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(),
##   condition = col_character(),
##   start_means = col_character(),
##   numeracy = col_integer(),
##   gender = col_character(),
##   age = col_character(),
##   education = col_character(),
##   chart_use = col_character(),
##   intervene = col_integer(),
##   outcome = col_character(),
##   pSup = col_integer(),
##   trial = col_character(),
##   trialIdx = col_character()
## )
```

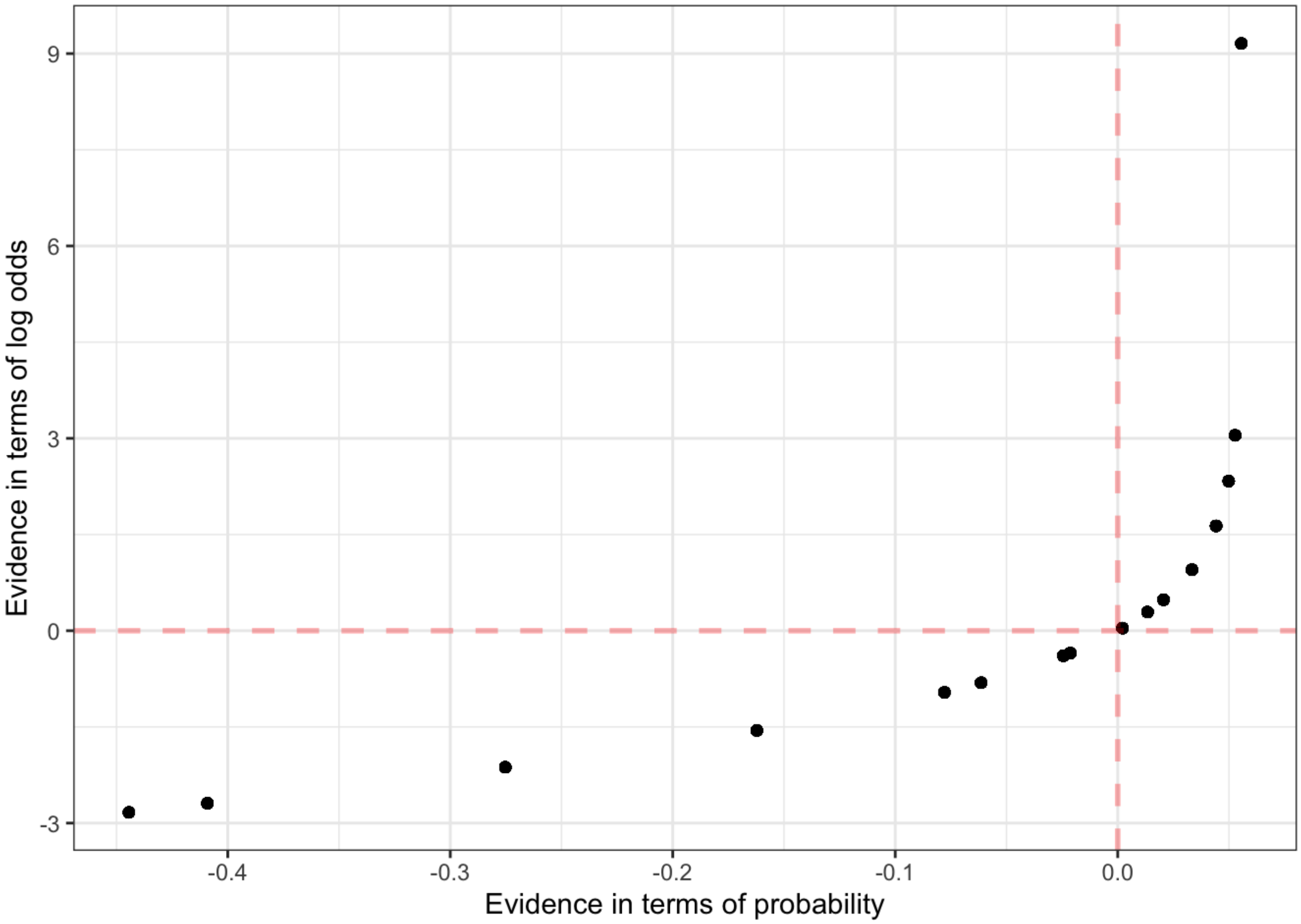
```
## See spec(...) for full column specifications.
```

```
# preprocessing
responses_df <- full_df %>%
  rename( # rename to convert away from camel case
    worker_id = workerId,
    account_value = accountValue,
    ground_truth = groundTruth,
    p_award_with = pAwardWith,
    p_award_without = pAwardWithout,
    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") %>%
  # add a variable to note whether the chart they viewed showed means
  mutate(
    means = as.factor((start_means == "True" & as.numeric(trial) < 16) | (start_means ==
"False" & as.numeric(trial) >= 16)),
    start_means = as.factor(start_means == "True")
  )
  # # mutate to rows where intervene == -1 for some reason
  # mutate(
  #   intervene = if_else(intervene == -1,
  #                       # repair
  #                       if_else((payoff == (award_value - 1) | 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
  #                       )
  # )

head(responses_df)
```

```
## # A tibble: 6 x 30
##   worker_id batch condition baseline award_value exchange start_means
##   <chr>     <int> <chr>        <dbl>       <dbl>    <dbl> <fct>
## 1 c3de4118      4 intervals      0.5        2.25      0.2 FALSE
## 2 c3de4118      4 intervals      0.5        2.25      0.2 FALSE
## 3 c3de4118      4 intervals      0.5        2.25      0.2 FALSE
## 4 c3de4118      4 intervals      0.5        2.25      0.2 FALSE
## 5 c3de4118      4 intervals      0.5        2.25      0.2 FALSE
## 6 c3de4118      4 intervals      0.5        2.25      0.2 FALSE
## # … with 23 more variables: total_bonus <dbl>, duration <dbl>,
## #   numeracy <int>, gender <chr>, age <chr>, education <chr>,
## #   chart_use <chr>, account_value <dbl>, ground_truth <dbl>,
## #   intervene <int>, outcome <chr>, pAwardCurrent <dbl>, pAwardNew <dbl>,
## #   p_award_with <dbl>, p_award_without <dbl>, p_superiority <int>,
## #   payoff <dbl>, resp_time <dbl>, start_time <dbl>, trial <chr>,
## #   trial_dur <dbl>, trial_idx <chr>, means <fct>
```

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 proabilities as log odds.

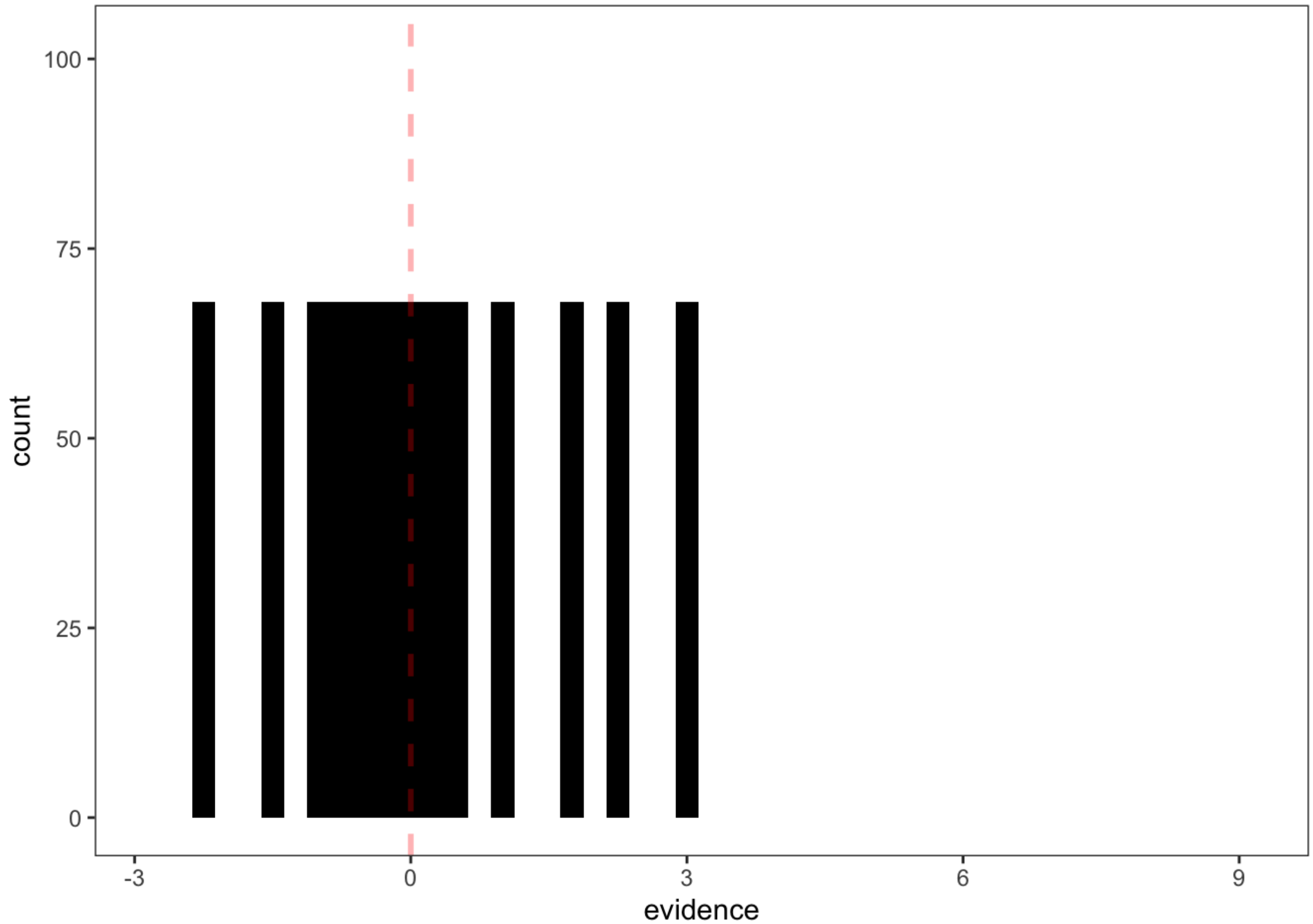
```
model_df <- responses_df %>%
  mutate(
    # evidence scale
    p_diff = p_award_with - (p_award_without + (1 / award_value)),
    evidence = qlogis(p_award_with) - qlogis(p_award_without + (1 / award_value))
  )

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 to modeling data set
model_df <- model_df %>% left_join(exclude_df, by = "worker_id") %>% filter(exclude == F
ALSE)
```

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.

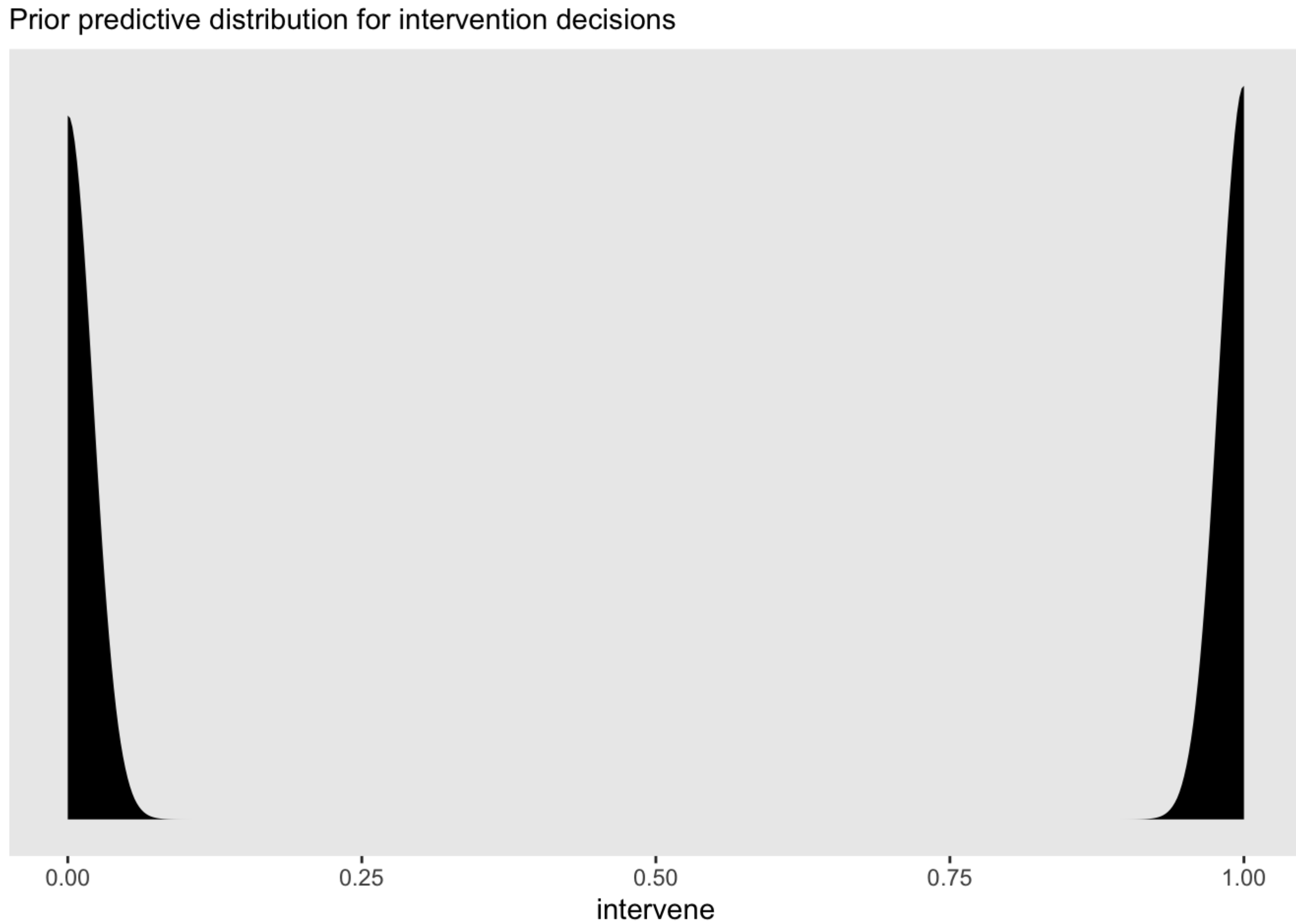
```
# get_prior(data = model_df,
#           family = bernoulli(link = "logit"),
#           formula = bf(intervene ~ 1))

# 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())
```



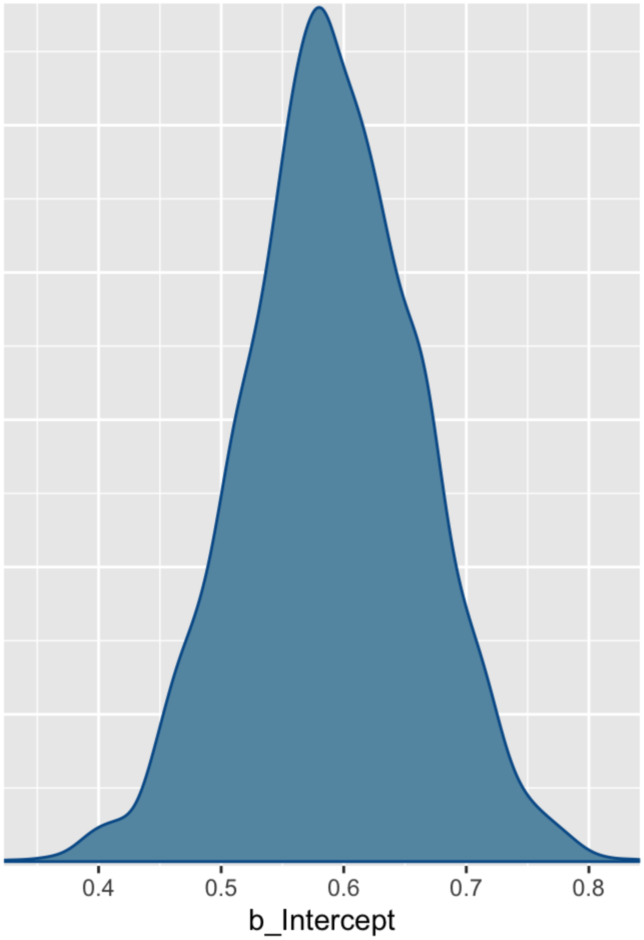
Now, let’s fit the model to our data.

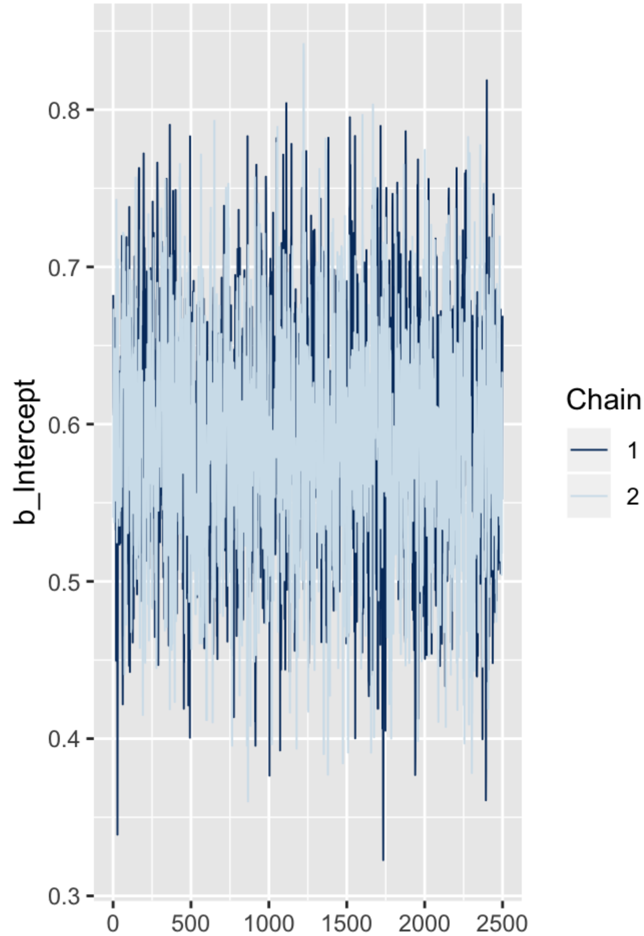
```
# starting as simple as possible: learn the distribution of decisions
m.logistic_intercept <- brm(data = model_df, family = bernoulli(link = "logit"),
              formula = bf(intervene ~ 1),
              prior = c(prior(normal(0, 1), class = Intercept)),
              iter = 3000, warmup = 500,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

Check diagnostics:

- Trace plots

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





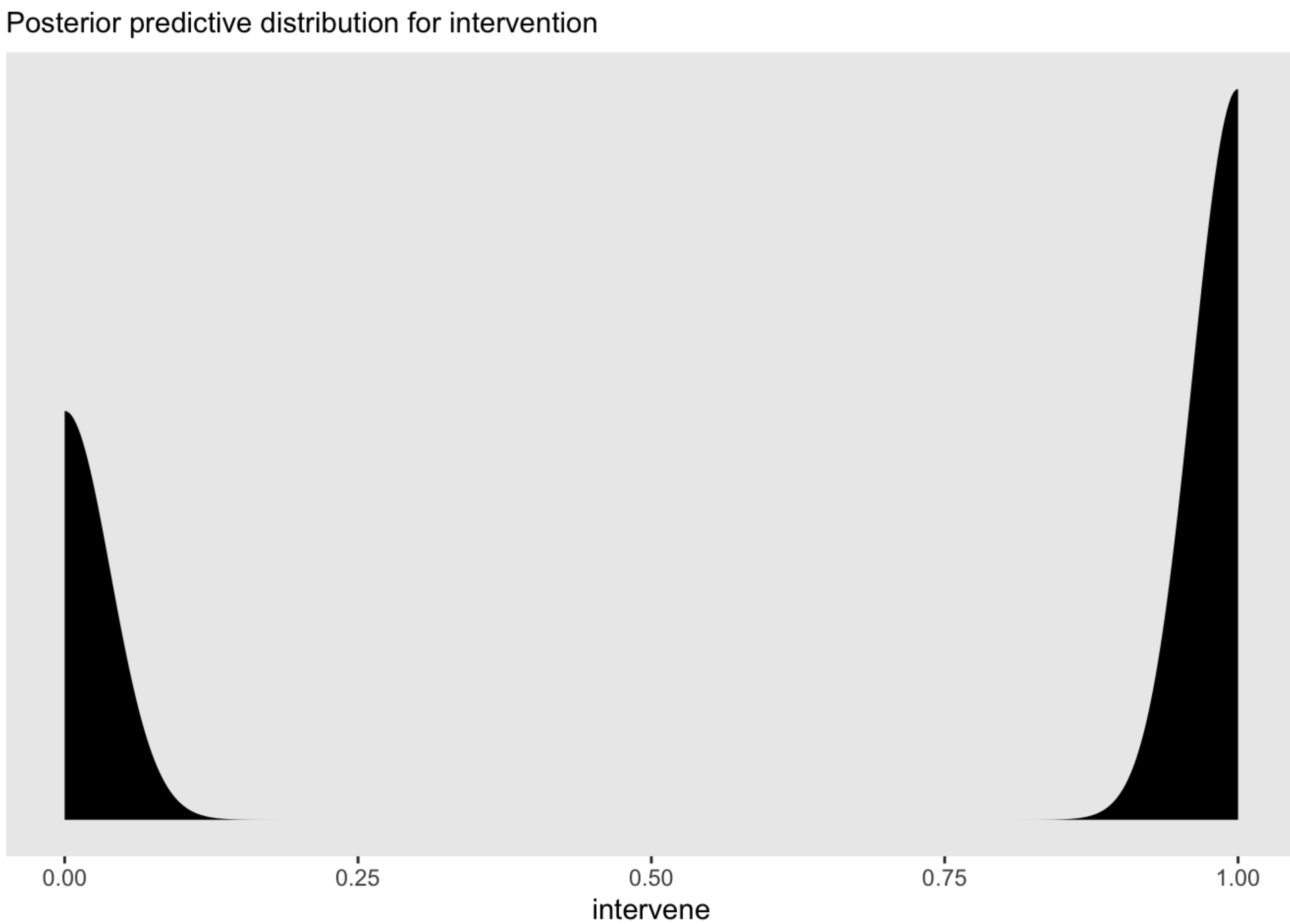
- Summary

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

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: intervene ~ 1
##    Data: model_df (Number of observations: 840)
## 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     0.59      0.07     0.45     0.73       1593 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.

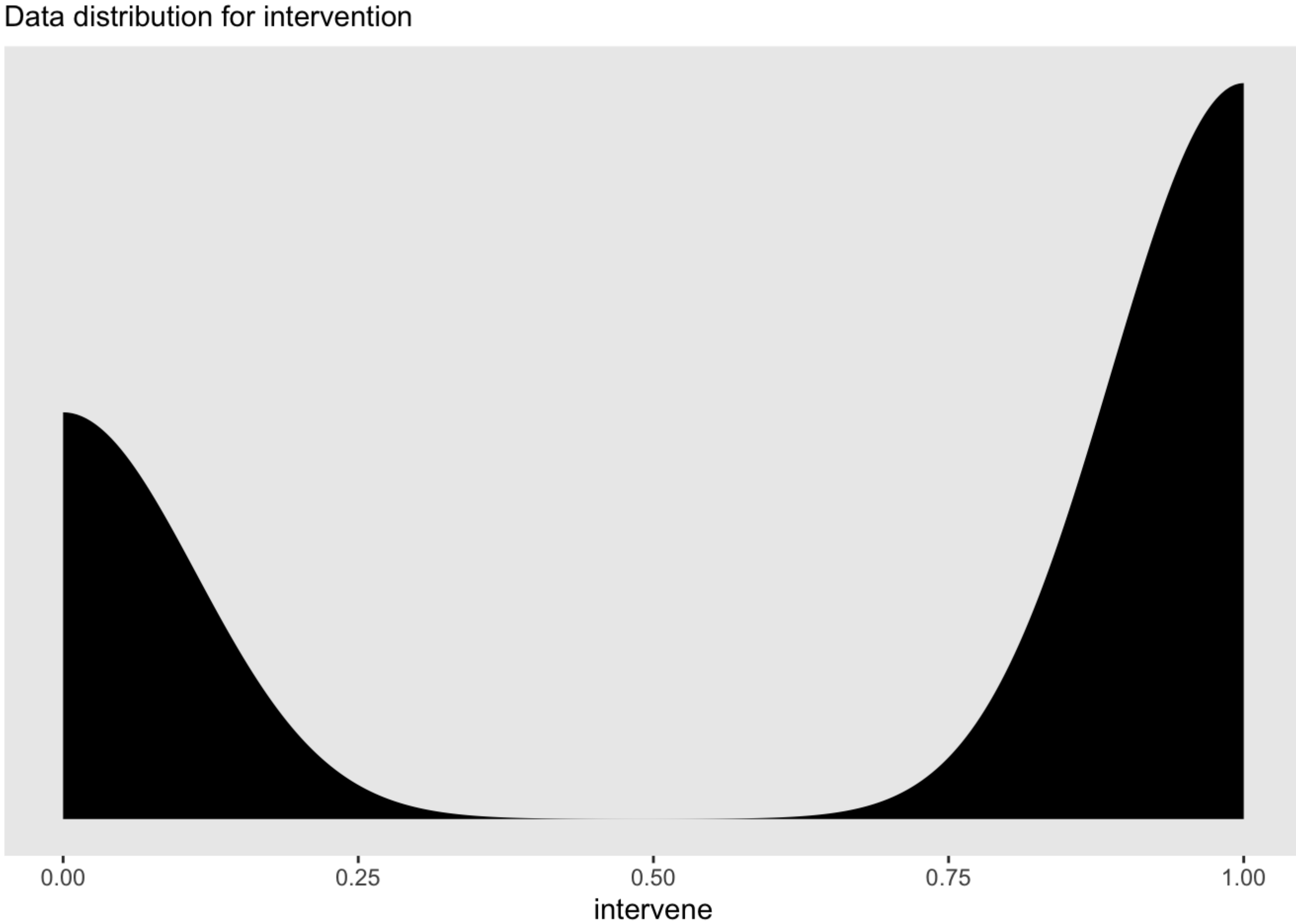
```
# posterior predictive check
model_df %>%
  select(-intervene) %>% # this model should not be sensitive to evidence
  add_predicted_draws(m.logistic_intercept, prediction = "intervene", seed = 1234, n = 2
00) %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL,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



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

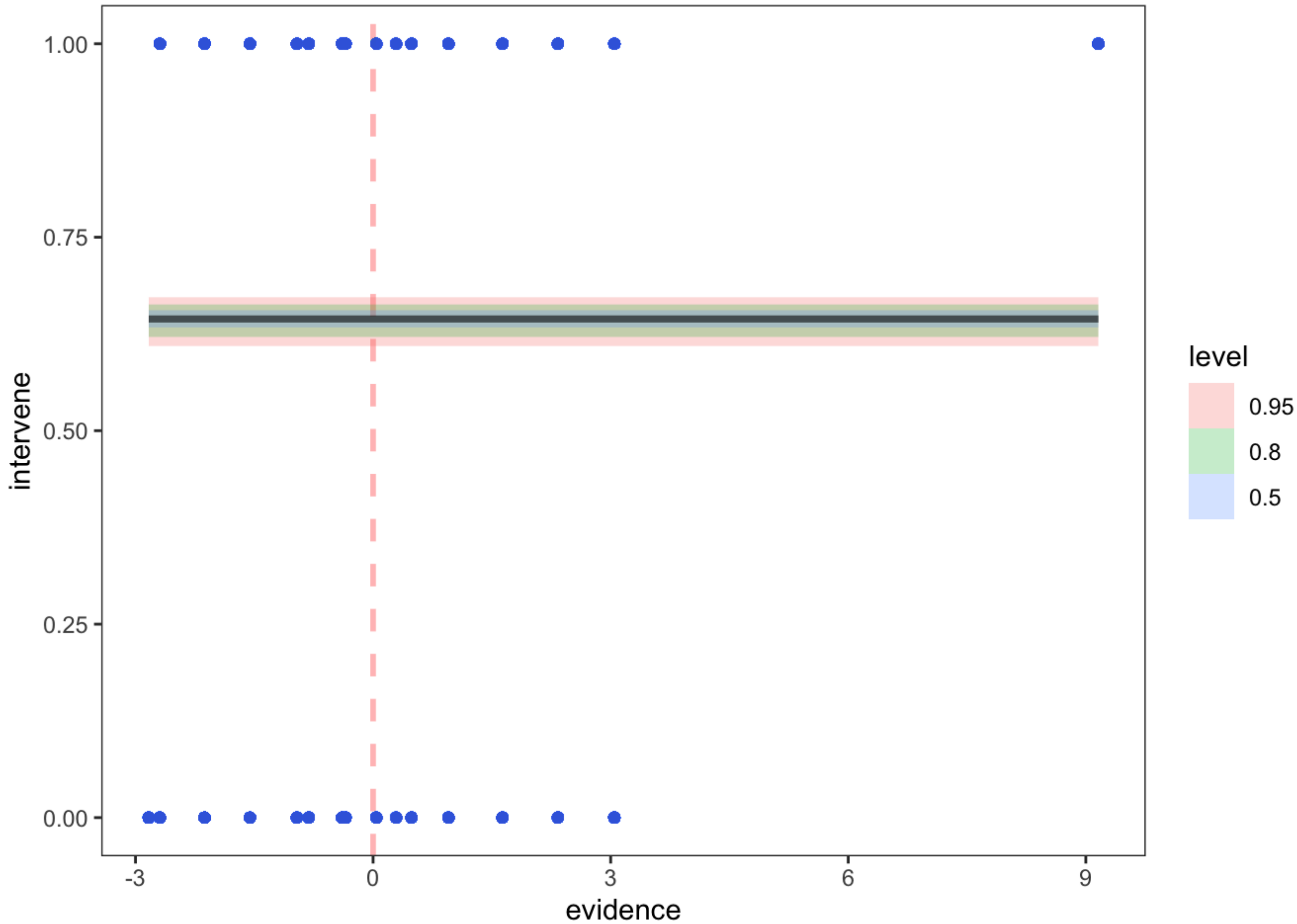
How do the posterior predictions compare to the observed data?

