

Supplemental Analyses

Jennah Gosciak

May 17, 2022

Contents

Analysis	1
Model with Gamma distribution	1
Hierarchical linear model	9
Hierarchical IV model	12
Conclusion	17

This PDF uses the same data as final_project.pdf, but uses the Gamma distribution with a log link and a hierarchical model where intercepts vary by state.

```
## load data
df <- read_dta("../00_data/sample1.dta")
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))

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

## # A tibble: 2 x 2
##   samesex     n
##   <dbl> <int>
## 1      0  1434
## 2      1  1566
```

Analysis

Model with Gamma distribution

- I explored using the gamma distribution with a log link function since the income data is right-skewed and positive.

- First, I tested out the program on a slightly larger sample using `rstanarm`.
- Then, I ran a smaller sample on a stan program I wrote myself, which is a simpler and slower version of the implementation in `rstanarm`. Because it takes so long to run, I only ran the model on 1,000 observations. The 95% credible intervals for the coefficients all contain 0 and consequently are not conclusive about any non-zero effect.

Using `rstanarm` for the Gamma distribution with log link

```

## running with rstanarm
post_gamma_rs <- stan_glm(incwage_mod ~ cnum_mt2 + age + age_fbirth + f_boy +
                           s_boy + r_black + hisp + r_oth, data = df_samp,
                           family = Gamma(link = "log"),
                           prior_intercept = normal(10, 1),
                           prior = normal(0, 1),
                           seed = 12345)

post_gamma_rs

## stan_glm
## family:      Gamma [log]
## formula:     incwage_mod ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black +
##               hisp + r_oth
## observations: 3000
## predictors:   9
## -----
##           Median MAD_SD
## (Intercept) 9.0    0.1
## cnum_mt2   -0.6    0.1
## age        0.1    0.0
## age_fbirth -0.1    0.0
## f_boy       0.0    0.1
## s_boy       0.0    0.1
## r_black    0.5    0.1
## hisp        0.0    0.2
## r_oth       0.3    0.3
##
## Auxiliary parameter(s):
##           Median MAD_SD
## shape 0.2    0.0
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg

loo_gamma <- loo(post_gamma_rs)
loo_gamma

##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate    SE

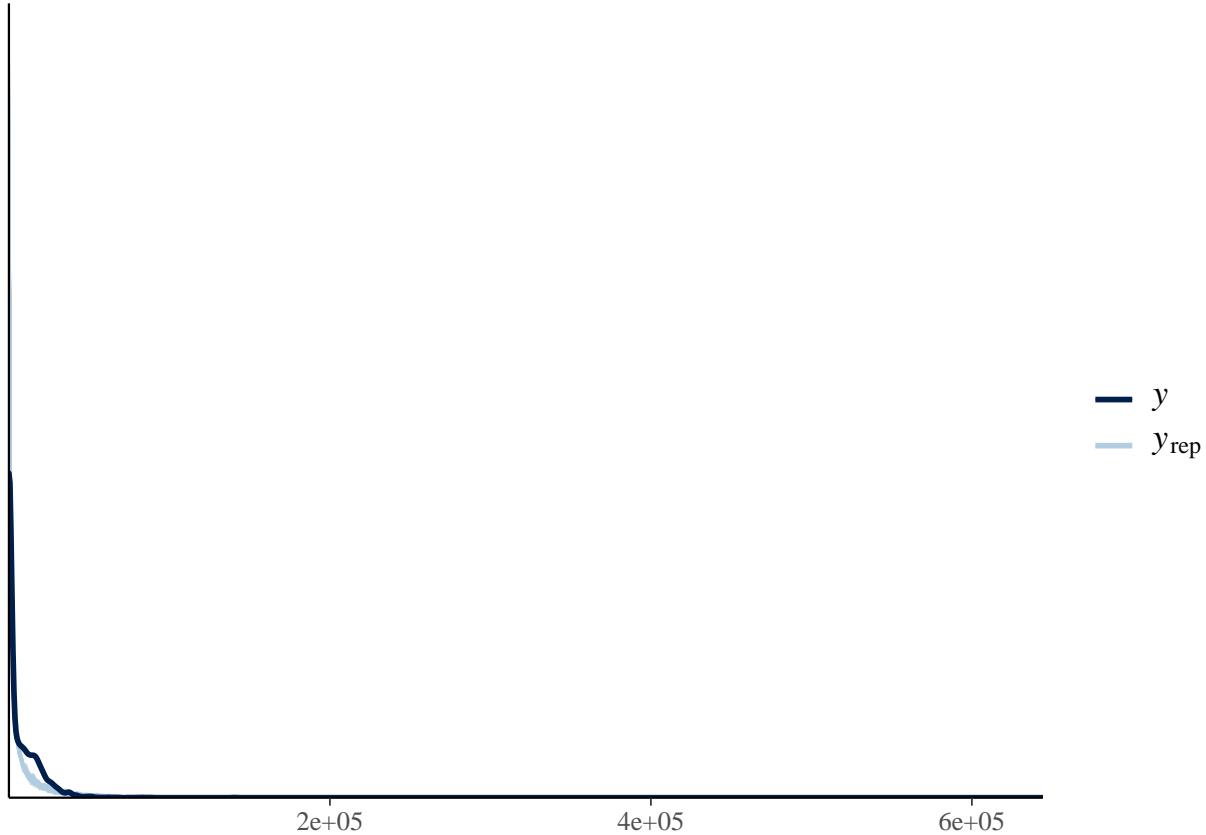
```

```

## elpd_loo -22810.5 216.0
## p_loo      5.2   0.8
## looic     45621.1 432.0
## -----
## 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.

pp_check(post_gamma_rs,
  plotfun = "ppc_dens_overlay"
)

