

# 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*, at which users see team equal utility with or without the new player. Differences between this parameter and the utility optimal decision rule for the intervention problem reflect bias in decision-making, either toward or away from intervening. 2. The slope or the amount of *noise in the perception of utility*, which reflects the consistency with which users evaluate prospects.

## 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_gain_frame = 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
  ) %>%
  filter(trial_idx != "practice", trial_idx != "mock") %>% # remove practice and mock trials from responses dataframe, leave in full version
  mutate( # mutate to rows where intervene == -1 for some reason
    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
    )
  ) #%>%
  # mutate( # create ground truth metric for evidence in favor of decision
  #   # evidence = qlogis(p_award_with) - qlogis(p_award_without)
  #   # evidence = log((p_award_with - p_award_without) / (1 / award_value))
  # )
}

head(responses_df)

```

```

## # A tibble: 6 x 29
##   worker_id batch condition baseline award_value exchange start_gain_frame
##   <chr>     <int> <chr>       <dbl>      <dbl>      <dbl> <chr>
## 1 bf797261     4 means_on...     0.5       2.25      0.5 True
## 2 bf797261     4 means_on...     0.5       2.25      0.5 True
## 3 bf797261     4 means_on...     0.5       2.25      0.5 True
## 4 bf797261     4 means_on...     0.5       2.25      0.5 True
## 5 bf797261     4 means_on...     0.5       2.25      0.5 True
## 6 bf797261     4 means_on...     0.5       2.25      0.5 True
## # ... with 22 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 <dbl>, 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>

```

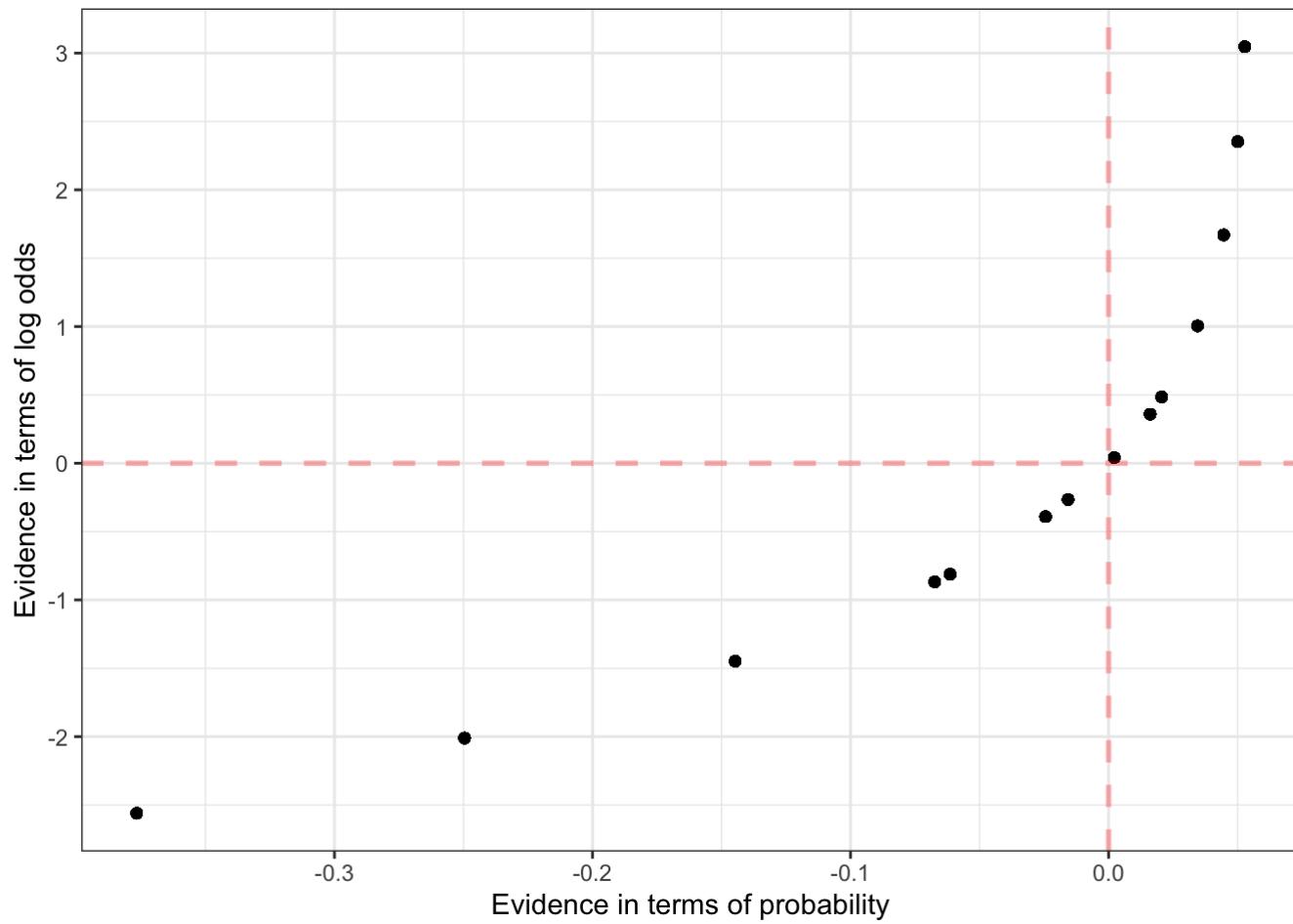
We need the data in a format where it is prepared for modeling. This means that we want problem framing as a factor.

```
# create data frame for model
model_df <- responses_df %>%
  mutate(
    frame = as.factor(if_else(ground_truth > 0.5, "gain", "loss"))
  )
```

We also want 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 <- model_df %>%
  mutate(
    p_diff = p_award_with - (p_award_without + (1 / award_value)),
    evidence = qlogis(p_award_with) - qlogis(p_award_without + (1 / award_value))
  )

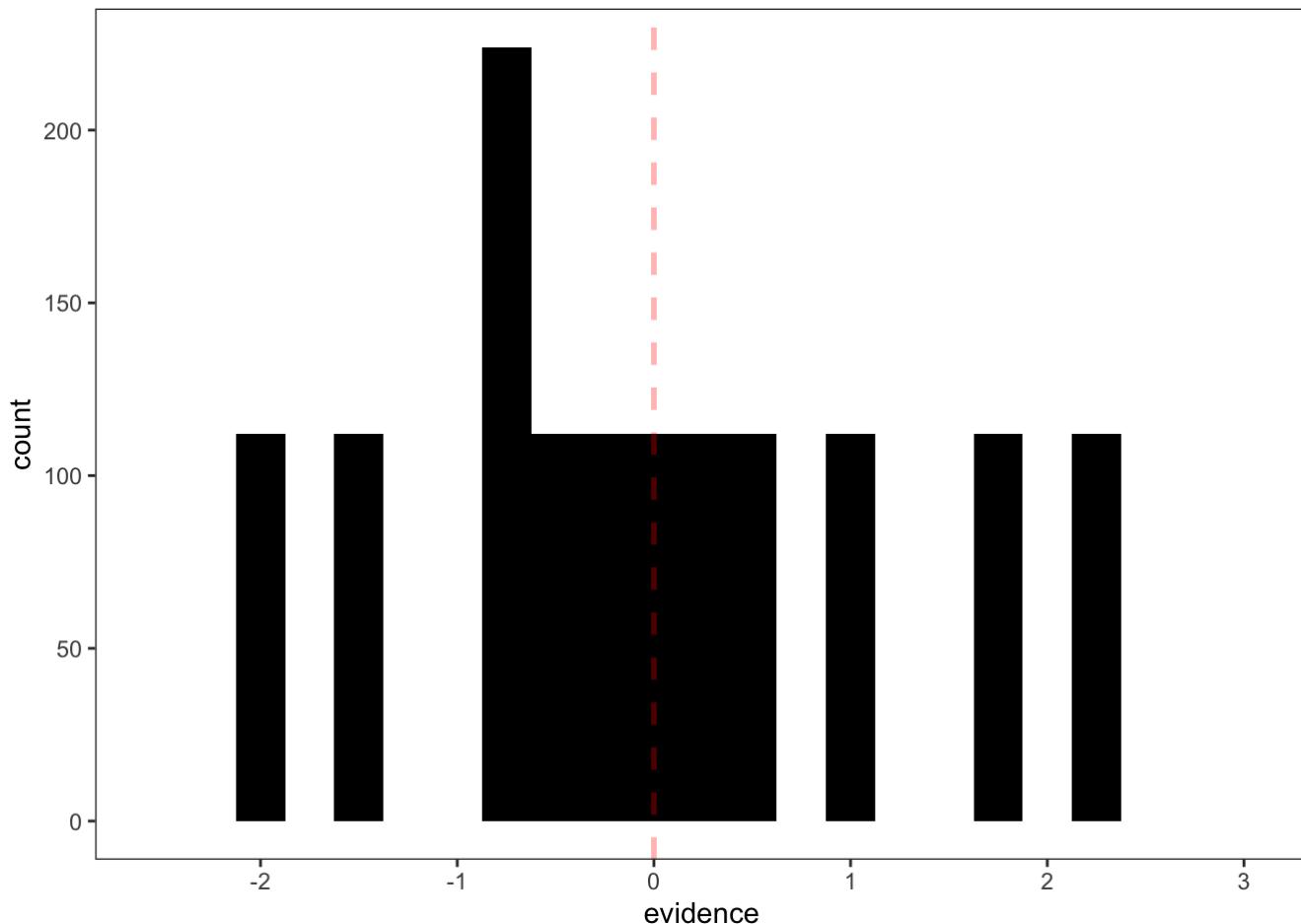
model_df %>%
  ggplot(aes(p_diff, evidence)) +
  geom_point() +
  geom_vline(xintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
# utility optimal decision rule
  geom_hline(yintercept = 0, size = 1, alpha = .3, color = "red", linetype = "dashed") +
  theme_bw() +
  labs(
    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.

```
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).
```



## Distribution of Decisions

We start as simply as possible by just modeling the distribution of decisions using a logit link function and a linear model with only an intercept.

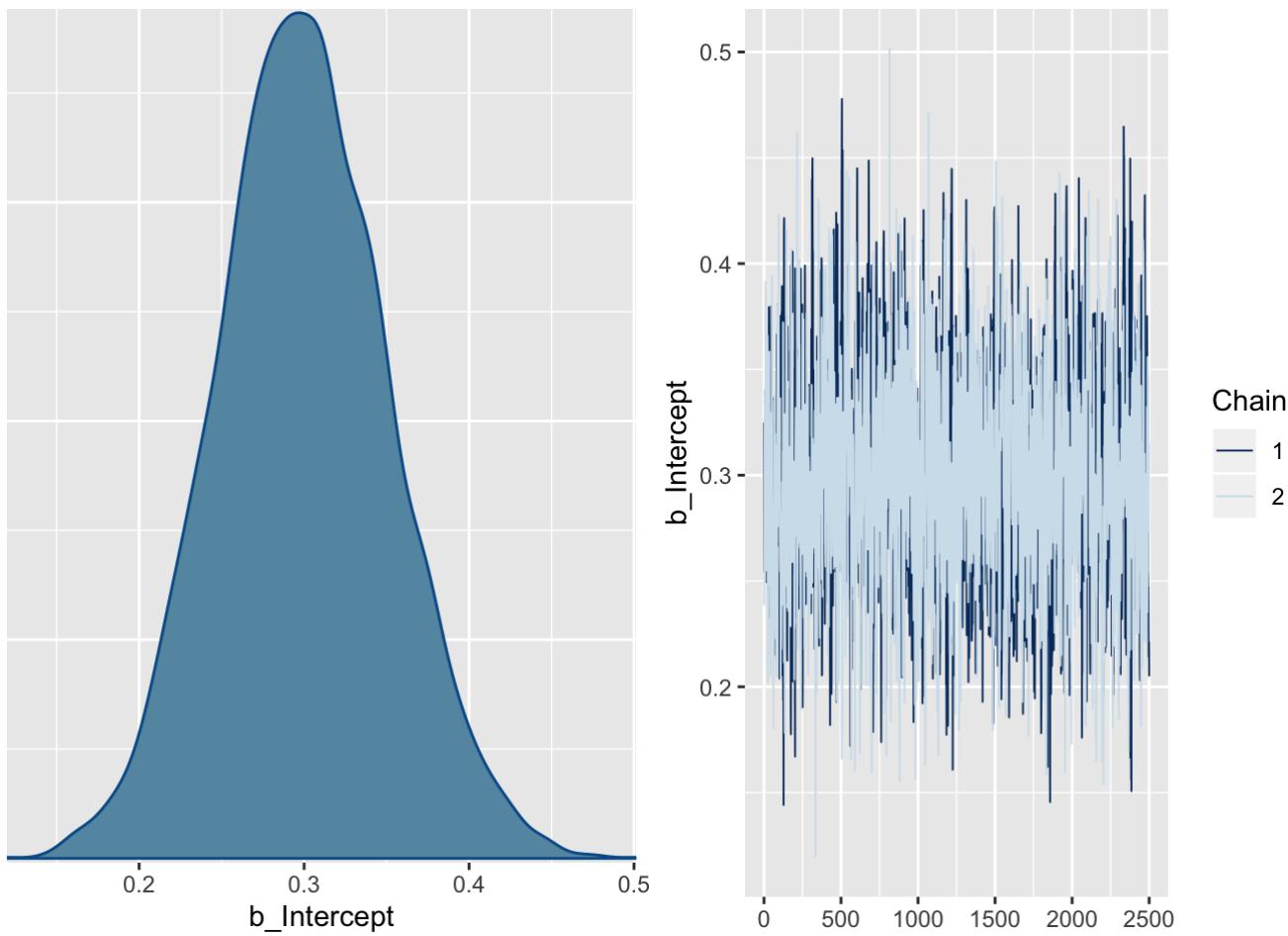
```
# get_prior(data = model_df,
#           family = bernoulli(link = "logit"),
#           formula = bf(intervene ~ 1))