```
# 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())
```



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) +
  geom_point(alpha = .15, color = "royalblue") +
  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())
```



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 our prior on a one-to-one correspondence between units of evidence and perceived evidence.

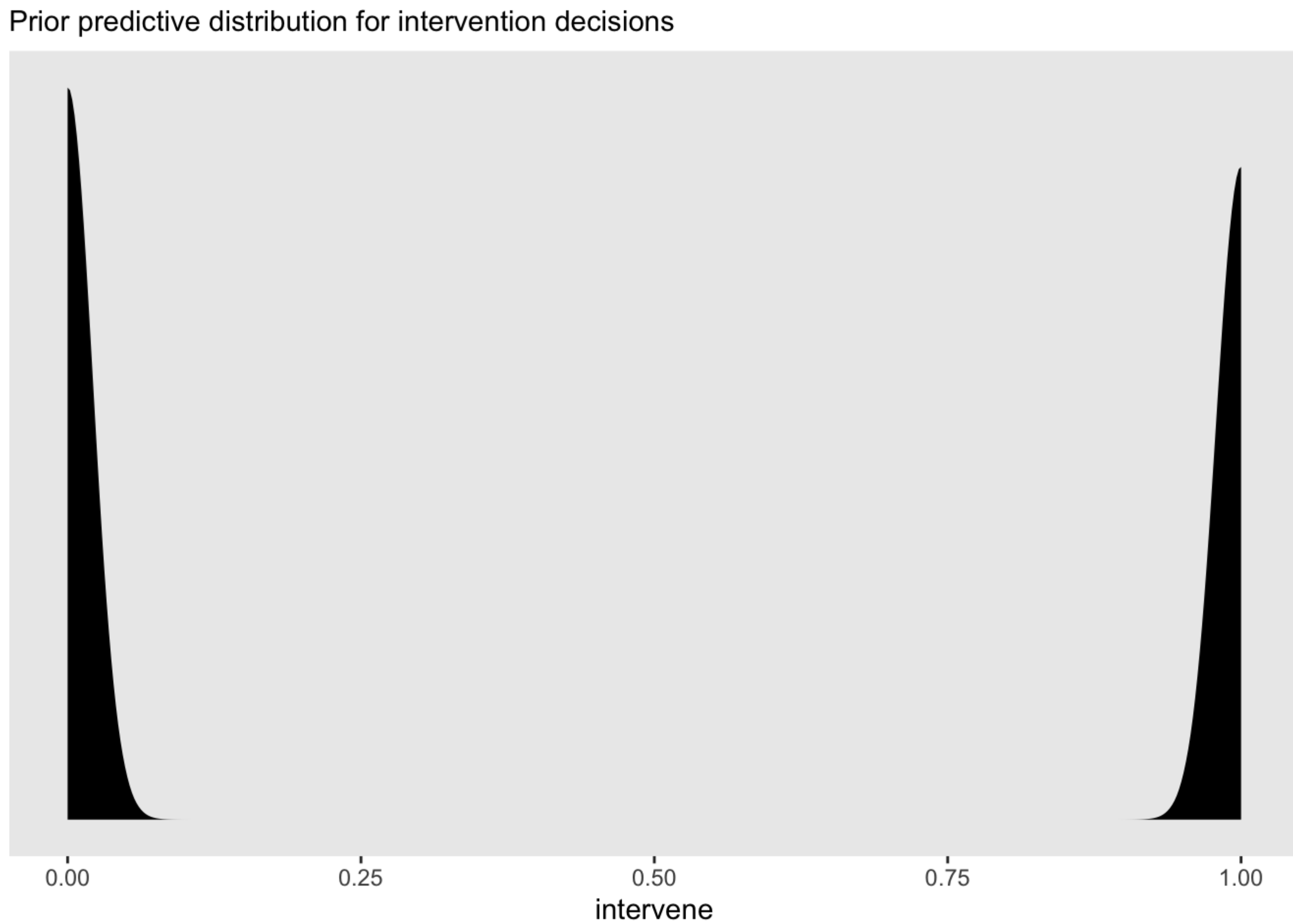
```
# get_prior(data = model_df, family = bernoulli(link = "logit"), formula = bf(intervene
 ~ 1 + evidence))

# starting as simple as possible: learn the distribution of decisions
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
```

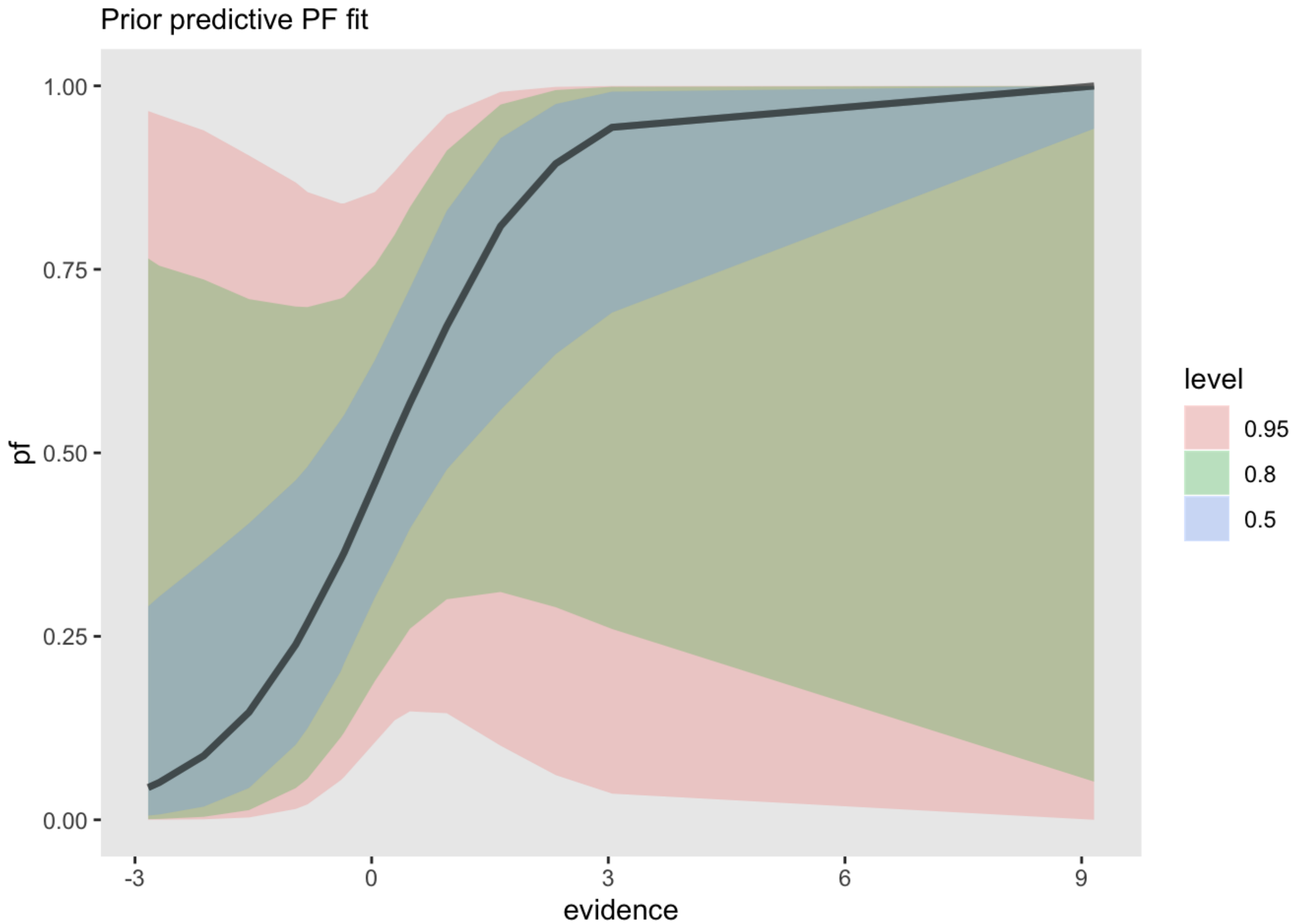
Let's look at our prior predictive distribution. This should still be an even split between 0 and 1, although now our model should be sensitive to evidence.

```
# 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())
```



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) +
  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())
```



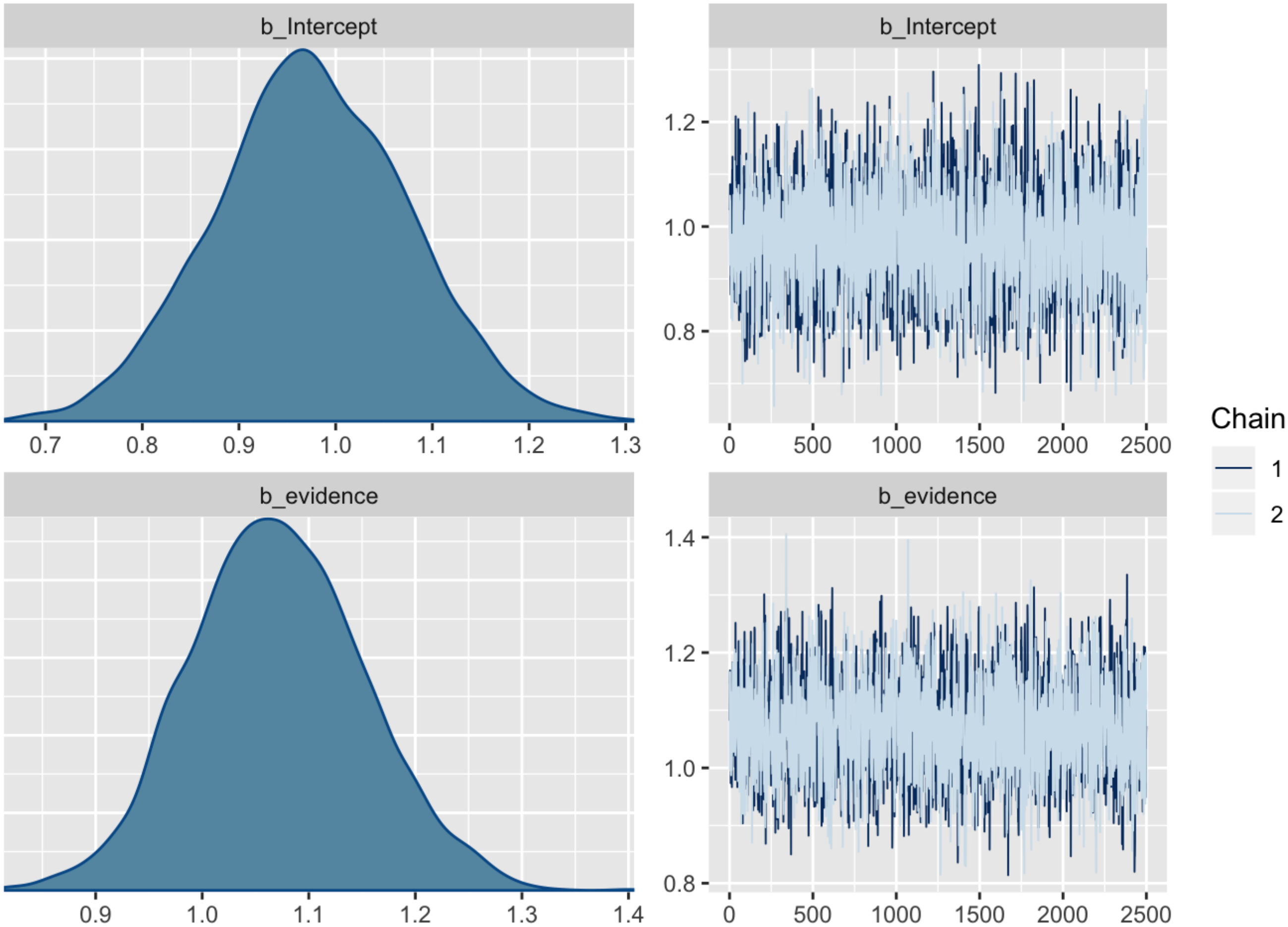
Now, let's fit the model.

```
# linear model with logit link
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:

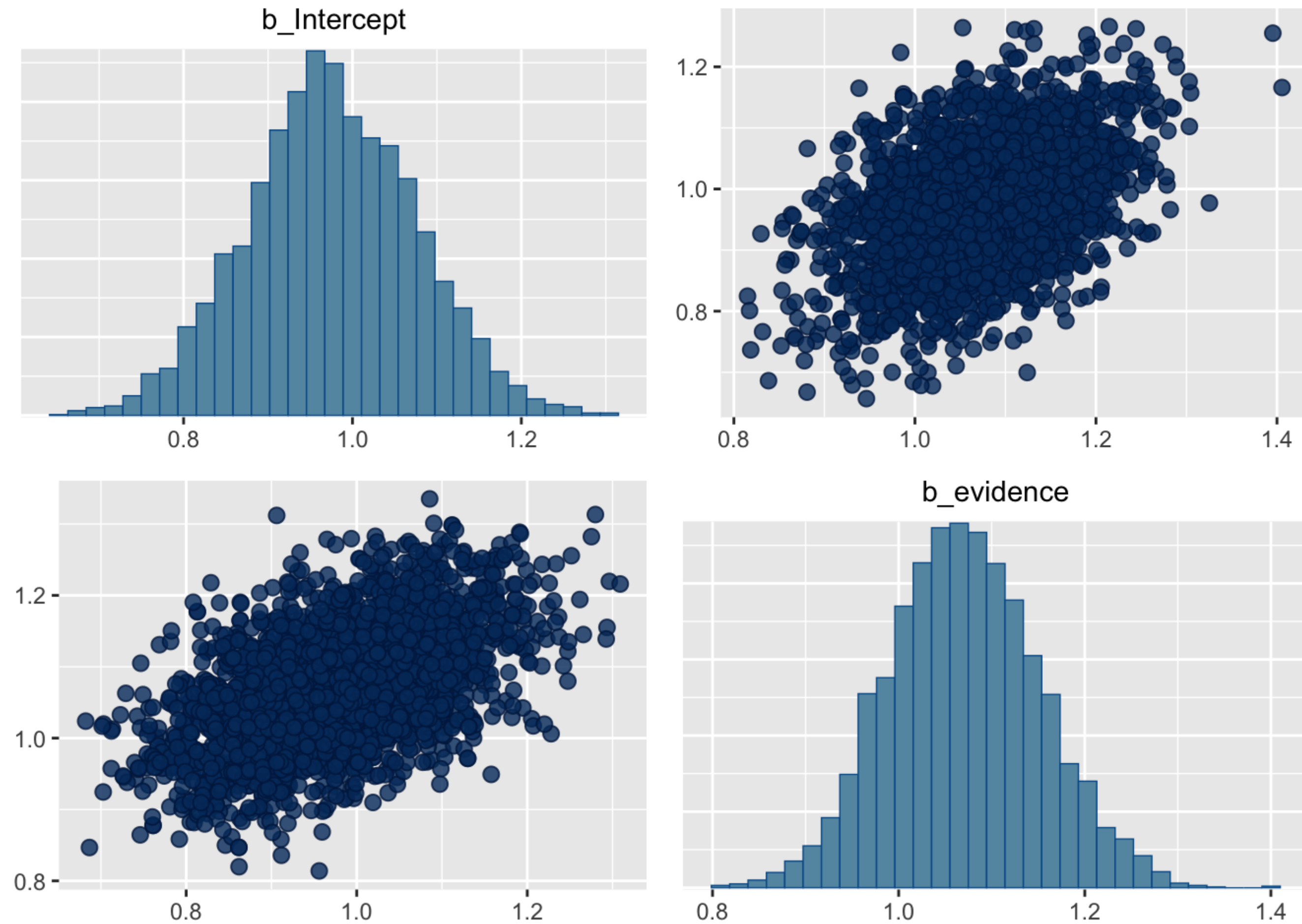
- Trace plots

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



- Pairs plot

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



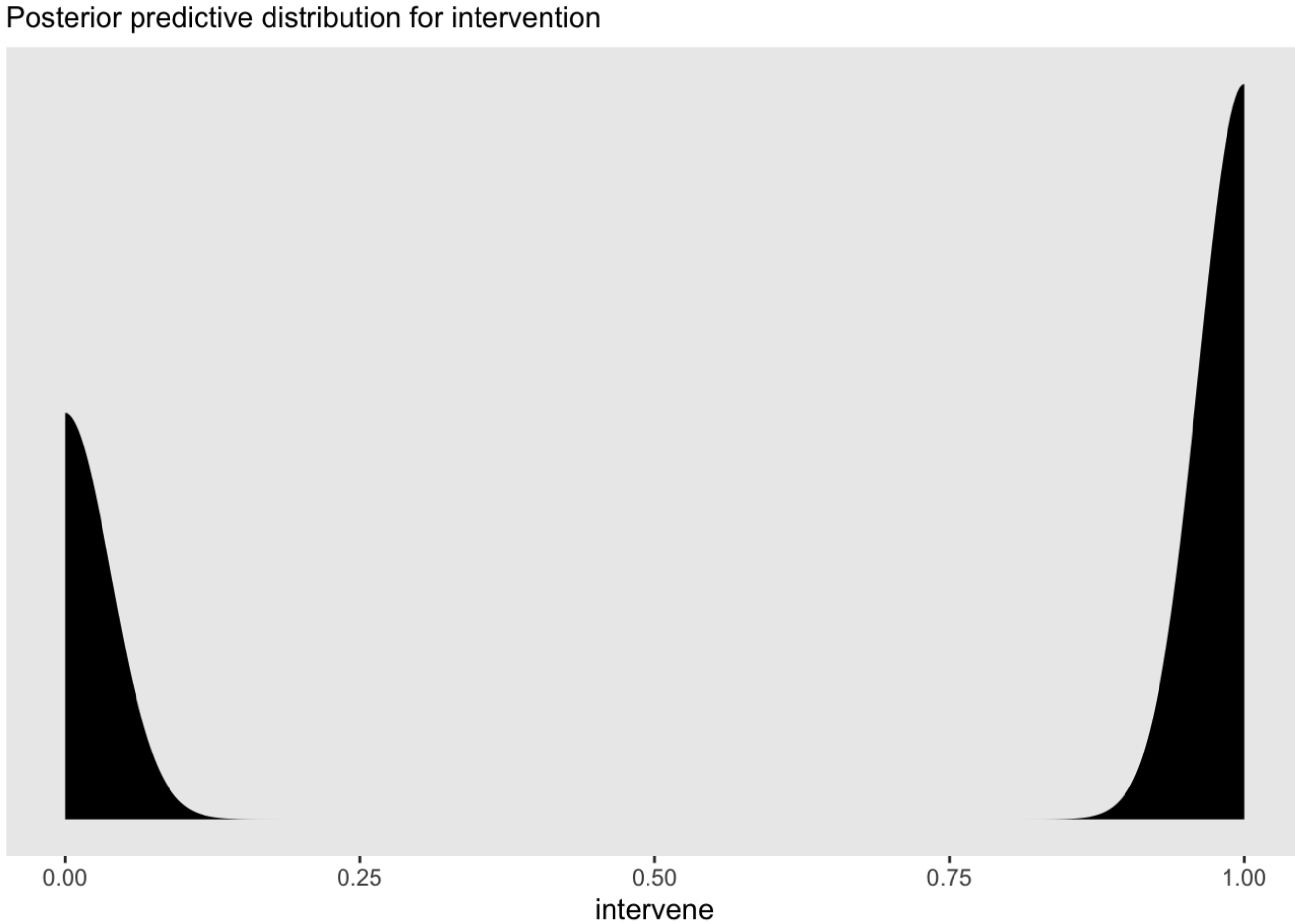
- Summary

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

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: intervene ~ 1 + evidence
##    Data: model_df (Number of observations: 840)
## 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     0.98      0.10     0.78     1.17       2423 1.00
## evidence      1.07      0.08     0.92     1.24       2169 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).
```

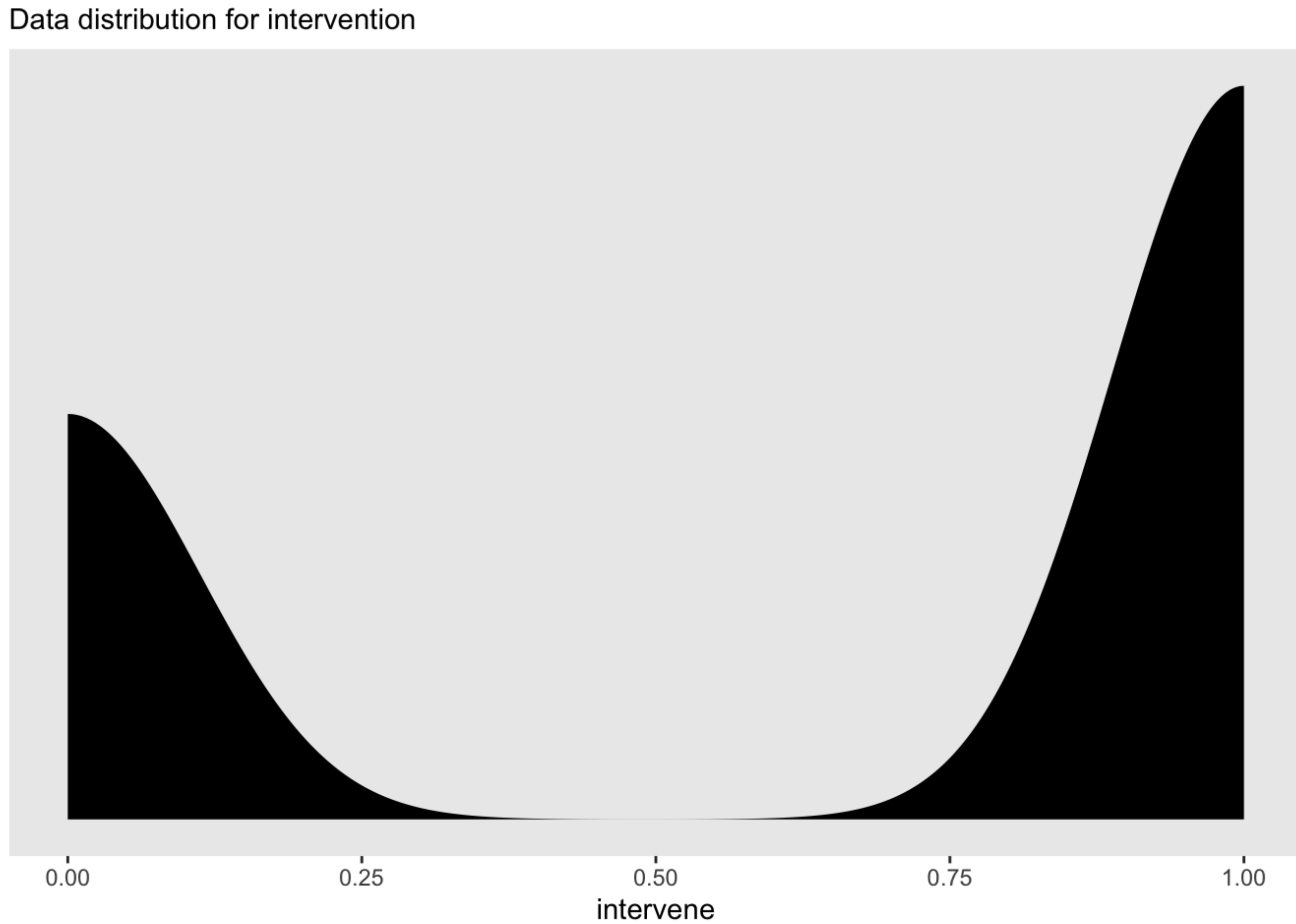
Let’s check out a posterior predictive distribution for intervention decisions.

```
# posterior predictive check
model_df %>%
  select(evidence) %>%
  add_predicted_draws(m.logistic, prediction = "intervene", seed = 1234, n = 200) %>%
  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())
```



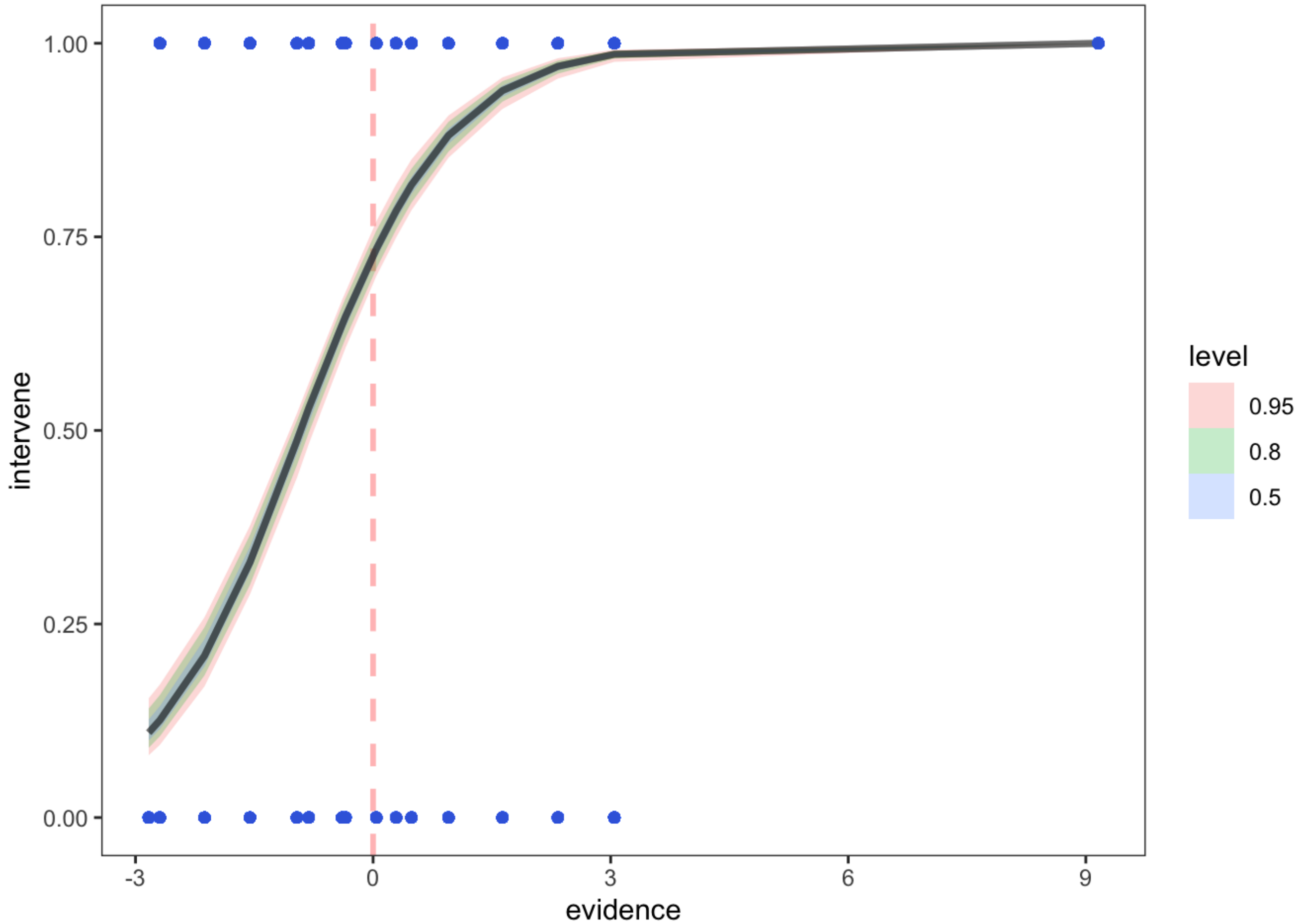
How do the posterior predictions compare to the observed data?

