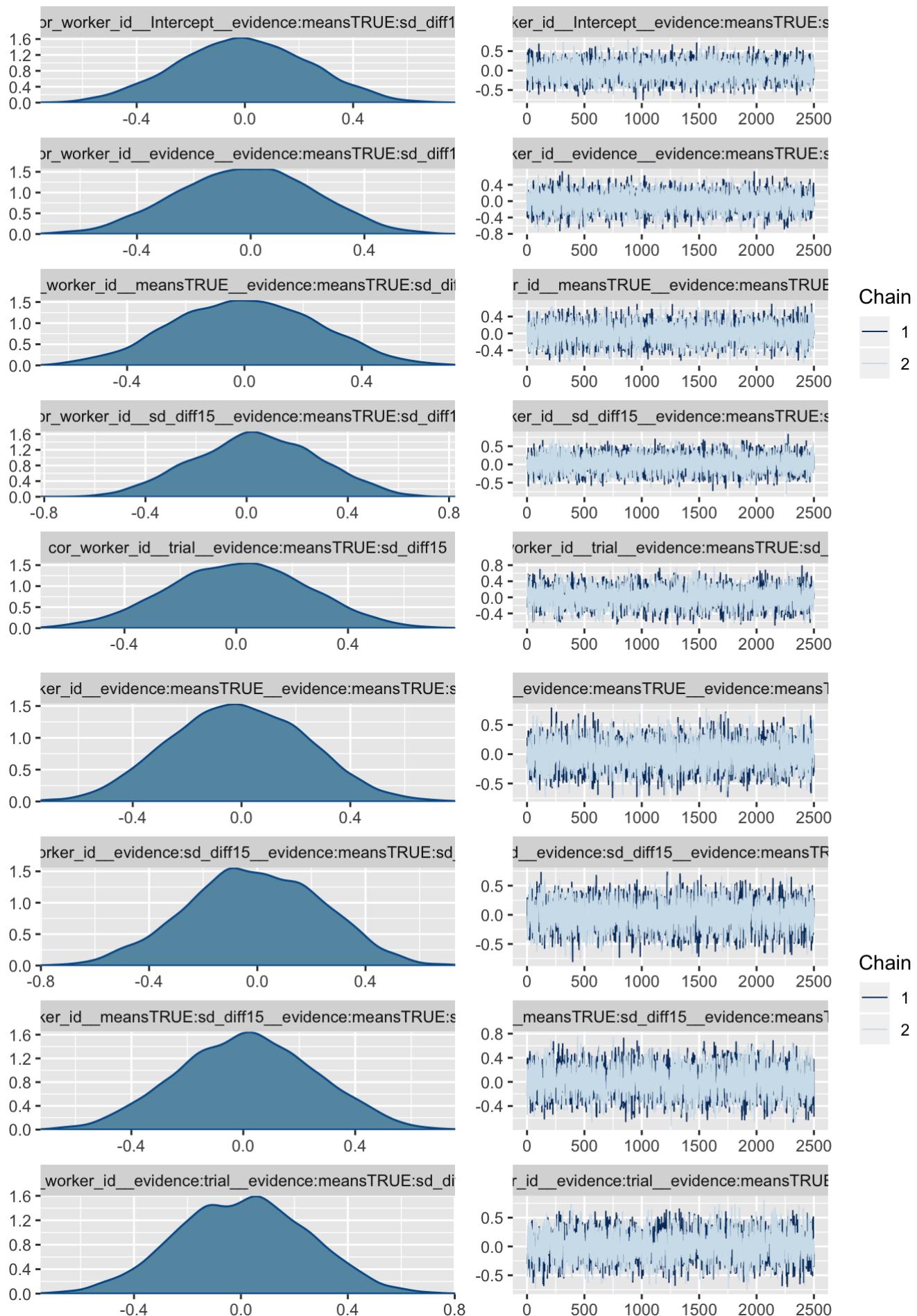












## Model Comparison

Does adding block order to our previous model improve WAIC?

```
waic(m.wrkr.means.sd.vis.logistic.r.means.sd.trial, m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial)
```

