Supplemental Analyses

Jennah Gosciak

May 16, 2022

```
## load data
df <- read_dta("../00_data/sample1.dta")
df_samp <- df %>%
  sample_n(1000)

df_samp <- df_samp %>%
  mutate(across(c("age", "age_fbirth"), ~ . - mean(., na.rm = T))) %>%
  mutate(l_incwage = if_else(incwage <= 0, log(1), log(incwage)),
         l_wkswork1 = if_else(wkswork1 <= 0, log(1), log(wkswork1)),
         incwage_mod = if_else(incwage <= 0, 1, incwage))

df_samp %>%
  group_by(samesex) %>%
  summarize(n = n())
```

```
## # A tibble: 2 x 2
##   samesex     n
##     <dbl> <int>
## 1       0   473
## 2       1   527
```

Model with Gamma distribution

- Uses log link
- Similar to implementation in `rstanarm`

```
writeLines(readLines("linear_gamma.stan"))
```

```
## #include quantile_functions.stan
## data {
##   int<lower = 0> N; // number of observations
##   int<lower = 0> K; // number of predictors
##   matrix[N, K] X;   // matrix of predictors
##   vector[N] y;      // outcomes
##   int<lower = 0, upper = 1> prior_only;   // ignore data?
##   vector[K + 1] m;                        // prior medians
##   vector<lower = 0>[K + 1] scale;             // prior IQRs
##   real r;
## }
## parameters {
```

```
##   vector[K] beta;
##   real alpha;
##   real<lower = 0> shape;
## }
##
## transformed parameters {
##   vector[N] mu = alpha + X * beta;
## }
##
## model { // log likelihood, equivalent to target += normal_lpdf(y | alpha + X * beta, sigma)
##   if (!prior_only) {
##     for (i in 1:N) target += gamma_lpdf(y | shape, shape/exp(mu[i]));
##   }
##   target += normal_lpdf(alpha | m[1],   scale[1]);
##   target += normal_lpdf(beta | m[2:K + 1], scale[2:K + 1]);
##   target += exponential_lpdf(shape | r); // exponential
## }
##
## generated quantities {
##   vector[N] log_lik;
##   vector[N] yrep;
##   {
##     for (n in 1:N) {
##       log_lik[n] = gamma_lpdf(y[n] | shape, shape / exp(mu[n]));
##       yrep[n] = gamma_rng(shape, shape / exp(mu[n]));
##     }
##   }
## }
```

```
# use normal priors
m <- rep(-0.1, 9)
s<- rep(1, 9)

stan_data <- list(N = nrow(df_samp), K = 8,
                                        y = df_samp$incwage,
                                        X = df_samp[, c("cnum_mt2", "age", "age_fbirth", "f_boy",
                                                  "s_boy", "r_black", "hisp", "r_oth")],
                                        prior_only = TRUE, m = m,
                                        scale = s, r = 0.5)
```