```
# 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())
```



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) +
  geom_point(alpha = .15, color = "royalblue") +
  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 heterogeniety 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.

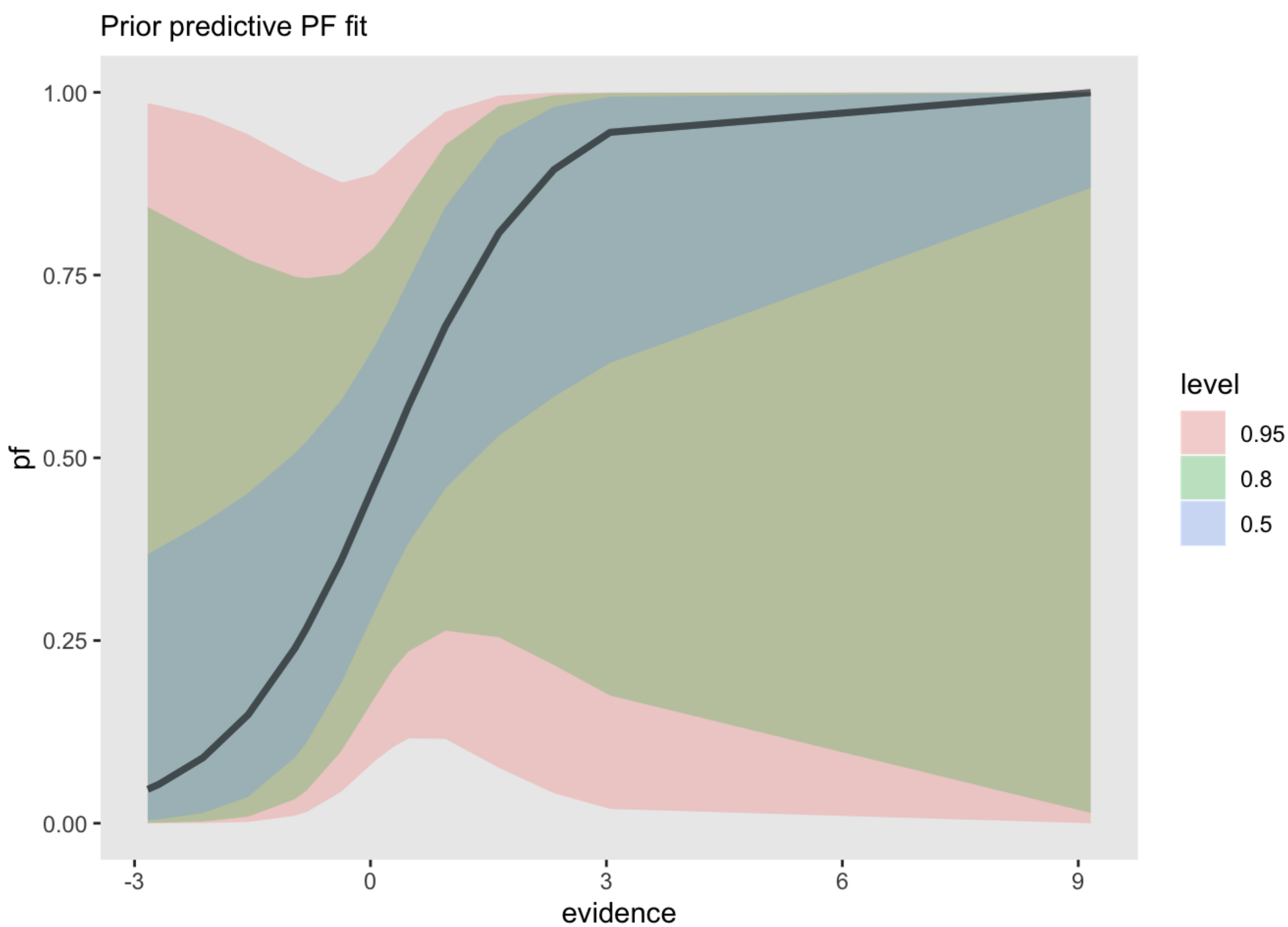
```
# get_prior(data = model_df, family = bernoulli(link = "logit"), formula = bf(intervene
 ~ (1 + evidence|worker_id) + evidence))

# 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) +
  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())
```

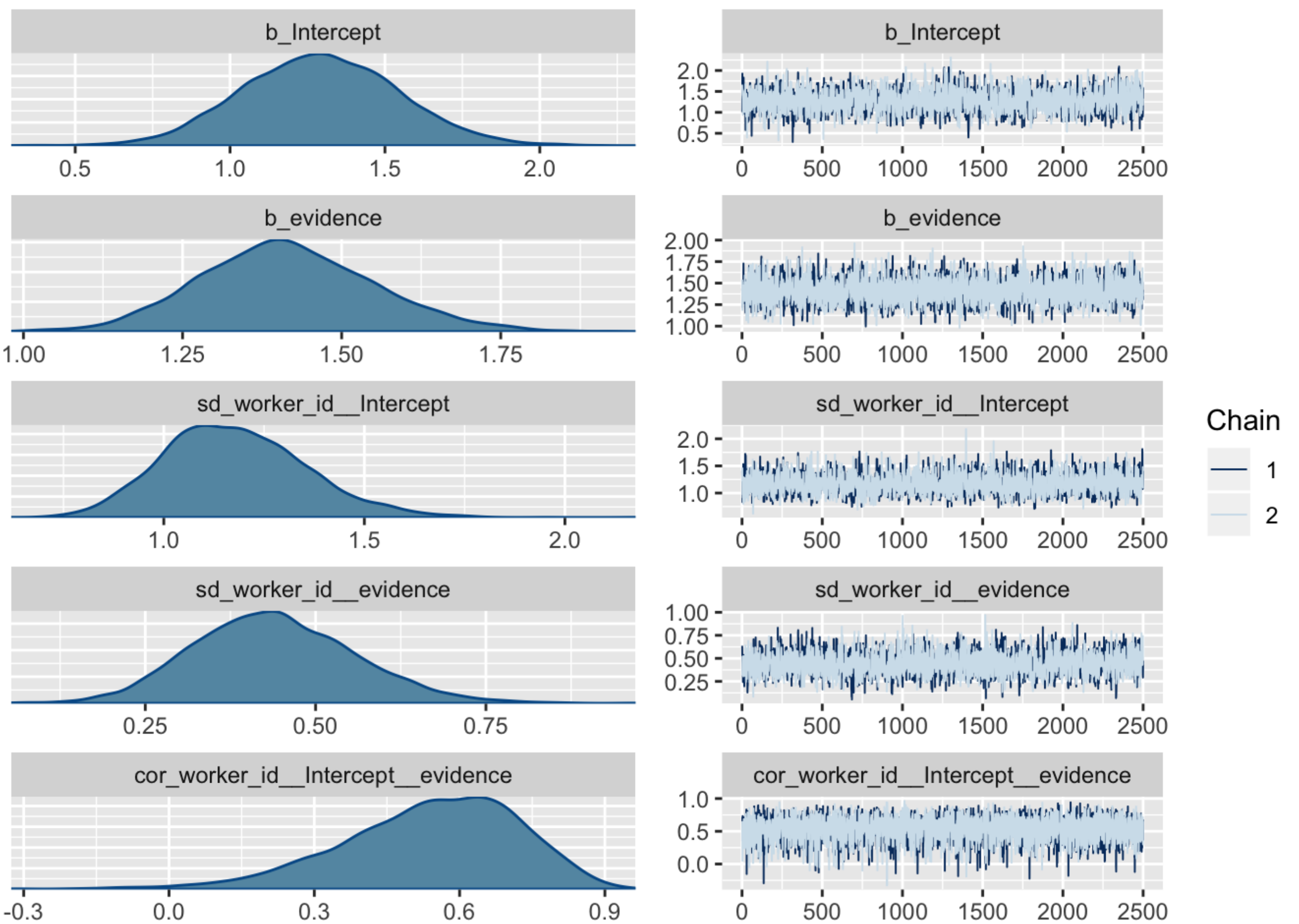


```
# hierarchical linear model with logit link
m.wrkr.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                       formula = bf(intervene ~ (1 + evidence|worker_id) + evidence),
                       prior = c(prior(normal(0, 1), class = Intercept),
                                 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:

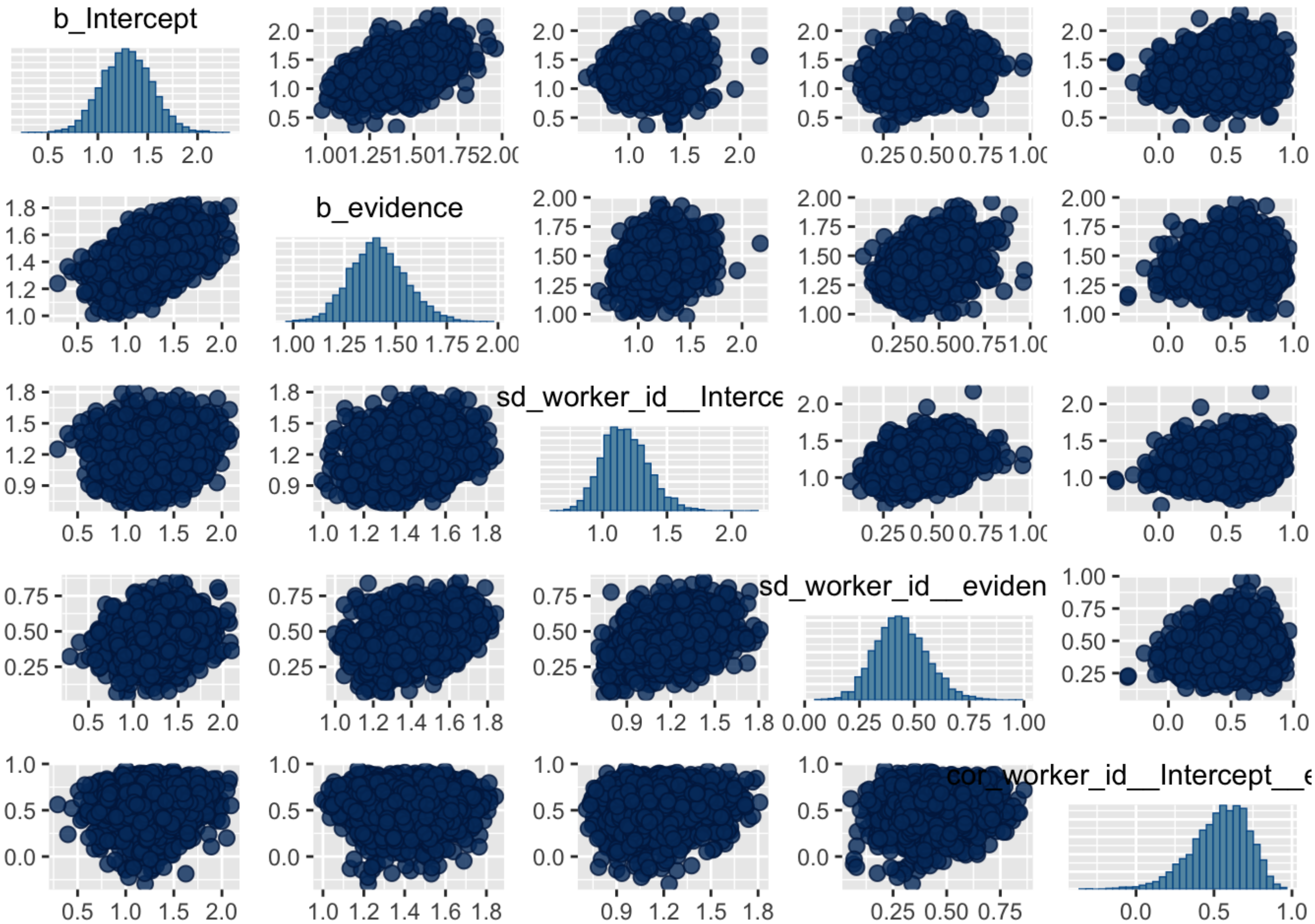
- Trace plots

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



- Pairs plot

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



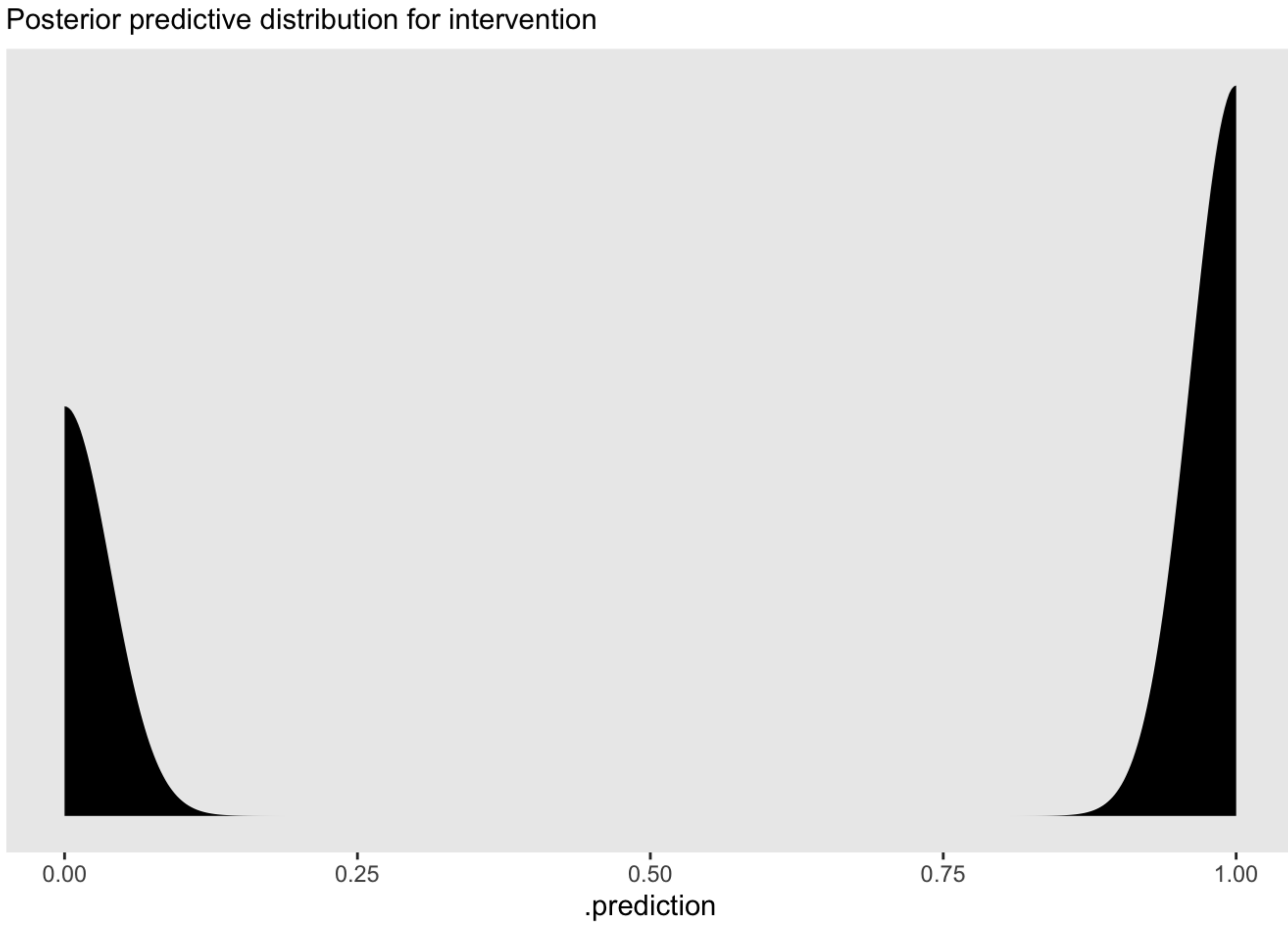
- Summary

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

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence
##    Data: model_df (Number of observations: 840)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 28)
##                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)               1.18      0.18     0.86     1.56       2335
## sd(evidence)                0.44      0.12     0.23     0.69       2417
## cor(Intercept,evidence)     0.54      0.18     0.13     0.84       4566
##                         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     1.29      0.25     0.81     1.79       2174 1.00
## evidence      1.41      0.14     1.16     1.70       3645 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).
```

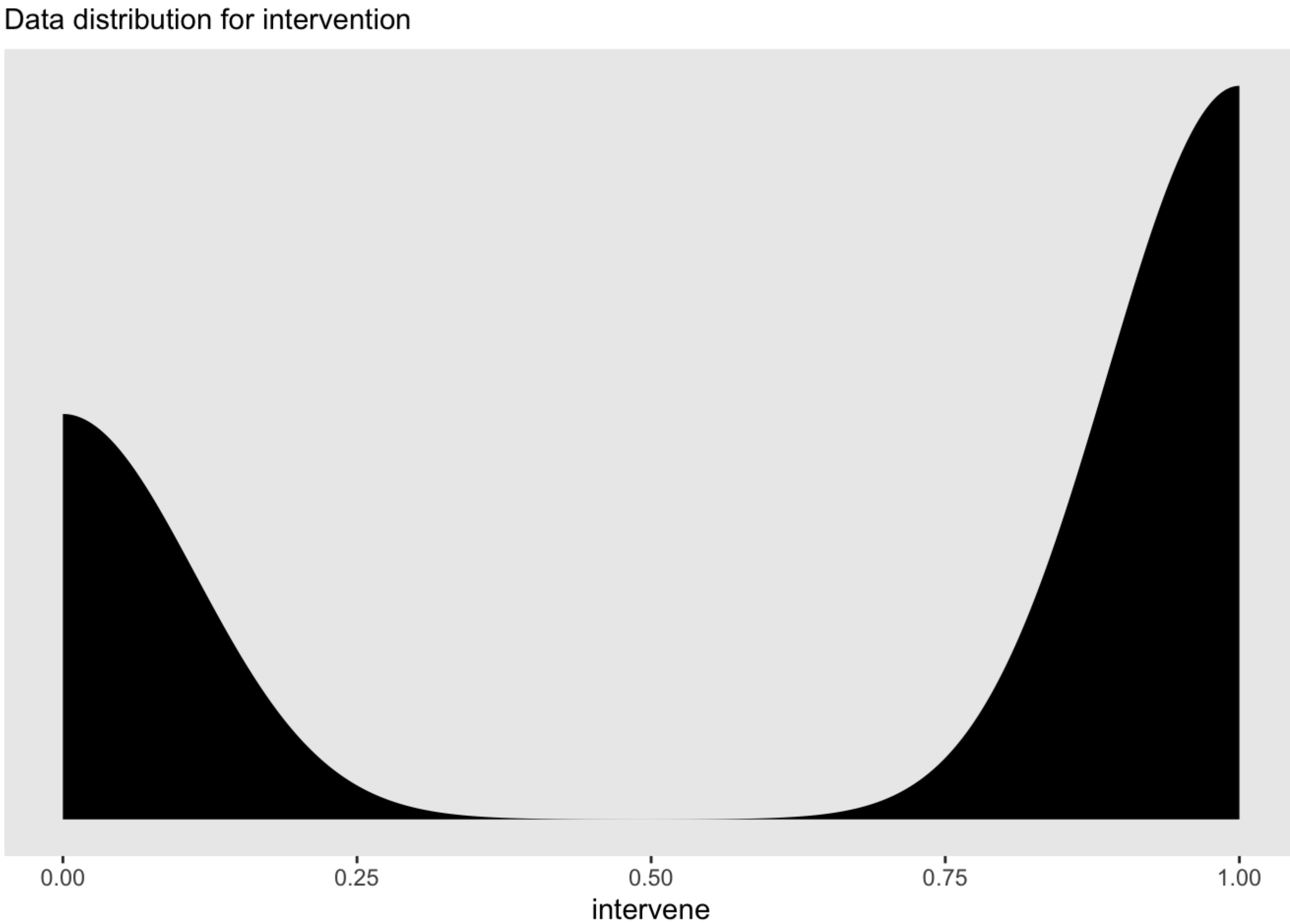
Let’s check out a posterior predictive distribution for intervention decisions.

```
# posterior predictive check
model_df %>%
  select(evidence, worker_id) %>%
  add_predicted_draws(m.wrkr.logistic, seed = 1234, n = 200) %>%
  ggplot(aes(x = .prediction)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior predictive distribution for intervention") +
  theme(panel.grid = element_blank())
```



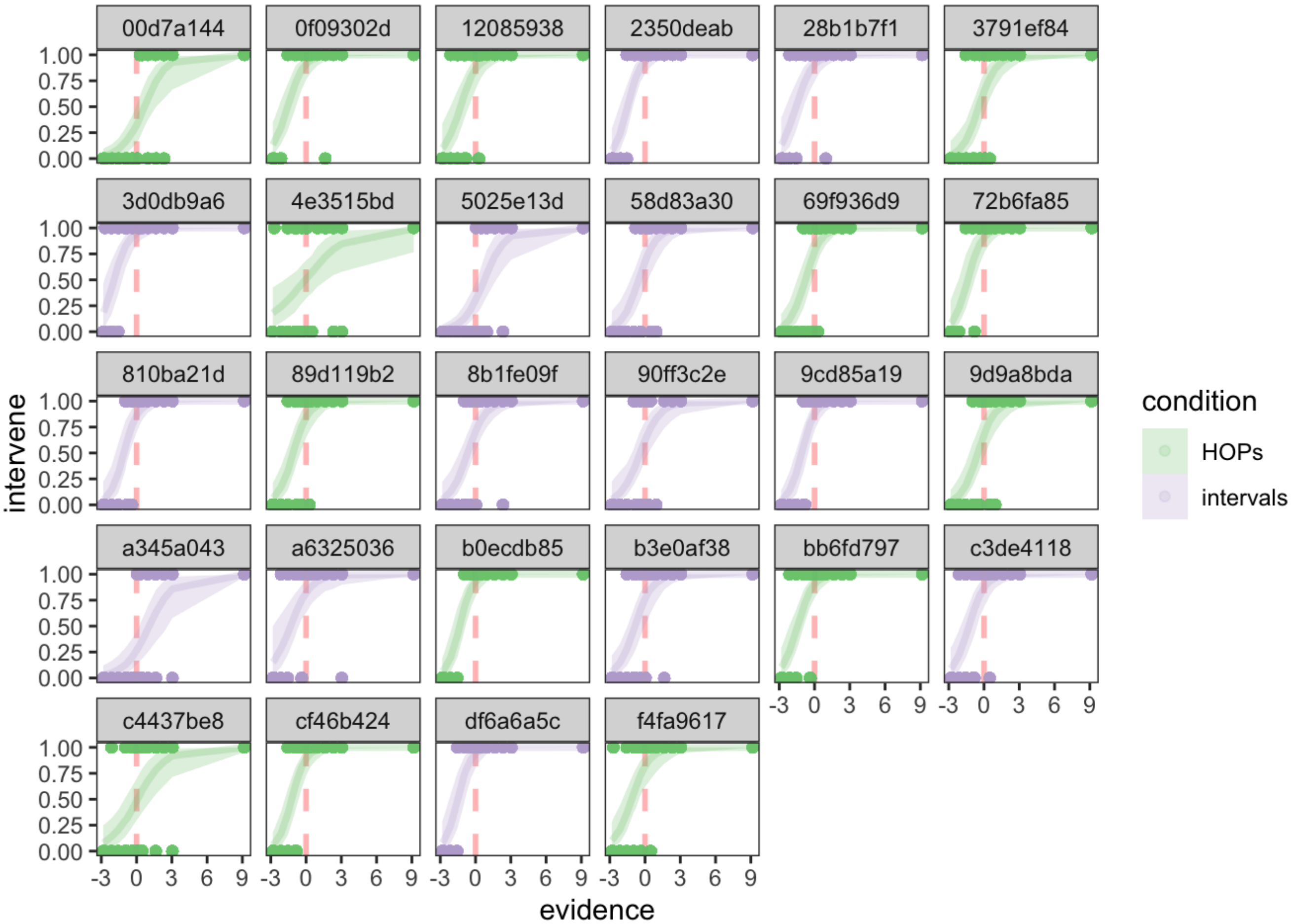
How do the posterior predictions compare to the observed data?

```
# 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())
```



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)
```



Add Predictors for the 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).

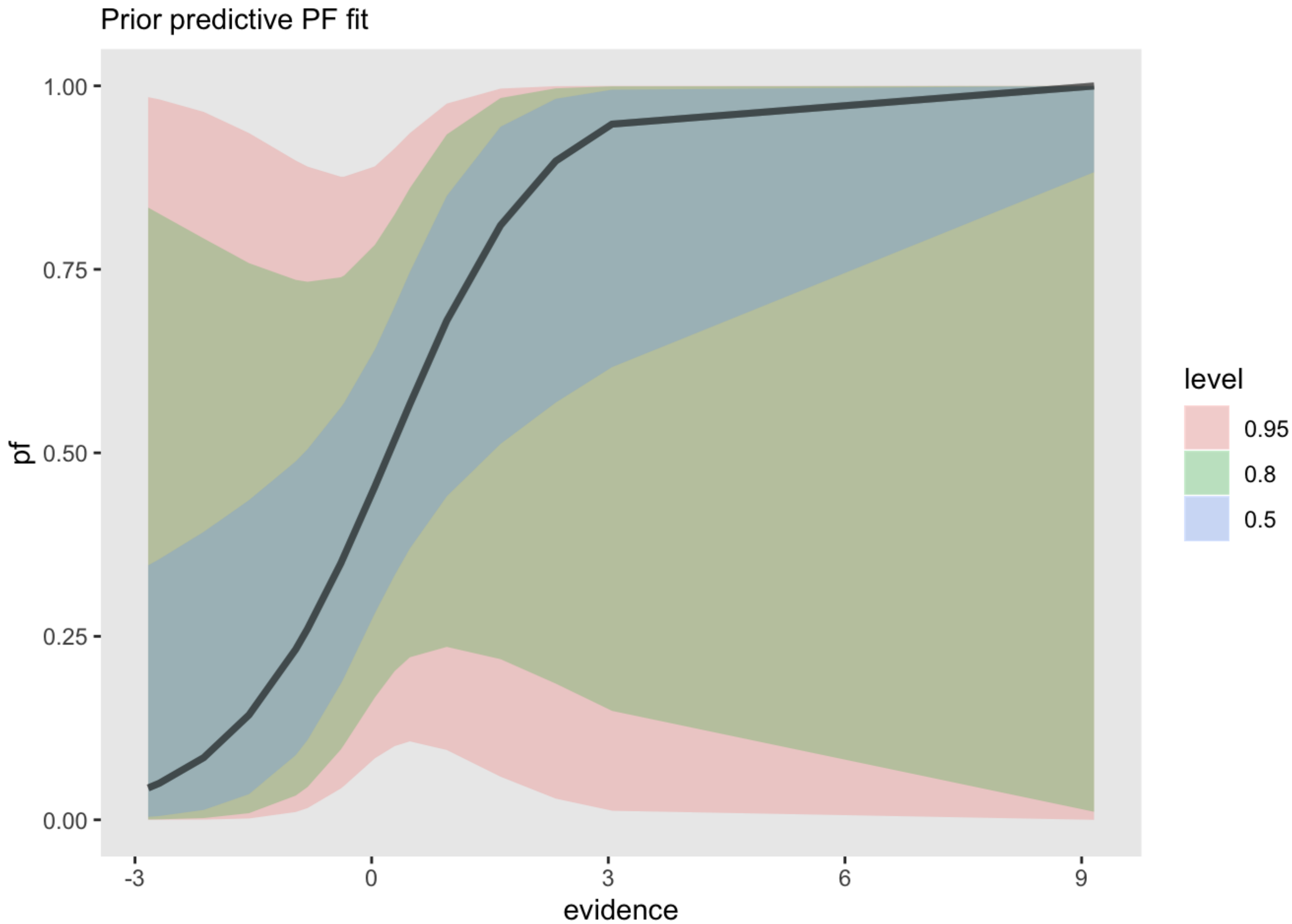
```
# get_prior(data = model_df, family = bernoulli(link = "logit"), formula = bf(intervene
 ~ (1 + evidence|worker_id) + evidence*means))

# 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) +
  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())
```