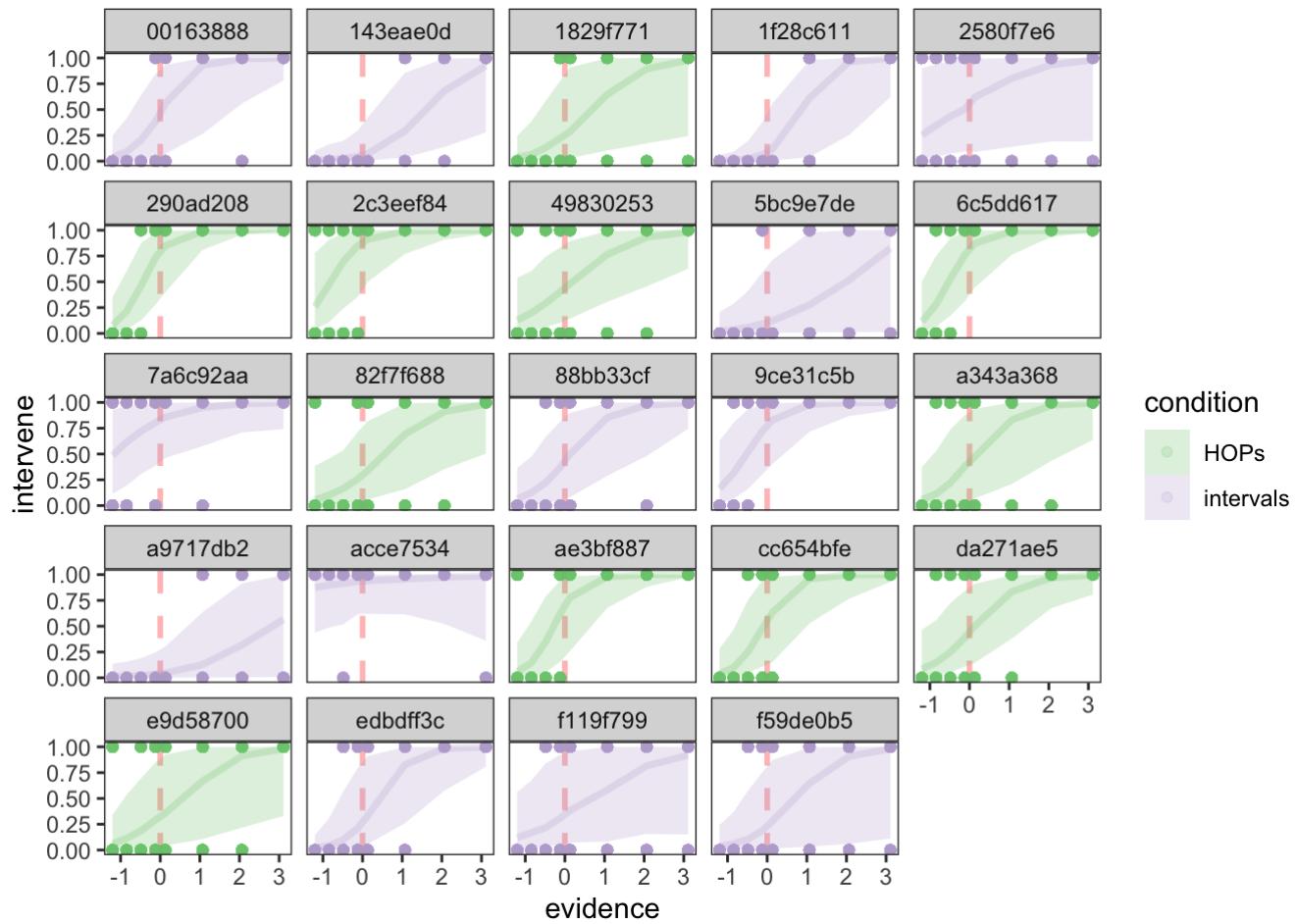
```
##  
WAIC  
## m.wrkr.means.sd.vis.logistic.r.means.sd.trial  
560.77  
## m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial  
557.17  
## m.wrkr.means.sd.vis.logistic.r.means.sd.trial - m.wrkr.means.sd.vis.order.logistic.r.  
means.sd.trial 3.60  
##  
SE  
## m.wrkr.means.sd.vis.logistic.r.means.sd.trial  
28.59  
## m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial  
29.06  
## m.wrkr.means.sd.vis.logistic.r.means.sd.trial - m.wrkr.means.sd.vis.order.logistic.r.  
means.sd.trial 4.03
```

Block order reduces WAIC slightly but less than the margin of error of the difference. Again, we will use the more maximal model in the spirit of model expansion.

## Predictive Checks

Let's take a look at the estimated psychometric functions for each worker.

```
model_df %>%  
  group_by(evidence, worker_id, means, sd_diff, condition, trial, start_means) %>%  
  add_fitted_draws(m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial, value = "pf", n  
= 200) %>%  
  ggplot(aes(x = evidence, y = intervene, color = condition, fill = condition)) +  
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +  
# utility optimal decision rule  
  stat_lineribbon(aes(y = pf), .width = c(.95), alpha = .25) +  
  geom_point(alpha = .15) +  
  scale_fill_brewer(type = "qual", palette = 1) +  
  scale_color_brewer(type = "qual", palette = 1) +  
  coord_cartesian(xlim = quantile(model_df$evidence, c(0, 1)),  
                  ylim = quantile(model_df$intervene, c(0, 1))) +  
  theme_bw() +  
  theme(panel.grid = element_blank()) +  
  facet_wrap(~ worker_id)
```



These look like the model is doing a better job of capturing variability for each individual worker.

What do the posteriors for just-noticeable differences (JND) and points of subjective equality (PSE) look like? We'll start by getting the slopes and intercepts of the linear model and using these to derive estimates of the JND and PSE for each fitted draw.

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(means, sd_diff, condition, trial, start_means) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial, re_formula = NA,
scale = "linear") %>%
  compare_levels(.value, by = evidence) %>%
  rename(slope = .value)

# get intercepts from linear model
intercepts_df <- model_df %>%
  group_by(means, sd_diff, condition, trial, start_means) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial, re_formula = NA,
scale = "linear") %>%
  rename(intercept = .value)