# 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

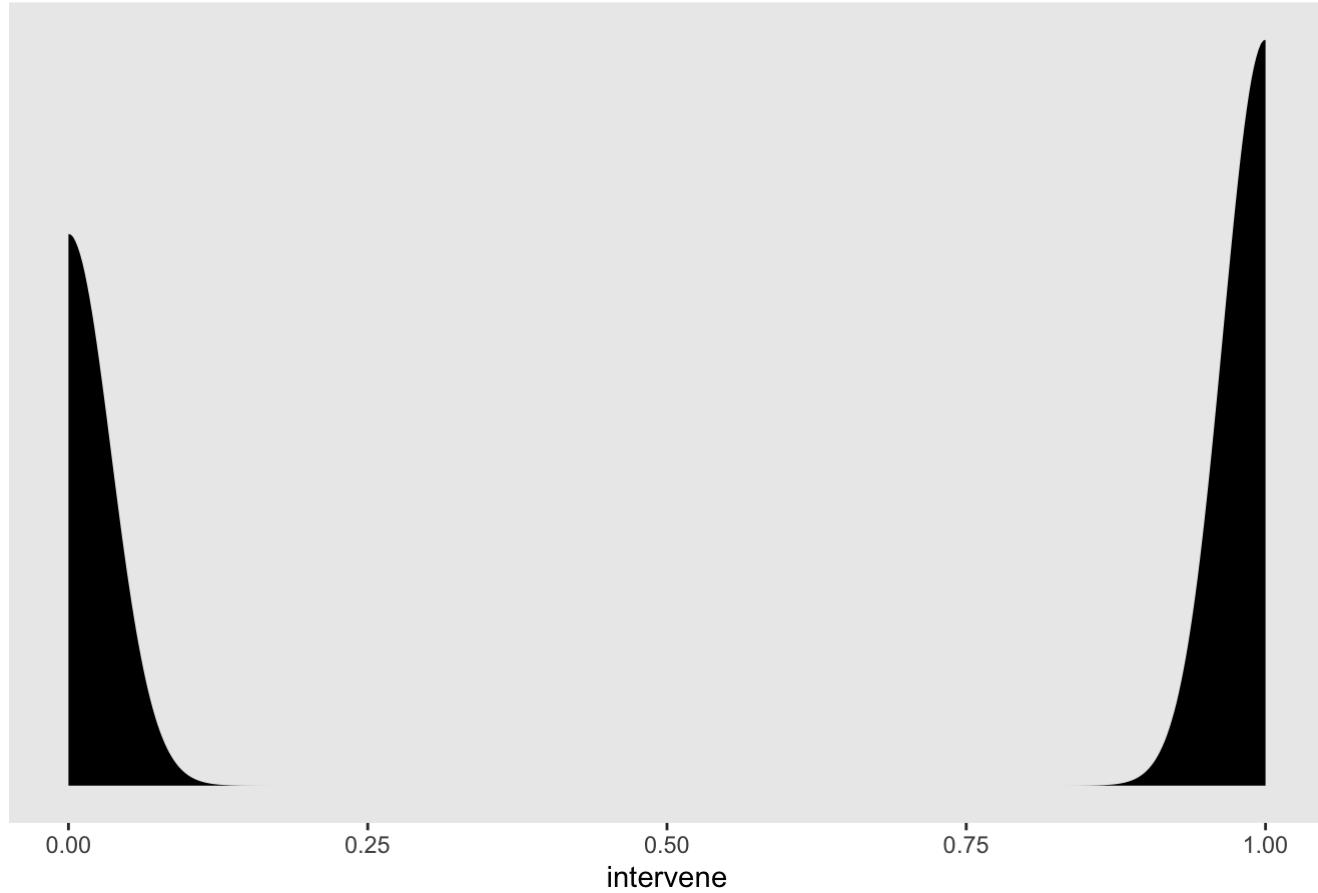
```
# model summary
print(m.logistic_intercept)
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ 1
## Data: model_df (Number of observations: 1568)
## 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     0.30      0.05     0.20      0.40       1517  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) %>% # 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

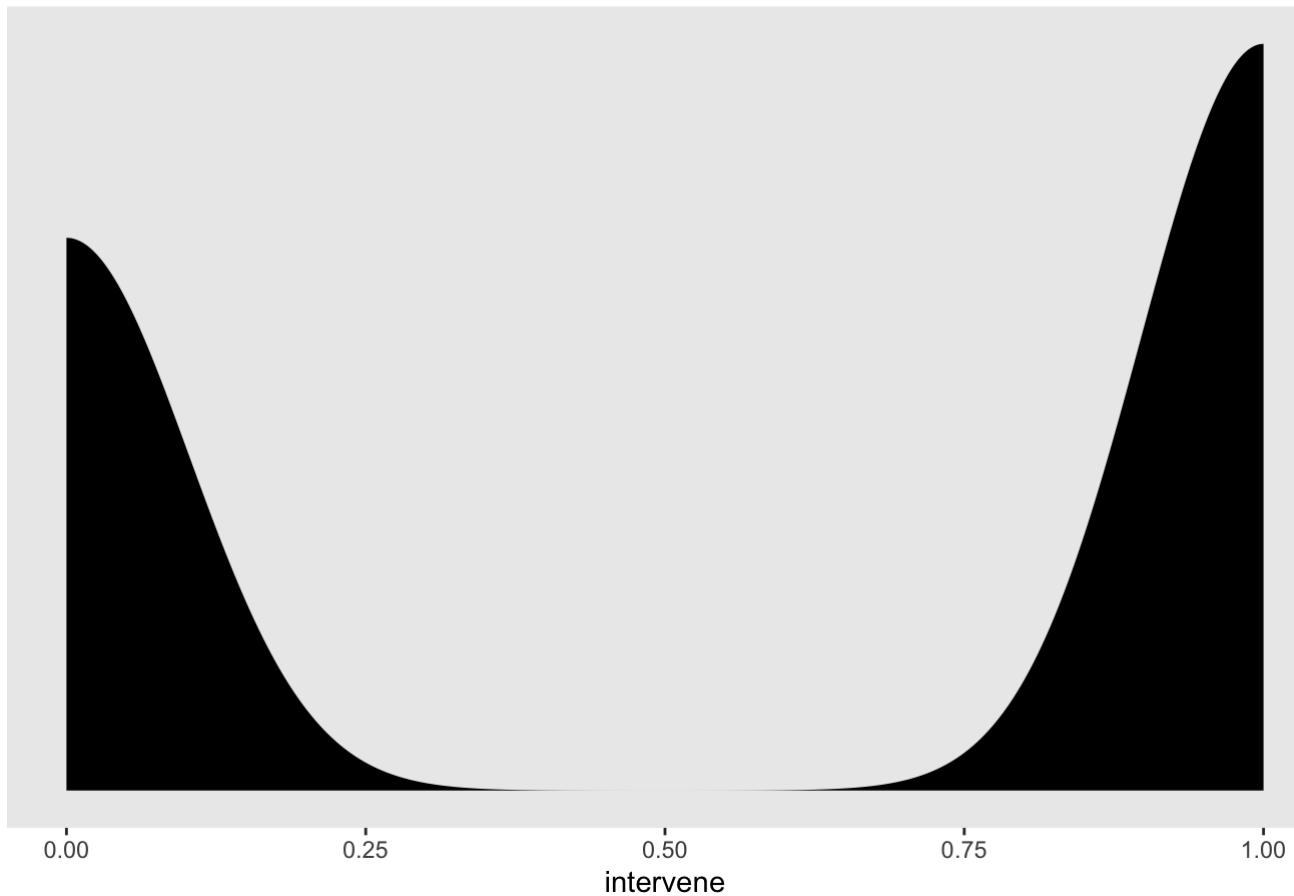


The posterior predictive distribution is about what we'd expect. The bias toward intervening is consistent with a positive intercept parameter.

How do the posterior predictions compare to the observed data?

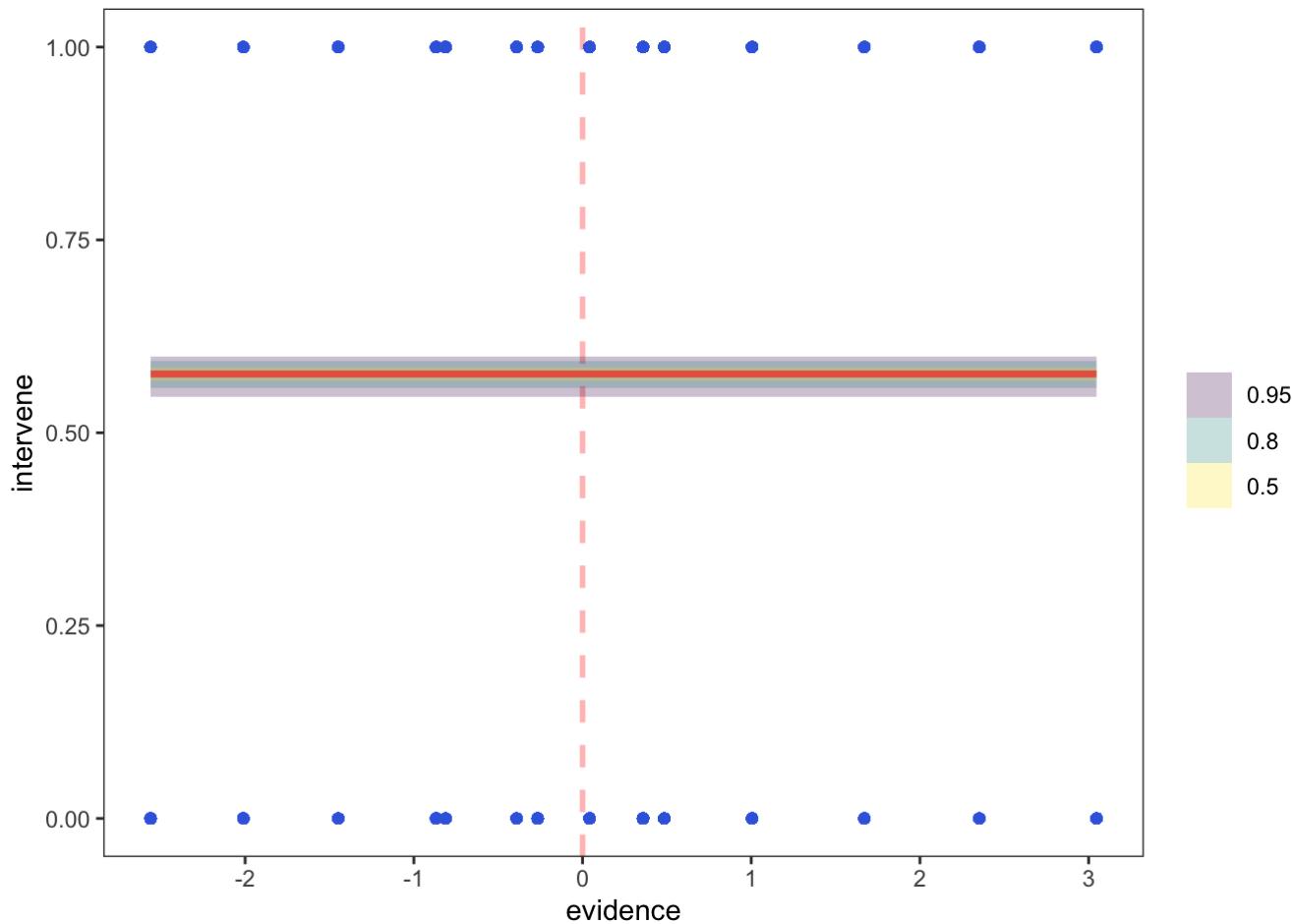
```
# data density
model_df %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Data distribution for intervention") +
  theme(panel.grid = element_blank())
```

### Data distribution for intervention



Let's take a look at the estimated psychometric function. This should not have any slope.

```
model_df %>%
  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

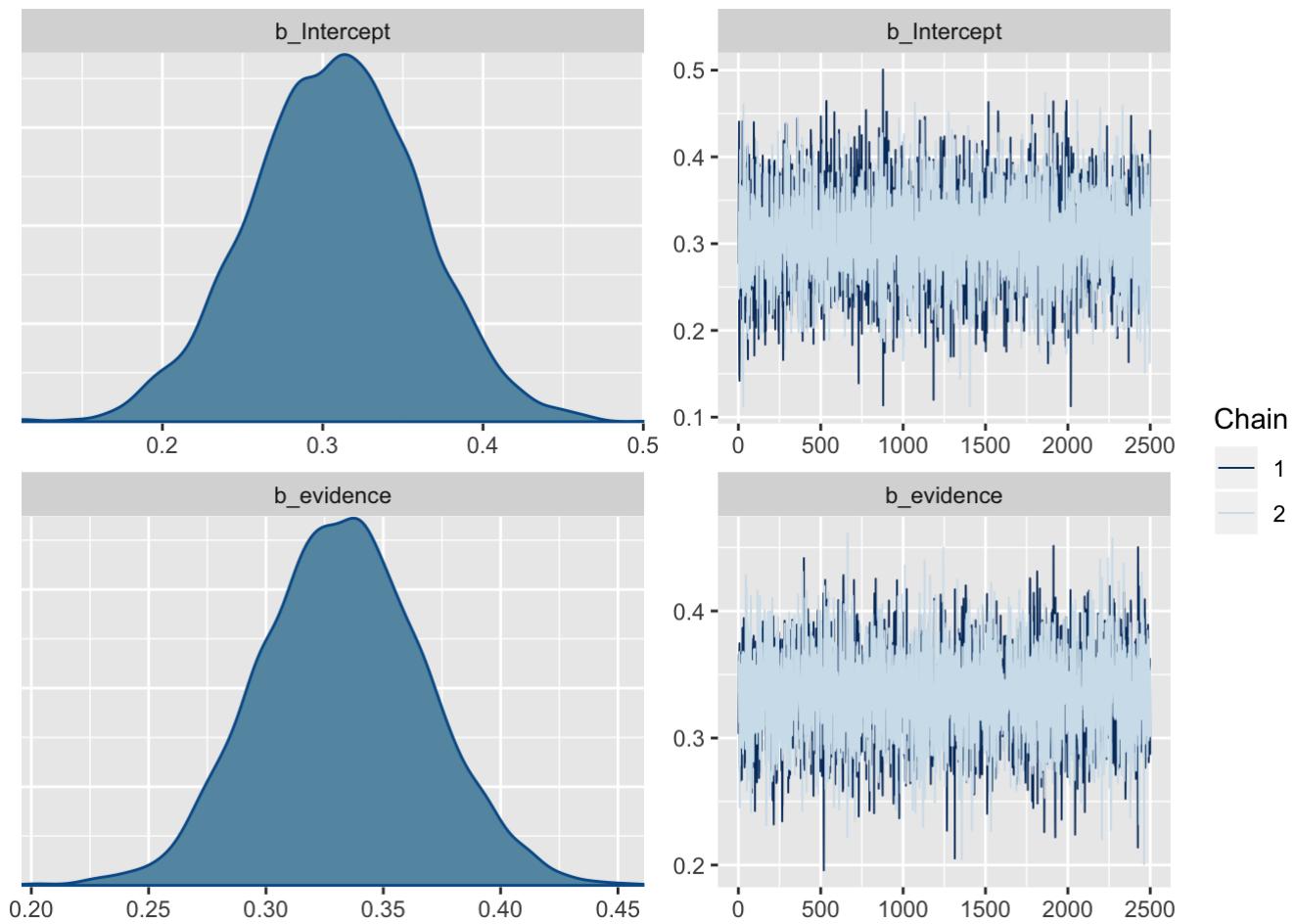
Now we'll add a slope parameter to our model to make it a simple linear model where decisions to intervene are a function of the probability of getting the award with the new player.

```
# 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(0, 1), class = b)),
  iter = 3000, warmup = 500, chains = 2, cores = 2,
  file = "model-fits/logistic_mdl")
```

Check diagnostics:

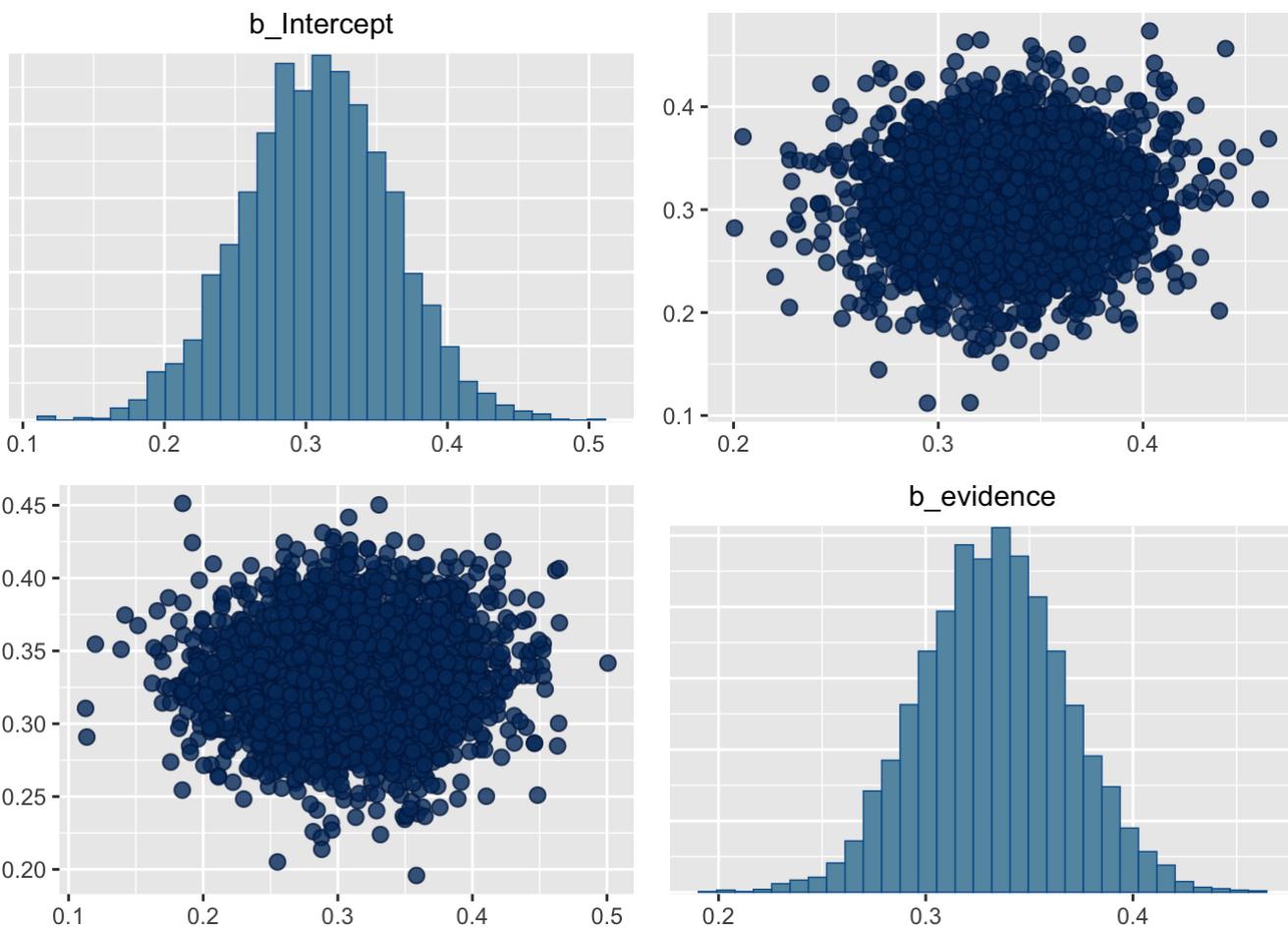
- Trace plots

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



- Pairs plot

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



- Summary

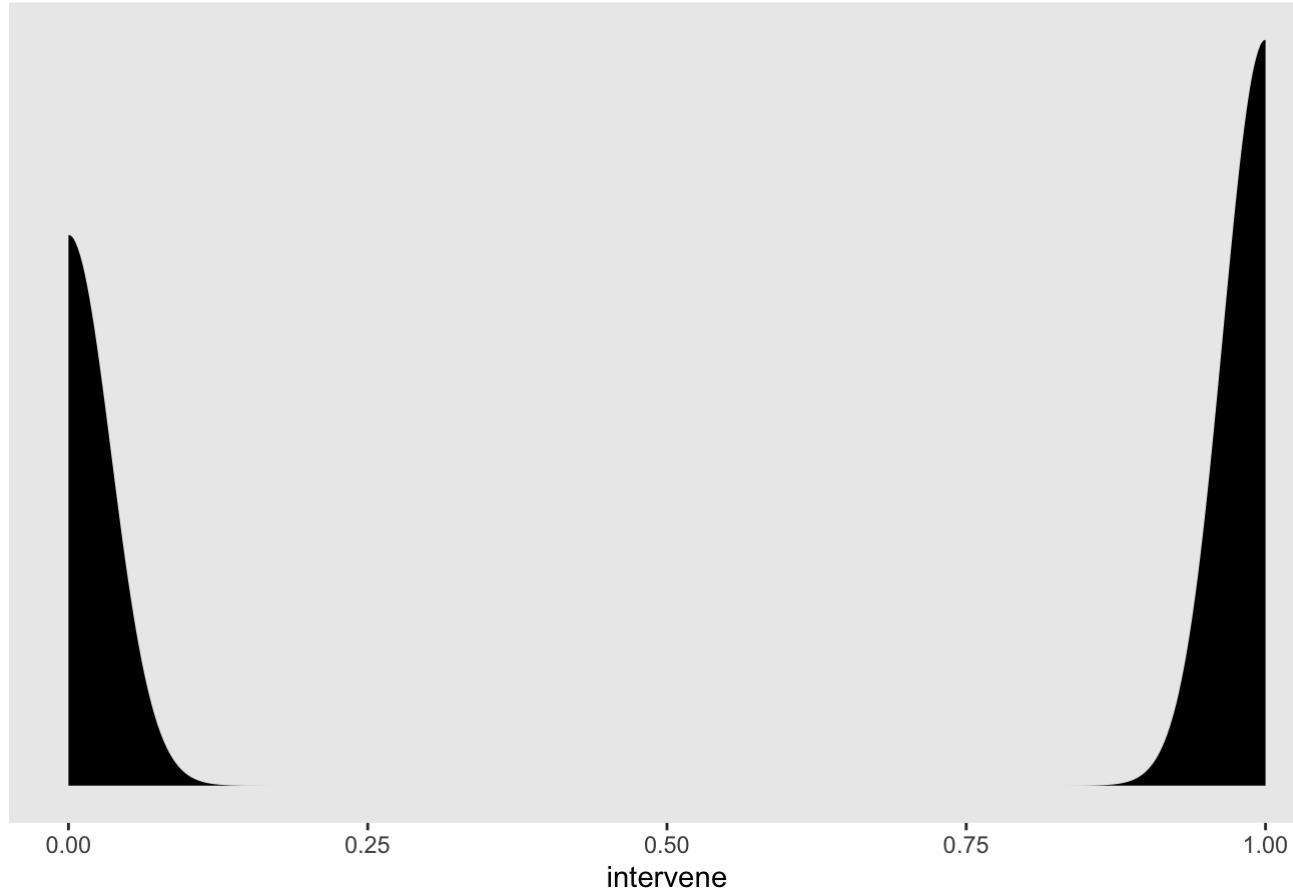
```
# model summary
print(m.logistic)
```

```
## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ 1 + evidence
## Data: model_df (Number of observations: 1568)
## 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     0.31      0.05    0.20     0.41       5315  1.00
## evidence      0.33      0.04    0.27     0.40       4281  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())
```

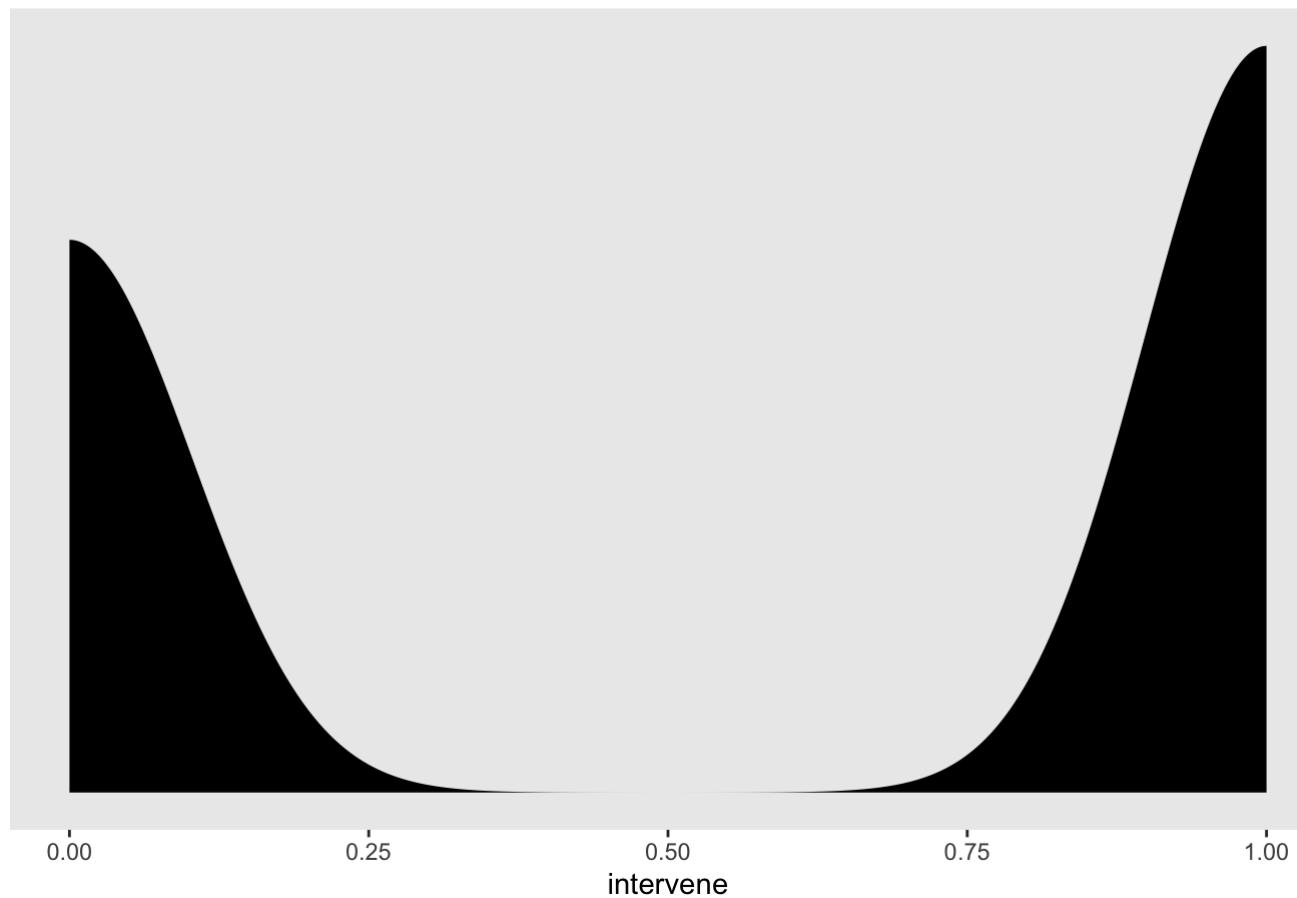
Posterior predictive distribution for intervention



How do the posterior predictions compare to the observed data?

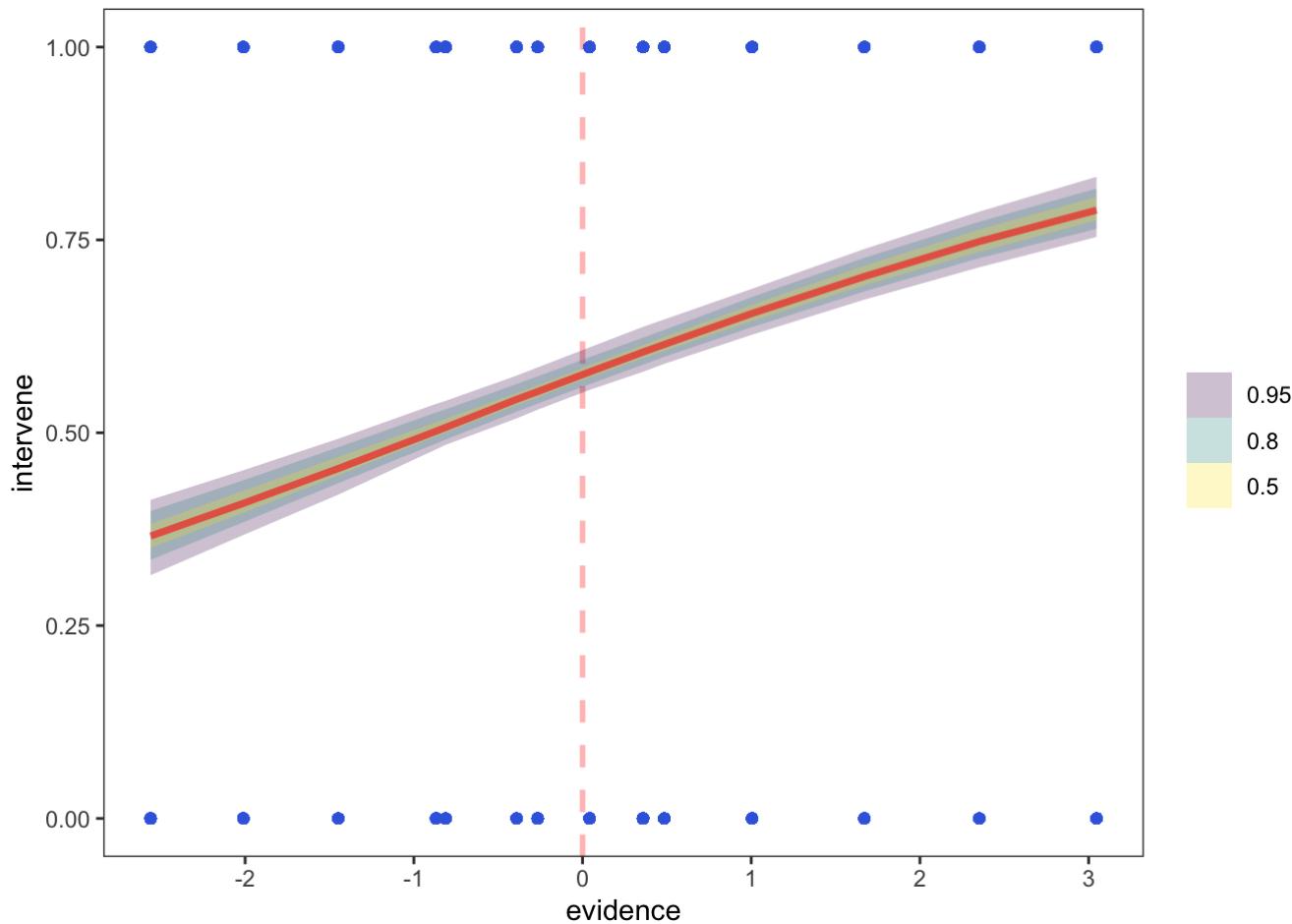
```
# data density
model_df %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Data distribution for intervention") +
  theme(panel.grid = element_blank())
```

### Data distribution for intervention



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 Different Linear Models per Visualization Condition

Now we add visualization condition as a predictor of both the point of subjective equality and the slope of the psychometric function.