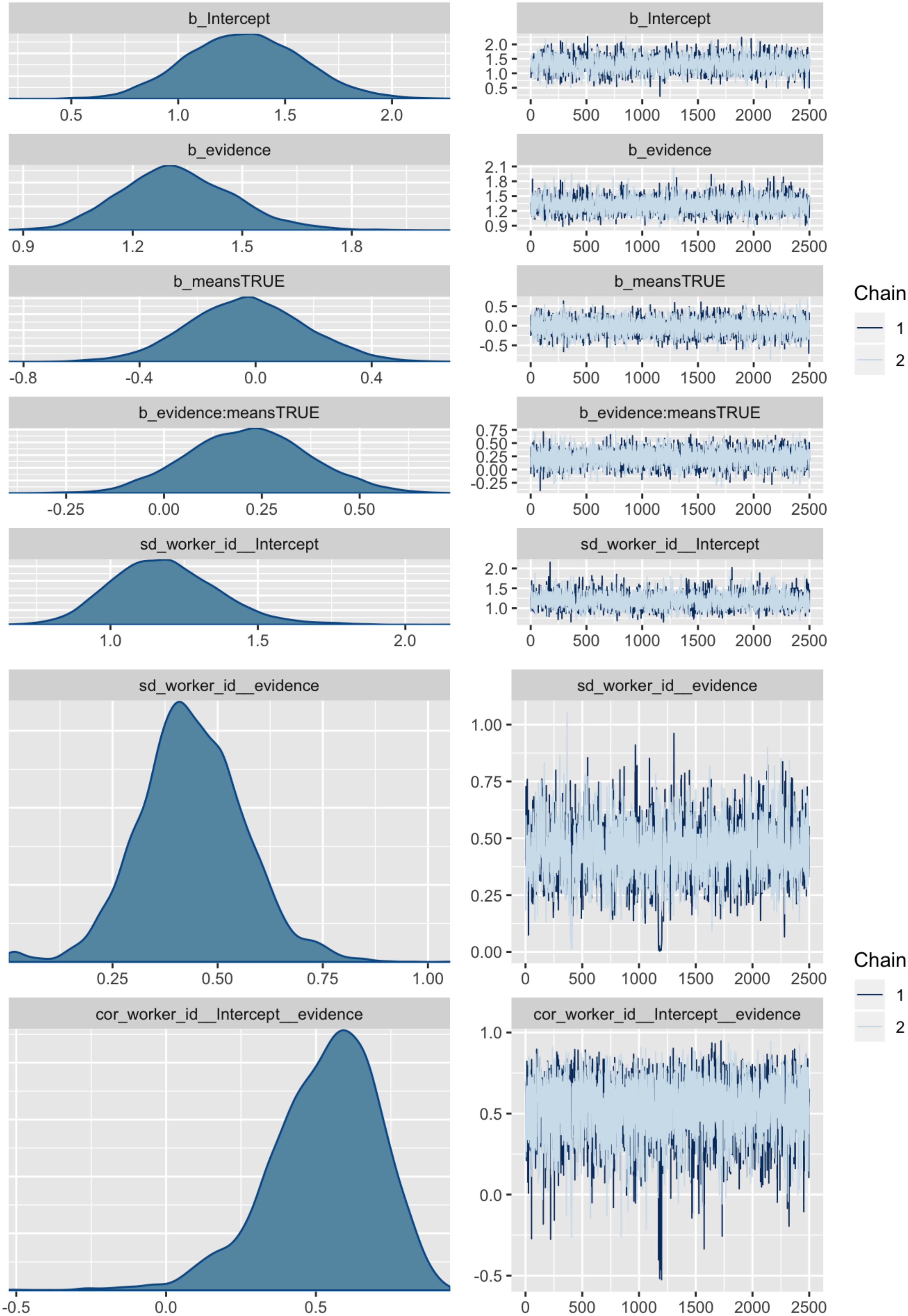



```
# hierarchical linear model with logit link and predictors for the presence/absence of m
eans
m.wrkr.means.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                             formula = bf(intervene ~ (1 + evidence|worker_id) + evidenc
e* means),
                             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:

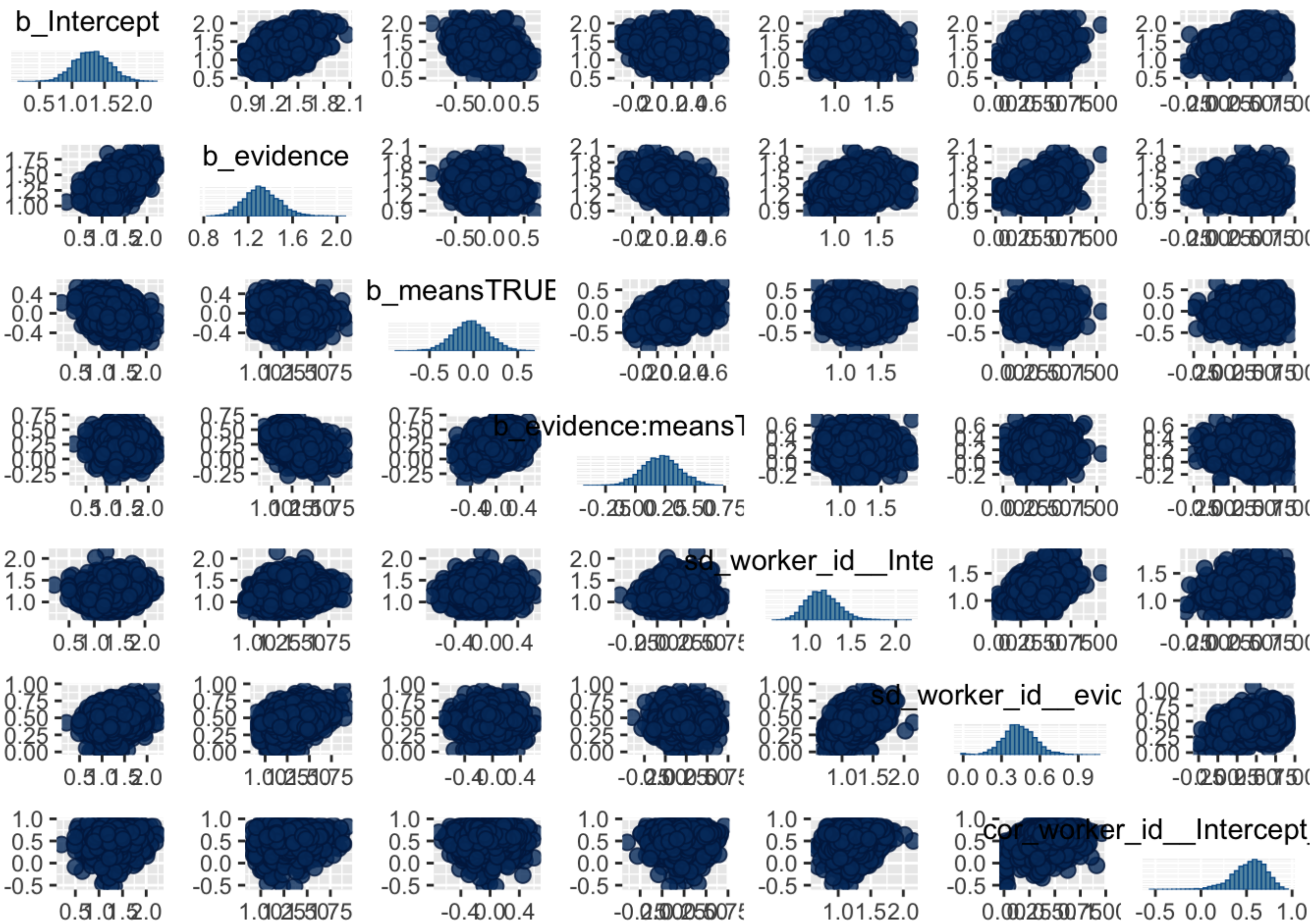
- Trace plots

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


b_Intercept
b_Intercept
b_evidence
b_evidence
b_meansTRUE
b_meansTRUE
b_evidence:meansTRUE
b_evidence:meansTRUE
sd_worker_id__Intercept
sd_worker_id__Intercept
sd_worker_id__evidence
sd_worker_id__evidence
cor_worker_id__Intercept__evidence
cor_worker_id__Intercept__evidence
Chain
1
2
Chain
1
2

- Pairs plot

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



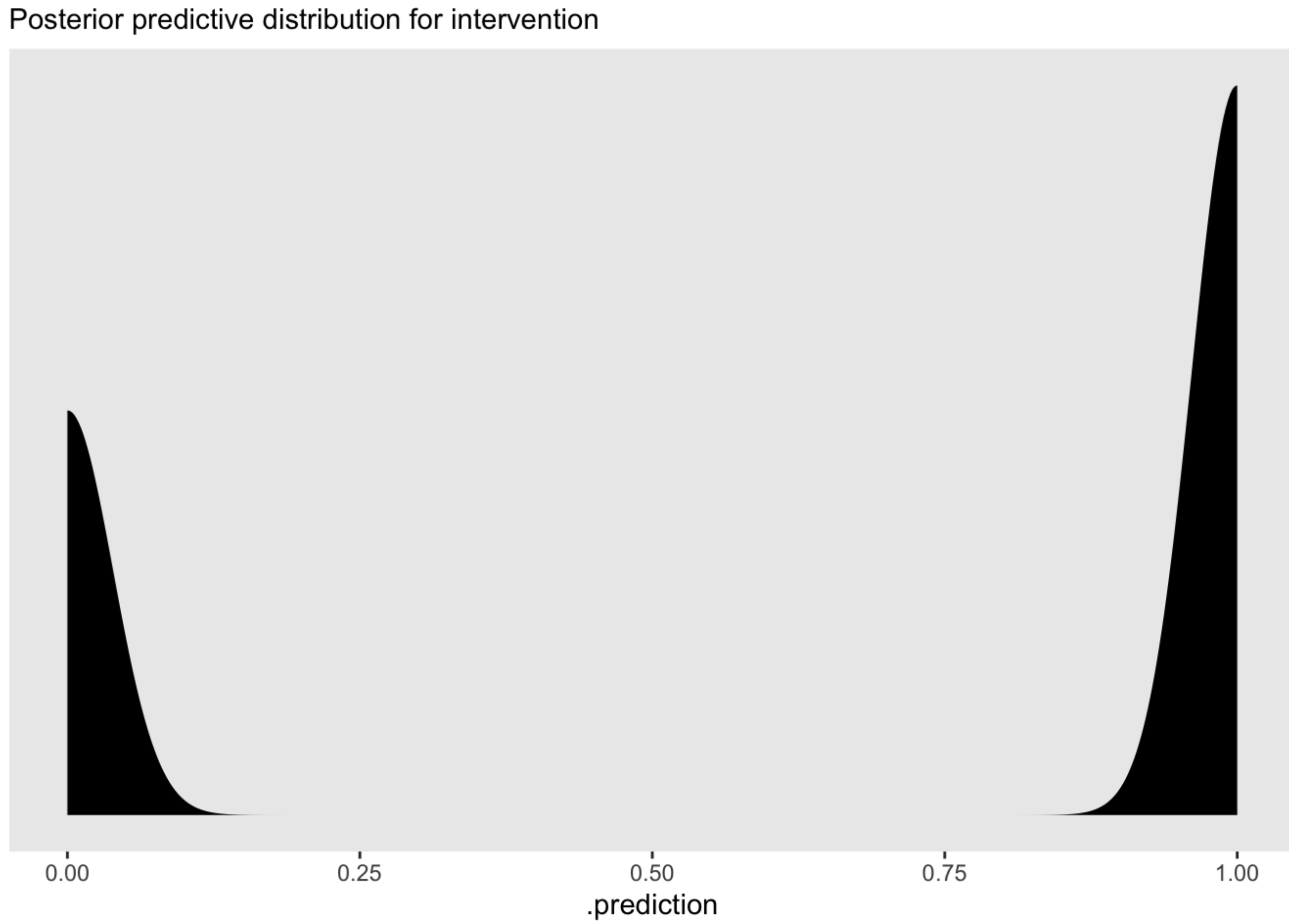
- Summary

```
# 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: 840)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 28)
##                          Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)                1.18      0.18     0.87     1.57       2376
## sd(evidence)                 0.44      0.13     0.20     0.70        946
## cor(Intercept,evidence)      0.53      0.18     0.11     0.83       2101
##                          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              1.31      0.27     0.80     1.85       2041 1.00
## evidence               1.32      0.16     1.04     1.65       2894 1.00
## meansTRUE             -0.03      0.20    -0.42     0.36       6881 1.00
## evidence:meansTRUE     0.22      0.16    -0.08     0.52       5340 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).
```

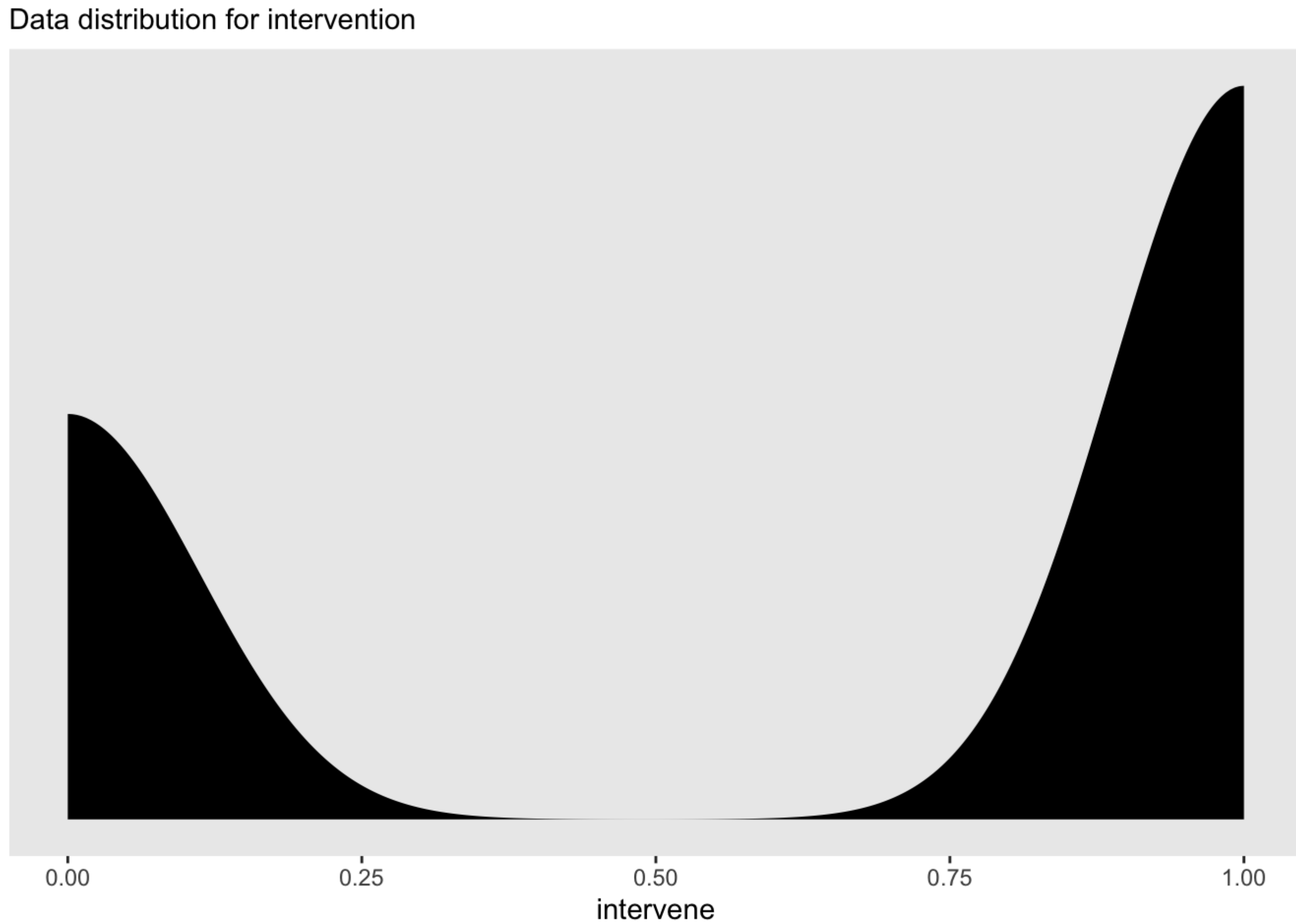
Let’s check out a posterior predictive distribution for intervention decisions.

```
# posterior predictive check
model_df %>%
  select(evidence, worker_id, means) %>%
  add_predicted_draws(m.wrkr.means.logistic, seed = 1234, n = 200) %>%
  ggplot(aes(x = .prediction)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior predictive distribution for intervention") +
  theme(panel.grid = element_blank())
```



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())
```

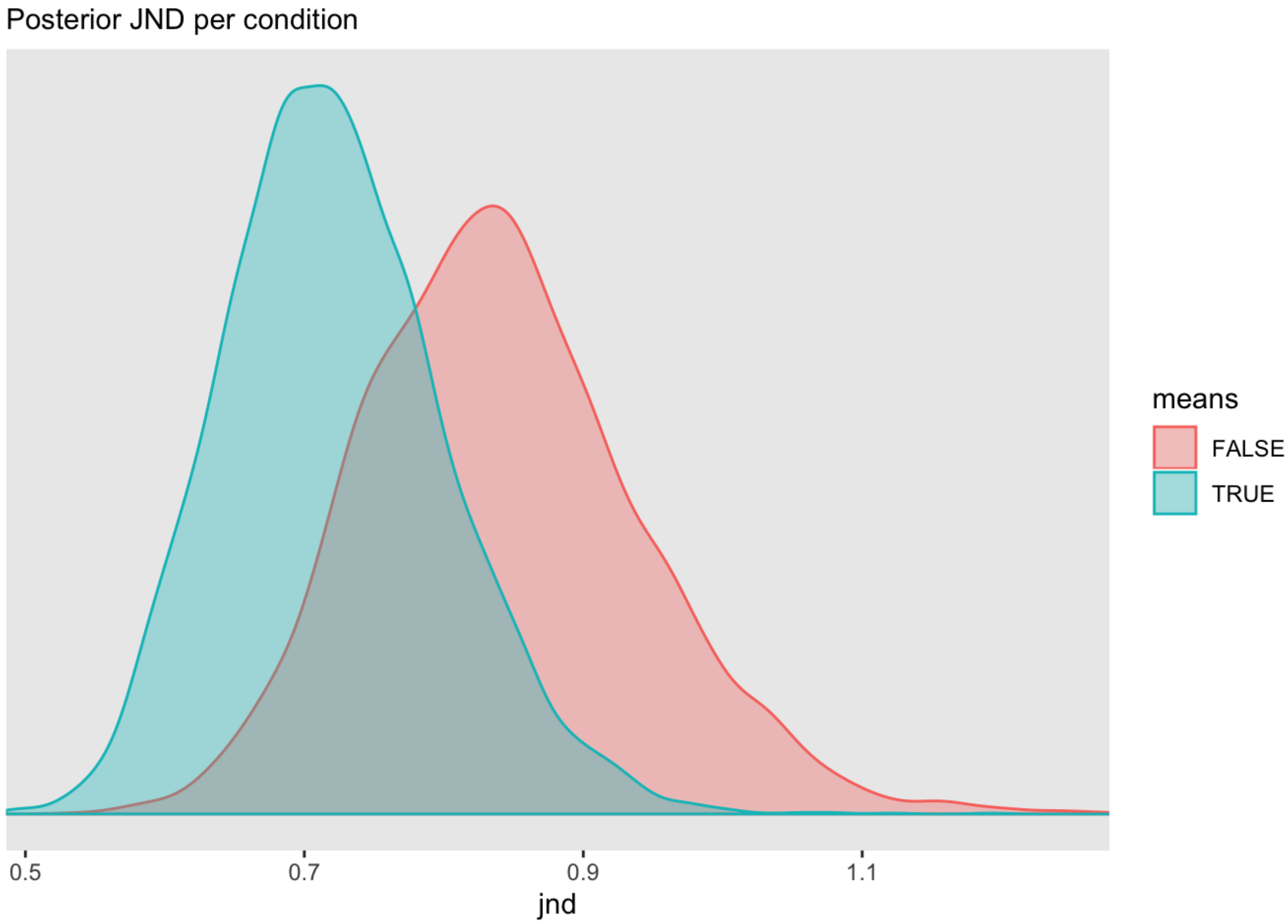


What do the posteriors for just-noticable differences (JND) and points of subjective equality (PSE) look like when means are present vs absent? Since we are probing conditional expectations, we'll forego calculating maringal effects by manually combining parameters. Instead we'll use add_fitted_draws 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.

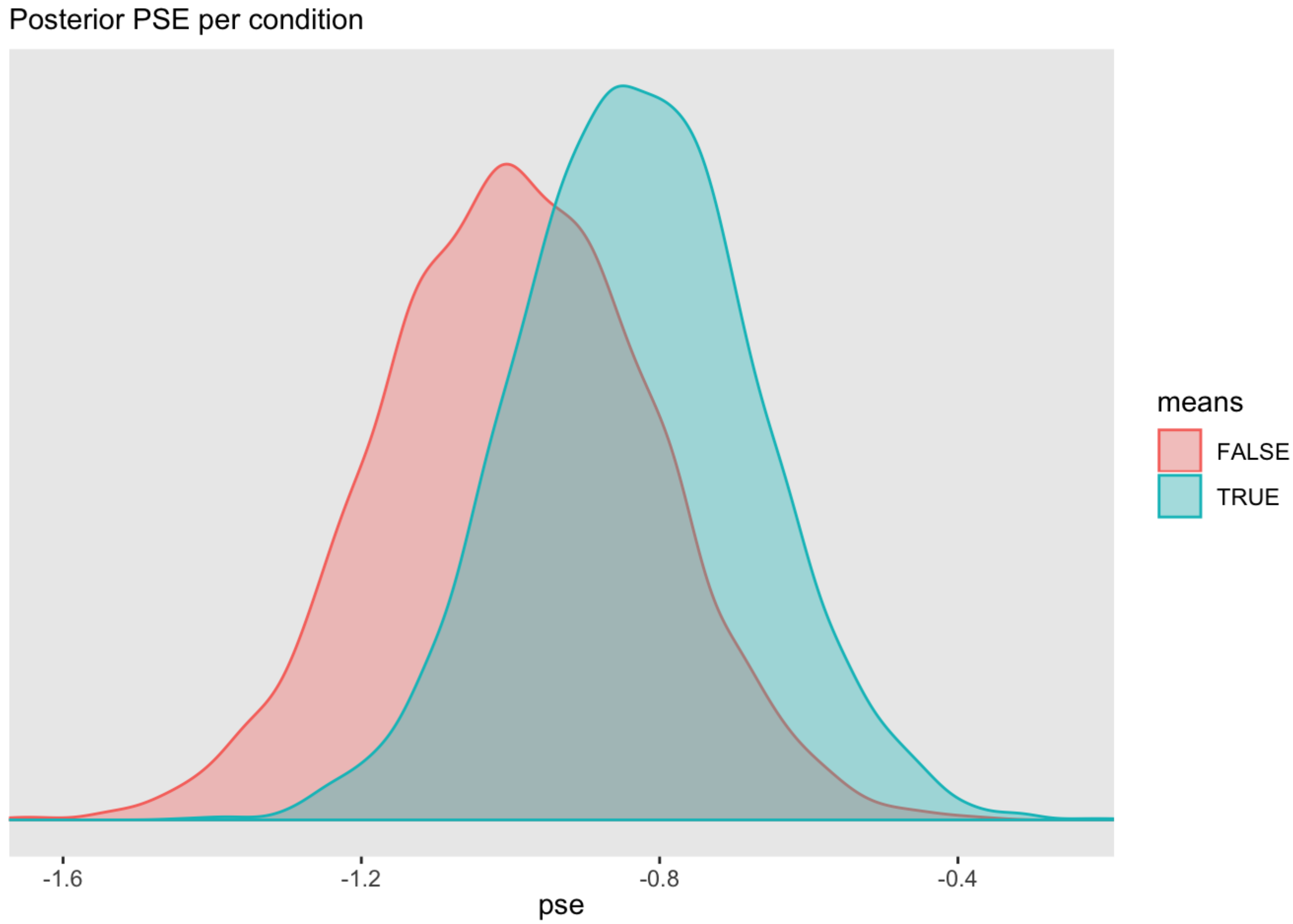
```
stats_df %>%
  ggplot(aes(x = jnd, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  theme(panel.grid = element_blank())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



It looks like users are more sensitive to evidence (i.e., JNDs are smaller) when means are present, ignoring the effect of visualization condition.

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

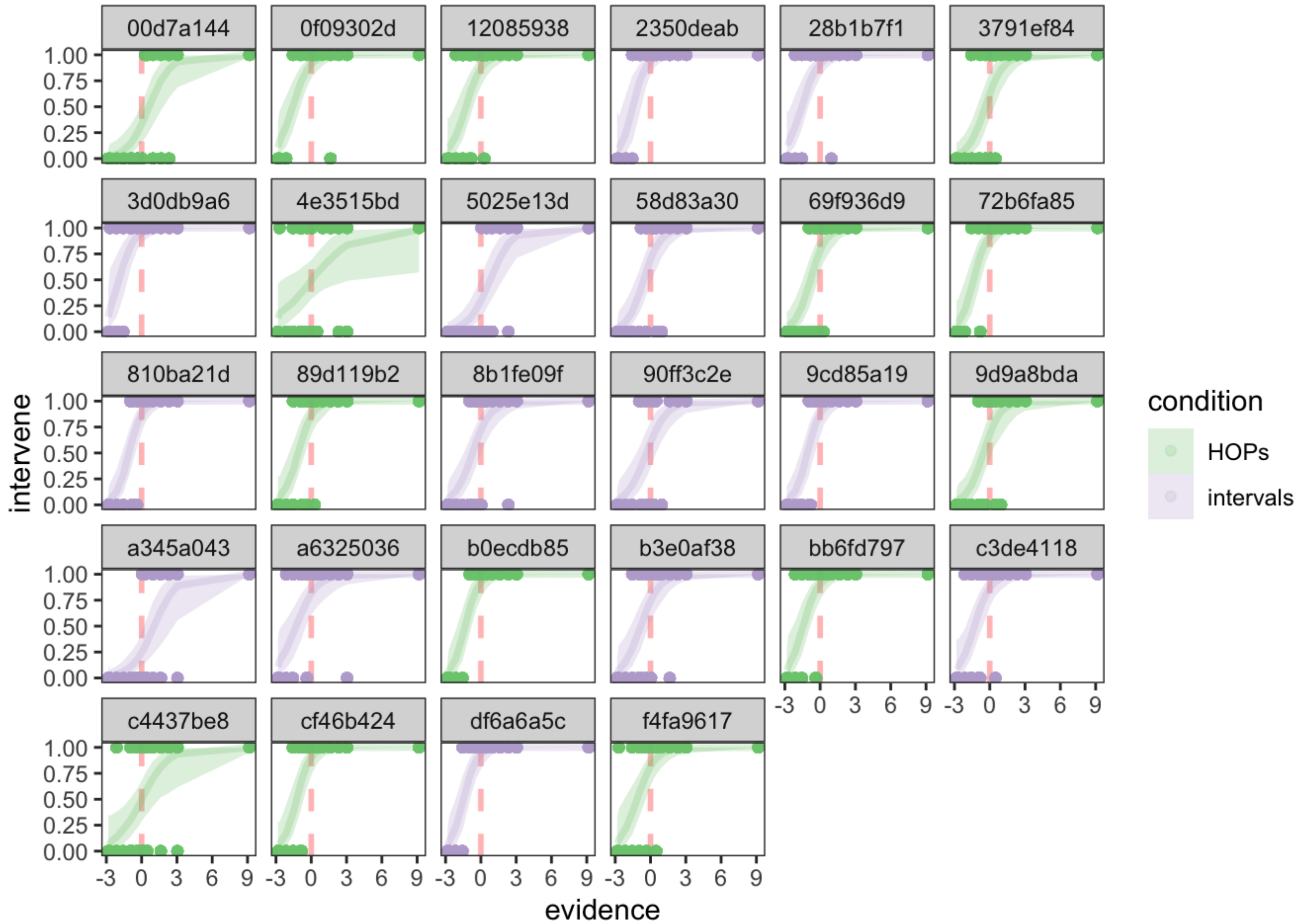
```
stats_df %>%
  ggplot(aes(x = pse, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per condition") +
  theme(panel.grid = element_blank())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



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

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.

Predictors for Visualization Condition

We also want to know whether difference 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).