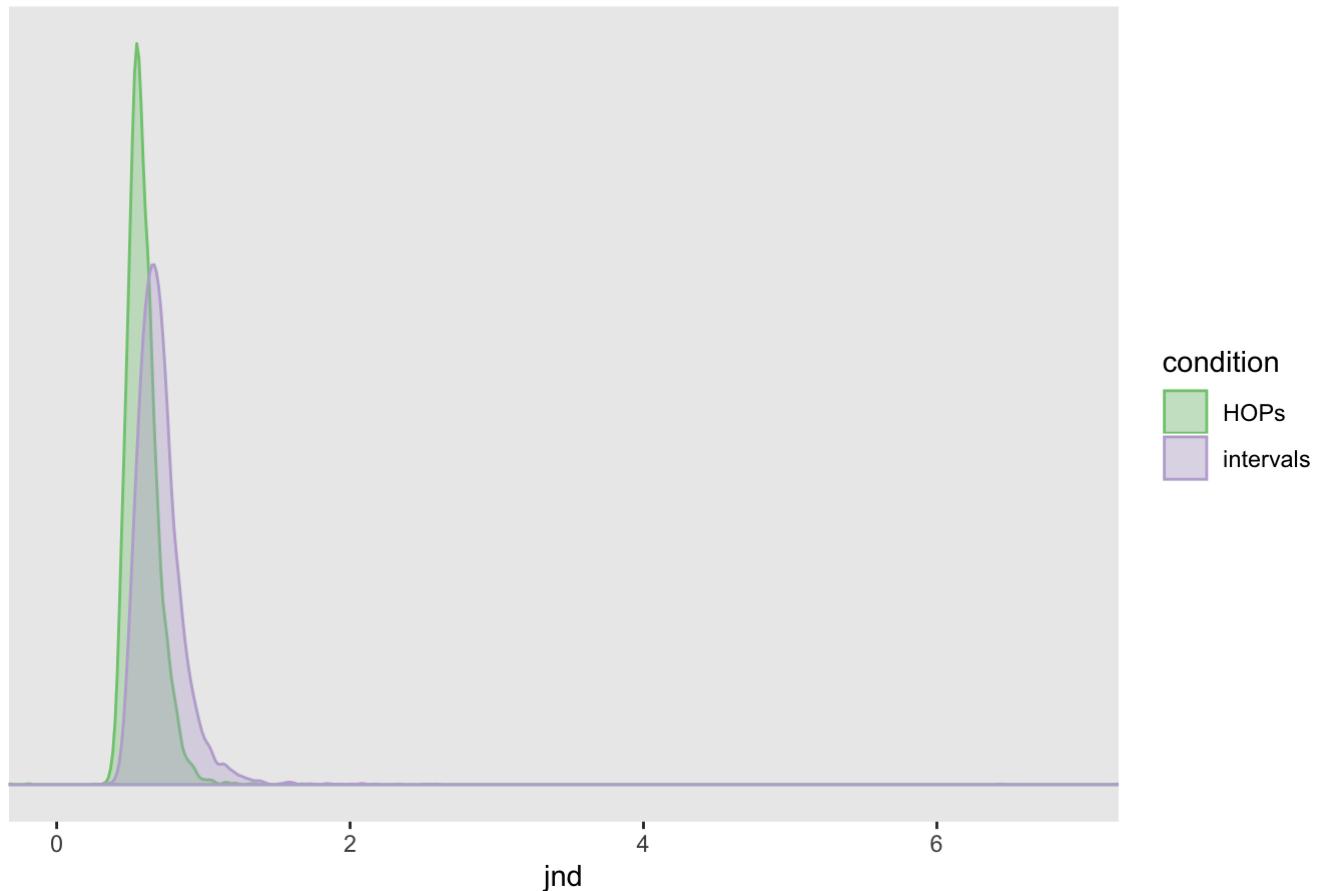
# join dataframes for slopes and intercepts, calculate PSE and JND
stats_df <- slopes_df %>%
  full_join(intercepts_df, by = c("means", "sd_diff", "condition", "trial", "start_mean
s", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )

```

First, let's look at the estimates of JNDs per visualization, marginalizing across other manipulations.

```
stats_df %>%
  group_by(condition, .draw) %>%          # marginalize out other manipulations
  summarise(jnd = weighted.mean(jnd)) %>%
  ggplot(aes(x = jnd, group = condition, color = condition, fill = condition)) +
  geom_density(alpha = 0.35) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per visualization") +
  theme(panel.grid = element_blank())
```

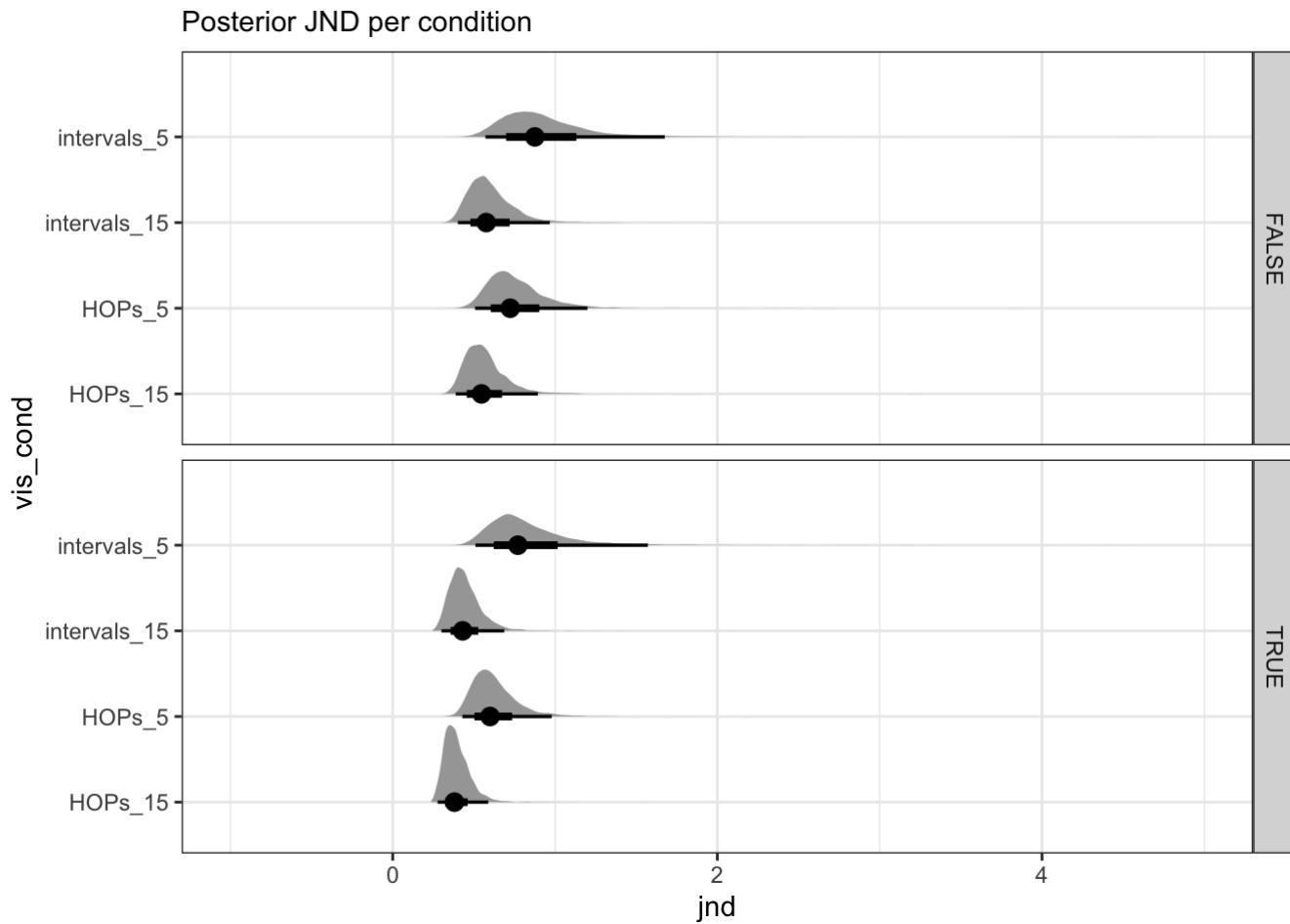
## Posterior JND per visualization



It looks like users are slightly more sensitive to evidence (i.e., JNDs are slightly smaller) in the HOPs condition, but not by much.

What if we look at the interaction effect on JNDs, marginalizing out block and trial order?