```



There are no Pareto k values > 0.5 , which indicates that importance sampling was able to estimate the `elpd_loo` accurately. Additionally, $p_{\text{loo}} < p$, which doesn't indicate any potential issues with model misspecification. The `elpd_loo` is small. Comparing the `elpd_loo` will allow for comparisons.

The posterior predictive distribution does appear to be a less smooth version of the observed values of y . However, since the distribution is so skewed, it's difficult to visually inspect the overlap at the extreme ends of the distribution.

Alternative approach using custom stan program

- The estimated coefficient values vary slightly from the above estimates generated by `rstanarm`. I include them for comparison.

```

## make sample smaller for running with user written stan program
df_samp2 <- df %>%
  sample_n(1000)

df_samp2 <- df_samp2 %>%
  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))

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<lower = 0>[N] y; // outcomes
##   int<lower = 0, upper = 1> prior_only; // ignore data?
##   vector[K + 1] m; // prior mean values
##   vector<lower = 0>[K + 1] scale; // prior scale values
##   real r;
## }
## parameters {
##   vector[K] beta;
##   real alpha;
##   real<lower = 0> shape; // shape cannot be less than 0
## }
##
## transformed parameters {
##   vector[N] mu = alpha + X * beta;
## }
##
## model { // log likelihood
##   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 for shape param
## }
##
## 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 <- c(10, rep(-0.1, 8))
s<- rep(1, 9)

stan_data <- list(N = nrow(df_samp2), K = 8,
                   y = df_samp2$incwage_mod,
                   X = df_samp2[, c("cnum_mt2", "age",
                                    "age_fbirth", "f_boy",
                                    "s_boy", "r_black",
                                    "hisp", "r_oth")],
                   prior_only = TRUE, m = m,
                   scale = s, r = 1)

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

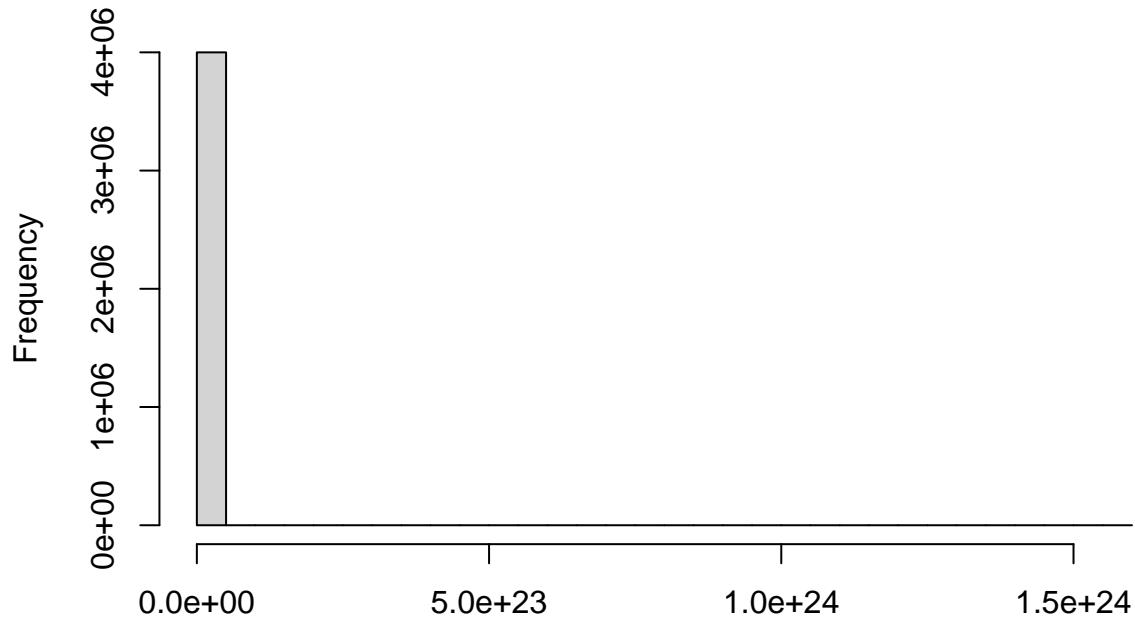
# print output
print(pre_gamma, pars = c("alpha", "beta", "shape"))

## 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    10.01    0.01 1.01  8.06  9.30 10.01 10.71 11.93  7191     1
## beta[1] -0.08    0.01 1.02 -2.10 -0.76 -0.10  0.59  1.96  7221     1
## beta[2] -0.10    0.01 1.00 -2.04 -0.77 -0.08  0.58  1.83  7847     1
## beta[3] -0.09    0.01 1.00 -2.08 -0.77 -0.10  0.60  1.86  7383     1
## beta[4] -0.09    0.01 1.01 -2.06 -0.80 -0.09  0.58  1.86  7796     1
## beta[5] -0.11    0.01 1.00 -2.05 -0.79 -0.11  0.55  1.86  7596     1
## beta[6] -0.09    0.01 1.00 -2.06 -0.76 -0.09  0.59  1.85  7814     1
## beta[7] -0.12    0.01 1.01 -2.14 -0.80 -0.12  0.56  1.88  8364     1
## beta[8] -0.10    0.01 0.98 -2.00 -0.75 -0.09  0.57  1.78  7993     1
## shape     1.00    0.01 1.02  0.02  0.27  0.69  1.36  3.77  6088     1
##
## Samples were drawn using NUTS(diag_e) at Tue May 17 14:29:58 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")