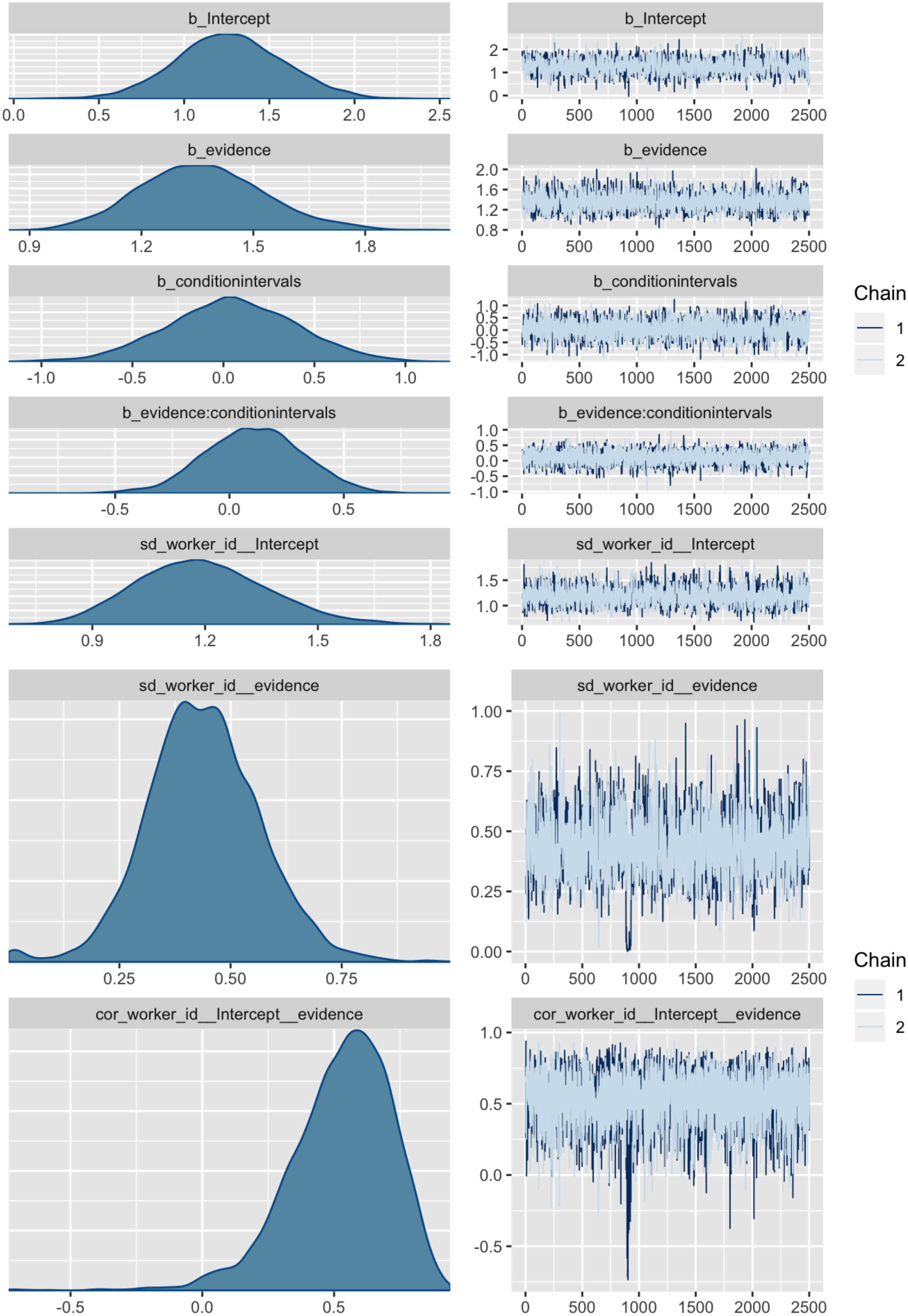
We'll use the same priors as we did for our previous model. Now, let's fit our model.

```
# hierarchical linear model with logit link and predictors per visualization condition
m.wrkr.vis.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                           formula = bf(intervene ~ (1 + evidence|worker_id) + evidence*
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_vis")
```

Check diagnostics:

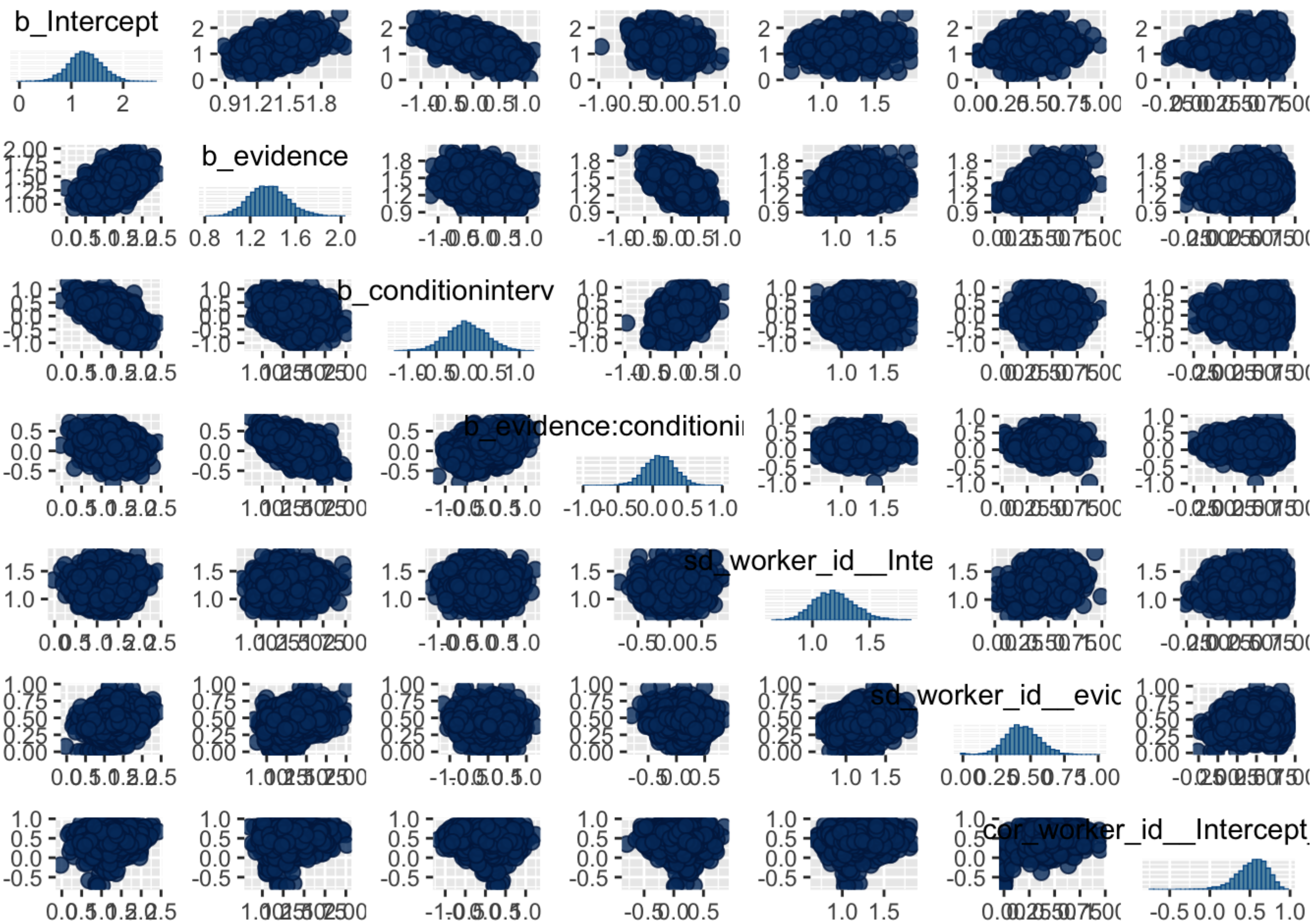
- Trace plots

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


b_Intercept
0.0 0.5 1.0 1.5 2.0 2.5
b_Intercept
0 1 2
0 500 1000 1500 2000 2500
b_evidence
0.9 1.2 1.5 1.8
b_evidence
0.8 1.2 1.6 2.0
0 500 1000 1500 2000 2500
b_conditionintervals
-1.0 -0.5 0.0 0.5 1.0
b_conditionintervals
-1.0 -0.5 0.0 0.5 1.0
0 500 1000 1500 2000 2500
Chain
1
2
b_evidence:conditionintervals
-0.5 0.0 0.5
b_evidence:conditionintervals
-1.0 -0.5 0.0 0.5 1.0
0 500 1000 1500 2000 2500
sd_worker_id__Intercept
0.9 1.2 1.5 1.8
sd_worker_id__Intercept
1.0 1.5
0 500 1000 1500 2000 2500
sd_worker_id__evidence
0.25 0.50 0.75
sd_worker_id__evidence
0.00 0.25 0.50 0.75 1.00
0 500 1000 1500 2000 2500
Chain
1
2
cor_worker_id__Intercept__evidence
-0.5 0.0 0.5
cor_worker_id__Intercept__evidence
-0.5 0.0 0.5 1.0
0 500 1000 1500 2000 2500

- Pairs plot

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



- Summary

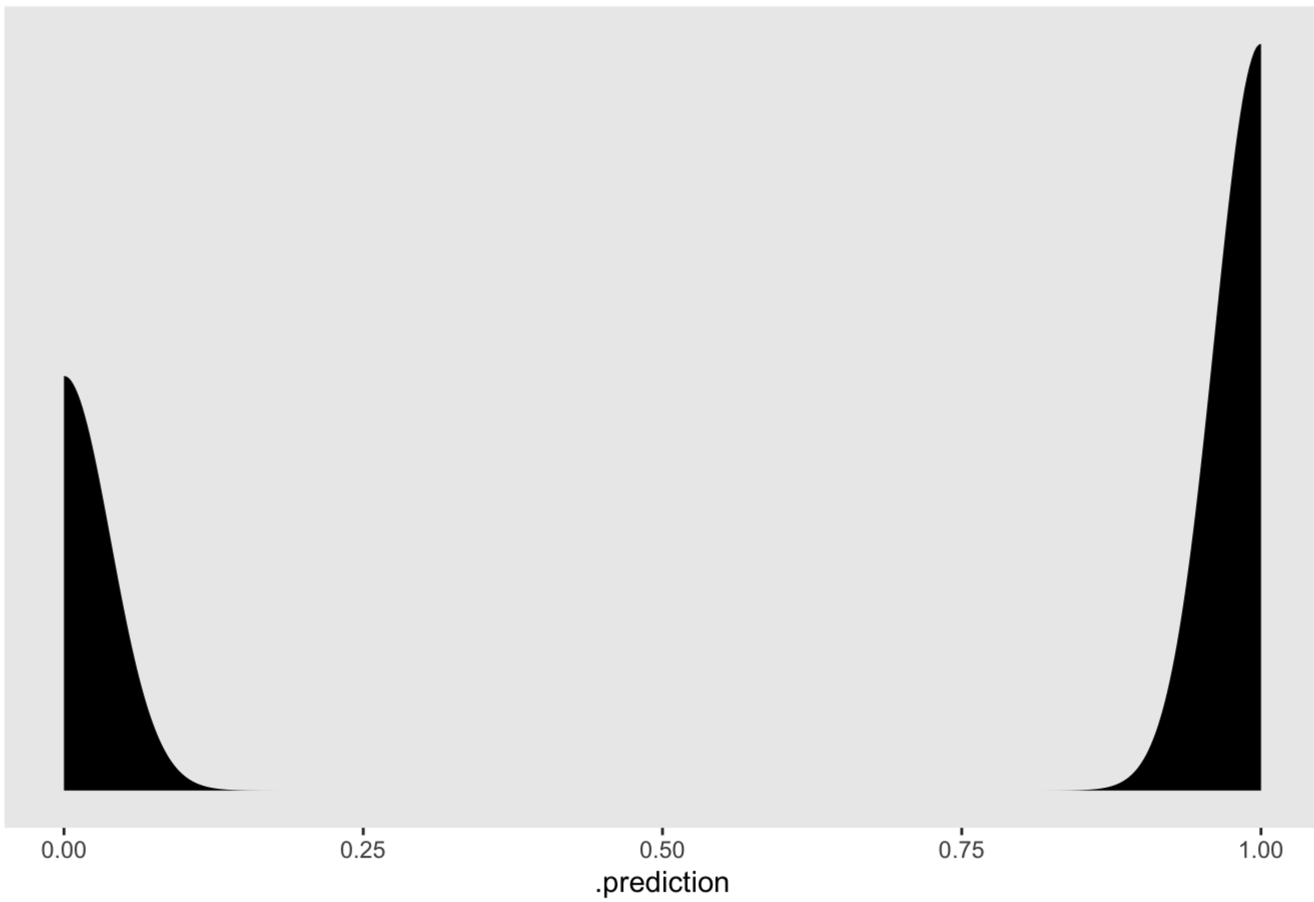
```
# model summary
print(m.wrkr.vis.logistic)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence * condition
##    Data: model_df (Number of observations: 840)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 28)
##                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)               1.19      0.18     0.88     1.57       2103
## sd(evidence)                0.44      0.13     0.19     0.69        823
## cor(Intercept,evidence)     0.53      0.19     0.08     0.83       1435
##                         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                       1.27      0.32     0.66     1.91
## evidence                        1.36      0.17     1.05     1.72
## conditionintervals              0.05      0.35    -0.65     0.74
## evidence:conditionintervals     0.11      0.21    -0.31     0.52
##                             Eff.Sample Rhat
## Intercept                         1651 1.00
## evidence                          2534 1.00
## conditionintervals                2550 1.00
## evidence:conditionintervals       3241 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.

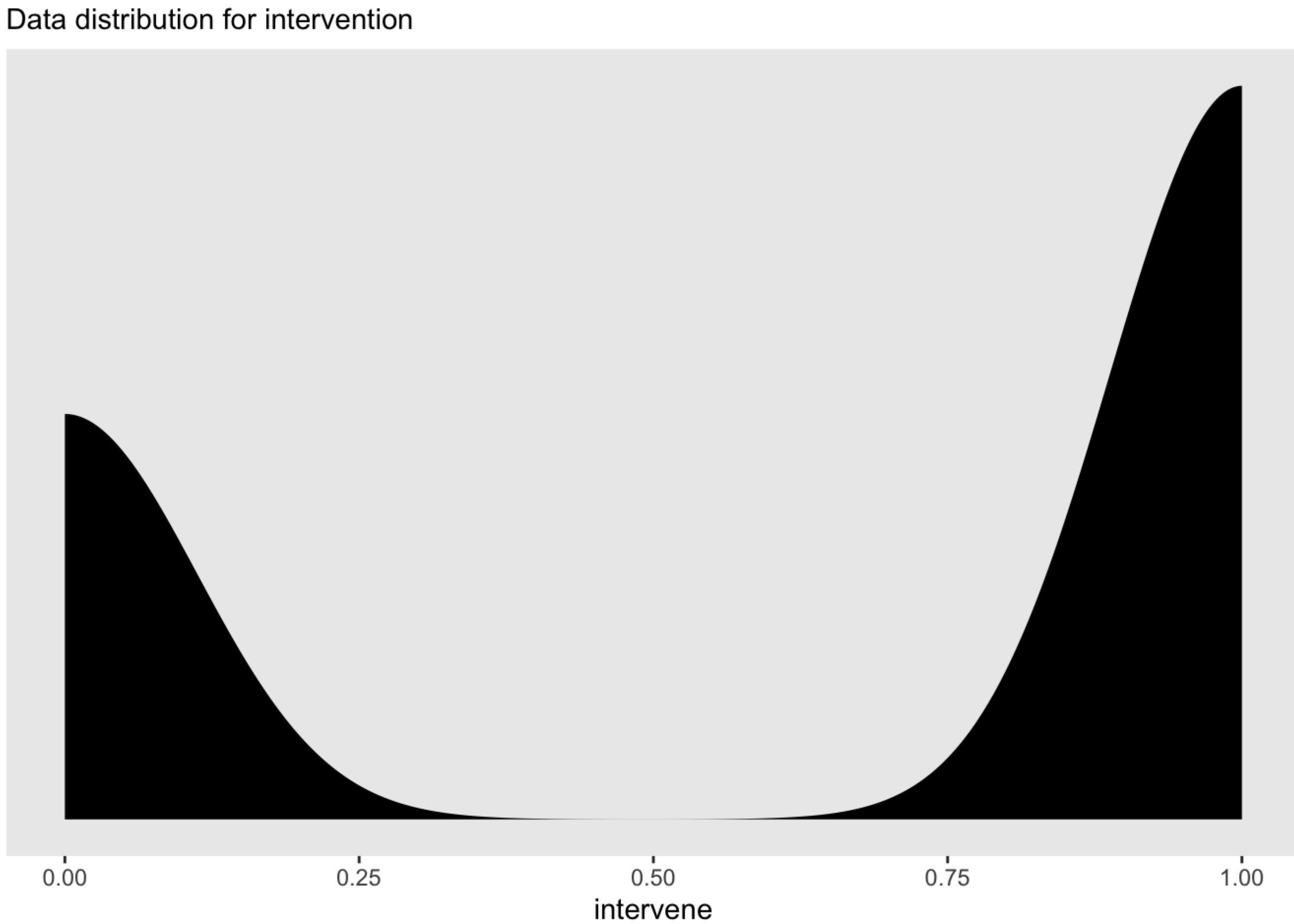
```
# posterior predictive check
model_df %>%
  select(evidence, worker_id, condition) %>%
  add_predicted_draws(m.wrkr.vis.logistic, seed = 1234, n = 200) %>%
  ggplot(aes(x = .prediction)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior predictive distribution for intervention") +
  theme(panel.grid = element_blank())
```

Posterior predictive distribution for intervention



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())
```

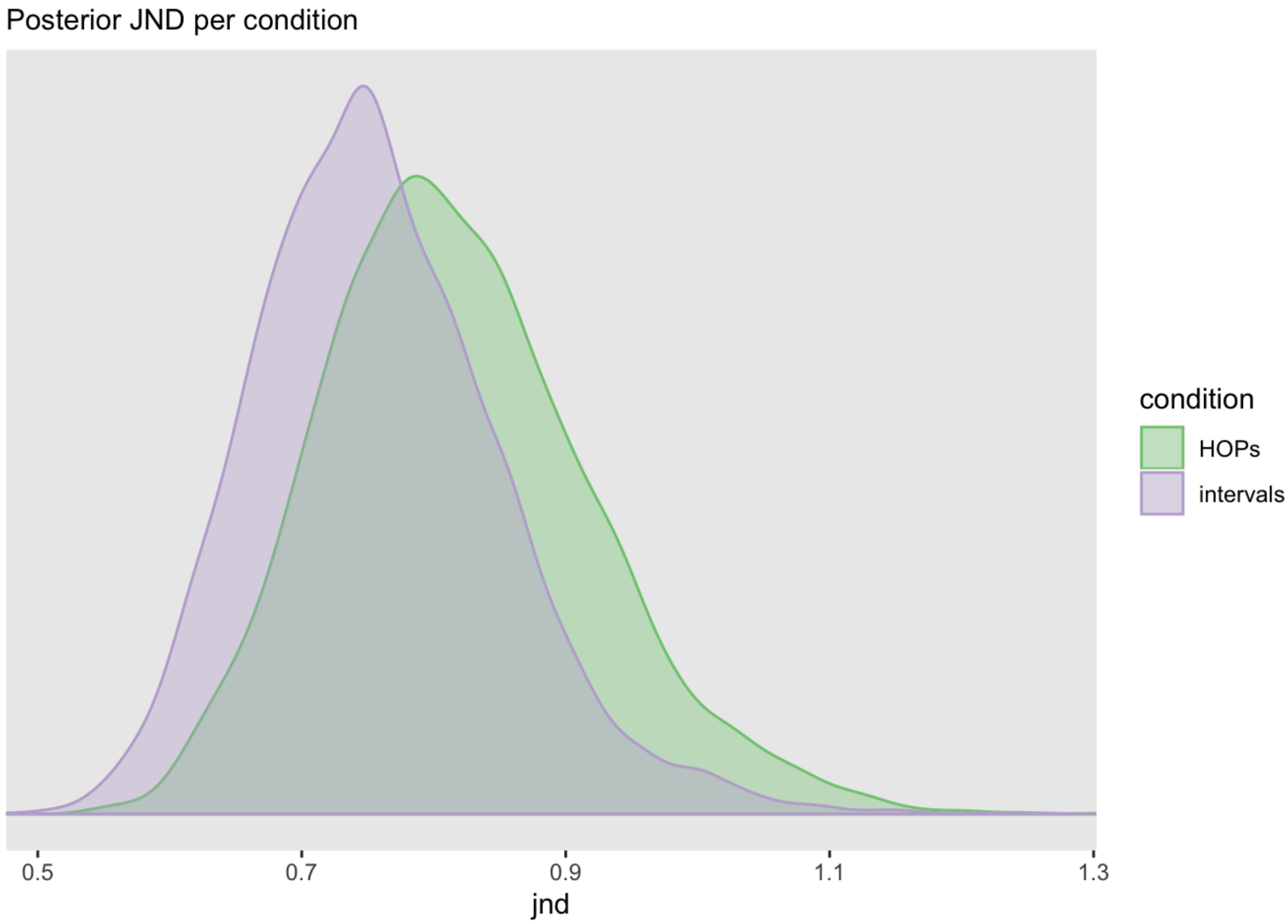


What do the posteriors for just-noticable differences (JND) and points of subjective equality (PSE) look like? We'll start by getting the slopes and intercepts of the linear model and using these to derive estimates of the JND and PSE for each fitted draw.

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(condition) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.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(condition) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.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("condition", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )
```

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

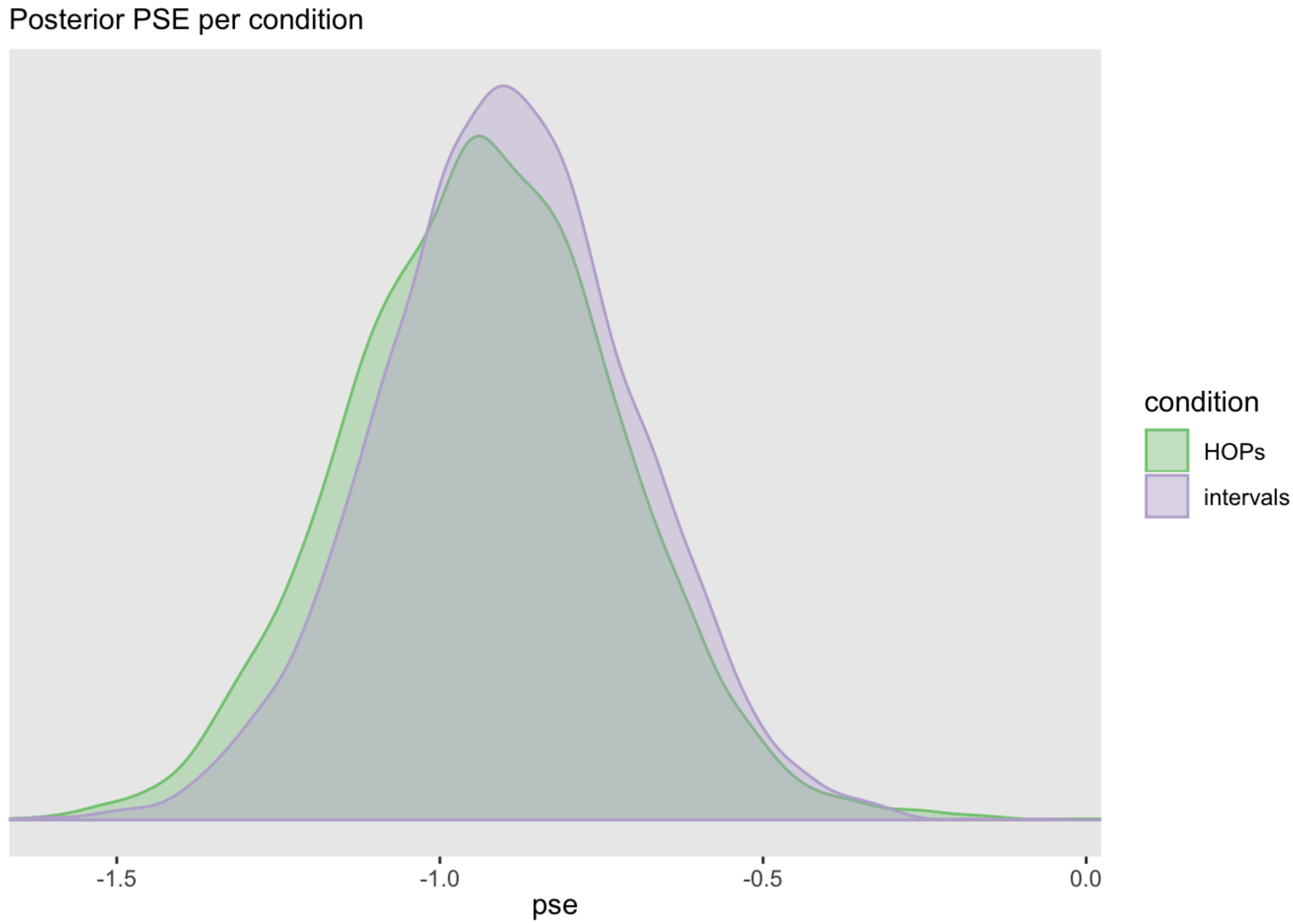
```
stats_df %>%
  ggplot(aes(x = jnd, group = condition, color = condition, fill = condition)) +
  geom_density(alpha = 0.35) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL,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



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

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

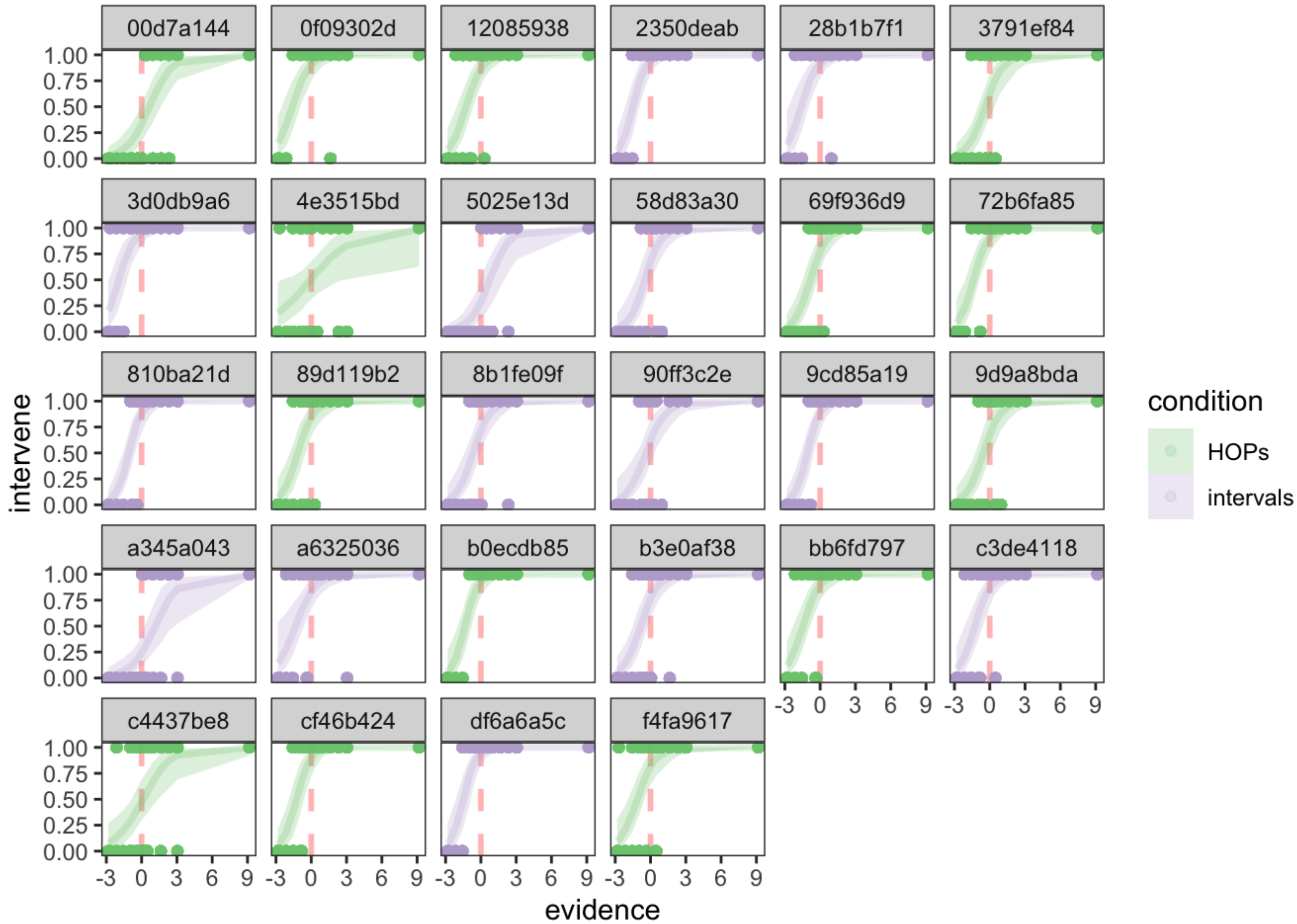
```
stats_df %>%
  ggplot(aes(x = pse, group = condition, color = condition, fill = condition)) +
  geom_density(alpha = 0.35) +
  scale_fill_brewer(type = "qual", palette = 1) +
  scale_color_brewer(type = "qual", palette = 1) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL,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



It looks like the point of subjective equality is about the same across conditions, ignoring the impact of extrinsic means.

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