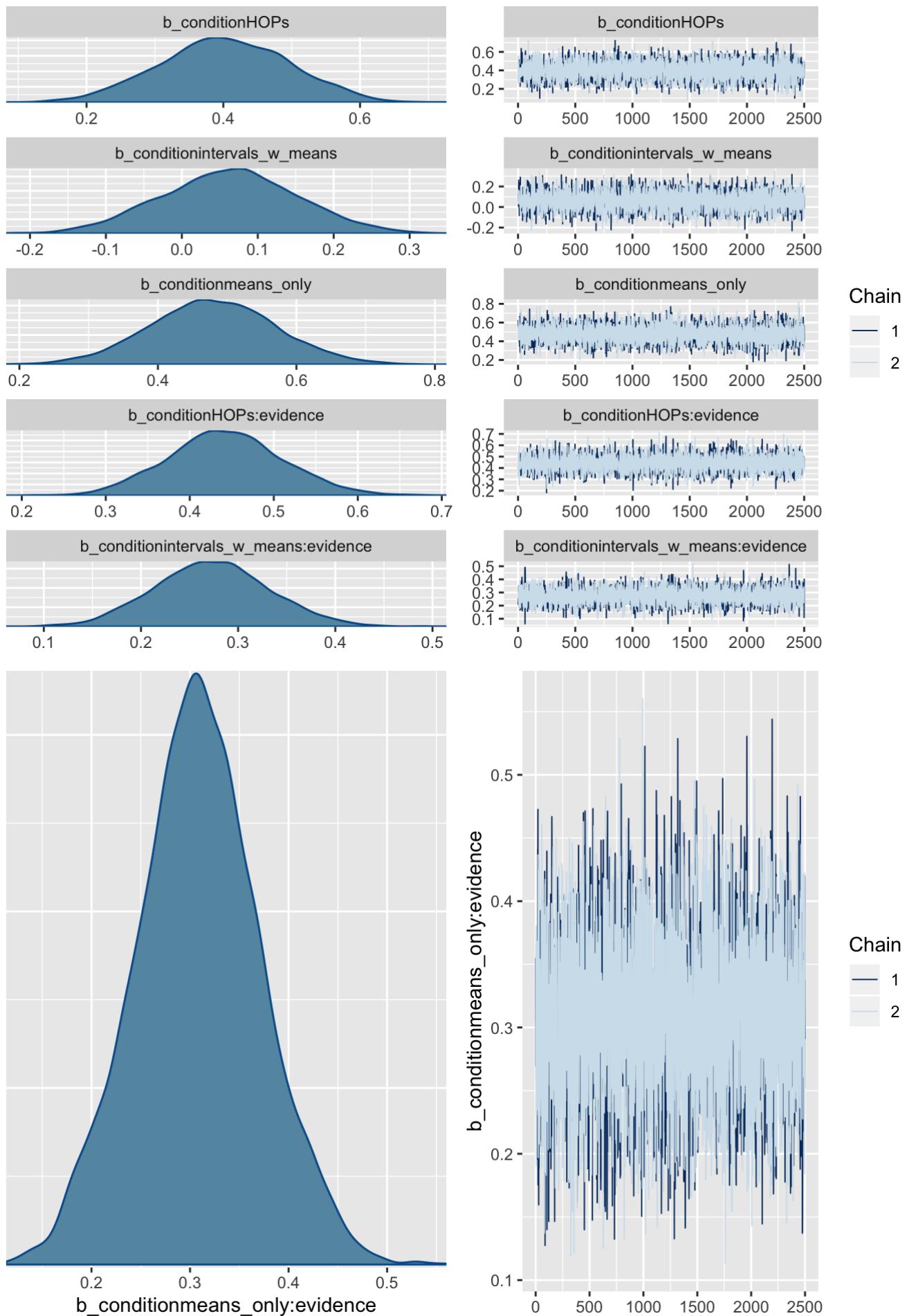
```
# get_prior(data = model_df, family = bernoulli(link = "logit"),
#           formula = bf(intervene ~ 0 + condition + evidence:condition))

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

Check diagnostics:

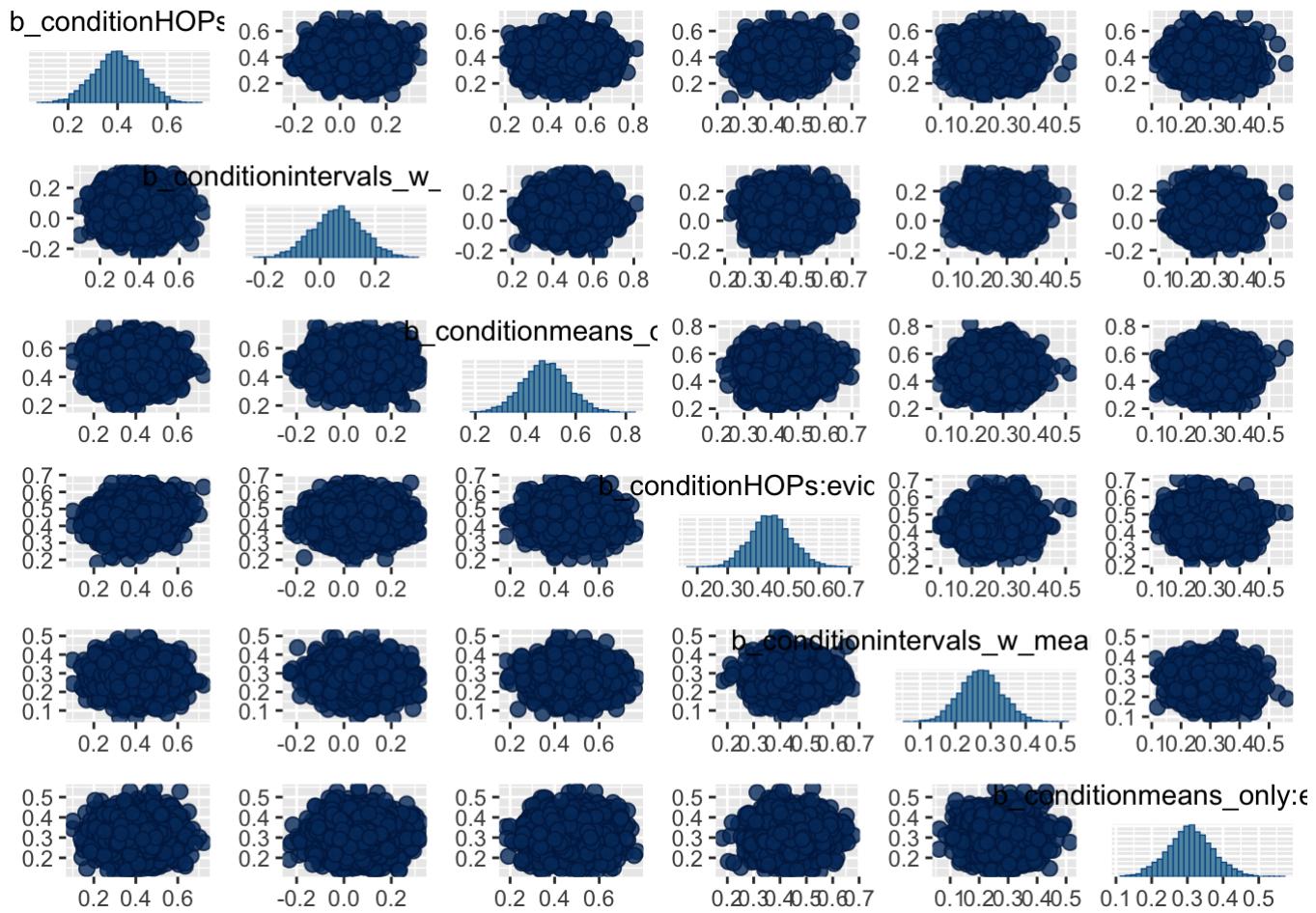
- Trace plots

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



- Pairs plot

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



- Summary

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

```

## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ 0 + condition + evidence:condition
## Data: model_df (Number of observations: 1568)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Population-Level Effects:
##                               Estimate Est.Error 1-95% CI u-95% CI
## conditionHOPs                  0.40     0.09   0.22    0.58
## conditionintervals_w_means     0.06     0.09  -0.11    0.24
## conditionmeans_only            0.48     0.09   0.30    0.66
## conditionHOPs:evidence        0.44     0.07   0.31    0.58
## conditionintervals_w_means:evidence  0.27     0.06   0.16    0.39
## conditionmeans_only:evidence   0.31     0.06   0.18    0.44
##                               Eff.Sample Rhat
## conditionHOPs                  6007 1.00
## conditionintervals_w_means     6469 1.00
## conditionmeans_only            6617 1.00
## conditionHOPs:evidence        7016 1.00
## conditionintervals_w_means:evidence  5479 1.00
## conditionmeans_only:evidence   4798 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).

```

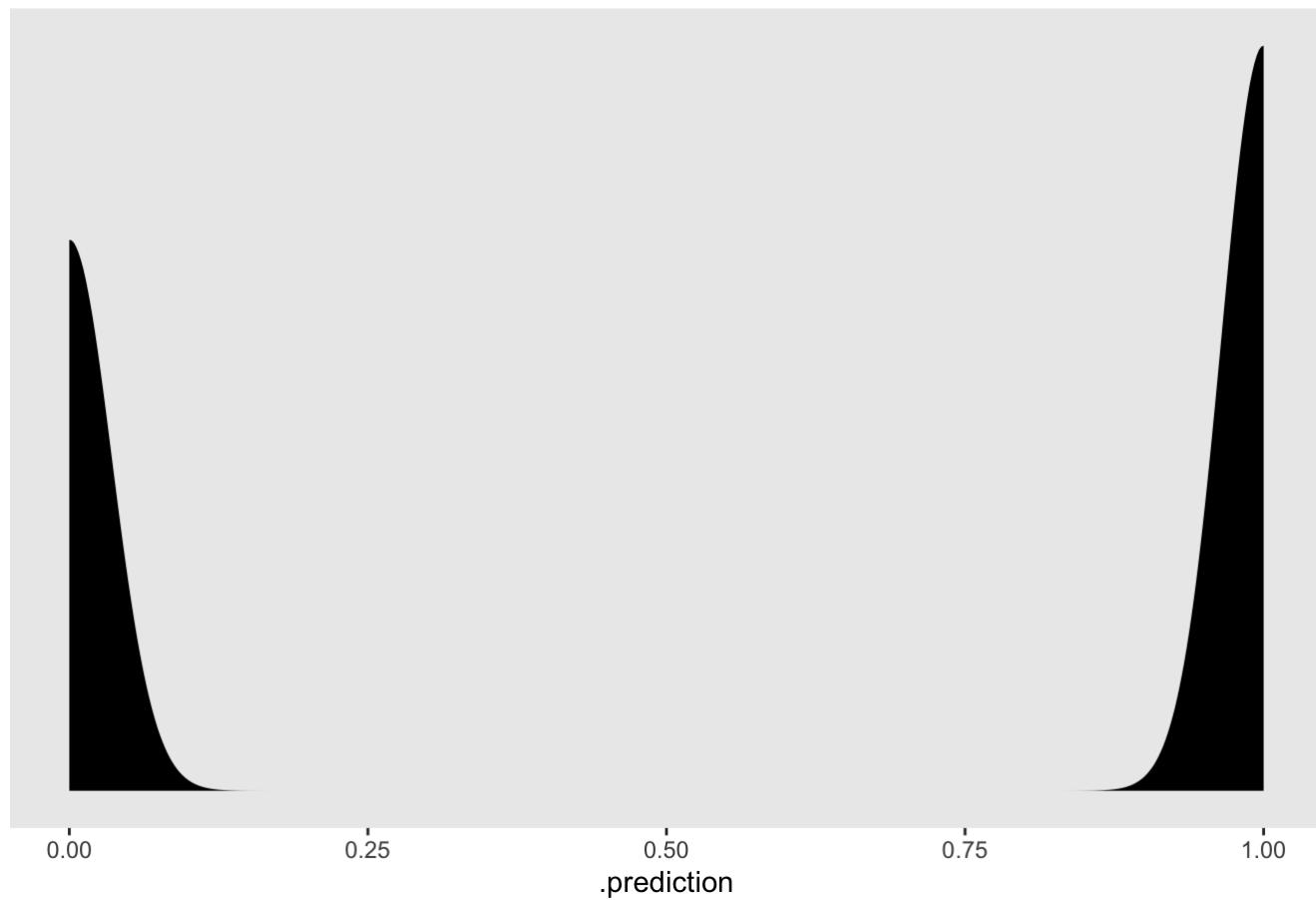
Let's check out a posterior predictive distribution for intervention decisions.