```

Prior predictive distribution



```
rstan::extract(pre_gamma, par = "yrep")$yrep
```

The prior predictive distribution generates large, extreme values of y . At the same time, most observations are clustered toward 0.

```
# set prior to false, run the full dist
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     8.87     0 0.00  8.87  8.87  8.88  8.88  2724     1
## beta[1]  0.00     0 0.00 -0.01  0.00  0.00  0.00  0.01 3609     1
## beta[2]  0.00     0 0.00  0.00  0.00  0.00  0.00  0.00 4438     1
## beta[3]  0.00     0 0.00  0.00  0.00  0.00  0.00  0.00 4235     1
## beta[4]  0.00     0 0.00 -0.01  0.00  0.00  0.00  0.01 3524     1
## beta[5]  0.00     0 0.00 -0.01  0.00  0.00  0.00  0.01 3659     1
## beta[6]  0.00     0 0.01 -0.01 -0.01  0.00  0.00  0.01 2955     1
## beta[7]  0.00     0 0.01 -0.02 -0.01  0.00  0.01  0.02 3424     1
## beta[8]  0.00     0 0.01 -0.03 -0.01  0.00  0.01  0.03 3244     1
## shape    0.18     0 0.00  0.18  0.18  0.18  0.19  0.19 5204     1
##
```