```
model_df %>%
  group_by(evidence, worker_id, condition) %>%
  add_fitted_draws(m.wrkr.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)
```



Interaction Between Presence/Absence of the Mean and Visualization Condition

Let's incorporate predictors for the presence/absence of the mean, visualization condition, and their interaction into one model.

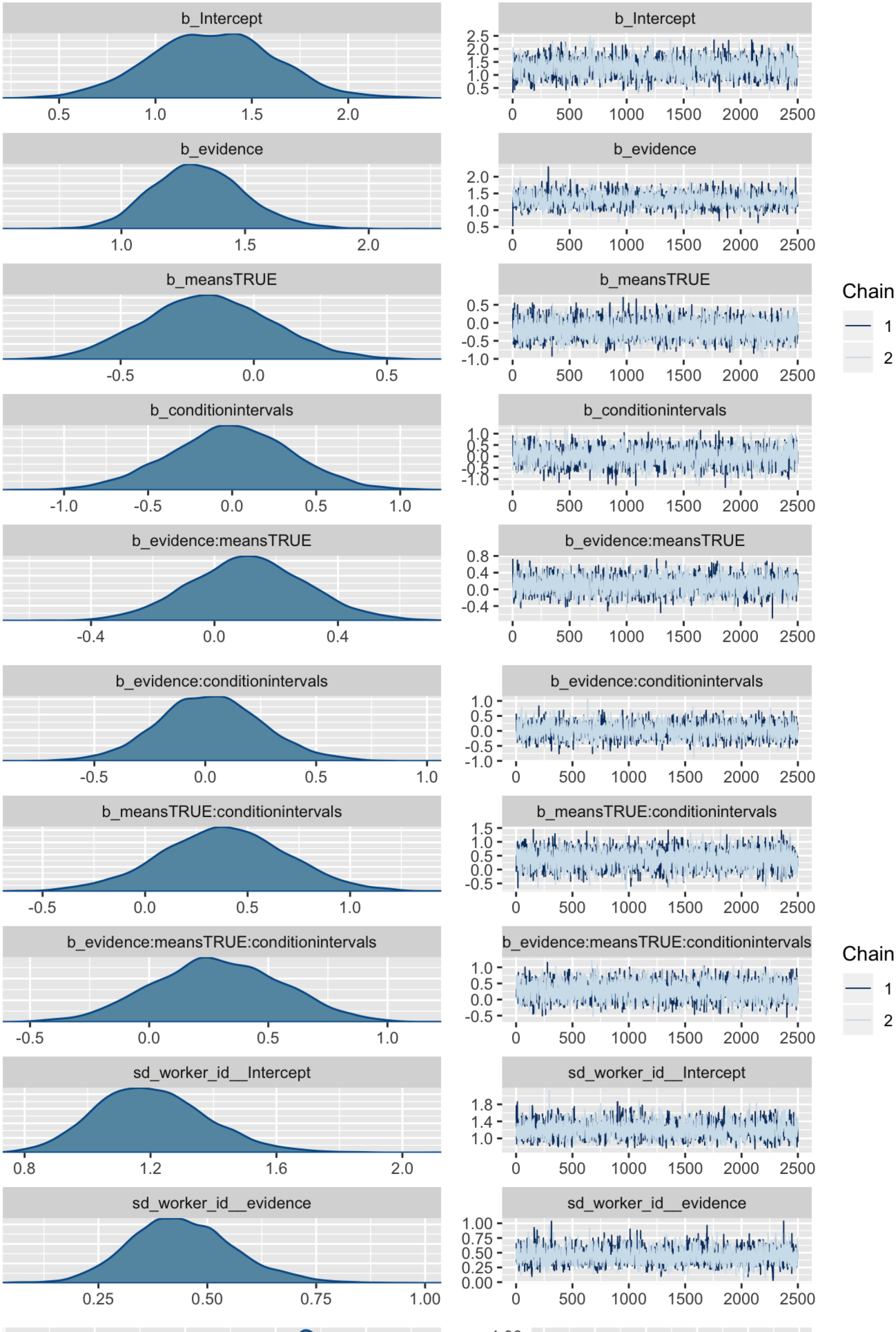
Again, we'll use the same priors as we did for our previous models. Now, let's fit our model.

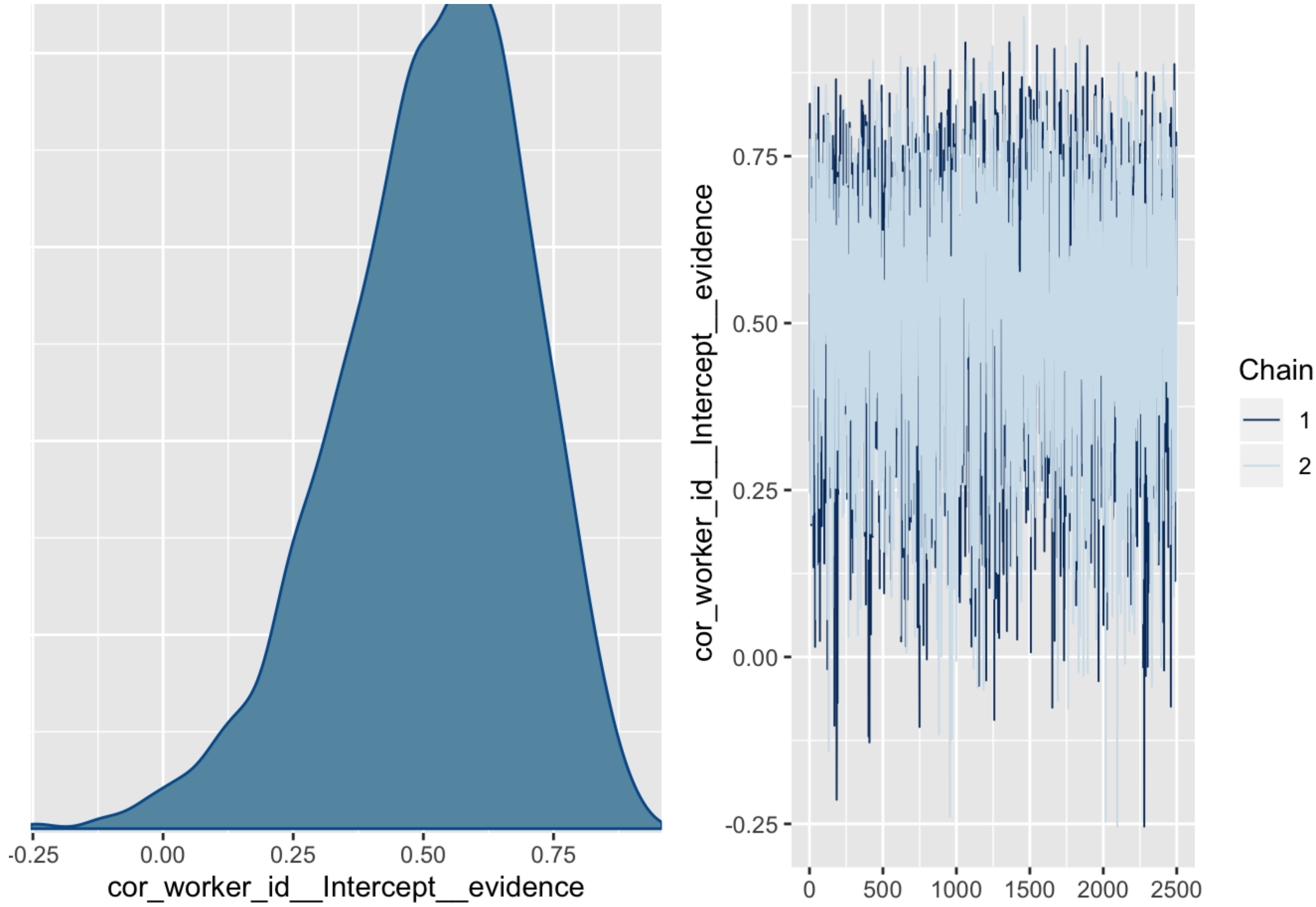
```
# hierarchical linear model with logit link and predictors for means present/absent, vis
ualization condition, and their interaction
m.wrkr.means.vis.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                                 formula = bf(intervene ~ (1 + evidence|worker_id) + evi
dence*means*condition),
                                 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)),
                                 iter = 3000, warmup = 500, chains = 2, cores = 2,
                                 file = "model-fits/logistic_mdl-wrkr_means_vis")
```

Check diagnostics:

- Trace plots

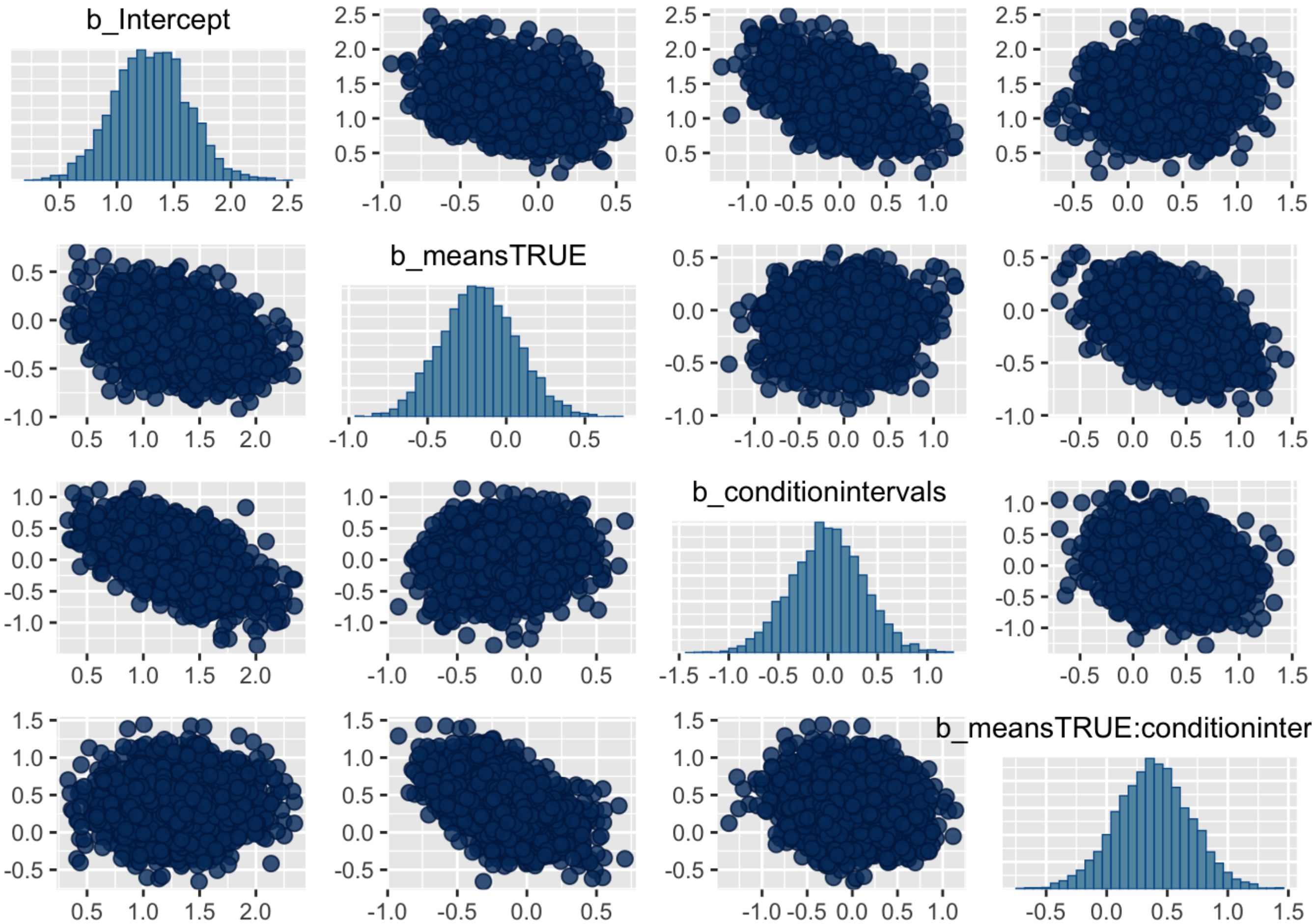
```
# trace plots
plot(m.wrkr.means.vis.logistic)
```


b_Intercept
0.5 1.0 1.5 2.0
b_Intercept
2.5 2.0 1.5 1.0 0.5
0 500 1000 1500 2000 2500
b_evidence
1.0 1.5 2.0
b_evidence
2.0 1.5 1.0 0.5
0 500 1000 1500 2000 2500
b_meansTRUE
-0.5 0.0 0.5
b_meansTRUE
0.5 0.0 -0.5 -1.0
0 500 1000 1500 2000 2500
Chain
1
2
b_conditionintervals
-1.0 -0.5 0.0 0.5 1.0
b_conditionintervals
1.0 0.5 0.0 -0.5 -1.0
0 500 1000 1500 2000 2500
b_evidence:meansTRUE
-0.4 0.0 0.4
b_evidence:meansTRUE
0.8 0.4 0.0 -0.4
0 500 1000 1500 2000 2500
b_evidence:conditionintervals
-0.5 0.0 0.5 1.0
b_evidence:conditionintervals
1.0 0.5 0.0 -0.5 -1.0
0 500 1000 1500 2000 2500
b_meansTRUE:conditionintervals
-0.5 0.0 0.5 1.0
b_meansTRUE:conditionintervals
1.5 1.0 0.5 0.0 -0.5
0 500 1000 1500 2000 2500
b_evidence:meansTRUE:conditionintervals
-0.5 0.0 0.5 1.0
b_evidence:meansTRUE:conditionintervals
1.0 0.5 0.0 -0.5
0 500 1000 1500 2000 2500
Chain
1
2
sd_worker_id__Intercept
0.8 1.2 1.6 2.0
sd_worker_id__Intercept
1.8 1.4 1.0
0 500 1000 1500 2000 2500
sd_worker_id__evidence
0.25 0.50 0.75 1.00
sd_worker_id__evidence
1.00 0.75 0.50 0.25 0.00
0 500 1000 1500 2000 2500
1.00

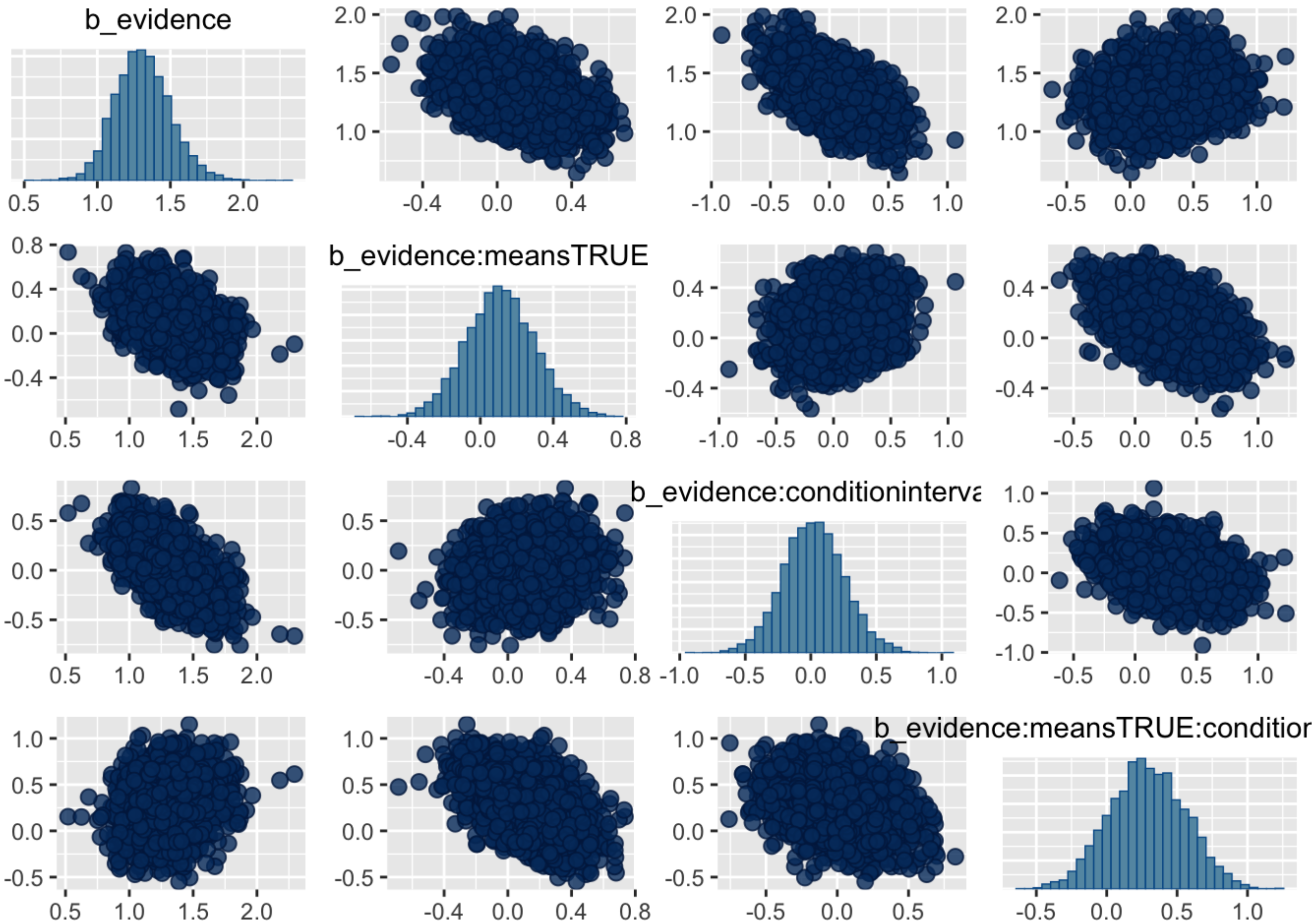


- Pairs plot

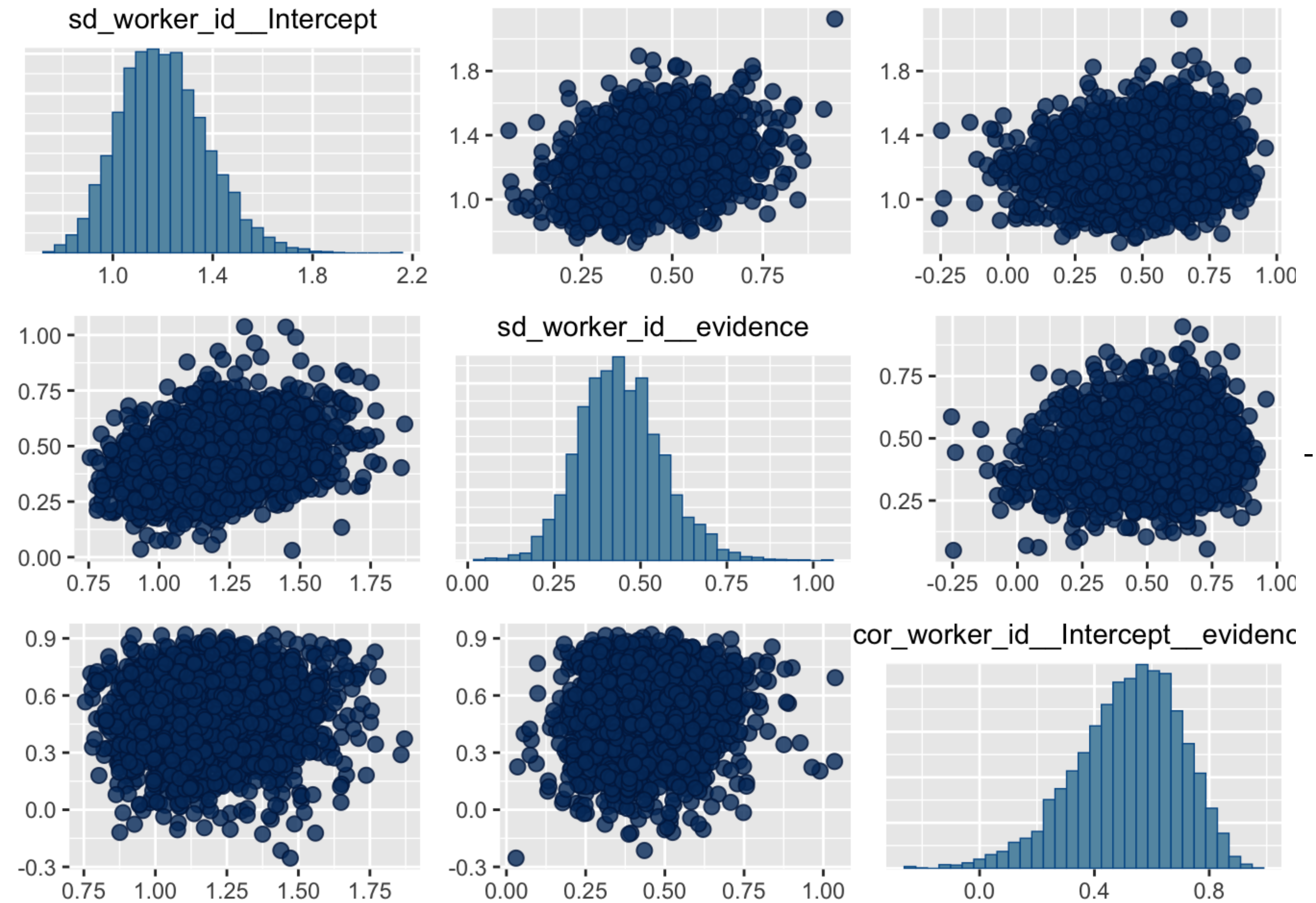
```
# pairs plot (fixed intercepts)
pairs(m.wrkr.means.vis.logistic, exact_match = TRUE, pars = c("b_Intercept", "b_meansTRU
E", "b_conditionintervals", "b_meansTRUE:conditionintervals"))
```




```
# pairs plot (fixed slopes)
pairs(m.wrkr.means.vis.logistic, exact_match = TRUE, pars = c("b_evidence", "b_evidence:
meansTRUE", "b_evidence:conditionintervals", "b_evidence:meansTRUE:conditionintervals"))
```




```
# pairs plot (random effects)
pairs(m.wrkr.means.vis.logistic, exact_match = TRUE, pars = c("sd_worker_id__Intercept",
"sd_worker_id__evidence", "cor_worker_id__Intercept__evidence"))
```



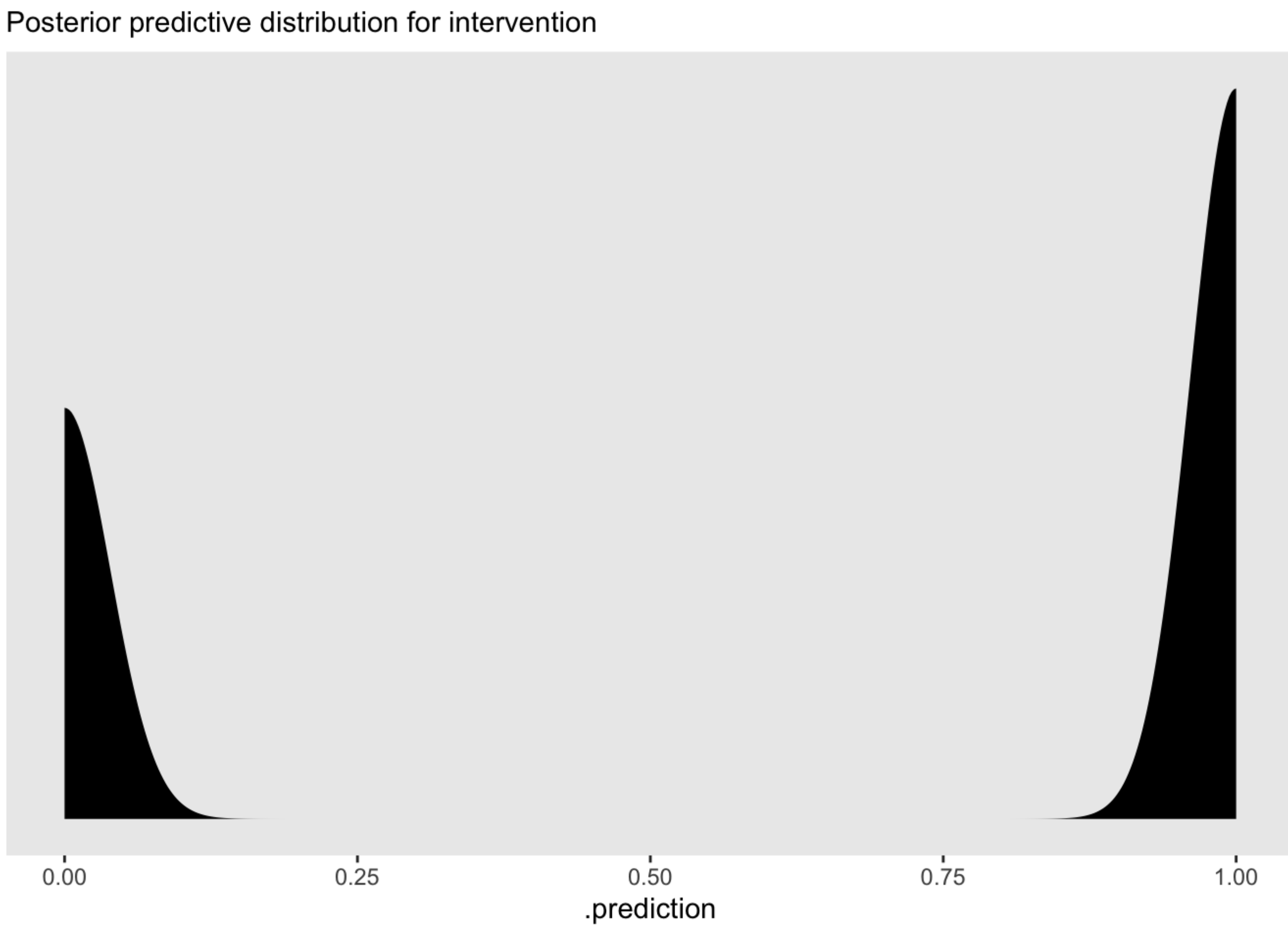
Summary

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

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + evidence * means * condition
##    Data: model_df (Number of observations: 840)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 28)
##                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)               1.20      0.18     0.89     1.58       2094
## sd(evidence)                0.44      0.12     0.22     0.71       1857
## cor(Intercept,evidence)     0.52      0.18     0.11     0.82       2951
##                         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                                1.30      0.32     0.67     1.93
## evidence                                 1.31      0.19     0.96     1.71
## meansTRUE                               -0.17      0.24    -0.62     0.31
## conditionintervals                      -0.01      0.36    -0.72     0.69
## evidence:meansTRUE                       0.11      0.19    -0.26     0.49
## evidence:conditionintervals              0.02      0.23    -0.42     0.50
## meansTRUE:conditionintervals             0.38      0.31    -0.24     0.99
## evidence:meansTRUE:conditionintervals    0.29      0.27    -0.23     0.81
##                                      Eff.Sample Rhat
## Intercept                                  1821 1.00
## evidence                                   1914 1.00
## meansTRUE                                  4514 1.00
## conditionintervals                         2401 1.00
## evidence:meansTRUE                         3332 1.00
## evidence:conditionintervals                2372 1.00
## meansTRUE:conditionintervals               4440 1.00
## evidence:meansTRUE:conditionintervals      3849 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).
```

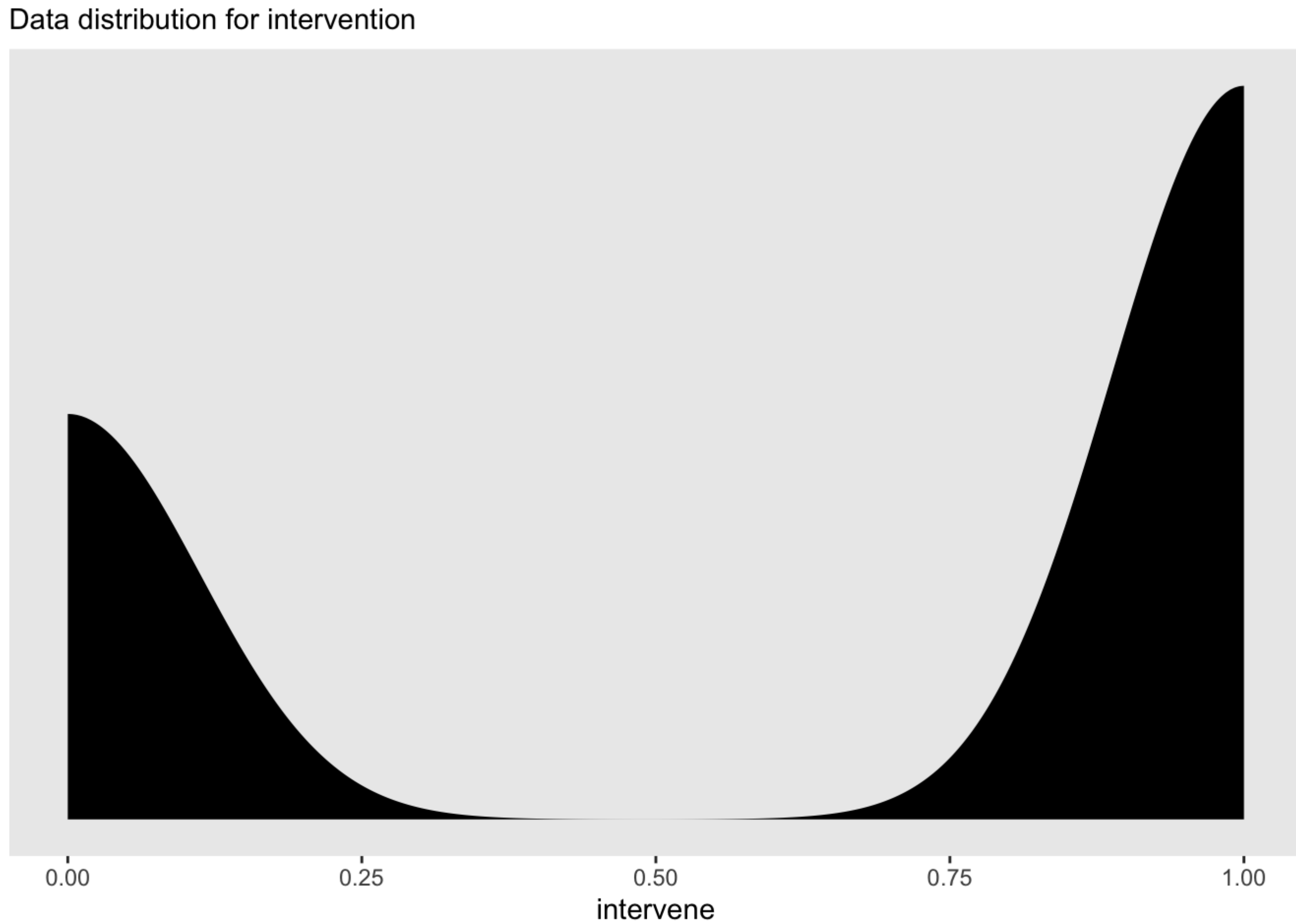
Let’s check out a posterior predictive distribution for intervention decisions.

```
# posterior predictive check
model_df %>%
  select(evidence, worker_id, means, condition) %>%
  add_predicted_draws(m.wrkr.means.vis.logistic, seed = 1234, n = 200) %>%
  ggplot(aes(x = .prediction)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior predictive distribution for intervention") +
  theme(panel.grid = element_blank())
```