```
stats_df %>%
  group_by(means, sd_diff, condition, .draw) %>%          # marginalize out other manipulations
  summarise(jnd = weighted.mean(jnd)) %>%
  unite(vis_cond, condition, sd_diff) %>%
  # unite(means_present, means, start_means) %>%
  ggplot(aes(x = jnd, y = vis_cond)) +
  stat_halfeyeh() +
  coord_cartesian(xlim = c(-1, 5)) +
  labs(subtitle = "Posterior JND per condition") +
  theme_bw() +
  facet_grid(means ~ .)
```

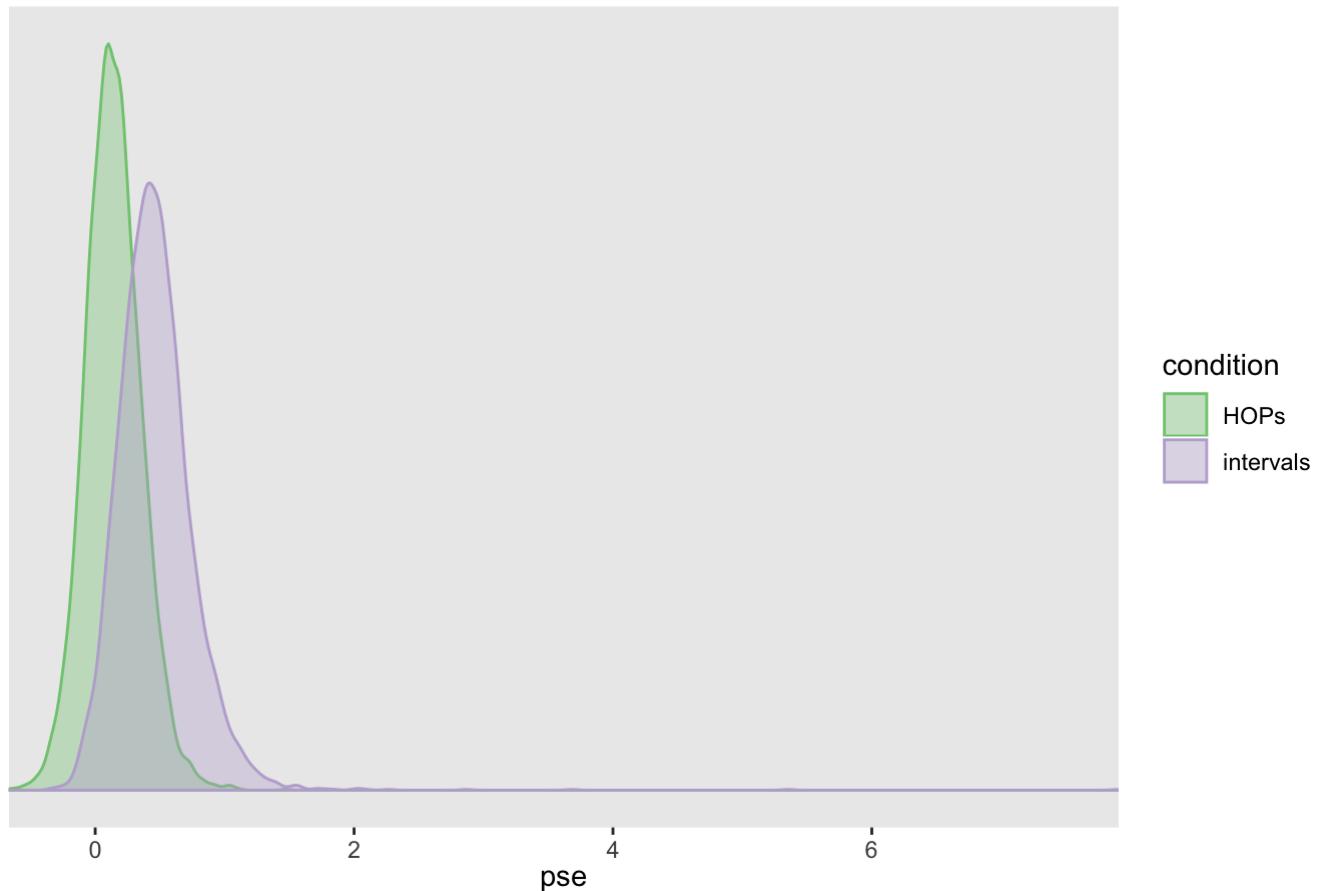


It seems like means help improve sensitivity to evidence a little bit, especially with HOPs when uncertainty is high.

Next, we'll look at the point of subjective equality for each visualization condition, marginalizing out other manipulations.