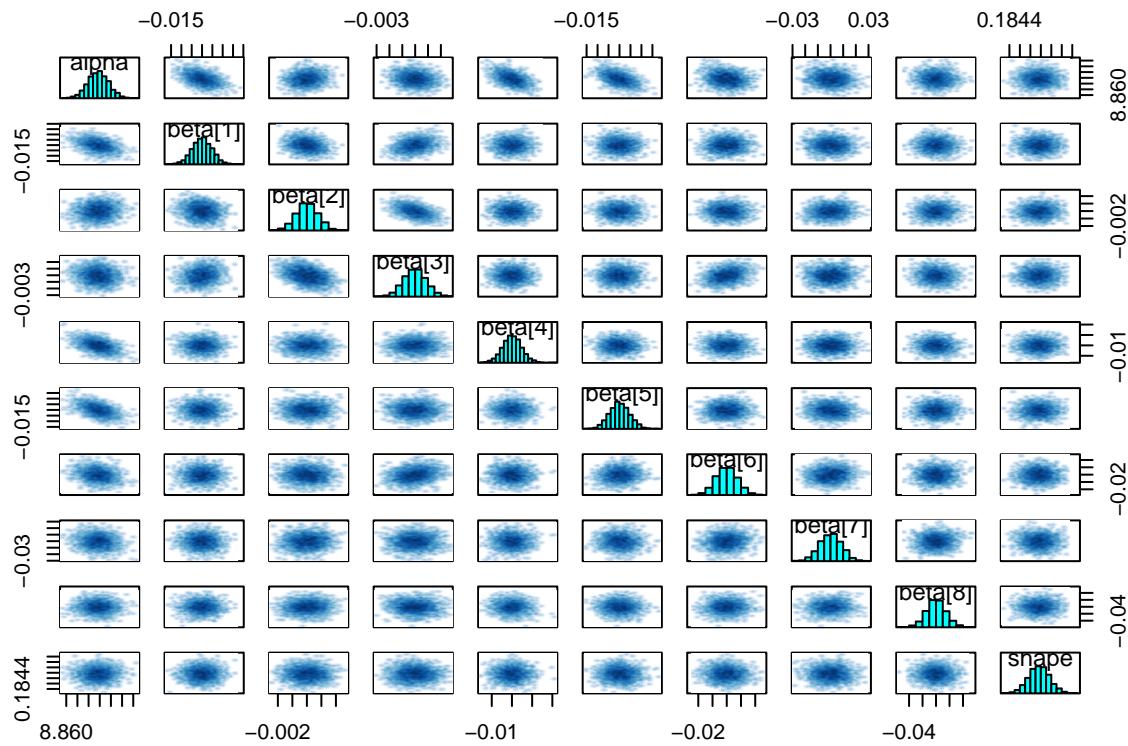
```

## Samples were drawn using NUTS(diag_e) at Tue May 17 15:22:22 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).

```

The posterior distribution when using a log link and the gamma distribution is not conclusive about the effect of additional children on earnings. The coefficient on `cnum_mt2` is on average 0 and the 95% credible interval is between -0.01 and 0.01. Almost all the predictors have values close to 0. The small values for the coefficients, as indicated by the marginal distributions, is likely due to the conversion from log earnings to earnings in dollars.