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())
```

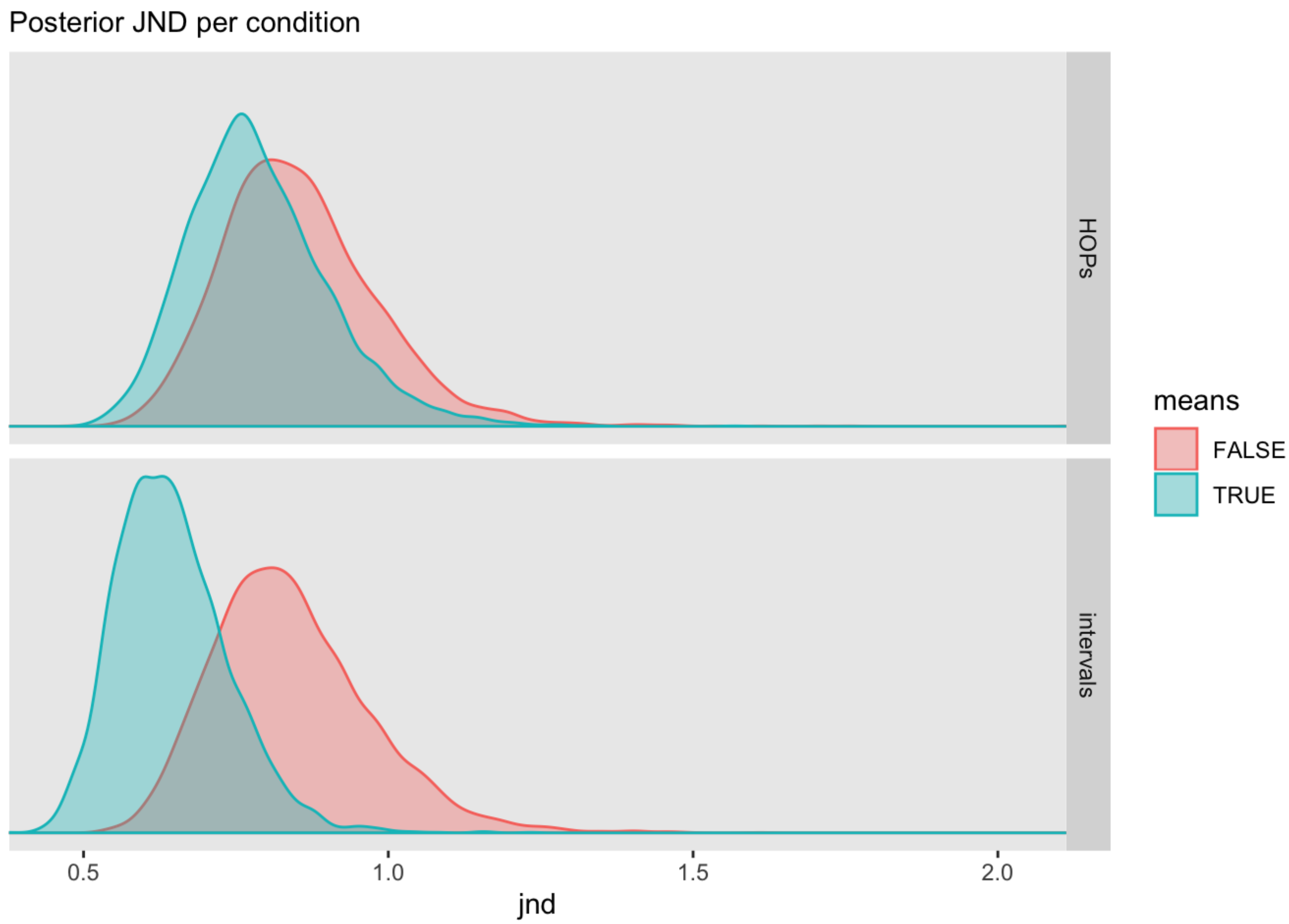


What do the posteriors for just-noticable differences (JND) and points of subjective equality (PSE) look like? We'll start by getting the slopes and intercepts of the linear model and using these to derive estimates of the JND and PSE for each fitted draw.

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(means, condition) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.means.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, condition) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.means.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", "condition", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )
```

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

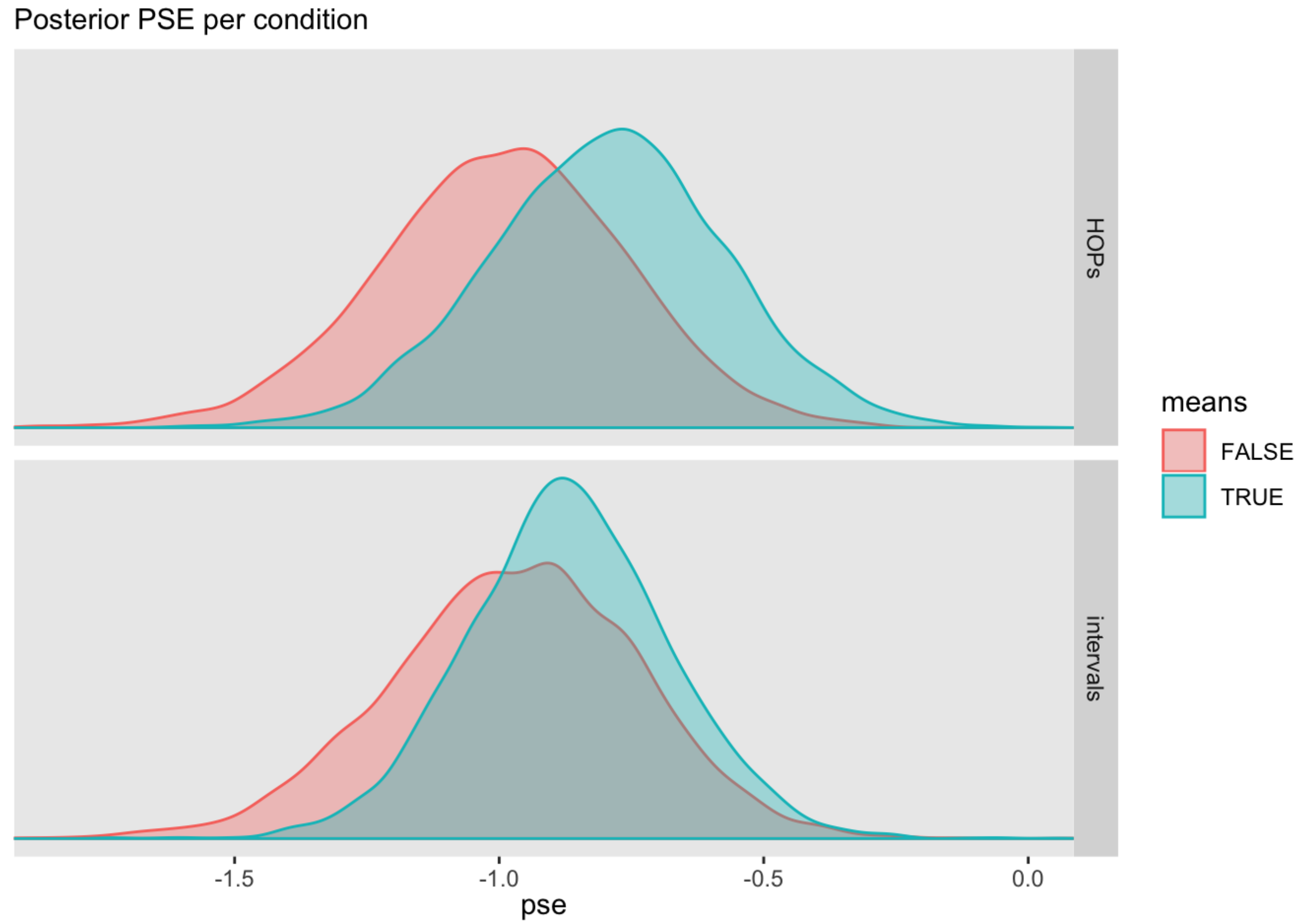
```
stats_df %>%
  ggplot(aes(x = jnd, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  theme(panel.grid = element_blank())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



It looks like people are more sensitivite to the evidence presented (i.e., smaller JNDs) when means are present, especially in the intervals condition.

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

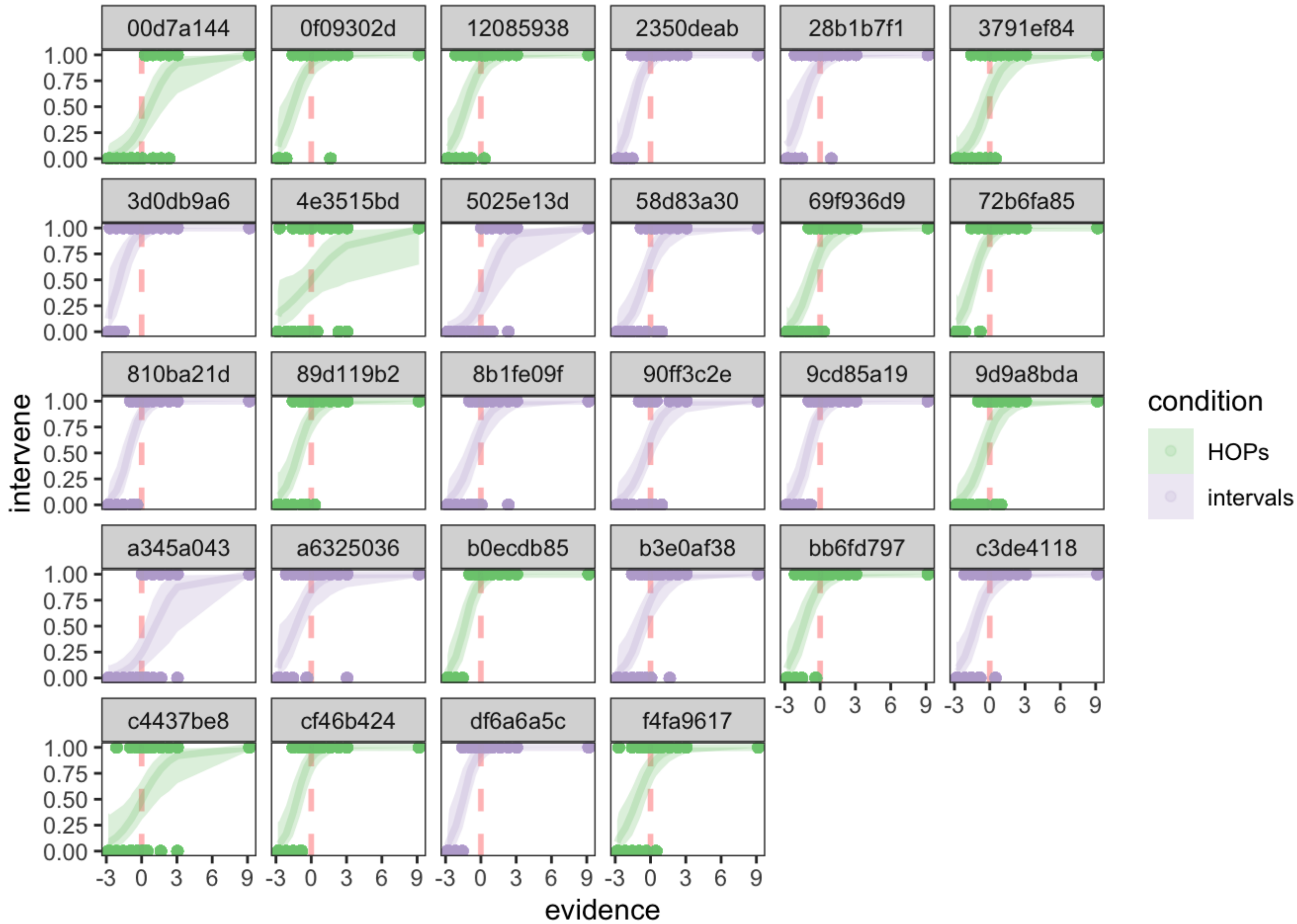
```
stats_df %>%
  ggplot(aes(x = pse, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per condition") +
  theme(panel.grid = element_blank())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



It looks like the point of subjective equality is slightly less biased when means are present, especially in the HOPs condition.

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

```
model_df %>%
  group_by(evidence, worker_id, means, condition) %>%
  add_fitted_draws(m.wrkr.means.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)
```



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.

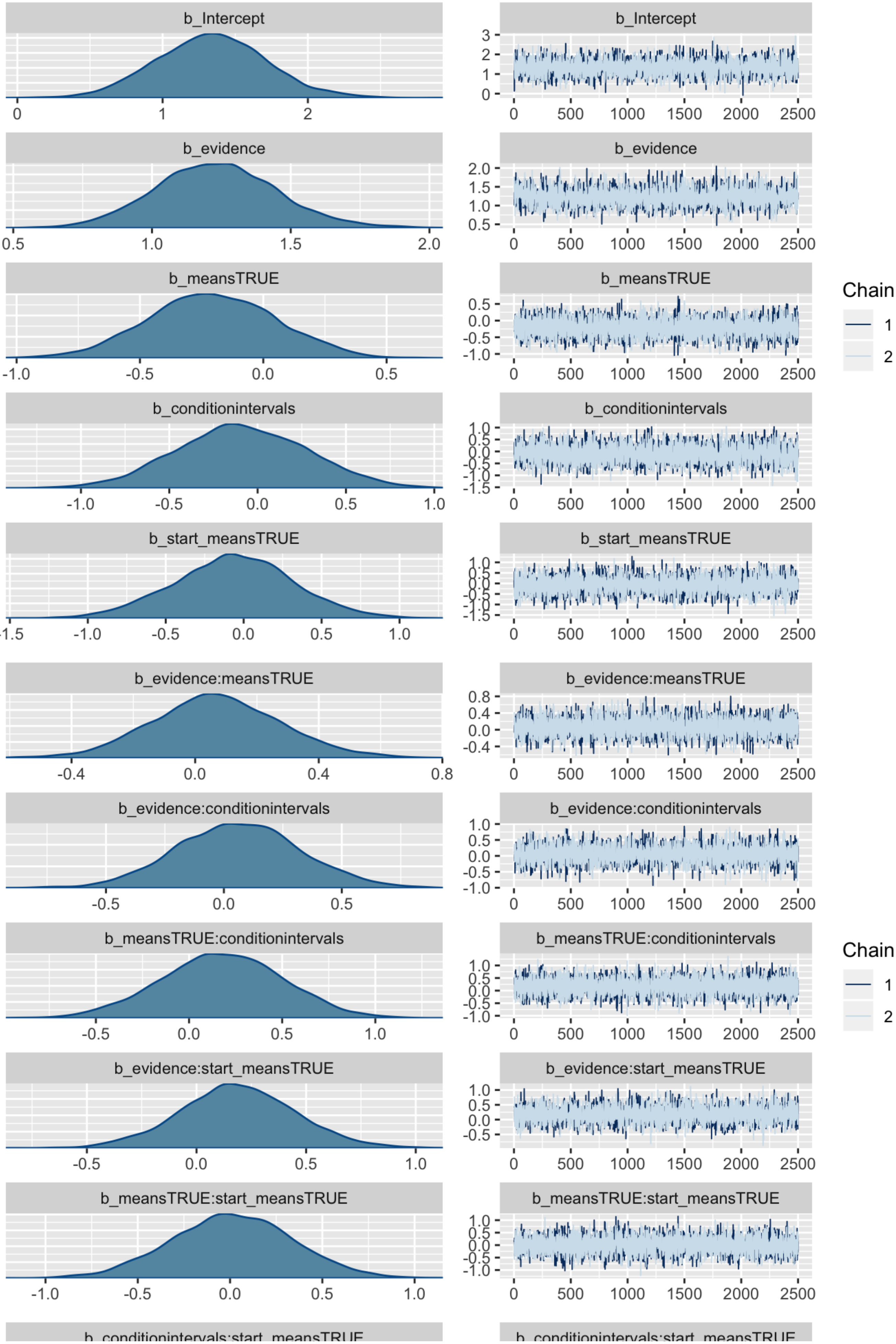
Again, we'll use the same priors as we did for our previous models. Now, let's fit our model.

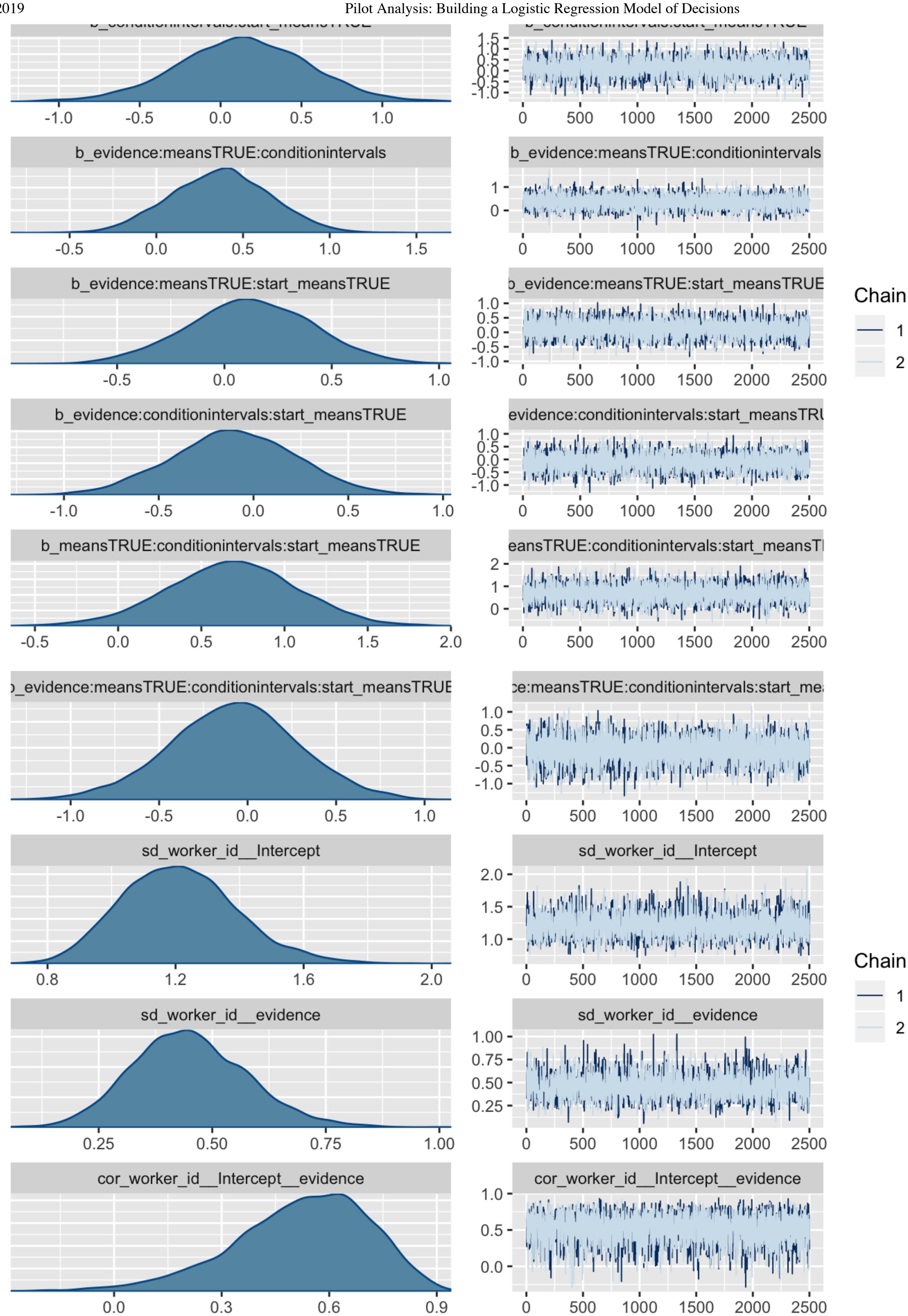
```
# hierarchical linear model with logit link and predictors for means present/absent, vis
ualization condition, and their interaction
m.wrkr.means.vis.order.logistic <- brm(data = model_df, family = bernoulli(link = "logi
t"),
                                       formula = bf(intervene ~ (1 + evidence|worker_id)
+ evidence*means*condition*start_means),
                                       prior = c(prior(normal(0, 1), class = Intercept),
                                                 prior(normal(1, 1), class = b, coef = e
vidence),
                                                 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_vis_or
der")
```

Check diagnostics:

- Trace plots

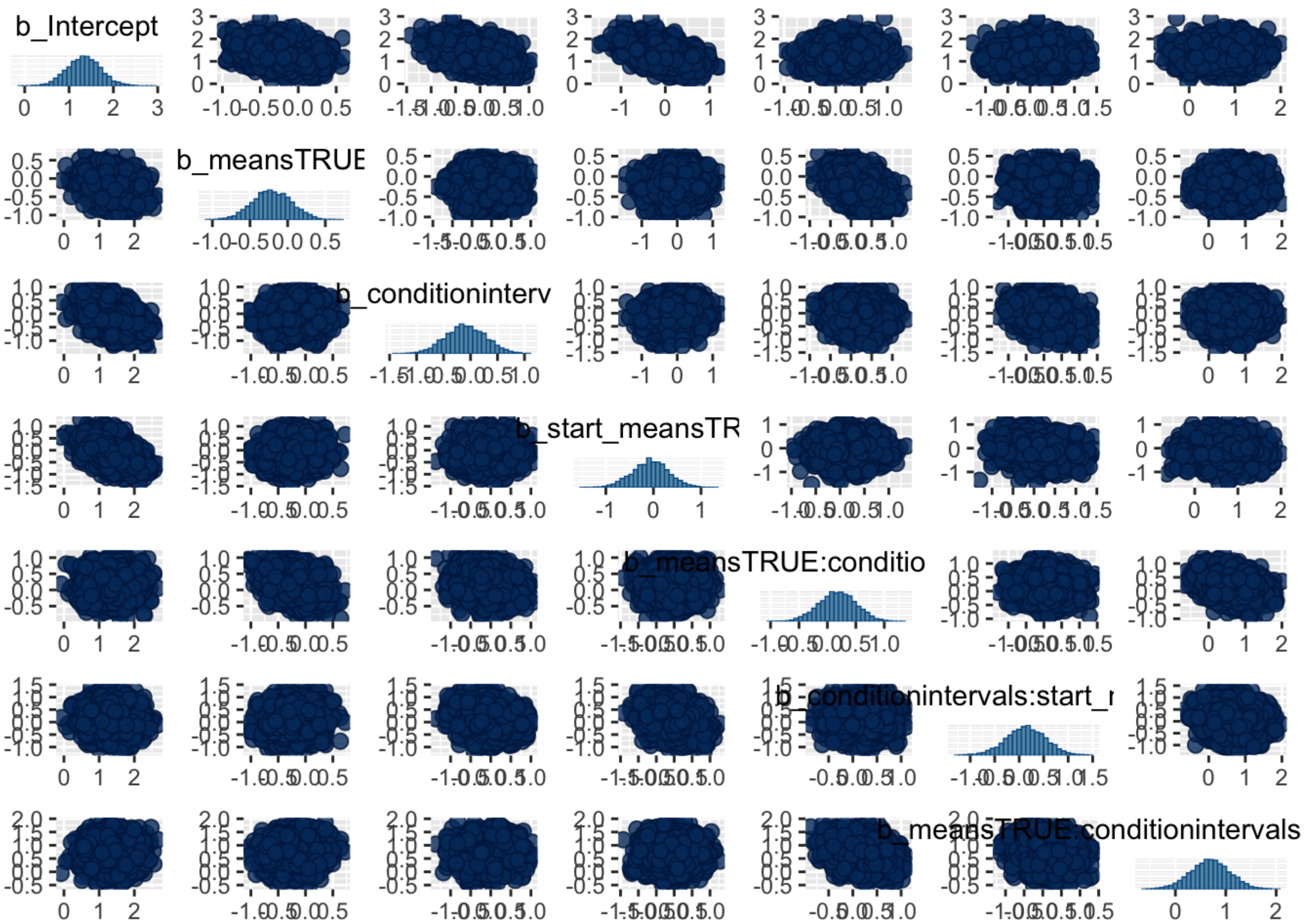
```
# trace plots
plot(m.wrkr.means.vis.order.logistic)
```


b_Intercept
b_Intercept
b_evidence
b_evidence
b_meansTRUE
b_meansTRUE
Chain
1
2
b_conditionintervals
b_conditionintervals
b_start_meansTRUE
b_start_meansTRUE
b_evidence:meansTRUE
b_evidence:meansTRUE
b_evidence:conditionintervals
b_evidence:conditionintervals
b_meansTRUE:conditionintervals
b_meansTRUE:conditionintervals
Chain
1
2
b_evidence:start_meansTRUE
b_evidence:start_meansTRUE
b_meansTRUE:start_meansTRUE
b_meansTRUE:start_meansTRUE

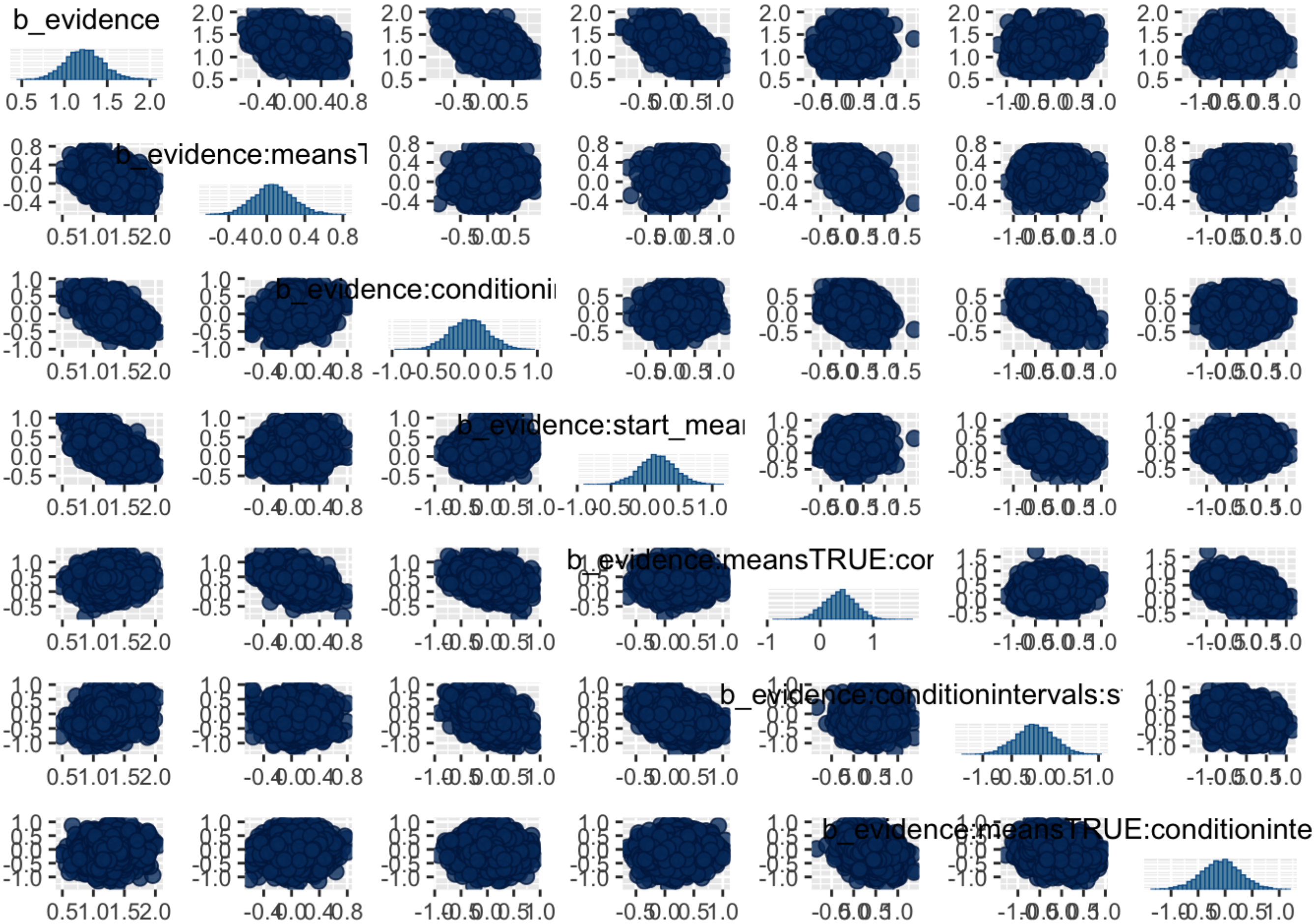

b_evidence:meansTRUE:conditionintervals
b_evidence:meansTRUE:conditionintervals
b_evidence:meansTRUE:start_meansTRUE
b_evidence:meansTRUE:start_meansTRUE
b_evidence:conditionintervals:start_meansTRUE
evidence:conditionintervals:start_meansTRU
b_meansTRUE:conditionintervals:start_meansTRUE
eansTRUE:conditionintervals:start_meansT
sd_worker_id__Intercept
sd_worker_id__Intercept
sd_worker_id__evidence
sd_worker_id__evidence
cor_worker_id__Intercept__evidence
cor_worker_id__Intercept__evidence
Chain
1
2
Chain
1
2