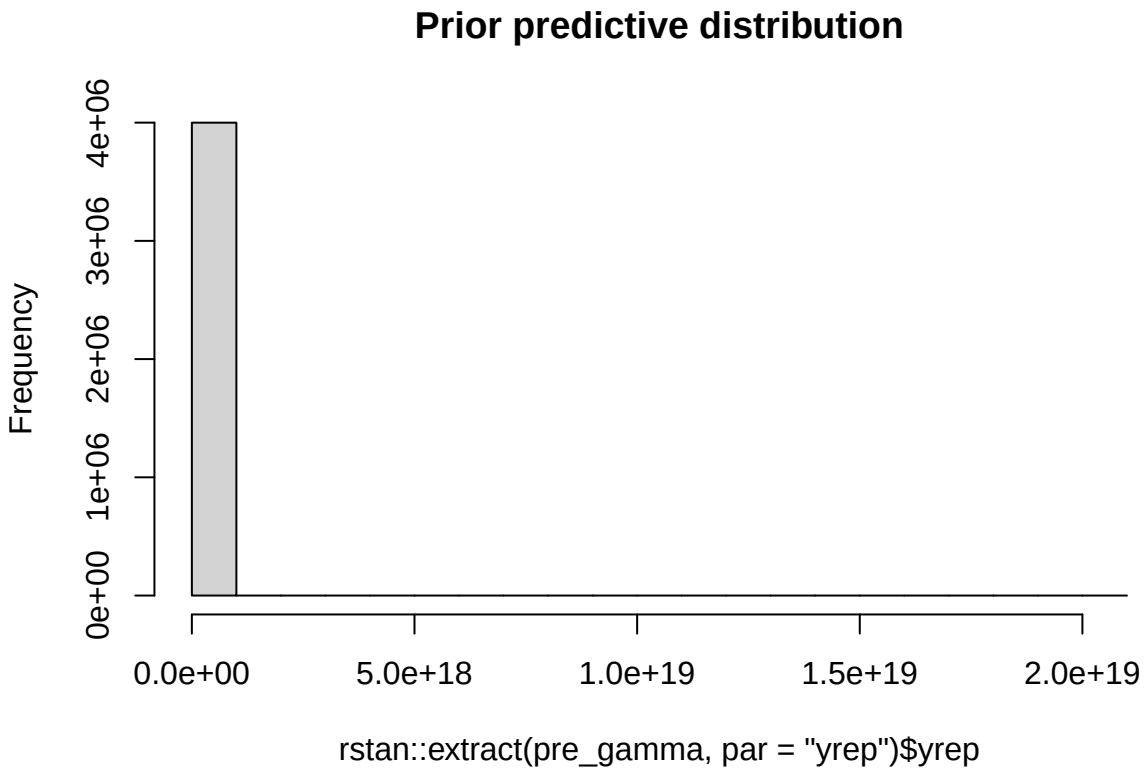
```
pre_gamma <- stan("linear_gamma.stan", data = stan_data, seed = 12345)
```

```
# print output
print(pre_gamma, pars = c("alpha", "beta", "shape"))
```

```
## Inference for Stan model: linear_gamma.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean   sd  2.5%   25%   50%  75% 97.5% n_eff Rhat
## alpha   -0.11    0.01 1.01 -2.05 -0.79 -0.13 0.57  1.86  7895    1
## beta[1] -0.08    0.01 0.99 -2.02 -0.73 -0.10 0.59  1.90  7723    1
## beta[2] -0.10    0.01 0.99 -1.99 -0.75 -0.11 0.56  1.86  8044    1
```

```
## beta[3] -0.11    0.01 1.01 -2.06 -0.80 -0.12 0.58  1.86  9162    1
## beta[4] -0.10    0.01 1.01 -2.08 -0.78 -0.11 0.58  1.85  8555    1
## beta[5] -0.12    0.01 1.01 -2.17 -0.78 -0.13 0.54  1.89  8612    1
## beta[6] -0.10    0.01 1.00 -2.09 -0.78 -0.09 0.57  1.86  7752    1
## beta[7] -0.09    0.01 1.02 -2.11 -0.79 -0.08 0.61  1.91  8154    1
## beta[8] -0.11    0.01 1.02 -2.10 -0.80 -0.11 0.59  1.92  6893    1
## shape    1.99    0.02 1.97  0.05  0.56  1.38 2.78  7.16  7238    1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 22:36:55 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
hist(rstan::extract(pre_gamma, par = "yrep")$yrep,
     main = "Prior predictive distribution")
```




```
stan_data$prior_only <- FALSE
```