```

# posterior predictive check
model_df %>%
  group_by(condition, evidence) %>%
  add_predicted_draws(m.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())

```

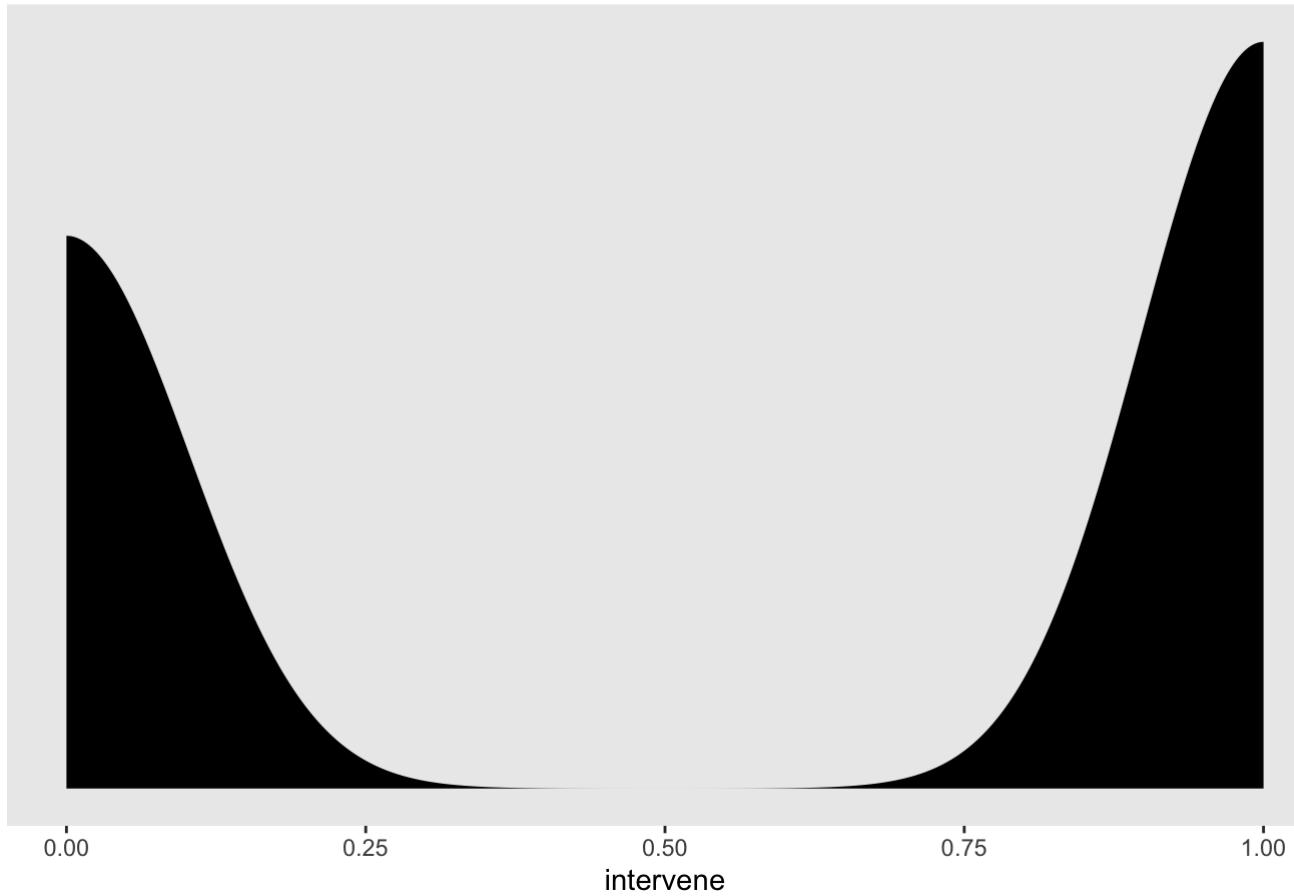
## Posterior predictive distribution for intervention



How do the posterior predictions compare to the observed data?

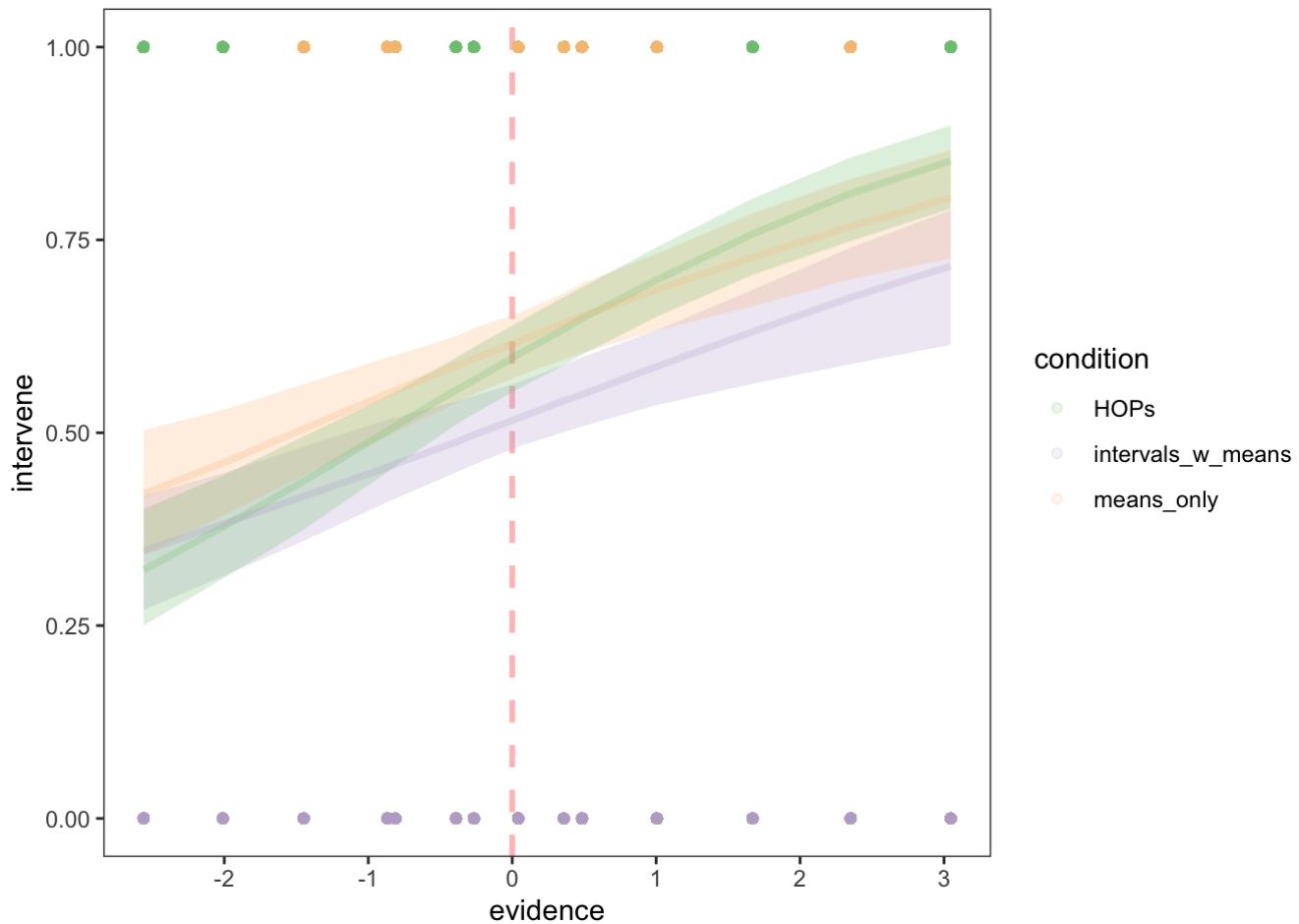
```
# data density
model_df %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Data distribution for intervention") +
  theme(panel.grid = element_blank())
```

### Data distribution for intervention



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

```
model_df %>%
  add_fitted_draws(m.vis.logistic, value = "pf", n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = condition)) +
  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, fill = condition), .width = .95, alpha = .25, show.legend =
  FALSE) +
  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())
```



## 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.

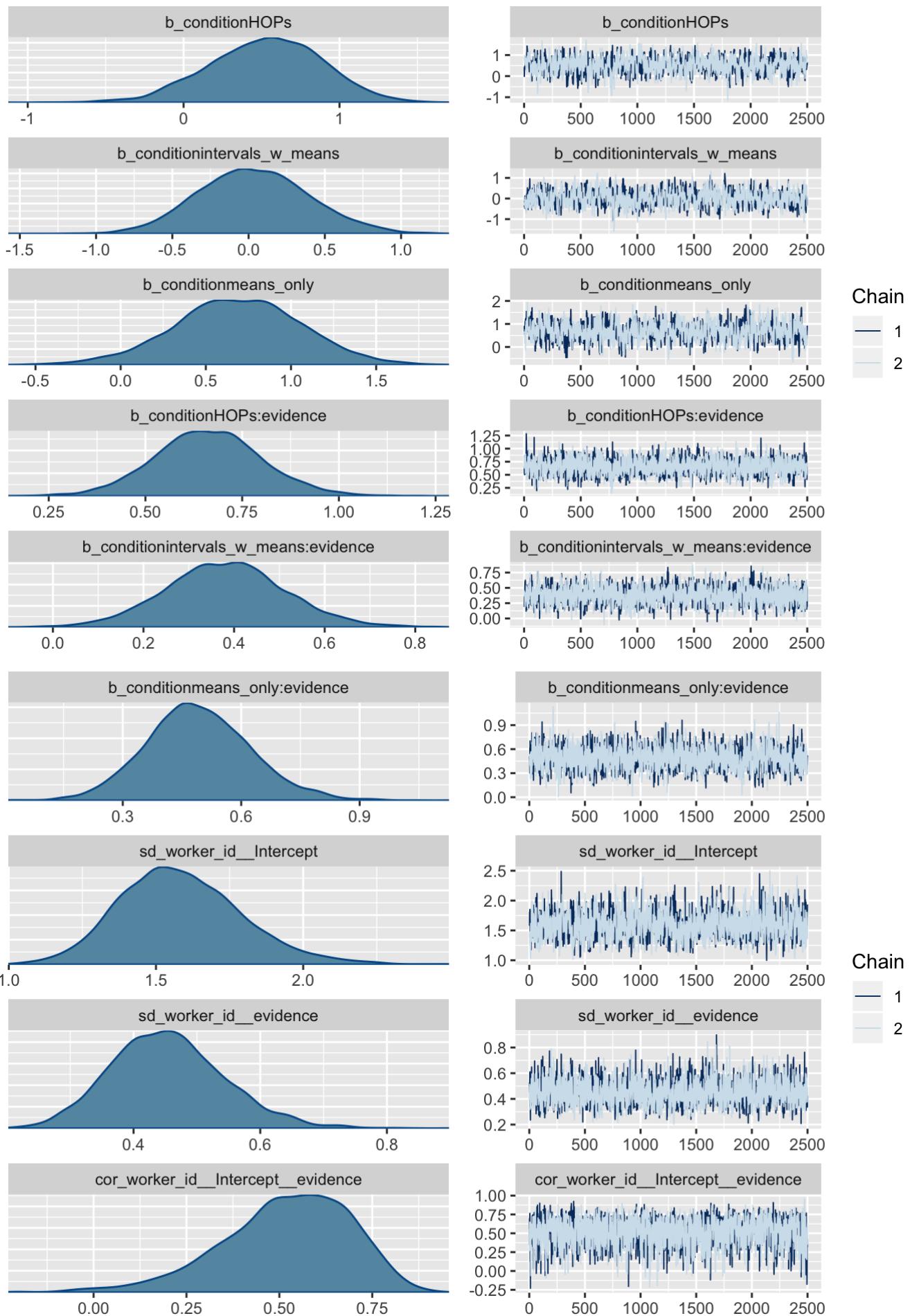
```
# get_prior(data = model_df, family = bernoulli(link = "logit"),
#           formula = bf(intervene ~ 0 + condition + evidence:condition))

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

Check diagnostics:

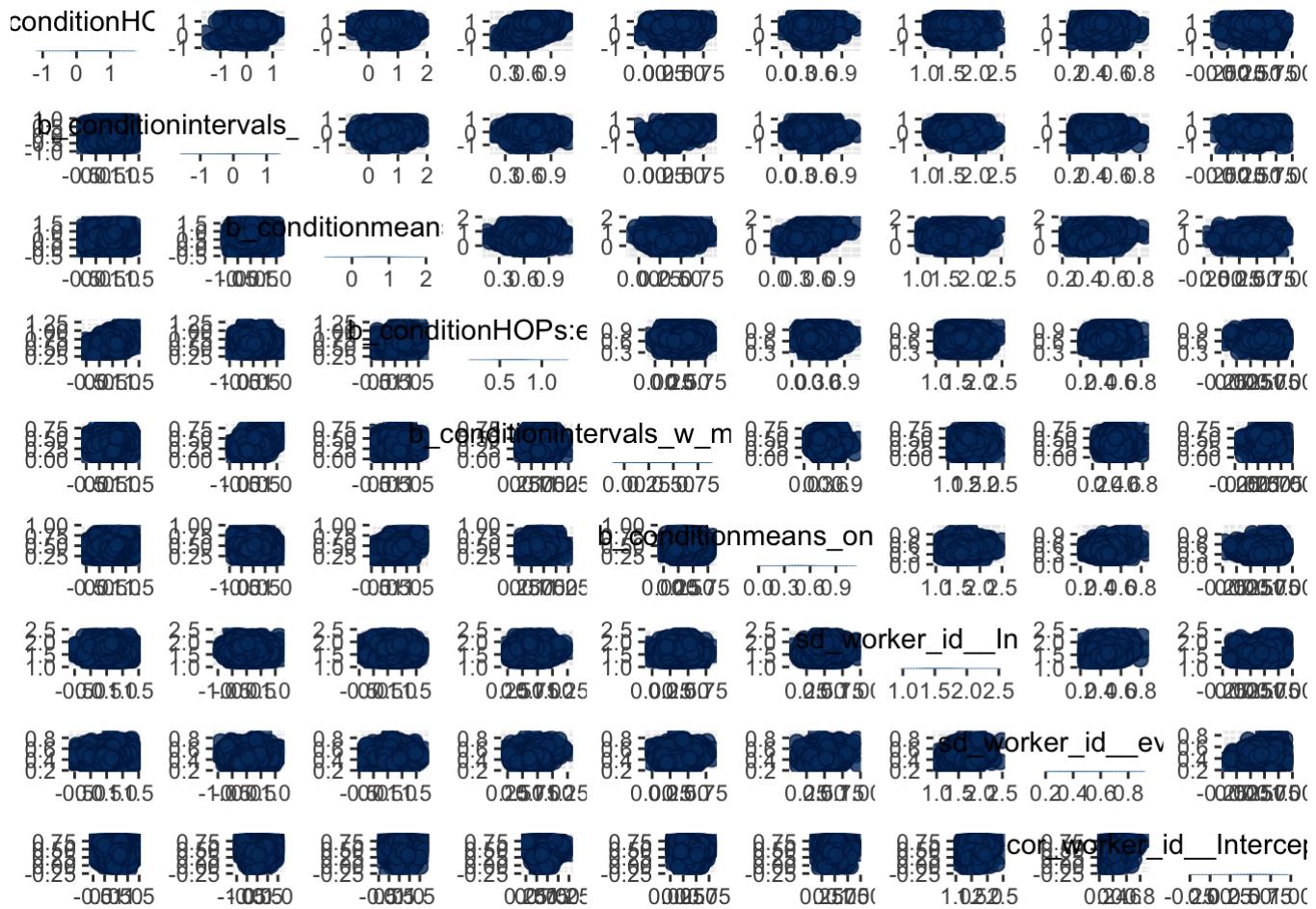
- Trace plots

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



- Pairs plot

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



- Summary

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

```

## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + 0 + condition + evidence:condition
## Data: model_df (Number of observations: 1568)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 56)
##                                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)                  1.58     0.21    1.22    2.04      978
## sd(evidence)                  0.45     0.09    0.29    0.64     1506
## cor(Intercept,evidence)       0.52     0.17    0.13    0.80     1640
##                                         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
## conditionHOPs                  0.53     0.36   -0.19    1.20
## conditionintervals_w_means    0.04     0.36   -0.65    0.77
## conditionmeans_only            0.69     0.37   -0.07    1.41
## conditionHOPs:evidence        0.66     0.14    0.39    0.94
## conditionintervals_w_means:evidence  0.38     0.13    0.13    0.65
## conditionmeans_only:evidence   0.49     0.13    0.24    0.77
##                                         Eff.Sample Rhat
## conditionHOPs                  675 1.00
## conditionintervals_w_means    595 1.00
## conditionmeans_only            661 1.00
## conditionHOPs:evidence        1397 1.00
## conditionintervals_w_means:evidence  1368 1.00
## conditionmeans_only:evidence   1325 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).