```
stats_df %>%
  group_by(condition, .draw) %>%          # marginalize out other manipulations
  summarise(pse = weighted.mean(pse)) %>%
  ggplot(aes(x = pse, group = condition, color = condition, fill = condition)) +
  geom_density(alpha = 0.35) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per visualization") +
  theme(panel.grid = element_blank())
```

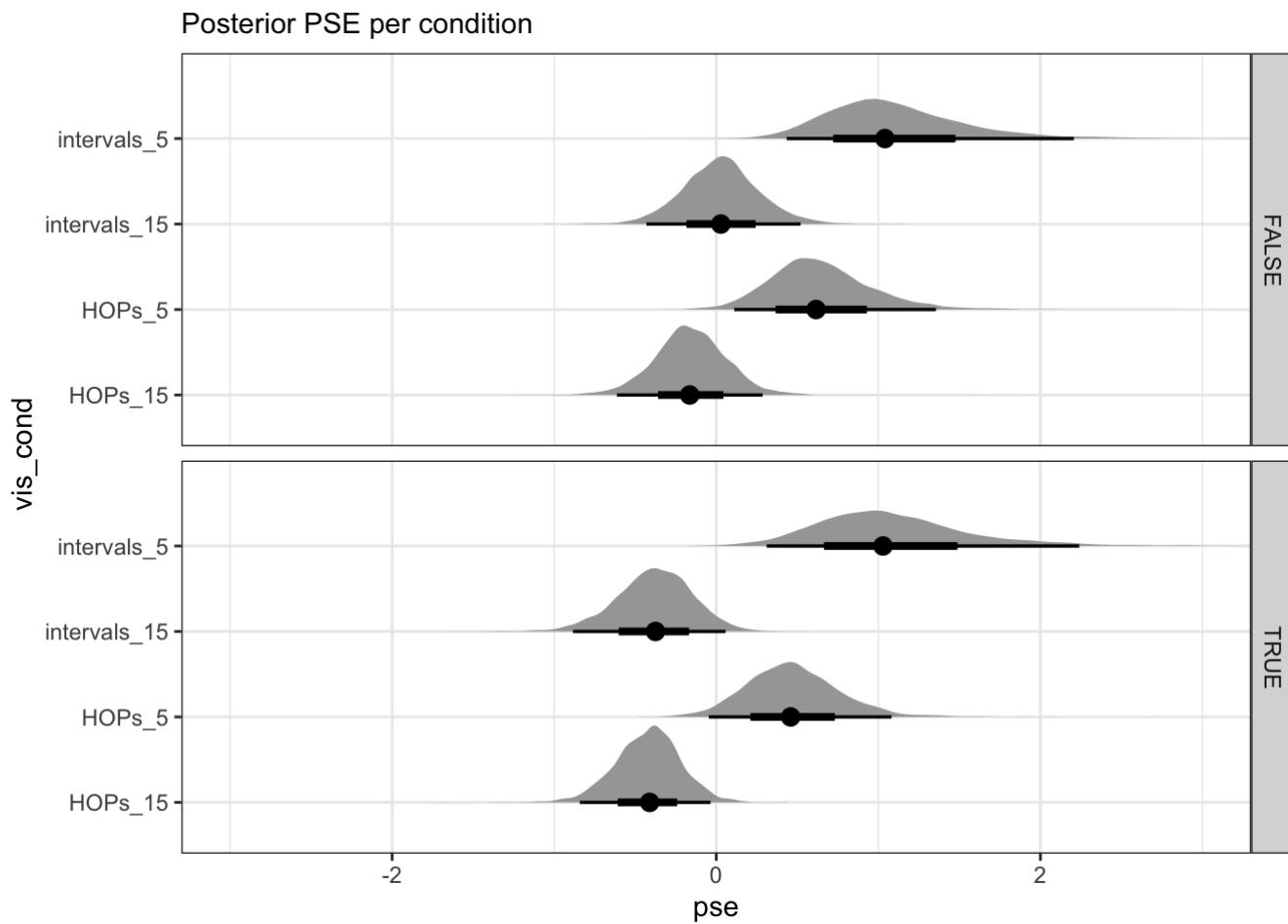
## Posterior PSE per visualization



It looks like the point of subjective equality is least biased with HOPs.

What if we look at the interaction effect on PSE, marginalizing out block and trial order?

```
stats_df %>%
  group_by(means, sd_diff, condition, .draw) %>%          # marginalize out other manipulations
  summarise(pse = weighted.mean(pse)) %>%
  unite(vis_cond, condition, sd_diff) %>%
  ggplot(aes(x = pse, y = vis_cond)) +
  stat_halfeyeh() +
  coord_cartesian(xlim = c(-3, 3)) +
  labs(subtitle = "Posterior PSE per condition") +
  theme_bw() +
  facet_grid(means ~ .)
```



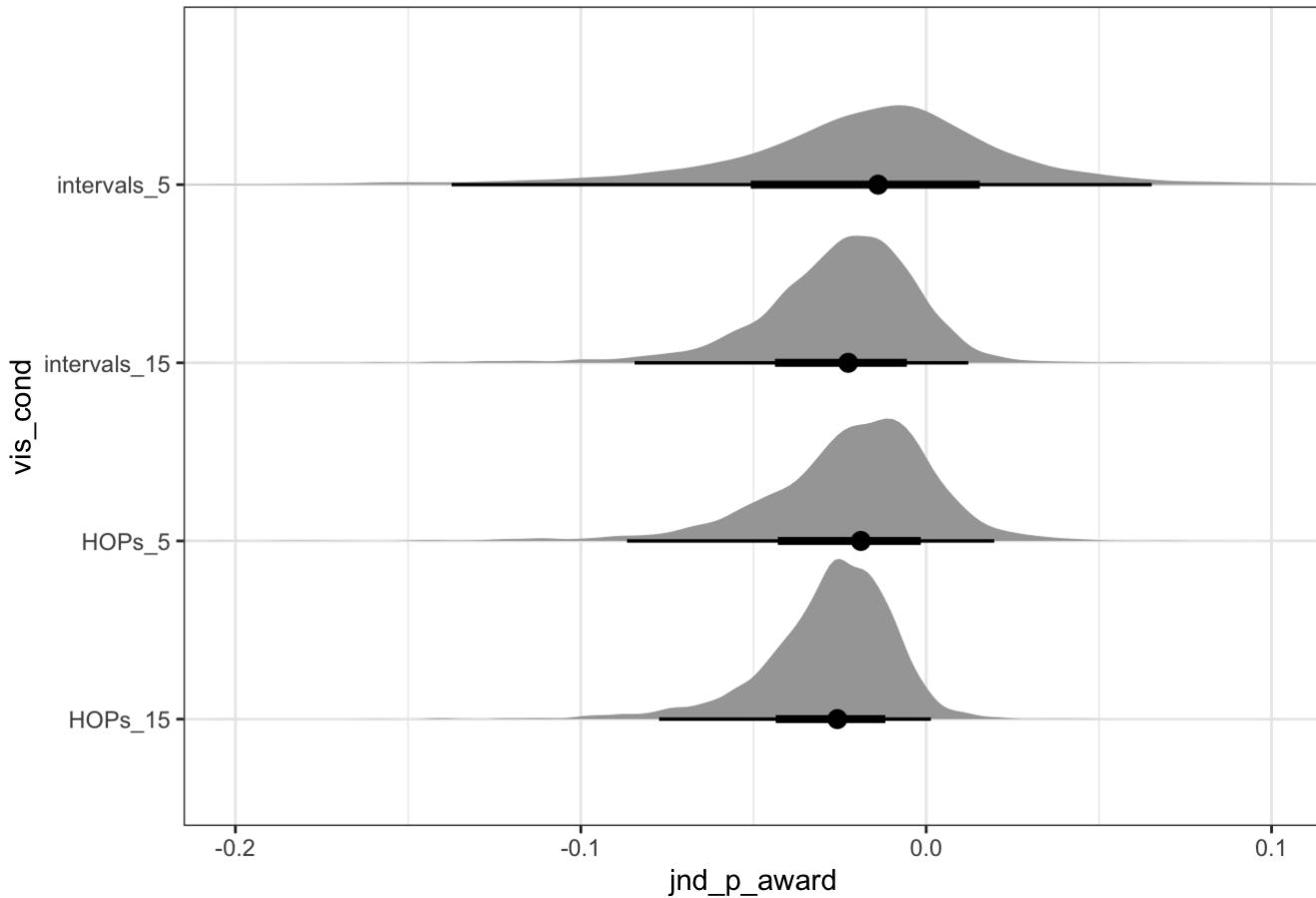
There are couple interesting things going on here:

1. People are more risk seeking when uncertainty is low.
2. People are more risk averse with HOPs.
3. Adding means might help a little when uncertainty is low, but it leads to excessive risk aversion when uncertainty is high, especially in the intervals condition.

Let's see if we can make versions of these charts that are easier to interpret since we expect our main findings to have a similar format. It should help to plot these as differences between means present minus absent. We'll also convert JND and PSE estimates into units of the probability of winning the award.