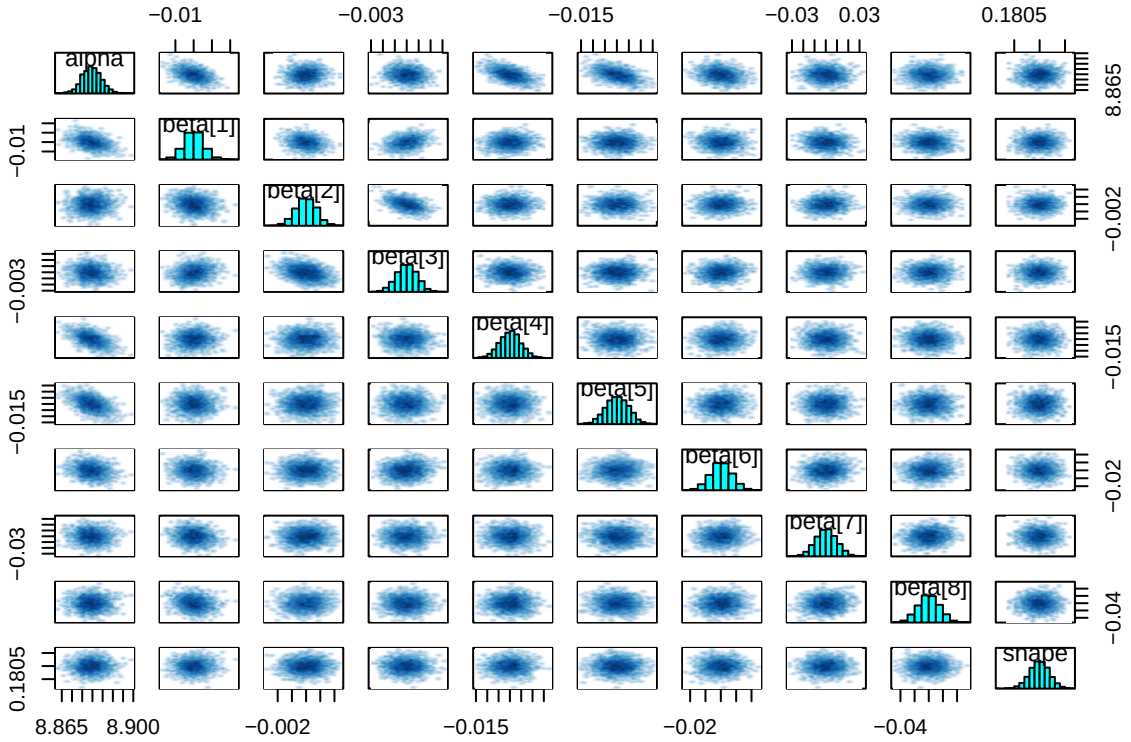
```
post_gamma <- stan("linear_gamma.stan", data = stan_data,
                                        seed = 12345)
```

```
# print output
print(post_gamma, pars = c("alpha", "beta", "shape"))
```