- Pairs plot

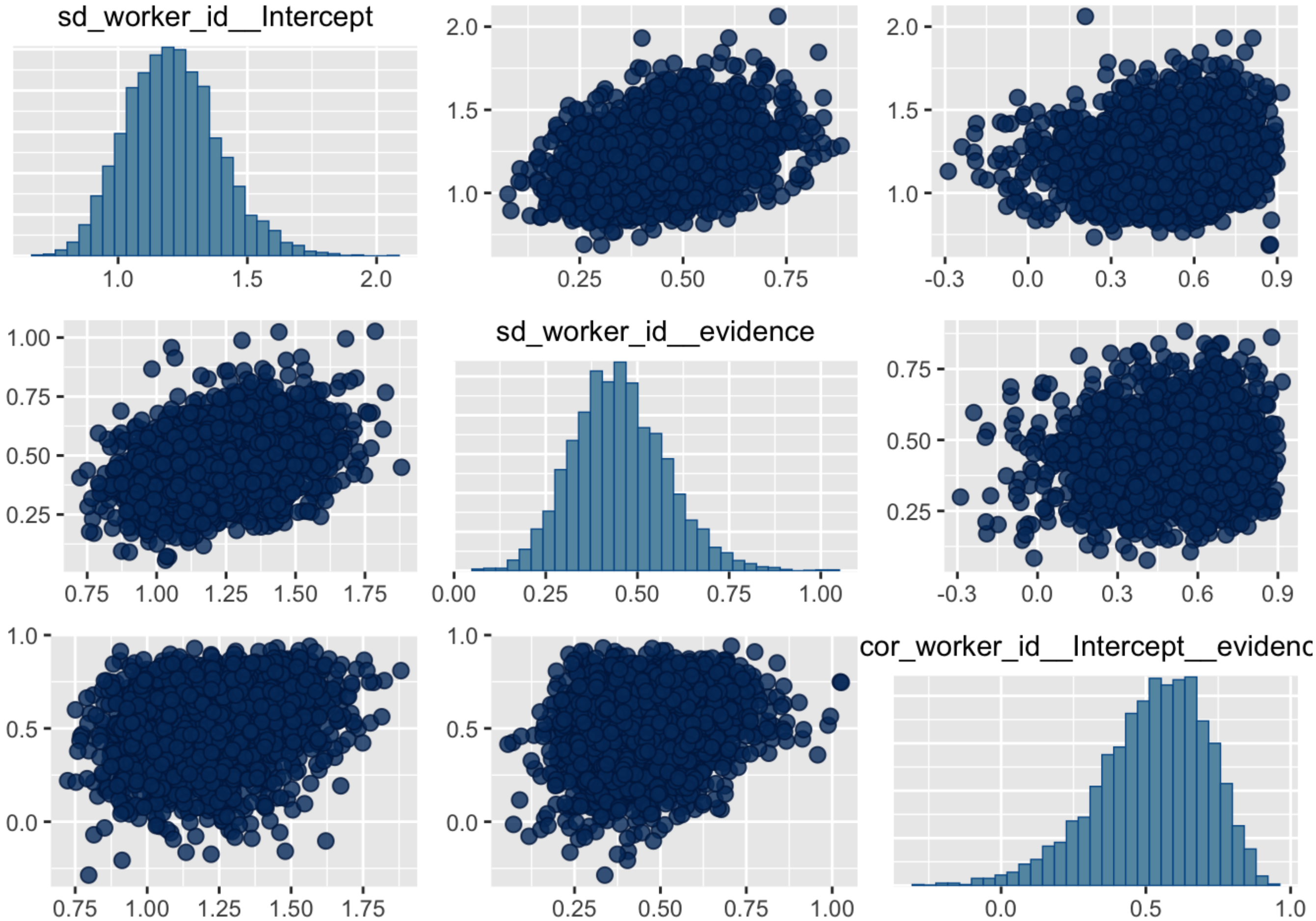
```
# pairs plot (fixed intercepts)
pairs(m.wrkr.means.vis.order.logistic, exact_match = TRUE, pars = c("b_Intercept", "b_meansTRUE", "b_conditionintervals", "b_start_meansTRUE", "b_meansTRUE:conditionintervals", "b_meansTRUE:b_start_meansTRUE", "b_conditionintervals:start_meansTRUE", "b_meansTRUE:conditionintervals:start_meansTRUE"))
```




```
# pairs plot (fixed slopes)
pairs(m.wrkr.means.vis.order.logistic, exact_match = TRUE, pars = c("b_evidence", "b_evidence:meansTRUE", "b_evidence:conditionintervals", "b_evidence:start_meansTRUE", "b_evidence:meansTRUE:conditionintervals", "b_evidence:meansTRUE:b_start_meansTRUE", "b_evidence:conditionintervals:start_meansTRUE", "b_evidence:meansTRUE:conditionintervals:start_meansTRUE"))
```




```
# pairs plot (random effects)
pairs(m.wrkr.means.vis.order.logistic, exact_match = TRUE, pars = c("sd_worker_id__Inter
cept", "sd_worker_id__evidence", "cor_worker_id__Intercept__evidence"))
```



- Summary

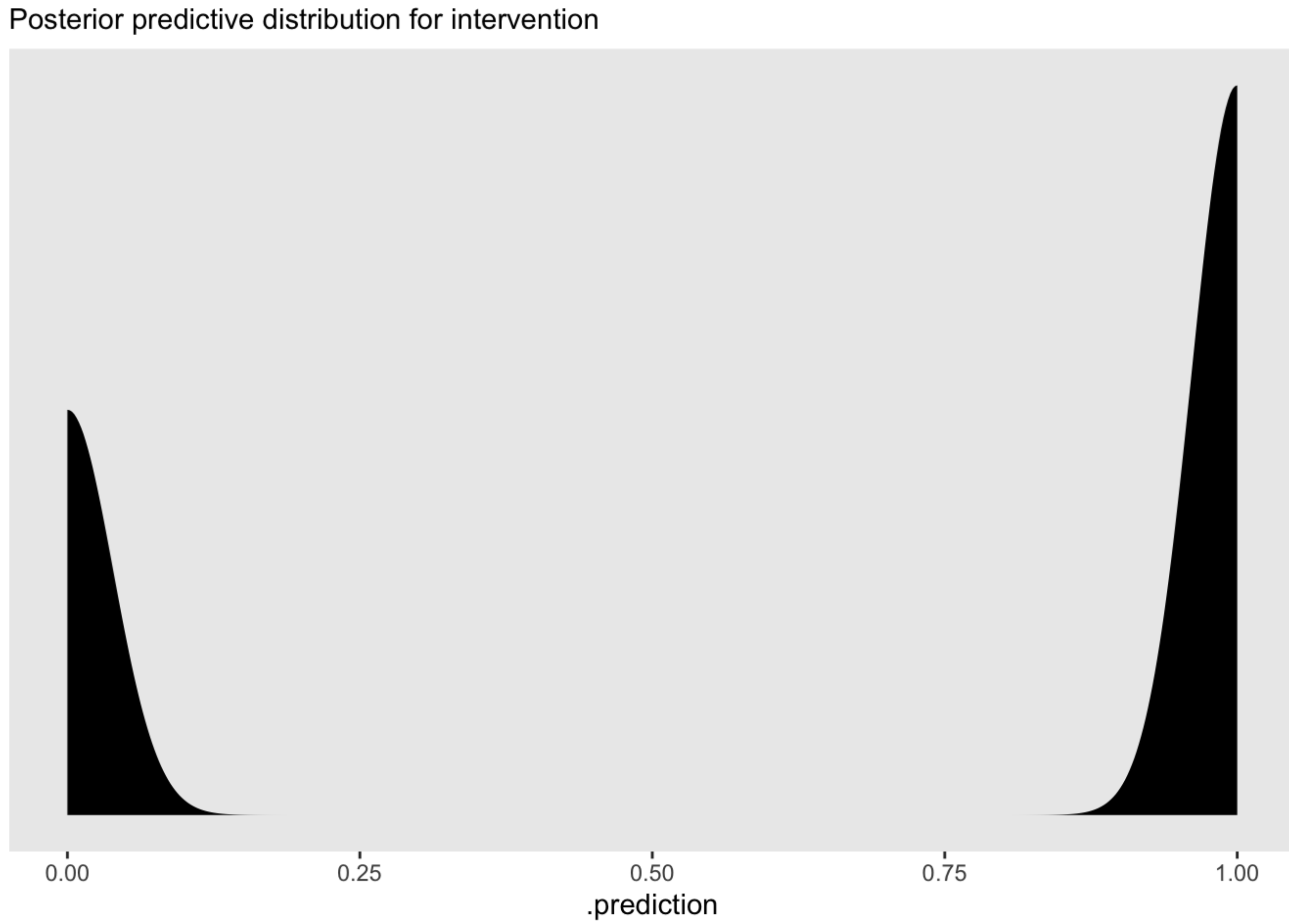
```
# model summary
print(m.wrkr.means.vis.order.logistic)
```

```
##  Family: bernoulli 
##   Links: mu = logit 
## Formula: intervene ~ (1 + evidence | worker_id) + evidence * means * condition * star
t_means 
##    Data: model_df (Number of observations: 840) 
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
## 
## Group-Level Effects: 
## ~worker_id (Number of levels: 28) 
##                           Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)                 1.21      0.18     0.89     1.60       3230
## sd(evidence)                  0.45      0.13     0.22     0.72       2494
## cor(Intercept,evidence)       0.52      0.19     0.09     0.83       4960
##                           Rhat
## sd(Intercept)             1.00
## sd(evidence)              1.00
## cor(Intercept,evidence) 1.00
## 
## Population-Level Effects: 
##                                                       Estimate Est.Error
## Intercept                                                 1.34      0.38
## evidence                                                  1.23      0.22
## meansTRUE                                                -0.21      0.26
## conditionintervals                                       -0.09      0.37
## start_meansTRUE                                          -0.06      0.38
## evidence:meansTRUE                                        0.07      0.21
## evidence:conditionintervals                               0.06      0.26
## meansTRUE:conditionintervals                              0.18      0.32
## evidence:start_meansTRUE                                  0.19      0.26
## meansTRUE:start_meansTRUE                                 0.01      0.33
## conditionintervals:start_meansTRUE                        0.14      0.40
## evidence:meansTRUE:conditionintervals                     0.37      0.29
## evidence:meansTRUE:start_meansTRUE                        0.12      0.29
## evidence:conditionintervals:start_meansTRUE              -0.10      0.33
## meansTRUE:conditionintervals:start_meansTRUE              0.69      0.39
## evidence:meansTRUE:conditionintervals:start_meansTRUE    -0.08      0.36
##                                                       l-95% CI u-95% CI
## Intercept                                                 0.61     2.10
## evidence                                                  0.82     1.69
## meansTRUE                                                -0.70     0.29
## conditionintervals                                       -0.81     0.63
## start_meansTRUE                                          -0.83     0.68
## evidence:meansTRUE                                       -0.33     0.49
## evidence:conditionintervals                              -0.44     0.58
## meansTRUE:conditionintervals                             -0.46     0.80
## evidence:start_meansTRUE                                 -0.33     0.72
## meansTRUE:start_meansTRUE                                -0.62     0.65
## conditionintervals:start_meansTRUE                       -0.67     0.93
## evidence:meansTRUE:conditionintervals                    -0.19     0.92
## evidence:meansTRUE:start_meansTRUE                       -0.45     0.69
## evidence:conditionintervals:start_meansTRUE              -0.74     0.54
## meansTRUE:conditionintervals:start_meansTRUE             -0.10     1.44
```

```
## evidence:meansTRUE:conditionintervals:start_meansTRUE    -0.81     0.63
##                                                       Eff.Sample Rhat
## Intercept                                                   3617 1.00
## evidence                                                    3895 1.00
## meansTRUE                                                   8038 1.00
## conditionintervals                                          4963 1.00
## start_meansTRUE                                             5084 1.00
## evidence:meansTRUE                                          7797 1.00
## evidence:conditionintervals                                 4856 1.00
## meansTRUE:conditionintervals                                7603 1.00
## evidence:start_meansTRUE                                    4934 1.00
## meansTRUE:start_meansTRUE                                   7319 1.00
## conditionintervals:start_meansTRUE                          6095 1.00
## evidence:meansTRUE:conditionintervals                       7728 1.00
## evidence:meansTRUE:start_meansTRUE                          7914 1.00
## evidence:conditionintervals:start_meansTRUE                 5915 1.00
## meansTRUE:conditionintervals:start_meansTRUE                8355 1.00
## evidence:meansTRUE:conditionintervals:start_meansTRUE       8095 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).
```

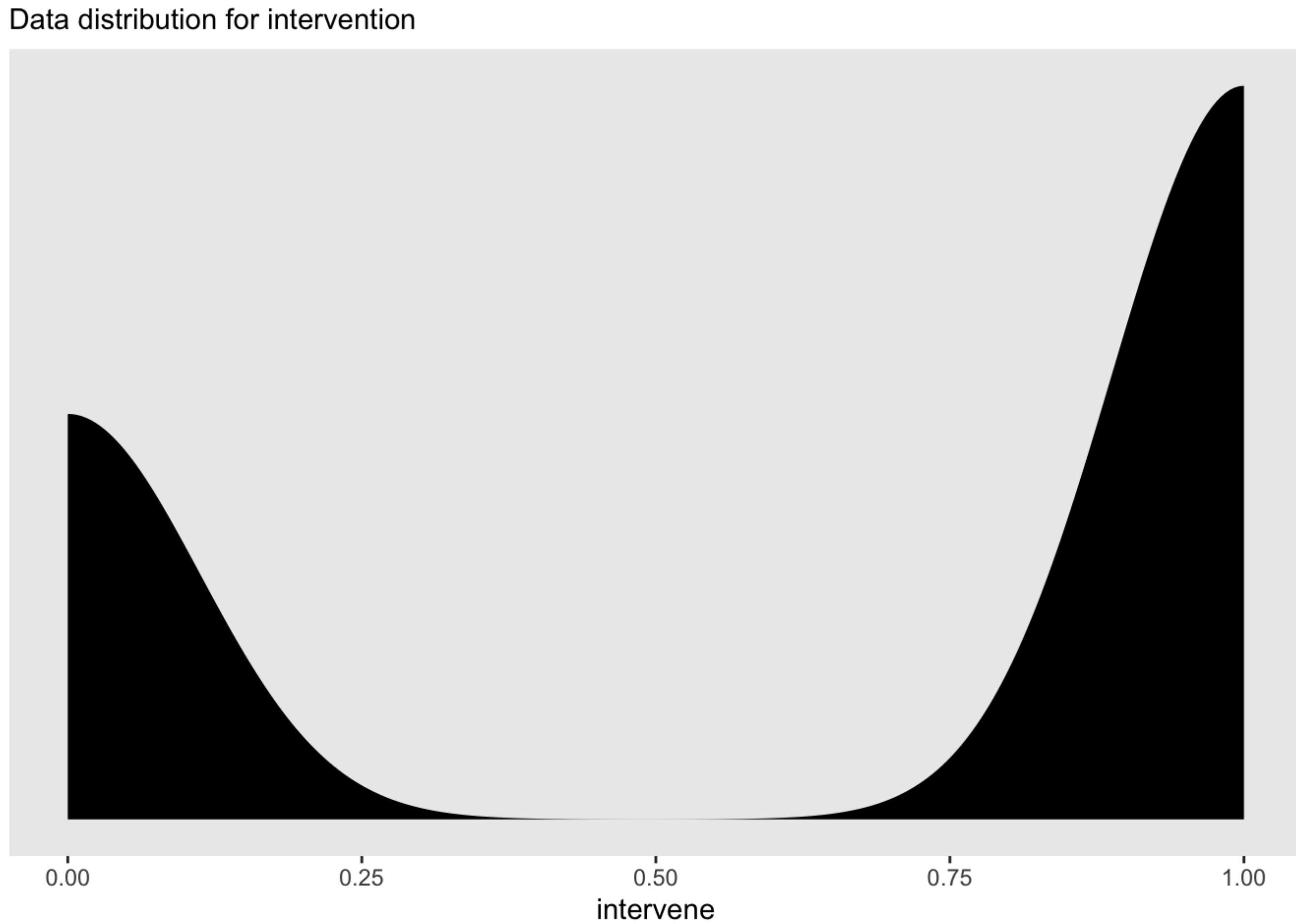
Let’s check out a posterior predictive distribution for intervention decisions.

```
# posterior predictive check
model_df %>%
  select(evidence, worker_id, means, condition, start_means) %>%
  add_predicted_draws(m.wrkr.means.vis.order.logistic, seed = 1234, n = 200) %>%
  ggplot(aes(x = .prediction)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior predictive distribution for intervention") +
  theme(panel.grid = element_blank())
```



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())
```

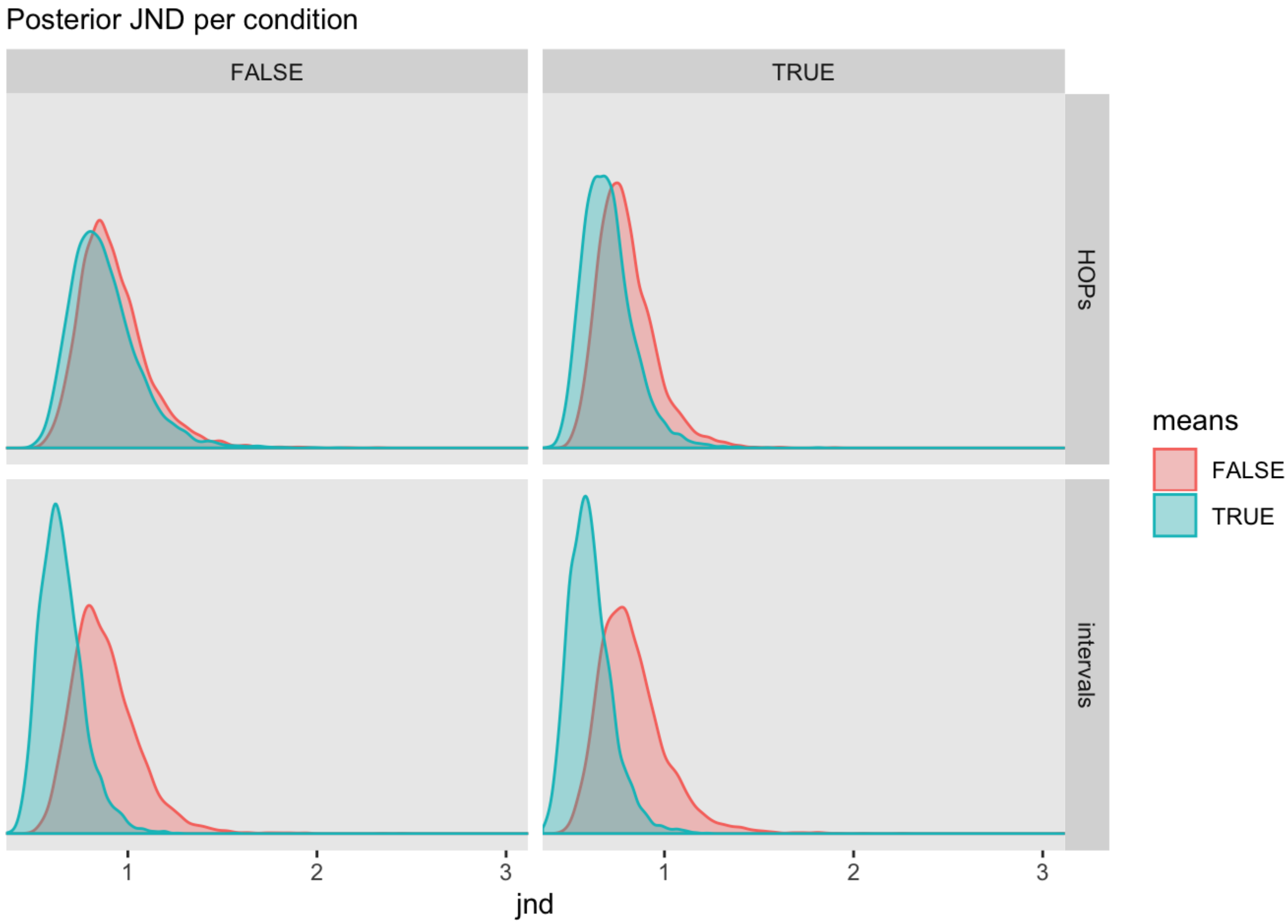


What do the posteriors for just-noticable differences (JND) and points of subjective equality (PSE) look like? We'll start by getting the slopes and intercepts of the linear model and using these to derive estimates of the JND and PSE for each fitted draw.

```
# get slopes from linear model
slopes_df <- model_df %>%
  group_by(means, condition, start_means) %>%
  data_grid(evidence = c(0, 1)) %>%
  add_fitted_draws(m.wrkr.means.vis.order.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, condition, start_means) %>%
  data_grid(evidence = 0) %>%
  add_fitted_draws(m.wrkr.means.vis.order.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", "condition", "start_means", ".draw")) %>%
  mutate(
    pse = -intercept / slope,
    jnd = qlogis(0.75) / slope
  )
```

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

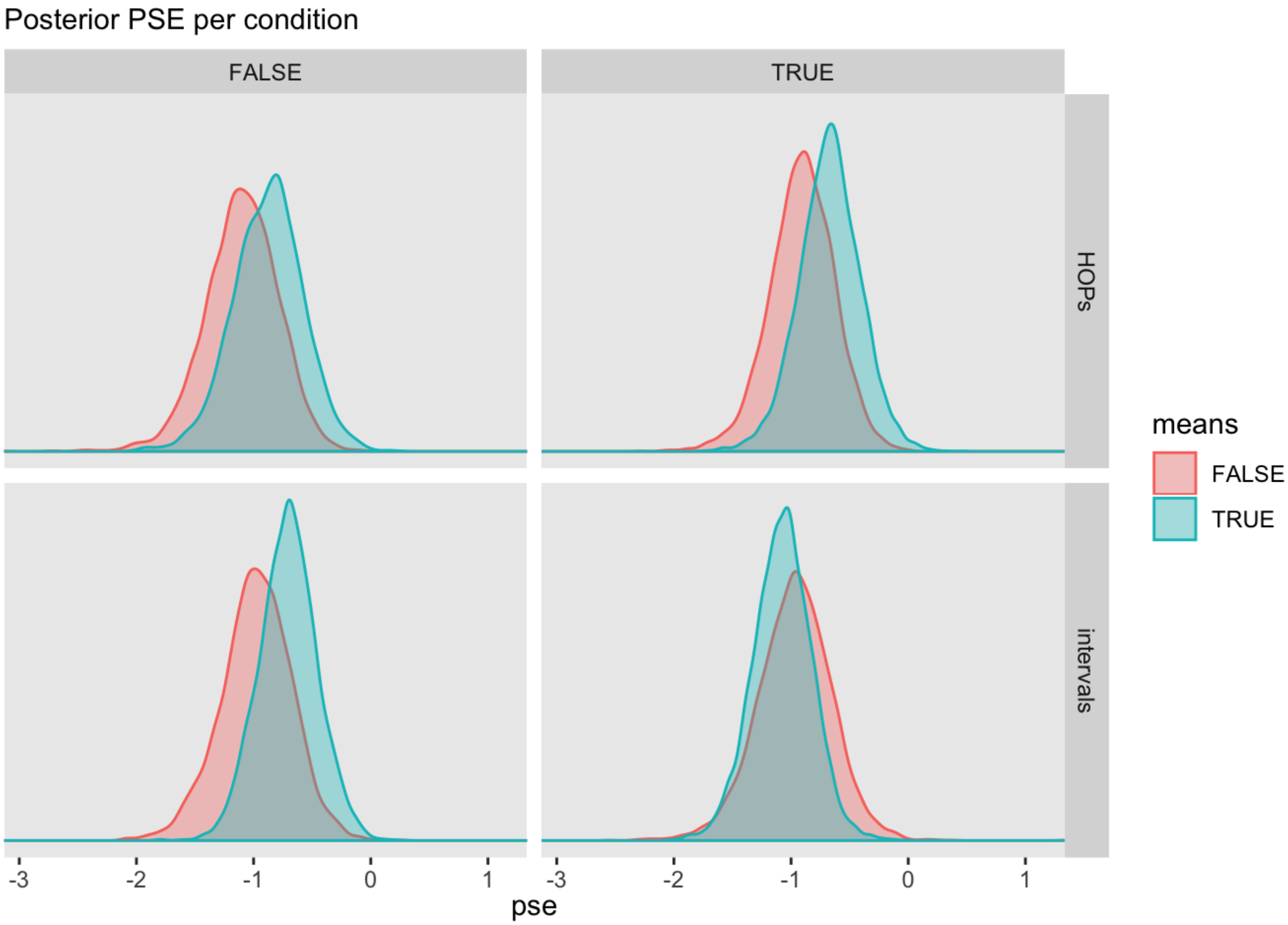
```
stats_df %>%
  ggplot(aes(x = jnd, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(jnd), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior JND per condition") +
  theme(panel.grid = element_blank())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



It looks like the effect of extrinsic means is stronger in the intervals condition and when participants are assigned to use means in the first block of trials.

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

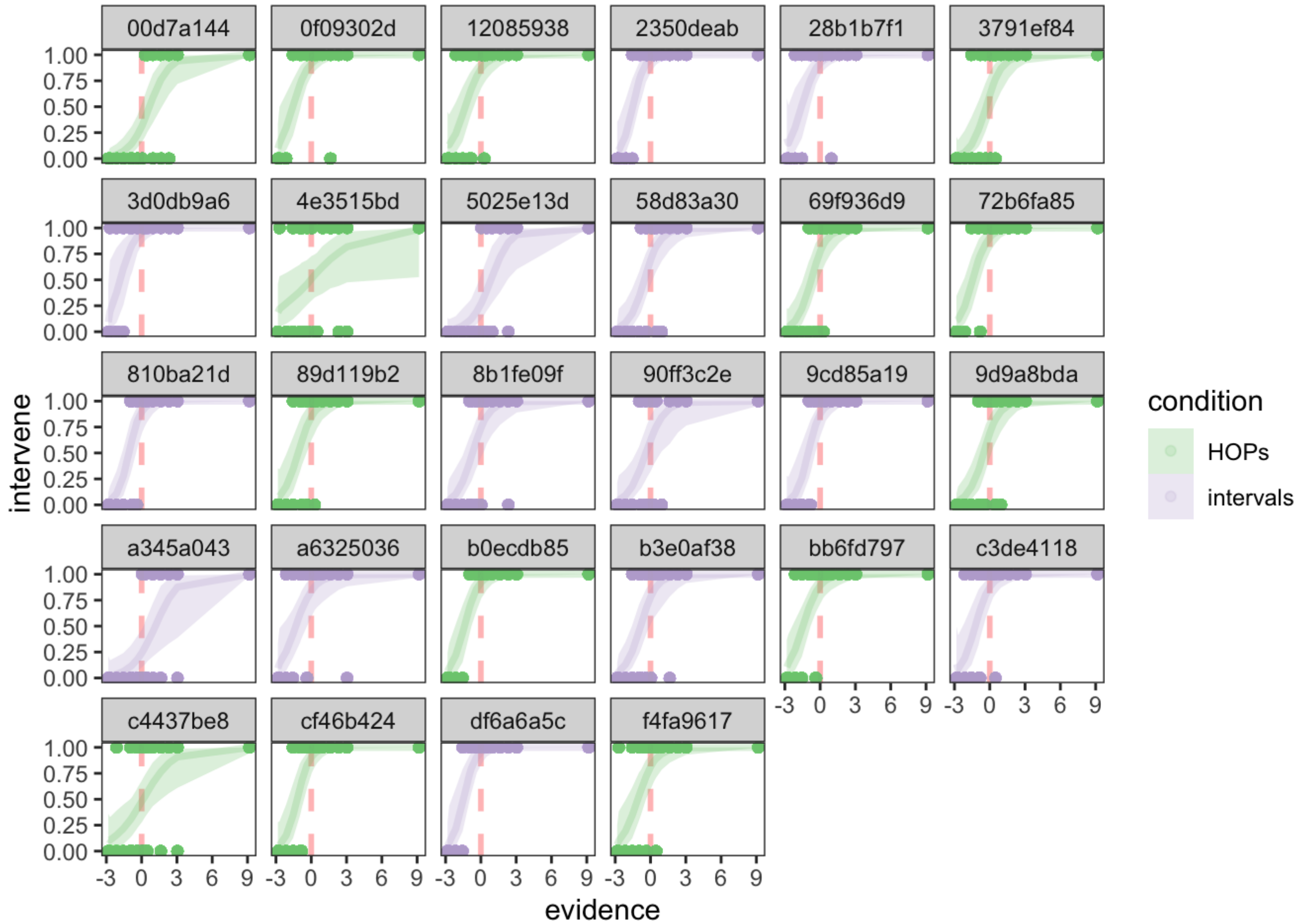
```
stats_df %>%
  ggplot(aes(x = pse, group = means, color = means, fill = means)) +
  geom_density(alpha = 0.35) +
  scale_x_continuous(expression(pse), expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Posterior PSE per condition") +
  theme(panel.grid = element_blank())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



There is a subtle reversal of the effect of extrinsic means in the intervals condition depending on block order. Means lead to more bias in the point of subjective equality when users start with means and less bias when means are added in the second block.

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

```
model_df %>%
  group_by(evidence, worker_id, means, condition, start_means) %>%
  add_fitted_draws(m.wrkr.means.vis.order.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)
```



Model Comparison

Let’s check check which of these two hierarchical models fits best insofar as the parameters contribute more to predictive validity than they contribute to overfitting. We’ll determine this by comparing the models according to the widely applicable information criterion (WAIC). Lower values of WAIC indicate a better fitting model.

```
waic(m.wrkr.logistic, m.wrkr.means.logistic, m.wrkr.vis.logistic, m.wrkr.means.vis.logis
tic, m.wrkr.means.vis.order.logistic)
```

```
##                                                                  WAIC    SE
## m.wrkr.logistic                                                648.23 34.39
## m.wrkr.means.logistic                                          649.59 34.64
## m.wrkr.vis.logistic                                            649.63 34.61
## m.wrkr.means.vis.logistic                                      649.53 34.47
## m.wrkr.means.vis.order.logistic                                649.07 34.89
## m.wrkr.logistic - m.wrkr.means.logistic                         -1.36  3.11
## m.wrkr.logistic - m.wrkr.vis.logistic                           -1.40  0.82
## m.wrkr.logistic - m.wrkr.means.vis.logistic                     -1.30  3.94
## m.wrkr.logistic - m.wrkr.means.vis.order.logistic               -0.84  4.73
## m.wrkr.means.logistic - m.wrkr.vis.logistic                     -0.04  3.37
## m.wrkr.means.logistic - m.wrkr.means.vis.logistic                0.06  2.61
## m.wrkr.means.logistic - m.wrkr.means.vis.order.logistic          0.52  3.97
## m.wrkr.vis.logistic - m.wrkr.means.vis.logistic                  0.11  4.04
## m.wrkr.vis.logistic - m.wrkr.means.vis.order.logistic            0.57  4.84
## m.wrkr.means.vis.logistic - m.wrkr.means.vis.order.logistic      0.46  2.92
```

Interestingly, none of these models have superior predictive validity vs the simple hierarchical model.