```
stats_df %>%
  group_by(means, sd_diff, condition, .draw) %>% # marginalize out other manipulations
  summarise(jnd = weighted.mean(jnd)) %>%
  compare_levels(jnd, by = means) %>%
  mutate(jnd_p_award = exp(jnd) / (1 / (unique(model_df$baseline) + 1 / unique(model_df$award_value)) - 1 + exp(jnd)) - unique(model_df$baseline) - 1 / unique(model_df$award_value)) %>%
  unite(vis_cond, condition, sd_diff, remove = FALSE) %>%
  ggplot(aes(x = jnd_p_award, y = vis_cond)) +
  stat_halfeyeh() +
  coord_cartesian(xlim = c(-0.2, 0.1)) +
  labs(subtitle = "Posterior difference in JND for means present - absent per condition") +
  theme_bw()
```

## Posterior difference in JND for means present - absent per condition

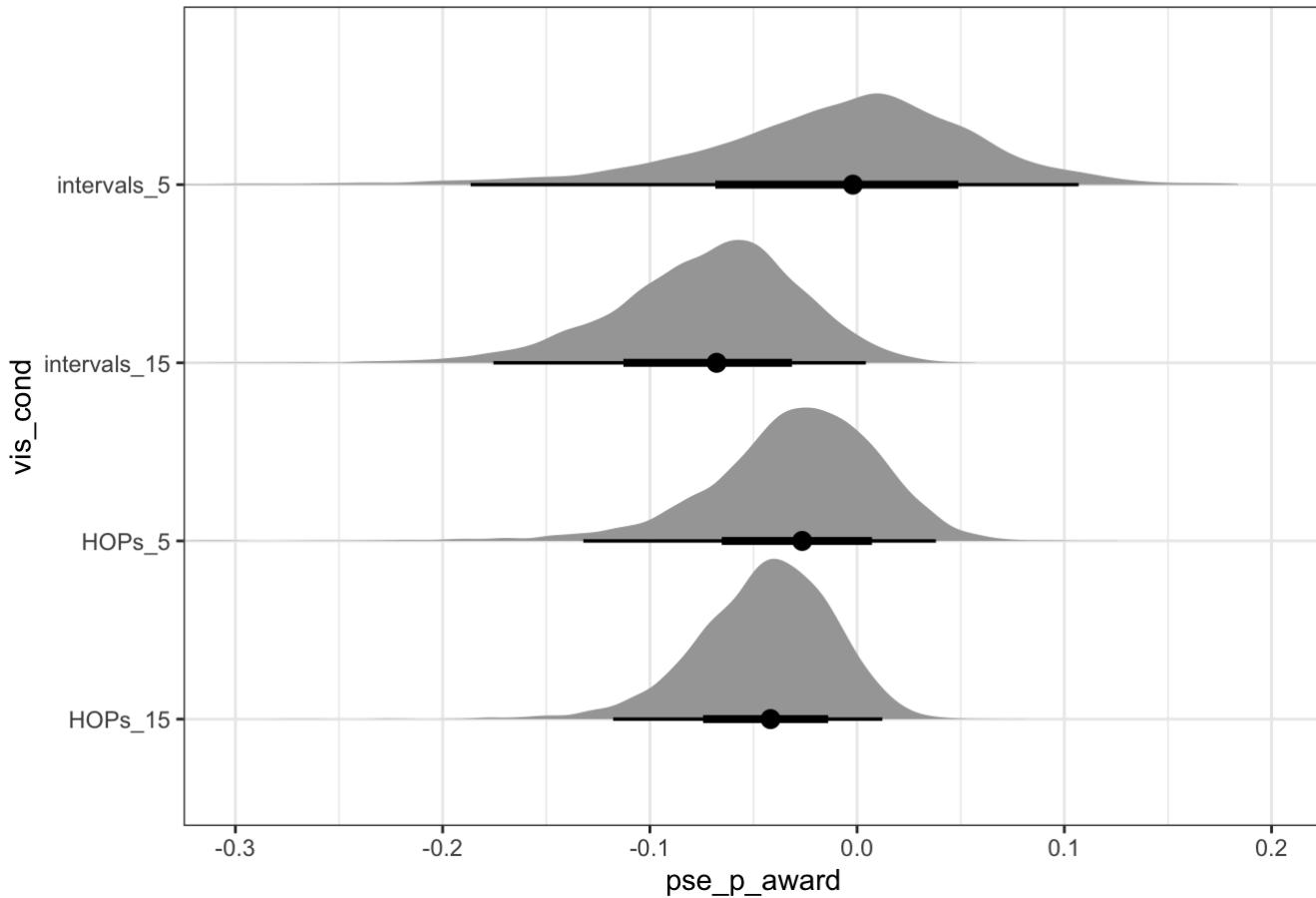


Here we can see that on average adding means reduces JNDs by about 0.025 in terms of the probability of winning the award. This means people will notice a difference in the odds of winning that is up to 2.5% less than the difference they would notice without means. This is a small difference, but it could be helpful for borderline cases where the effect size is near the decision threshold.

Let's do something similar for the effect of extrinsic means on PSE.