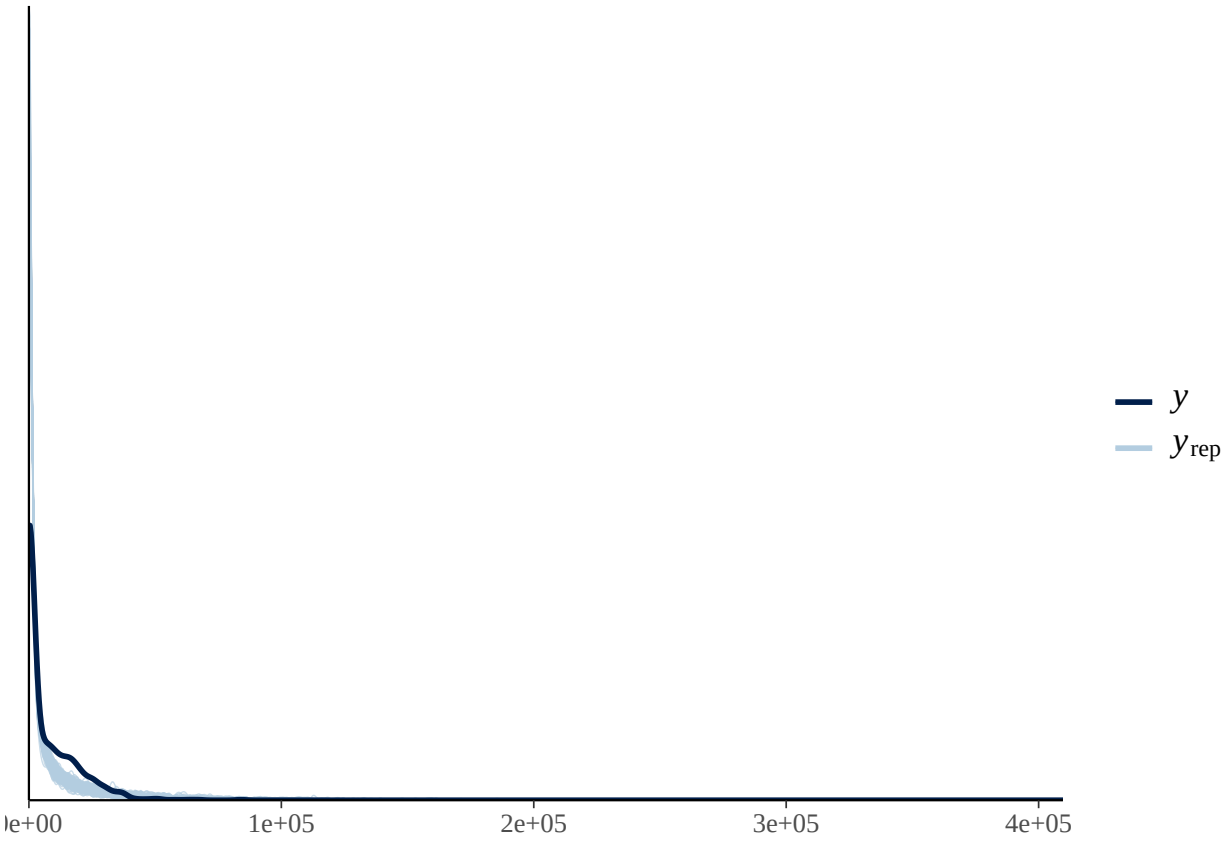
```
## Inference for Stan model: linear_gamma.
```

```
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##         mean se_mean   sd  2.5%   25%  50%  75% 97.5% n_eff Rhat
## alpha   8.88       0 0.00  8.87  8.88 8.88 8.88  8.89  2327    1
## beta[1] 0.00       0 0.01 -0.01  0.00 0.00 0.00  0.01  3360    1
## beta[2] 0.00       0 0.00  0.00  0.00 0.00 0.00  0.00  4100    1
## beta[3] 0.00       0 0.00  0.00  0.00 0.00 0.00  0.00  4644    1
## beta[4] 0.00       0 0.00 -0.01  0.00 0.00 0.00  0.01  3130    1
## beta[5] 0.00       0 0.00 -0.01  0.00 0.00 0.00  0.01  3274    1
## beta[6] 0.00       0 0.01 -0.01  0.00 0.00 0.00  0.01  3250    1
## beta[7] 0.00       0 0.01 -0.02 -0.01 0.00 0.01  0.02  3238    1
## beta[8] 0.00       0 0.02 -0.03 -0.01 0.00 0.01  0.03  3406    1
## shape   0.18       0 0.00  0.18  0.18 0.18 0.18  0.18  4890    1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 23:39:00 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
pairs(post_gamma, pars = c("alpha", "beta", "shape"))
```




```
pp_check(as.numeric(stan_data$y),
  rstan::extract(post_gamma, par = "yrep")$yrep[sample(1:length(stan_data$y), size = 150), ],
  ppc_dens_overlay
)
```




```
loo_gamma <- loo(post_gamma)
loo_gamma
```

```
##
## Computed from 4000 by 1000 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -7618.3 125.7
## p_loo         0.0   0.0
## looic     15236.6 251.5
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

Hierarchical linear model

- Allows intercepts to shift based on state

```
df_samp <- df %>%
  sample_n(3000)

df_samp <- df_samp %>%
```

```
  mutate(across(c("age", "age_fbirth"), ~ . - mean(., na.rm = T))) %>%
  mutate(l_incwage = if_else(incwage <= 0, log(1), log(incwage)),
         l_wkswork1 = if_else(wkswork1 <= 0, log(1), log(wkswork1)),
         incwage_mod = if_else(incwage <= 0, 1, incwage))
```

```
# set generic priors
m <- rep(0, 9)
s<- rep(1, 9)