```

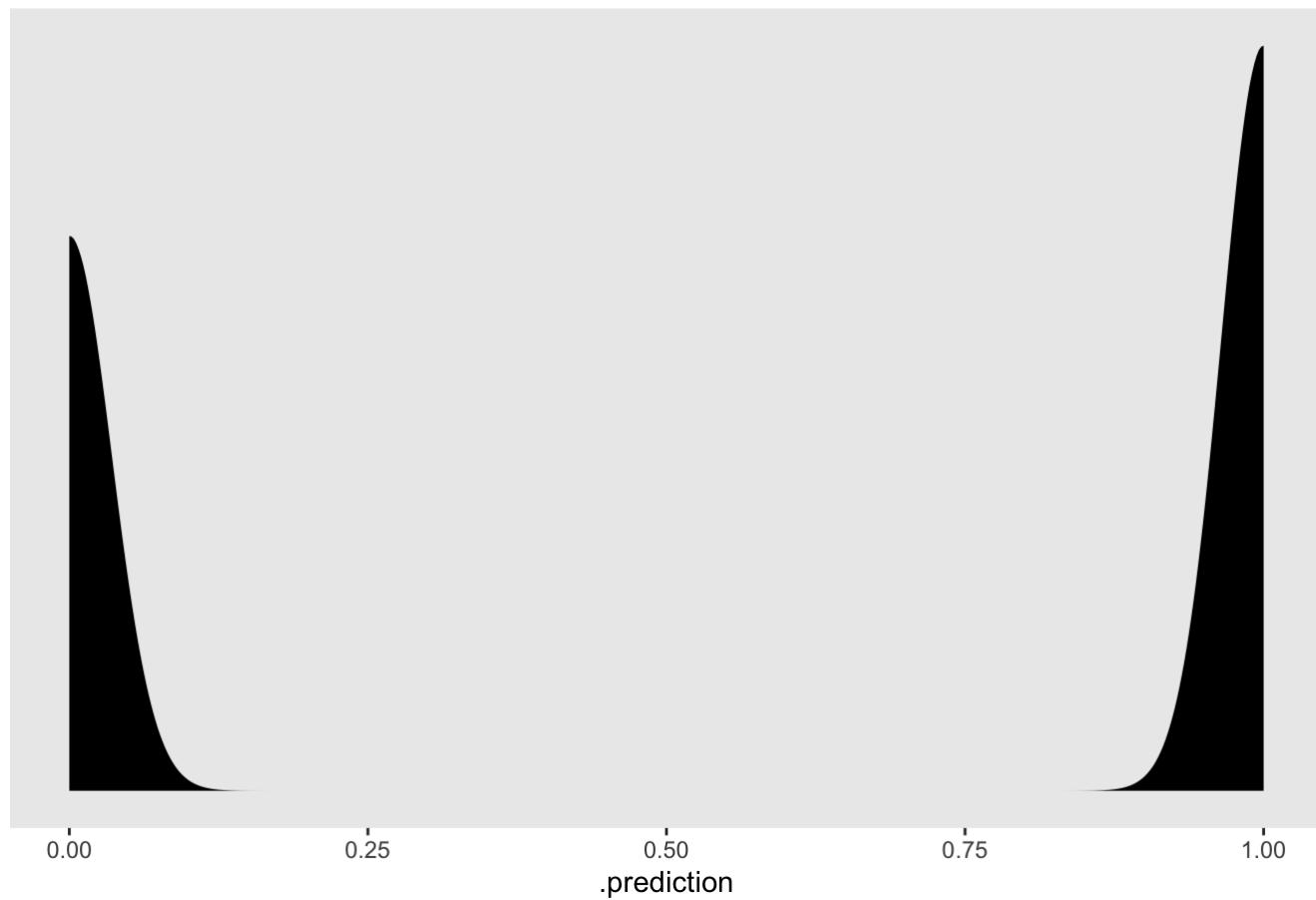
Let's check out a posterior predictive distribution for intervention decisions.

```

# posterior predictive check
model_df %>%
  group_by(worker_id, condition, evidence) %>%
  add_predicted_draws(m.vis.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())

```

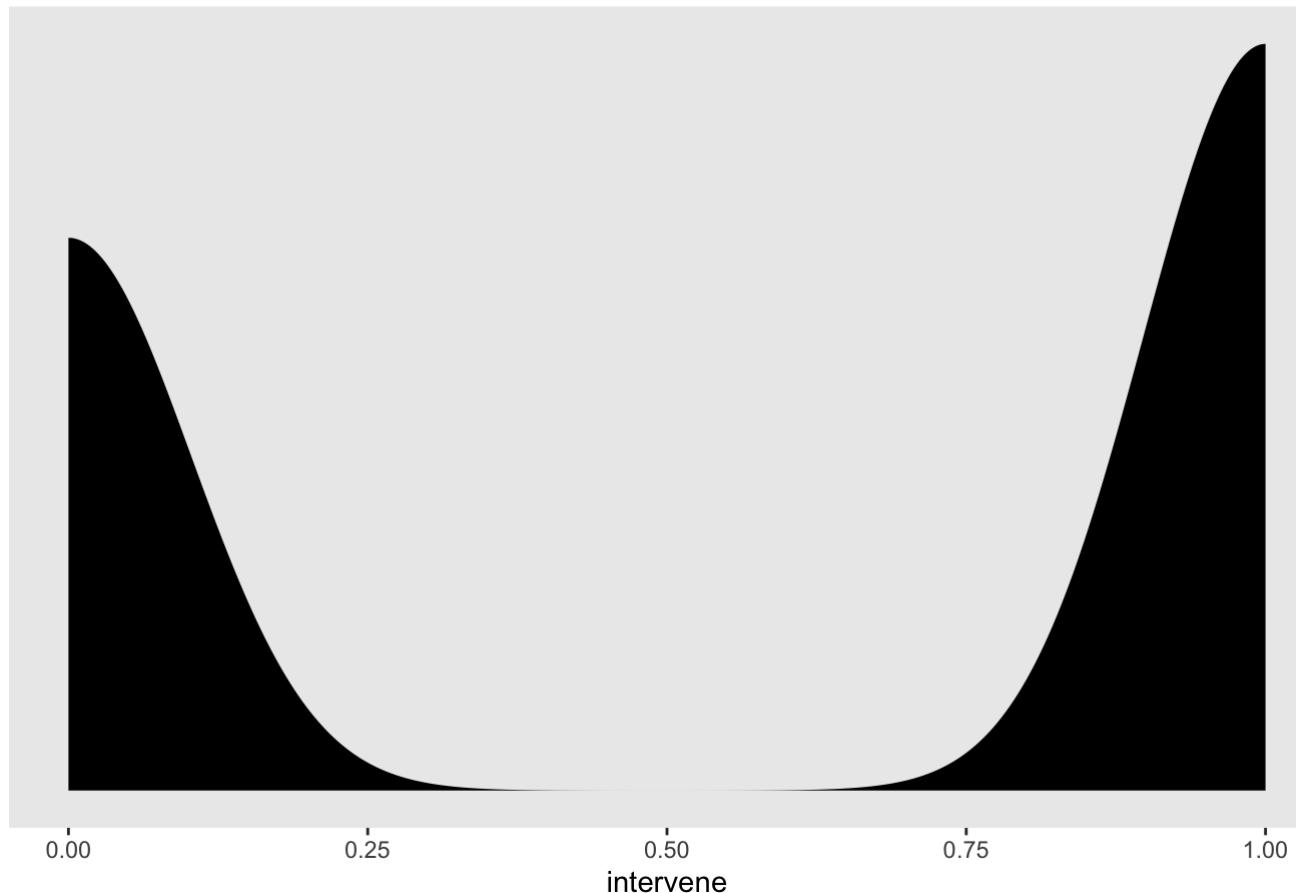
## Posterior predictive distribution for intervention



How do the posterior predictions compare to the observed data?

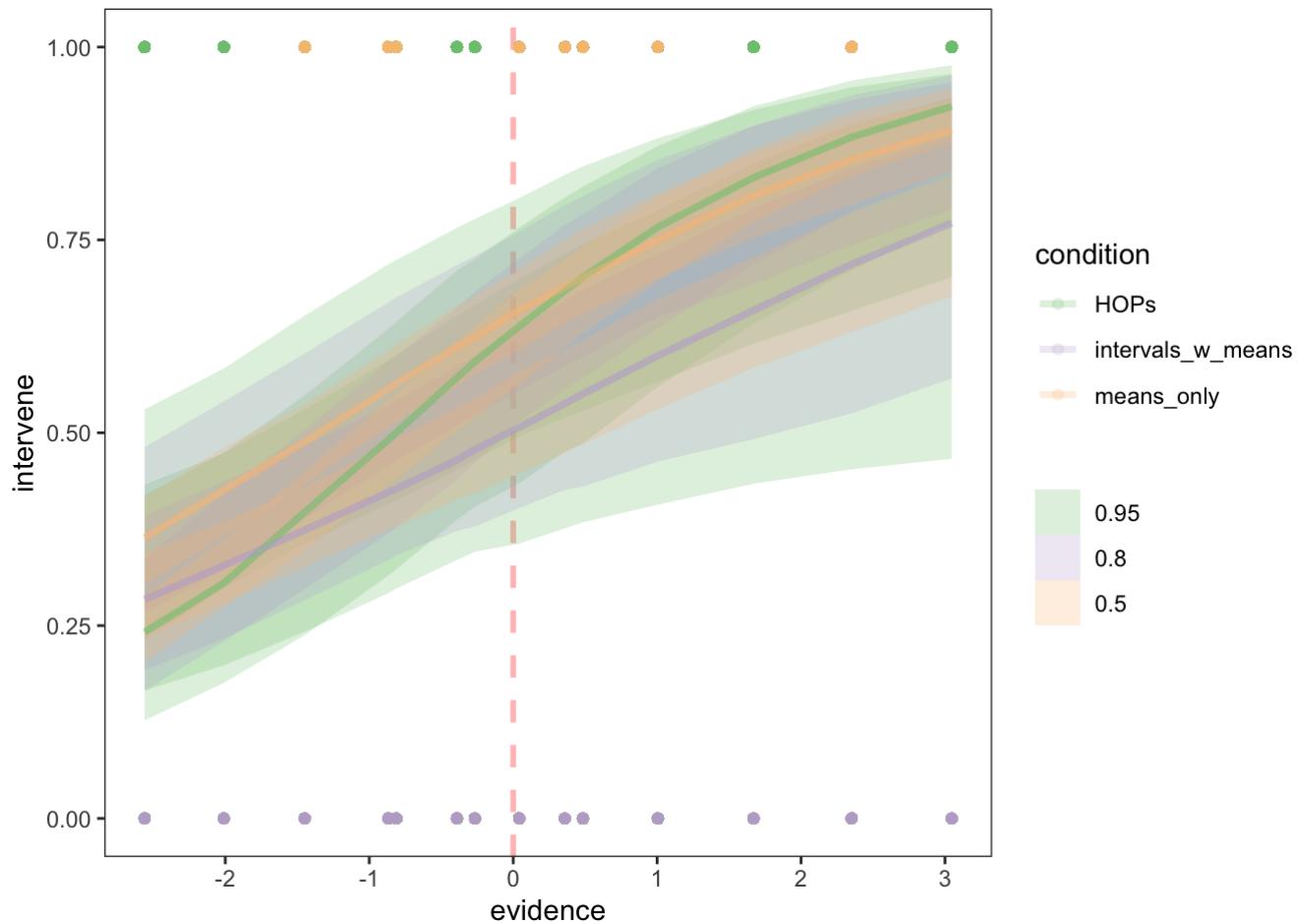
```
# data density
model_df %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Data distribution for intervention") +
  theme(panel.grid = element_blank())
```

### Data distribution for intervention



Let's take a look at the estimated psychometric function in each condition for the average observer.

```
model_df %>%
  group_by(evidence, condition, worker_id) %>%
  add_fitted_draws(m.vis.wrkr.logistic, value = "pf", re_formula = NA, n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = 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, .80, .50), 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())
```



Now we're seeing more separation between visualization conditions. However, these differences are highly uncertain.

## Add Predictors for Problem Framing

Similar to how we model the effects of visualization conditions on the location and slope of the psychometric function, we add predictors for gain vs loss framing. This gives us a three way interaction with evidence.

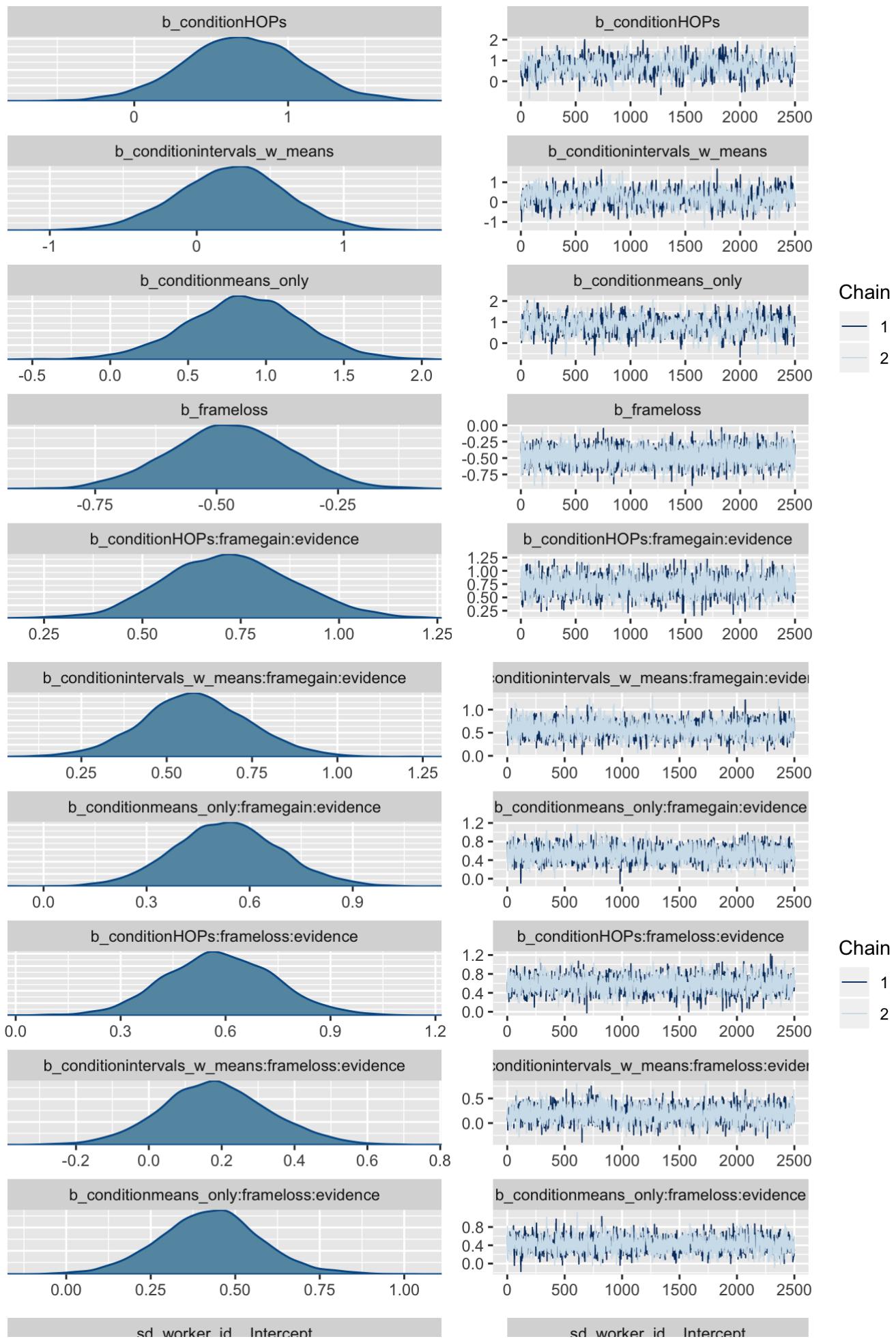
```
# linear model with logit link
m.vis.frame.wrkr.logistic <- brm(data = model_df, family = bernoulli(link = "logit"),
                                    formula = bf(intervene ~ (1 + evidence|worker_id) + 0 + co
                                    ndition + frame + evidence:condition:frame),
                                    prior = c(prior(normal(0, 1), class = b)),
                                    iter = 3000, warmup = 500, chains = 2, cores = 2,
                                    file = "model-fits/logistic_mdl_vis_frame_wrkr")
```