```
stats_df %>%
  group_by(means, sd_diff, condition, .draw) %>%          # marginalize out other manipulations
  summarise(pse = weighted.mean(pse)) %>%
  compare_levels(pse, by = means) %>%
  mutate(pse_p_award = exp(pse) / (1 / (unique(model_df$baseline) + 1 / unique(model_df$award_value)) - 1 + exp(pse)) - unique(model_df$baseline) - 1 / unique(model_df$award_value)) %>%
  unite(vis_cond, condition, sd_diff, remove = FALSE) %>%
  ggplot(aes(x = pse_p_award, y = vis_cond)) +
  stat_halfeyeh() +
  coord_cartesian(xlim = c(-0.3, 0.2)) +
  labs(subtitle = "Posterior differences in PSE for means present - absent per condition") +
  theme_bw()
```

## Posterior differences in PSE for means present - absent per condition

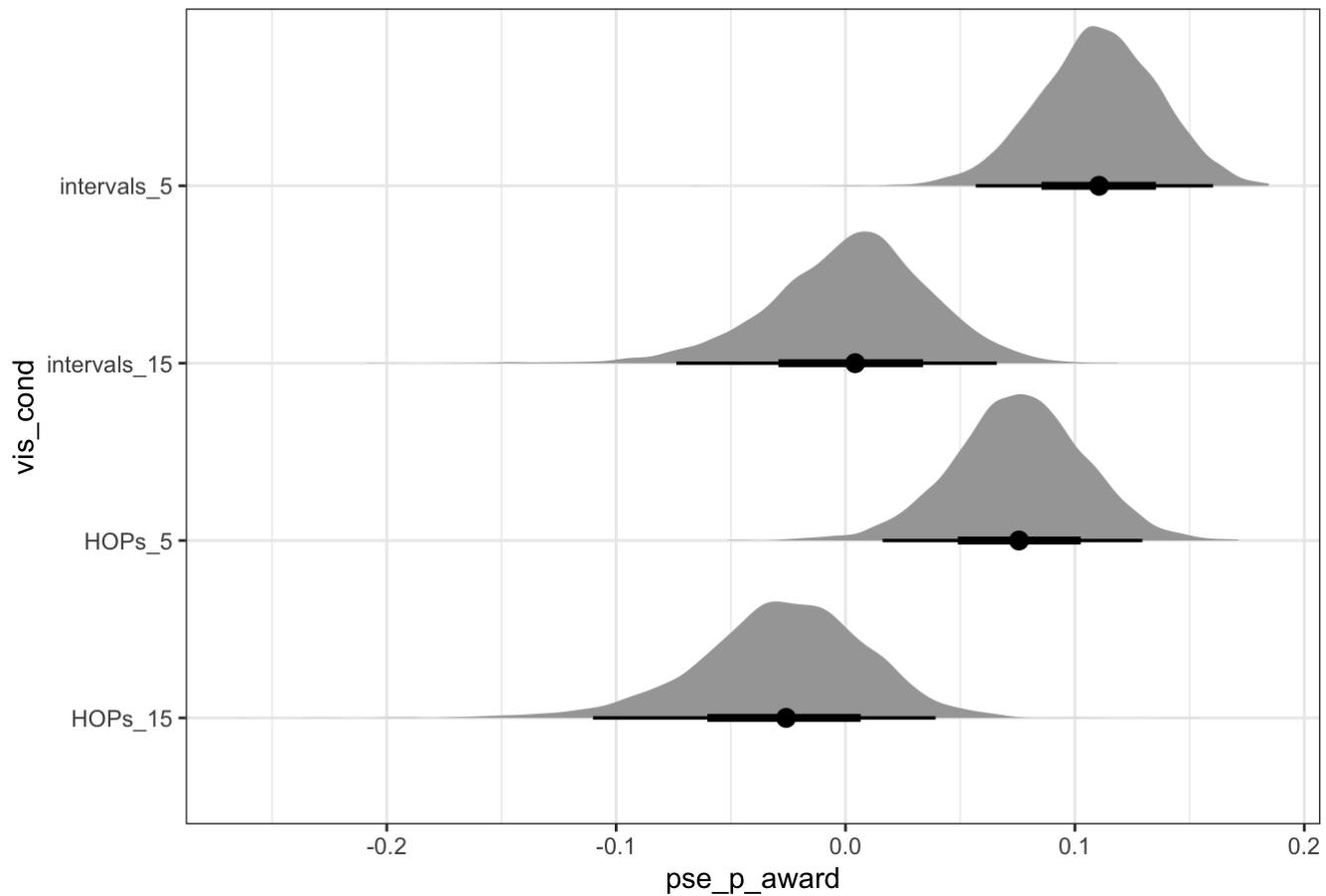


On average, adding means leads to more risk averse decisions. Specifically, people see the better option as having 3% to 7% less probability of producing a desirable outcome with mean compared to without means. This could lead to suboptimal decisions or reflect a debiasing effect depending on the baseline PSE in each condition.

What is the baseline PSE in each condition?