# set states as ordered factor
states <- as.integer(as.factor(df_samp$statefip))


stan_data <- list(N = nrow(df_samp), K = 8, J = 51,
                                        states = states,
                                        y = df_samp$l_incwage,
                                        X = df_samp[, c("cnum_mt2", "age", "age_fbirth", "f_boy",
                                                  "s_boy", "r_black", "hisp", "r_oth")],
                                        prior_only = FALSE, m = m,
                                        scale = s, r = 1)
```

```
post_mlm <- stan("linear_mlm.stan",
                 data = stan_data, seed = 1234)
```

```
print(post_mlm, pars = c("beta"))
```

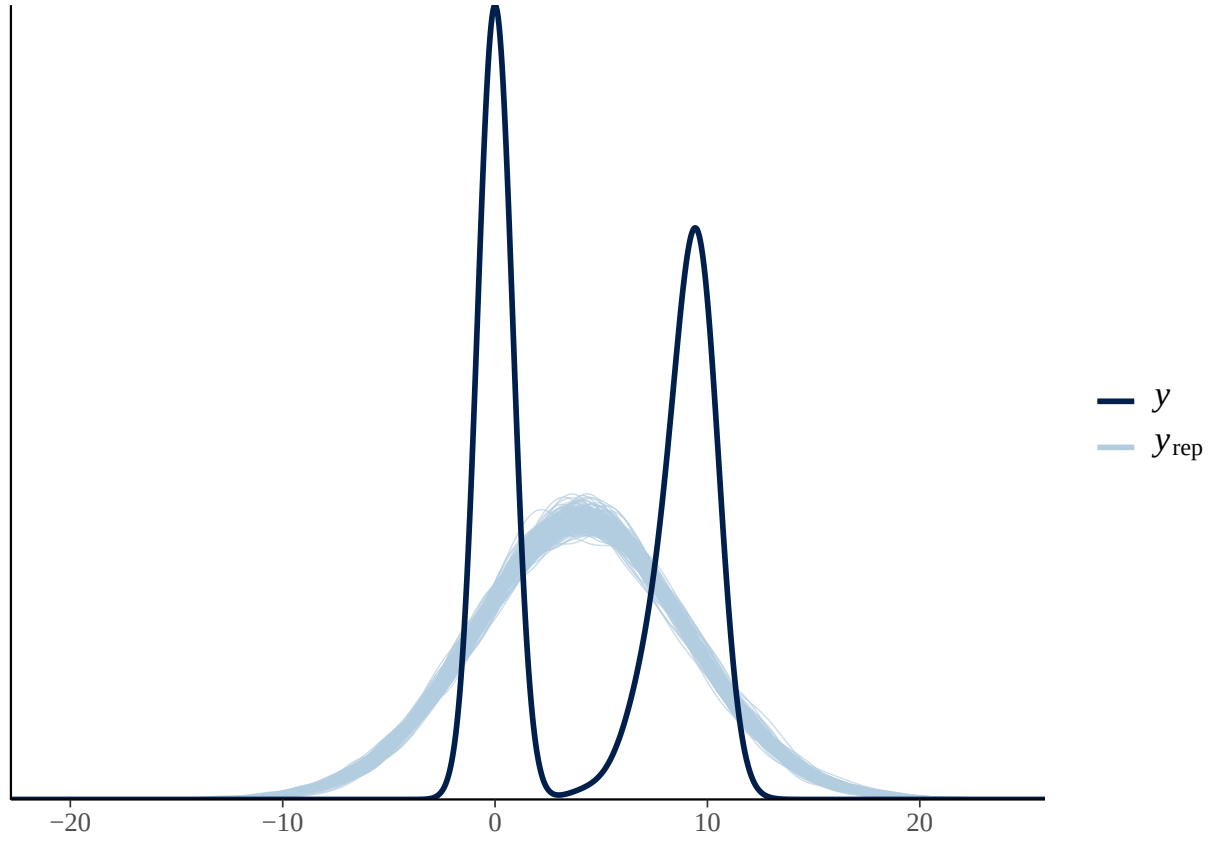
```
## Inference for Stan model: linear_mlm.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta[1] -0.19    0.00 0.17 -0.54 -0.31 -0.19 -0.08  0.15  4853    1
## beta[2]  0.19    0.00 0.03  0.14  0.17  0.19  0.21  0.24  5707    1
## beta[3] -0.18    0.00 0.03 -0.25 -0.21 -0.18 -0.16 -0.12  4128    1
## beta[4]  1.41    0.00 0.16  1.10  1.30  1.41  1.51  1.71  4755    1
## beta[5]  1.51    0.00 0.16  1.20  1.41  1.52  1.62  1.81  4777    1
## beta[6]  1.94    0.00 0.26  1.42  1.76  1.93  2.12  2.47  5415    1
## beta[7]  0.56    0.00 0.30 -0.03  0.36  0.56  0.77  1.14  6459    1
## beta[8]  0.47    0.01 0.51 -0.53  0.13  0.46  0.80  1.49  6680    1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 23:42:17 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
loo_mlm <- loo(post_mlm)
loo_mlm
```