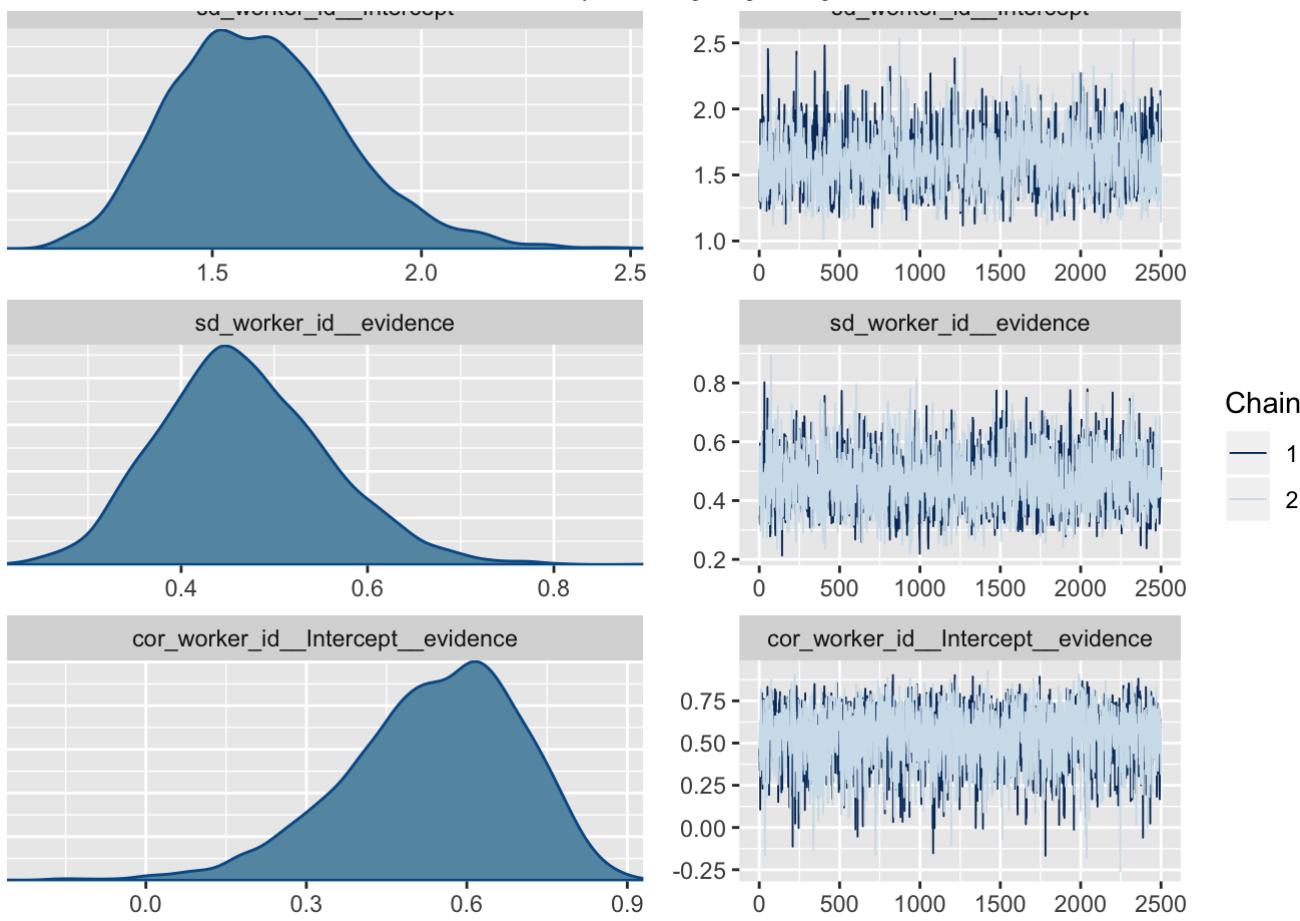
Check diagnostics:

- Trace plots

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

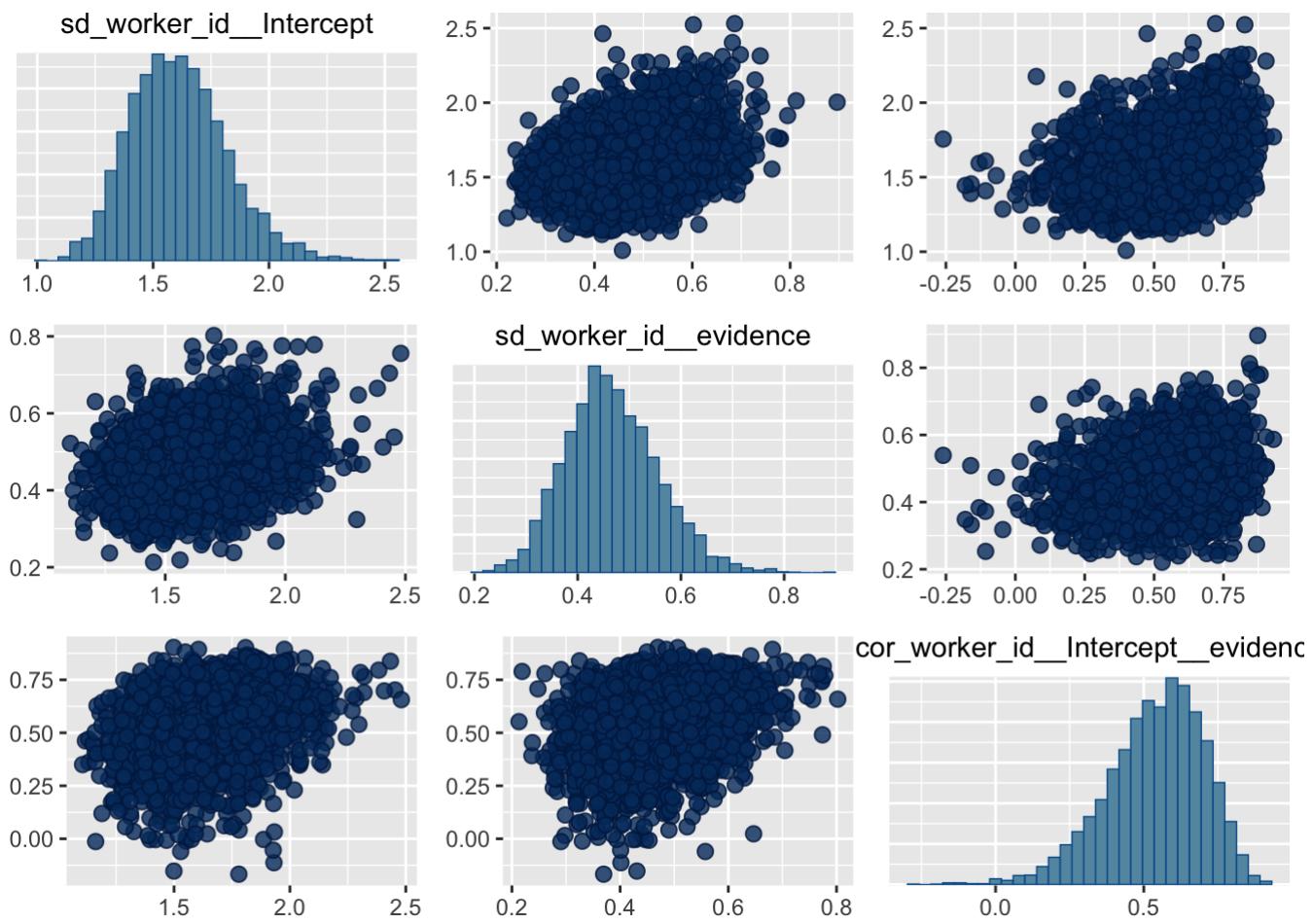






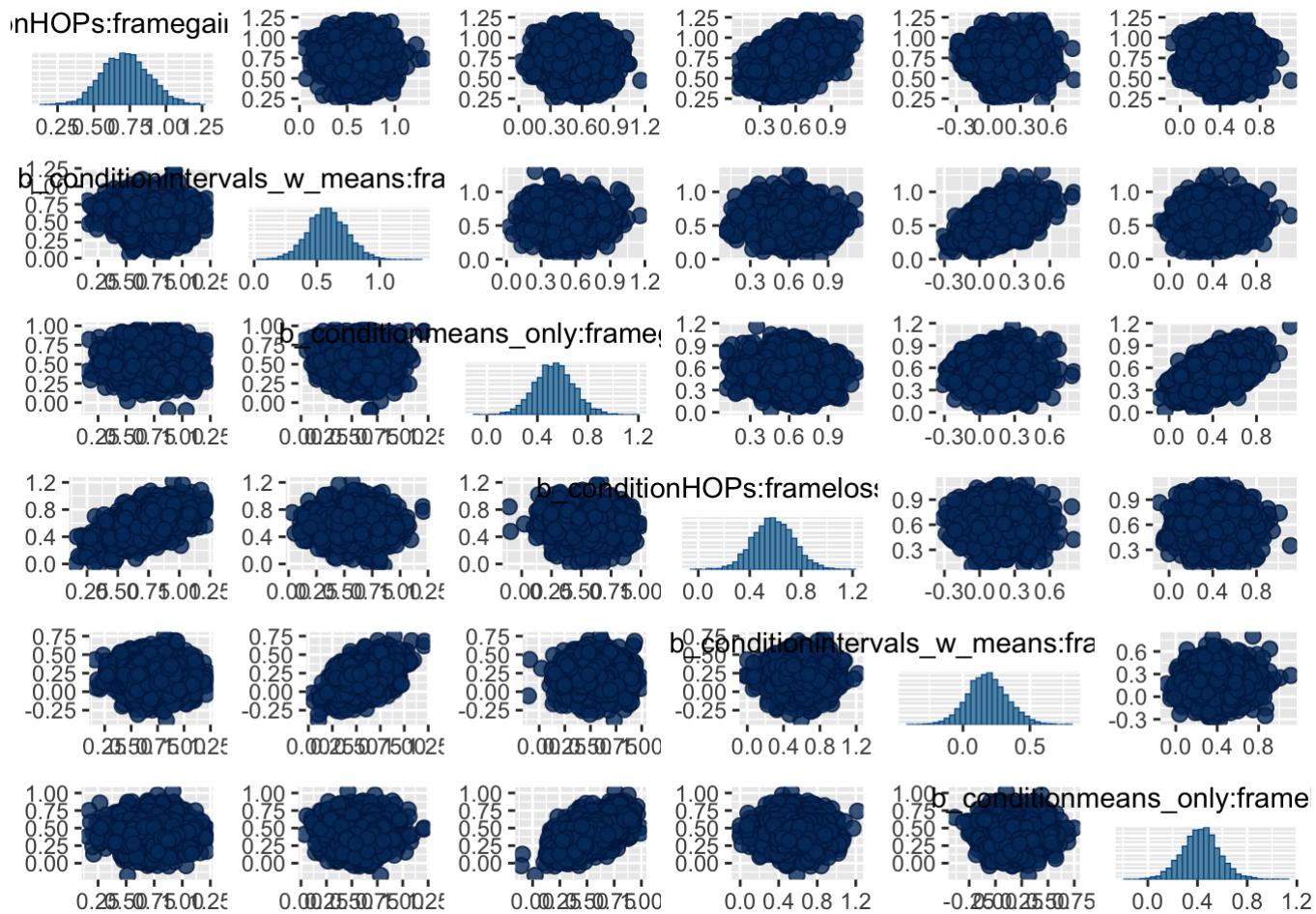
- Pairs plot

```
# pairs plot (too many things to view at once, so we've grouped them)
# hyperparameters
pairs(m.vis.frame.wrkr.logistic, pars = c("sd_worker_id_Intercept",
                                           "sd_worker_id_evidence",
                                           "cor_worker_id_Intercept_evidence"))
```

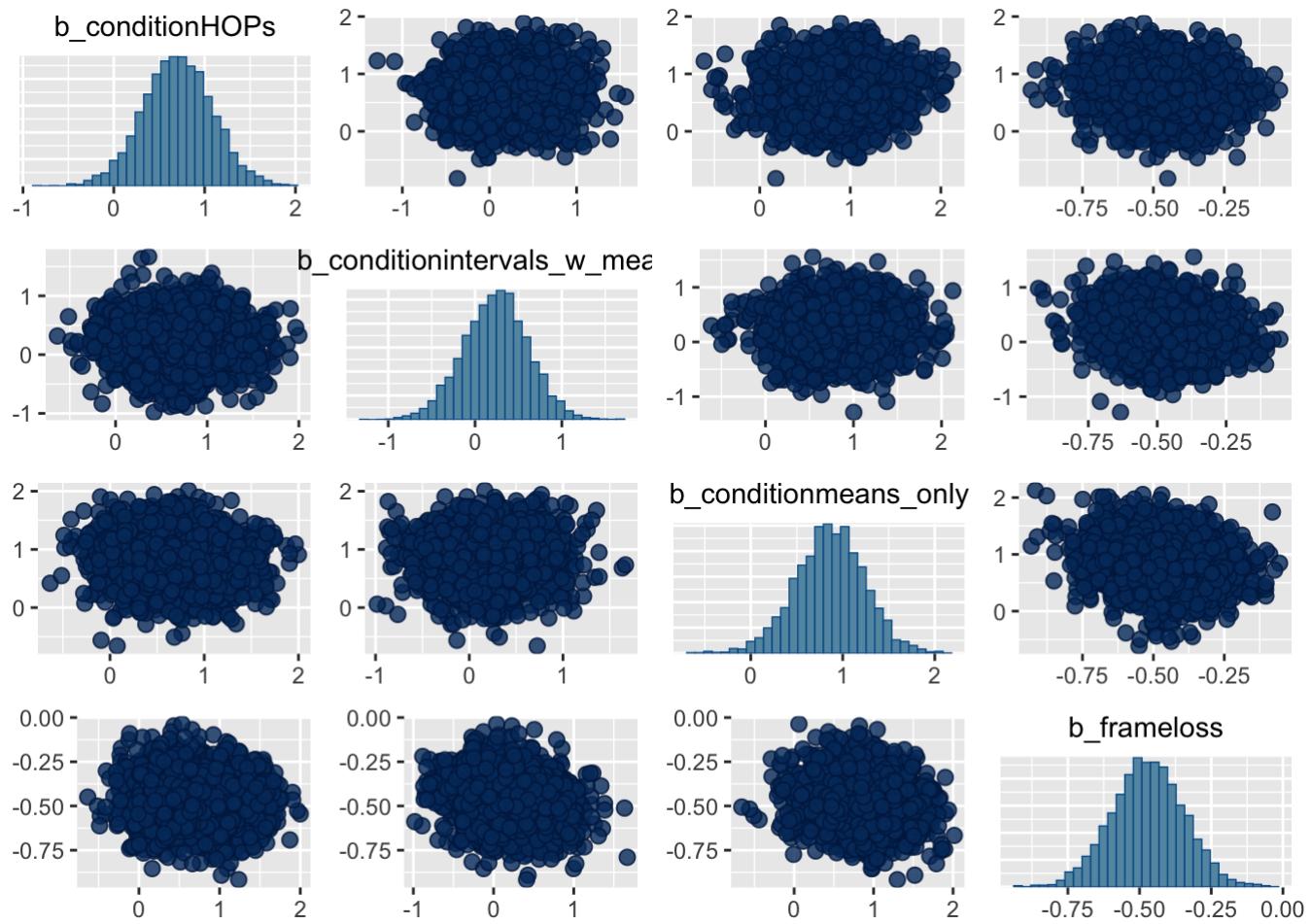


```
# pairs plot (too many things to view at once, so we've grouped them)
# slope effects
pairs(m.vis.frame.wrkr.logistic, pars = c("b_conditionHOPs:framegain:evidence",
                                           "b_conditionintervals_w_means:framegain:evidence"
                                           ,
                                           "b_conditionmeans_only:framegain:evidence",
                                           "b_conditionHOPs:frameloss:evidence",
                                           "b_conditionintervals_w_means:frameloss:evidence"
                                           ,
                                           "b_conditionmeans_only:frameloss:evidence"))