```
stats_df %>%
  filter(means == FALSE) %>%
  group_by(sd_diff, condition, .draw) %>%           # marginalize out other manipulations
  summarise(pse = weighted.mean(pse)) %>%
  mutate(pse_p_award = exp(pse) / (1 / (unique(model_df$baseline) + 1 / unique(model_df$award_value)) - 1 + exp(pse)) - unique(model_df$baseline) - 1 / unique(model_df$award_value)) %>%
  unite(vis_cond, condition, sd_diff, remove = FALSE) %>%
  ggplot(aes(x = pse_p_award, y = vis_cond)) +
  stat_halfeyeh() +
  # coord_cartesian(xlim = c(-0.3, 0.2)) +
  labs(subtitle = "Posterior PSE without means per condition") +
  theme_bw()
```

## Posterior PSE without means per condition



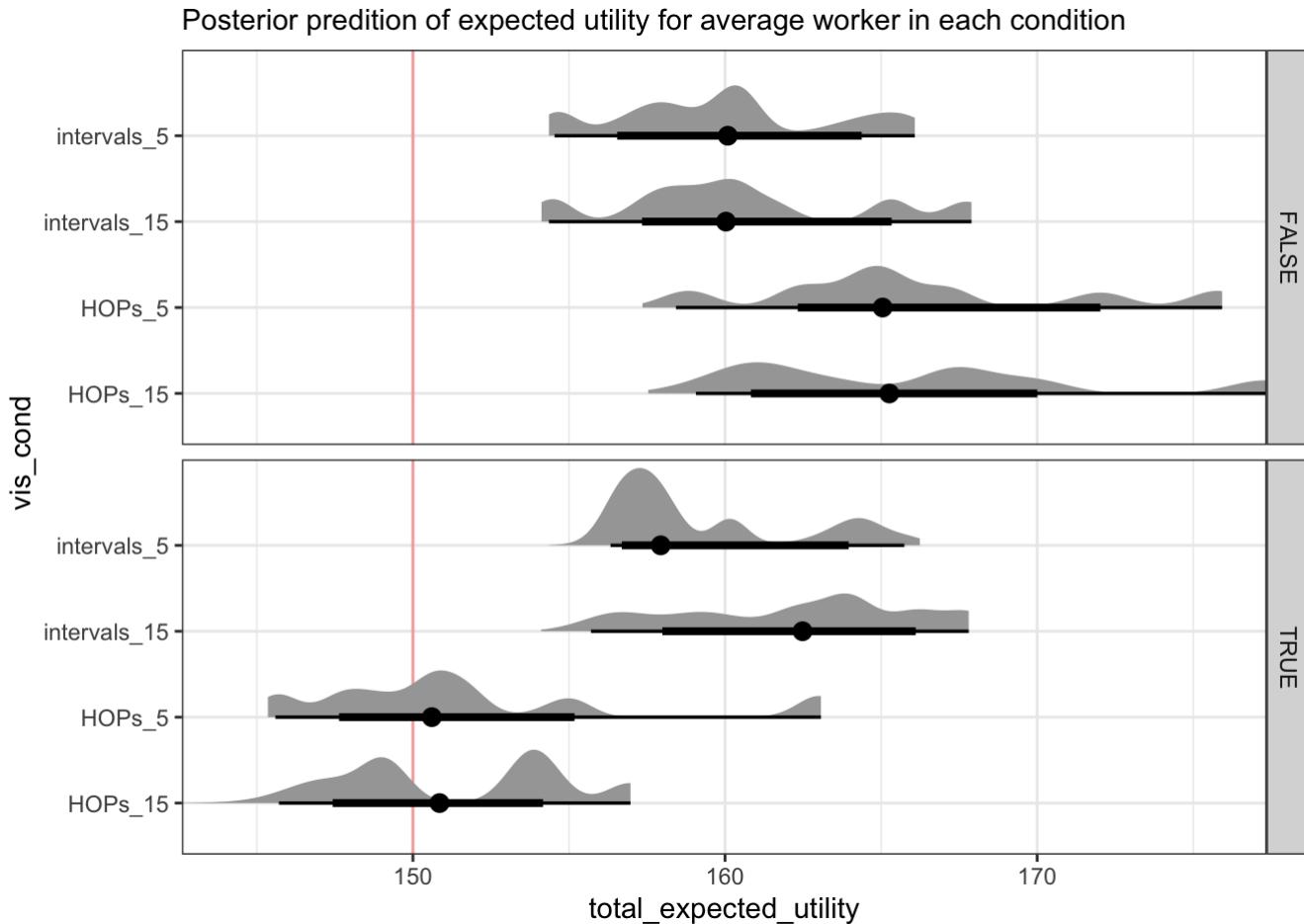
This suggests that means have a debasing effect for intervals and a biasing effect on HOPs.

What if we look at the expected utility in each condition?

```

model_df %>%
  add_predicted_draws(m.wrkr.means.sd.vis.order.logistic.r.means.sd.trial, re_formula =
NA, n = 200) %>%
  group_by(evidence, means, sd_diff, condition, trial, start_means, .draw) %>%
  mutate(
    expected_payoff = if_else(.prediction == 1,
                               award_value * p_award_with - 1,
                               award_value * p_award_without)
  ) %>%
  group_by(evidence, means, sd_diff, condition, .draw) %>% # aggregate across trials and
block order
  summarize(
    total_expected_utility = weighted.mean(account_value + sum(expected_payoff)),
    cutoff = mean(cutoff)
  ) %>%
  unite(vis_cond, condition, sd_diff, remove = FALSE) %>%
# unite(means_present, means, start_means, remove = FALSE) %>%
  ggplot(aes(x = total_expected_utility, y = vis_cond)) +
  geom_vline(aes(xintercept = cutoff), color = "red", alpha = 0.35) +
  stat_halfeyeh() +
  scale_x_continuous(expression(total_expected_utility), expand = c(0, 0)) +
  labs(subtitle = "Posterior prediction of expected utility for average worker in each co
ndition") +
  theme_bw() +
  facet_grid(means ~ .)

```



This might look a little off because of the improper counterbalancing of block order, but this kind of thing will eventually be a good way to communicate results in aggregate.