```
##
## Computed from 4000 by 3000 log-likelihood matrix
##
```

```
##          Estimate   SE
## elpd_loo  -8870.2 20.1
## p_loo        38.8  0.8
## looic     17740.3 40.2
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
pp_check(as.numeric(stan_data$y),
  rstan::extract(post_mlm, par = "yrep")$yrep[sample(1:length(stan_data$y), size = 150), ],
  ppc_dens_overlay
)
```



Hierarchical IV model

```
# set generic priors
m <- rep(0, 9)
s<- rep(1, 9)

# set states as ordered factor
df_samp$states <- as.integer(as.factor(df_samp$statefip))
```

```r
# subset the data
df_child <- df_samp %>%
  filter(cnum_mt2 == 1)

df_nochild <- df_samp %>%
  filter(cnum_mt2 == 0)

# reset covariates
cov <- c("age", "age_fbirth", "f_boy", "s_boy", "r_black", "hisp", "r_oth")

# set stan data
stan_data_iv <- list(N = nrow(df_samp),
                     N_child = nrow(df_child),
                     N_nochild = nrow(df_nochild),
                     K = 7,
                     J = 51,
                     states_child = df_child$states,
                     states_nochild = df_nochild$states,
                     X_child_s = df_child[, c("samesex", cov)],
                     X_nochild_s = df_nochild[, c("samesex", cov)],
                     X_child = df_child[, c(cov)],
                     X_nochild = df_nochild[, c(cov)],
                     y_child = df_child$l_incwage,
                     y_nochild = df_nochild$l_incwage,
                     prior_only = FALSE,
                     m = rep(-0.1, 5),
                     scale = rep(0.3, 5))
```

```r
post_iv_mlm <- stan("iv_bin_mlm.stan",
                    data = stan_data_iv, seed = 1234)
```

```r
print(post_iv_mlm, pars = c("beta1", "beta2"))
```

```
## Inference for Stan model: iv_bin_mlm.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd  2.5%   25%   50%  75% 97.5% n_eff Rhat
## beta1[1]  0.07       0 0.03  0.00  0.05  0.07 0.09  0.13  6731    1
## beta1[2]  0.02       0 0.04 -0.05 -0.01  0.02 0.04  0.09  5725    1
## beta1[3]  1.61       0 0.15  1.32  1.52  1.61 1.71  1.90  6807    1
## beta1[4]  1.69       0 0.15  1.39  1.59  1.69 1.79  1.98  7486    1
## beta1[5]  1.01       0 0.22  0.59  0.86  1.01 1.17  1.42  7690    1
## beta1[6]  0.12       0 0.23 -0.35 -0.03  0.12 0.28  0.58  8372    1
## beta1[7] -0.01       0 0.27 -0.53 -0.19 -0.01 0.17  0.53  9975    1
## beta2     4.13       0 0.20  3.74  4.00  4.13 4.27  4.52  4213    1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 23:47:01 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