```



```
# pairs plot (too many things to view at once, so we've grouped them)
# intercept effects
pairs(m.vis.frame.wrkr.logistic, exact_match = TRUE, pars = c("b_conditionHOPs",
                                                               "b_conditionintervals_w_mean
s",
                                                               "b_conditionmeans_only",
                                                               "b_frameloss"))
```



- Summary

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

```

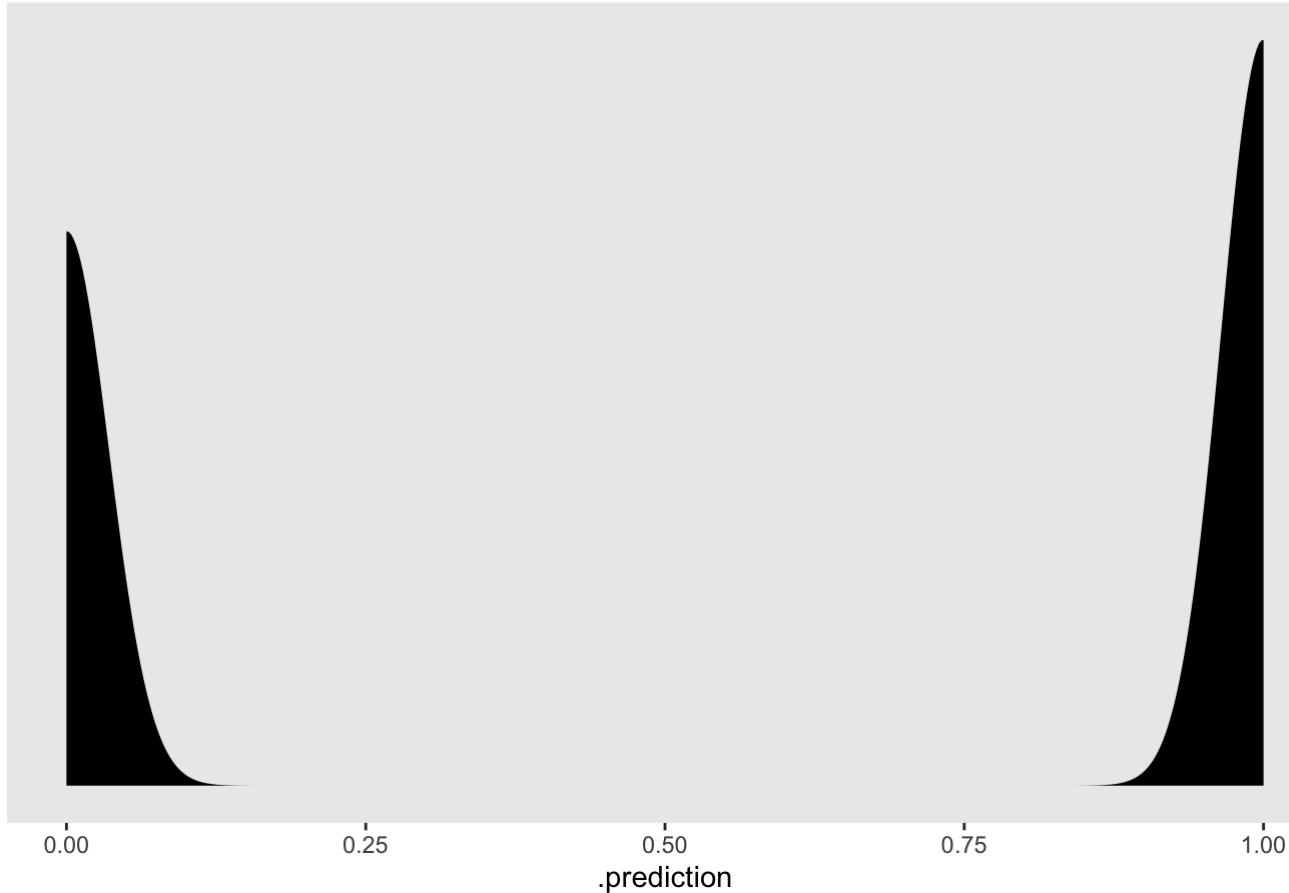
## Family: bernoulli
## Links: mu = logit
## Formula: intervene ~ (1 + evidence | worker_id) + 0 + condition + frame + evidence:condition:frame
## Data: model_df (Number of observations: 1568)
## Samples: 2 chains, each with iter = 3000; warmup = 500; thin = 1;
##          total post-warmup samples = 5000
##
## Group-Level Effects:
## ~worker_id (Number of levels: 56)
##                               Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## sd(Intercept)             1.61     0.21    1.26    2.07      1140
## sd(evidence)              0.47     0.09    0.31    0.65      1862
## cor(Intercept,evidence)   0.54     0.16    0.18    0.80      2054
##                               Rhat
## sd(Intercept)             1.00
## sd(evidence)              1.00
## cor(Intercept,evidence)   1.00
##
## Population-Level Effects:
##                               Estimate Est.Error l-95% CI
## conditionHOPs                0.70     0.38    -0.04
## conditionintervals_w_means   0.24     0.37    -0.51
## conditionmeans_only           0.86     0.38    0.10
## frameloss                     -0.47    0.13    -0.72
## conditionHOPs:framegain:evidence  0.72     0.16    0.42
## conditionintervals_w_means:framegain:evidence  0.59     0.16    0.27
## conditionmeans_only:framegain:evidence  0.53     0.15    0.24
## conditionHOPs:frameloss:evidence        0.58     0.16    0.29
## conditionintervals_w_means:frameloss:evidence  0.18     0.15    -0.11
## conditionmeans_only:frameloss:evidence        0.43     0.15    0.13
##                               u-95% CI Eff.Sample Rhat
## conditionHOPs                 1.46      738 1.00
## conditionintervals_w_means   0.96      667 1.00
## conditionmeans_only           1.58      599 1.00
## frameloss                     -0.23     4883 1.00
## conditionHOPs:framegain:evidence  1.05     1894 1.00
## conditionintervals_w_means:framegain:evidence  0.90     1949 1.00
## conditionmeans_only:framegain:evidence  0.84     1616 1.00
## conditionHOPs:frameloss:evidence        0.90     1774 1.00
## conditionintervals_w_means:frameloss:evidence  0.49     1782 1.00
## conditionmeans_only:frameloss:evidence        0.72     1636 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 %>%
  group_by(worker_id, condition, evidence) %>%
  add_predicted_draws(m.vis.frame.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())
```

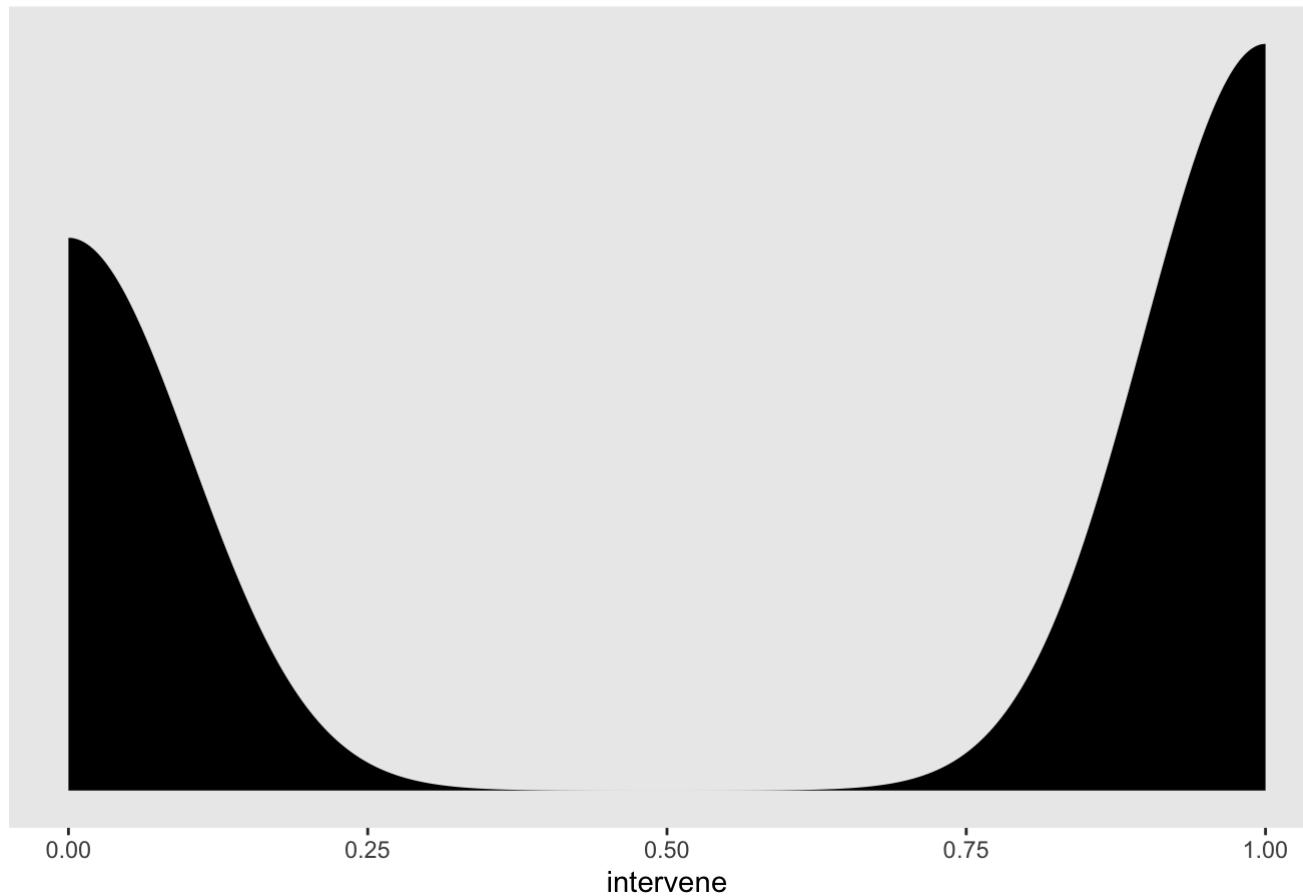
Posterior predictive distribution for intervention



How do the posterior predictions compare to the observed data?

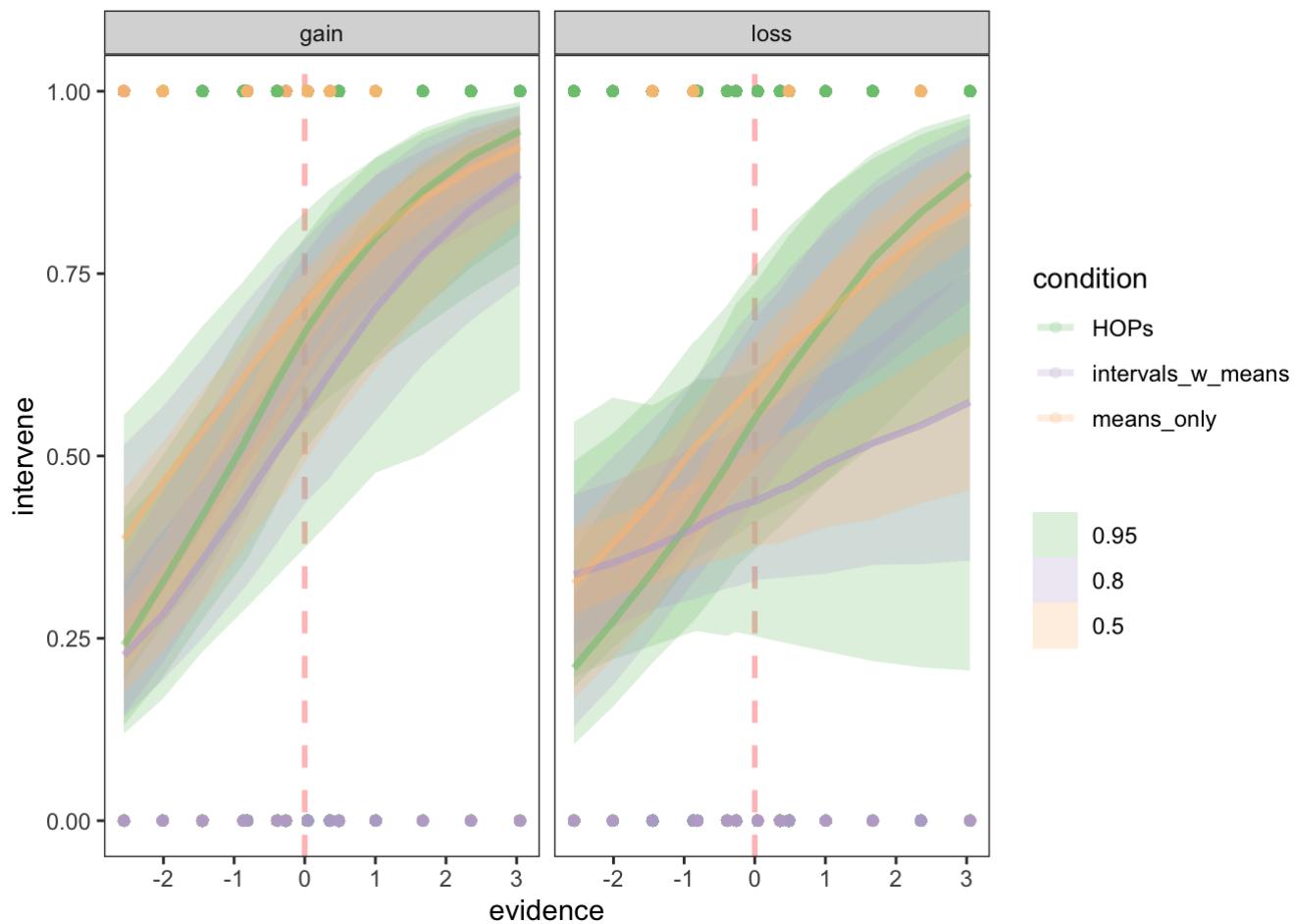
```
# data density
model_df %>%
  ggplot(aes(x = intervene)) +
  geom_density(fill = "black", size = 0) +
  scale_y_continuous(NULL, breaks = NULL) +
  labs(subtitle = "Data distribution for intervention") +
  theme(panel.grid = element_blank())
```

### Data distribution for intervention



Let's take a look at the estimated psychometric function for the average observer in each visualization condition and each level of problem framing.

```
model_df %>%
  group_by(evidence, condition, frame, worker_id) %>%
  add_fitted_draws(m.vis.frame.wrkr.logistic, value = "pf", re_formula = NA, n = 200) %>%
  ggplot(aes(x = evidence, y = intervene, color = 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, .80, .50), 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_grid(. ~ frame)
```



Here we see a slight difference in psychometric functions for visualization conditions across problem frames, with more convergence of performance in the loss frame.

## Model Comparison

Let's check which of these two hierarchical models, with and without framing as a predictor, 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.vis.wrkr.logistic, m.vis.frame.wrkr.logistic)
```

	WAIC	SE
## m.vis.wrkr.logistic	1651.41	38.57
## m.vis.frame.wrkr.logistic	1637.27	38.78
## m.vis.wrkr.logistic - m.vis.frame.wrkr.logistic	14.14	9.77

The model with the lowest WAIC value (i.e., the best fitting model) is the one with predictors for visualization condition and problem framing. The importance of problem framing is consistent with prior work in behavioral economics.