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

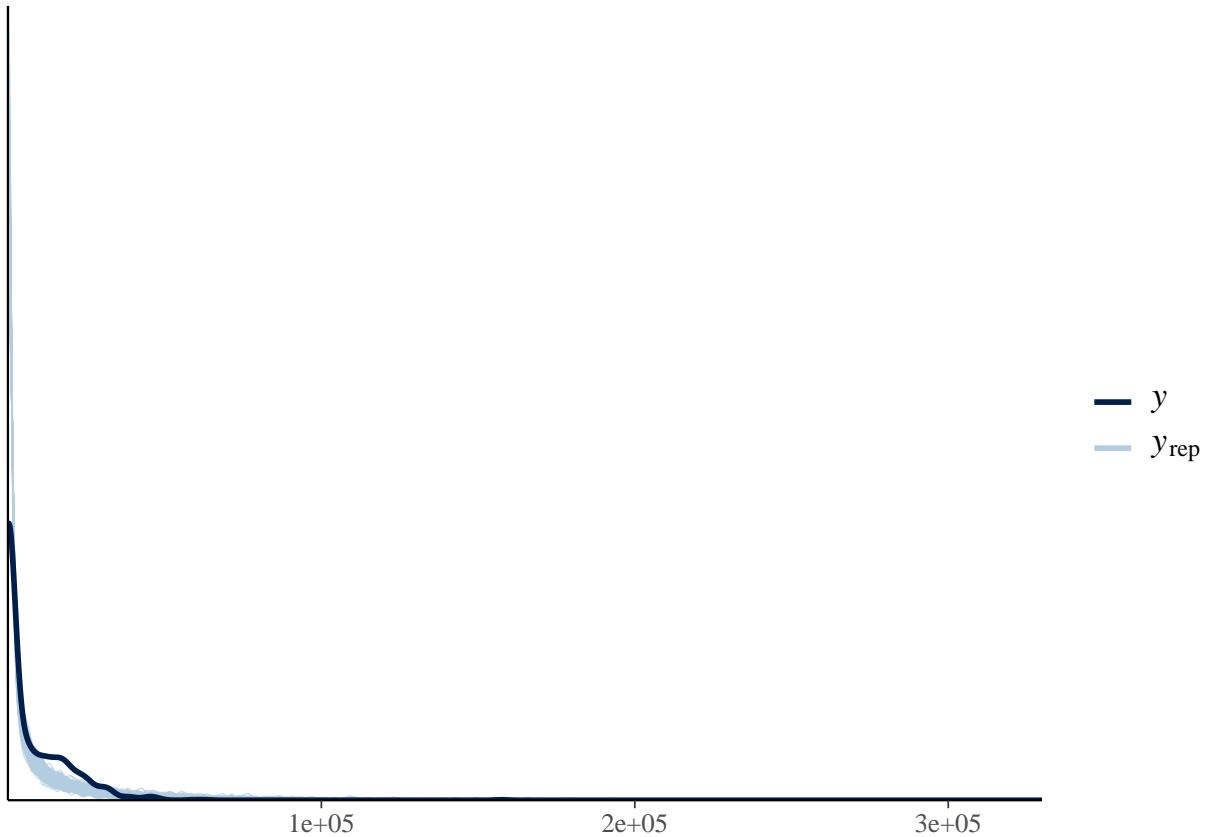


The marginal distribution of the parameters all seem approximately normal, even for shape.

```

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
)

```



The posterior predictive distribution does appear to closely follow the observed values of y , which are less smooth. However, it's difficult to tell in the tails if the observed values and the predicted values diverge.

```
# comparison of predictions
yrep <- rstan::extract(post_gamma, "yrep")[[1]]
low <- apply(yrep, MARGIN = 2, FUN = quantile, probs = 1 / 3)
high <- apply(yrep, MARGIN = 2, FUN = quantile, probs = 2 / 3)

y <- stan_data$y

c(too_low = mean(y < low),
  just_right = mean(y > low & y < high),
  too_high = mean(y > high))

##      too_low just_right   too_high
##        0.461       0.102      0.437
```

The model is predicting too many extreme values (bottom and top third percentiles). Only 8% of observed y -values fall in only the middle third percentile.

```
loo_gamma2 <- loo(post_gamma)
loo_gamma2

##
## Computed from 4000 by 1000 log-likelihood matrix
```

```

##          Estimate    SE
## elpd_loo  -7696.0 124.3
## p_loo      0.0   0.0
## looic     15392.0 248.5
## -----
## 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.

```

All the Pareto k values are less than 0.5, which indicates that importance sampling can generate a good estimate. However, `p_loo` << `p`, which might be evidence that the model is misspecified.

Hierarchical linear model

- Allowing intercepts to shift based on state
- Replicates the linear model and the IV model

```

# 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,
                    # use fewer predictors
                    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("alpha", "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
## alpha[1]  1.64    0.01 0.55  0.56  1.28  1.64  2.00  2.70  3921    1
## alpha[2]  0.87    0.01 0.87 -0.81  0.29  0.87  1.46  2.59  5300    1
## alpha[3]  2.18    0.01 0.61  0.98  1.76  2.18  2.61  3.39  5403    1
## alpha[4]  1.35    0.01 0.63  0.12  0.91  1.34  1.79  2.57  5178    1