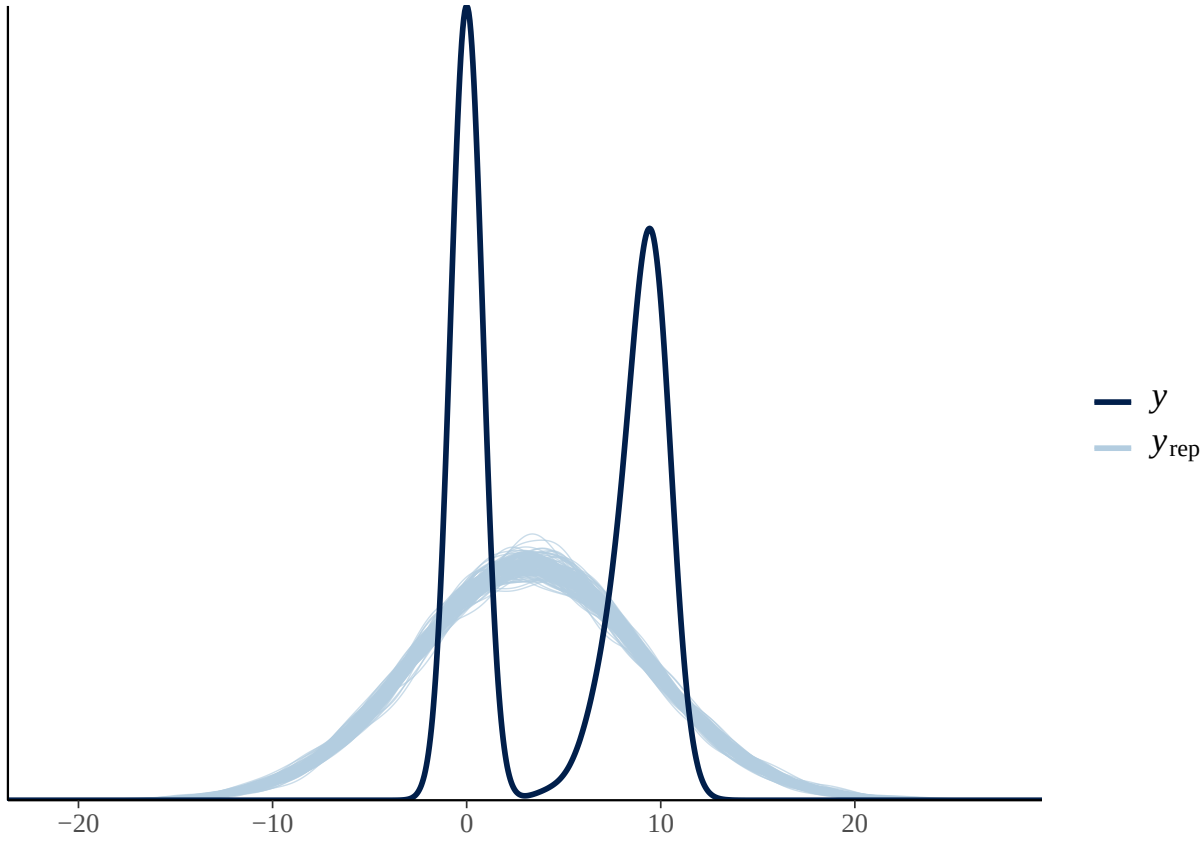
```
loo_iv_mlm <- loo(post_iv_mlm)
loo_iv_mlm
```

```
##
## Computed from 4000 by 3000 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo -10720.2 26.8
## p_loo        55.3  1.4
## looic     21440.4 53.6
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_compare(loo_iv_mlm, loo_mlm)
```

```
##        elpd_diff se_diff
## model2     0.0       0.0
## model1 -1850.0      33.7
```

```
pp_check(c(as.numeric(stan_data_iv$y_child),
           as.numeric(stan_data_iv$y_nochild)),
  rstan::extract(post_iv_mlm, par = "yrep")$yrep[sample(1:nrow(df_samp),
                                                        size = 150), ],
  ppc_dens_overlay
)
```


−20
−10
0
10
20
y
y rep