```

## alpha[5]	2.62	0.00	0.30	2.02	2.41	2.62	2.82	3.21	4402	1
## alpha[6]	2.91	0.01	0.61	1.71	2.49	2.91	3.33	4.09	5289	1
## alpha[7]	2.10	0.01	0.64	0.83	1.67	2.10	2.53	3.31	5935	1
## alpha[8]	-0.04	0.01	0.88	-1.77	-0.63	-0.05	0.54	1.63	6069	1
## alpha[9]	0.09	0.01	0.93	-1.70	-0.55	0.10	0.70	1.93	5811	1
## alpha[10]	2.85	0.01	0.44	1.99	2.56	2.85	3.15	3.70	4563	1
## alpha[11]	3.25	0.01	0.51	2.24	2.91	3.25	3.60	4.24	4587	1
## alpha[12]	1.11	0.01	0.76	-0.41	0.60	1.11	1.62	2.59	5465	1
## alpha[13]	0.75	0.01	0.81	-0.87	0.21	0.75	1.30	2.34	5116	1
## alpha[14]	2.35	0.01	0.36	1.66	2.10	2.34	2.58	3.06	4596	1
## alpha[15]	3.23	0.01	0.41	2.41	2.96	3.23	3.51	4.04	4663	1
## alpha[16]	2.65	0.01	0.53	1.62	2.30	2.66	3.01	3.70	4327	1
## alpha[17]	2.48	0.01	0.66	1.20	2.04	2.49	2.93	3.73	5711	1
## alpha[18]	1.90	0.01	0.56	0.83	1.52	1.90	2.26	3.00	5033	1
## alpha[19]	1.71	0.01	0.51	0.70	1.38	1.71	2.05	2.71	5212	1
## alpha[20]	1.25	0.01	0.73	-0.16	0.76	1.25	1.74	2.65	4777	1
## alpha[21]	2.69	0.01	0.58	1.60	2.30	2.67	3.09	3.82	5348	1
## alpha[22]	3.31	0.01	0.52	2.31	2.97	3.31	3.67	4.33	4925	1
## alpha[23]	2.39	0.01	0.35	1.70	2.16	2.39	2.63	3.09	4730	1
## alpha[24]	3.18	0.01	0.51	2.16	2.85	3.18	3.53	4.18	5368	1
## alpha[25]	2.70	0.01	0.59	1.53	2.30	2.69	3.09	3.88	5268	1
## alpha[26]	3.50	0.01	0.47	2.61	3.17	3.48	3.82	4.44	4881	1
## alpha[27]	0.63	0.01	0.79	-0.93	0.12	0.64	1.15	2.23	4846	1
## alpha[28]	1.47	0.01	0.67	0.15	1.01	1.47	1.92	2.75	6044	1
## alpha[29]	1.43	0.01	0.82	-0.17	0.89	1.44	1.96	3.00	5362	1
## alpha[30]	1.53	0.01	0.85	-0.13	0.95	1.53	2.11	3.18	5238	1
## alpha[31]	1.82	0.01	0.48	0.88	1.49	1.82	2.15	2.75	4789	1
## alpha[32]	1.41	0.01	0.72	0.02	0.92	1.39	1.88	2.84	4429	1
## alpha[33]	1.75	0.00	0.33	1.11	1.53	1.75	1.97	2.38	4391	1
## alpha[34]	2.55	0.01	0.49	1.60	2.21	2.54	2.89	3.50	4392	1
## alpha[35]	0.58	0.01	0.86	-1.10	0.02	0.57	1.17	2.26	5194	1
## alpha[36]	2.49	0.01	0.37	1.76	2.24	2.50	2.75	3.20	4087	1
## alpha[37]	3.05	0.01	0.58	1.93	2.66	3.04	3.44	4.19	4820	1
## alpha[38]	1.89	0.01	0.62	0.66	1.49	1.90	2.30	3.14	5916	1
## alpha[39]	2.33	0.01	0.36	1.63	2.08	2.33	2.57	3.03	5081	1
## alpha[40]	1.35	0.01	0.86	-0.33	0.76	1.36	1.94	2.98	4387	1
## alpha[41]	2.59	0.01	0.59	1.41	2.20	2.60	2.99	3.73	5391	1
## alpha[42]	0.55	0.01	0.84	-1.12	-0.02	0.56	1.14	2.14	4459	1
## alpha[43]	2.37	0.01	0.50	1.37	2.04	2.37	2.71	3.39	4735	1
## alpha[44]	2.38	0.00	0.33	1.73	2.15	2.38	2.60	3.02	4629	1
## alpha[45]	1.82	0.01	0.60	0.65	1.40	1.81	2.23	3.00	4465	1
## alpha[46]	1.19	0.01	0.86	-0.47	0.61	1.18	1.78	2.84	4706	1
## alpha[47]	2.67	0.01	0.53	1.66	2.31	2.67	3.04	3.69	5532	1
## alpha[48]	2.48	0.01	0.54	1.41	2.11	2.48	2.84	3.53	5117	1
## alpha[49]	0.82	0.01	0.65	-0.45	0.37	0.83	1.27	2.08	5510	1
## alpha[50]	3.13	0.01	0.55	2.04	2.76	3.13	3.50	4.21	5976	1
## alpha[51]	1.33	0.01	0.80	-0.22	0.78	1.33	1.86	2.91	5115	1
## beta[1]	-0.23	0.00	0.17	-0.56	-0.35	-0.23	-0.12	0.11	3673	1
## beta[2]	0.21	0.00	0.03	0.16	0.20	0.21	0.23	0.26	4418	1
## beta[3]	-0.22	0.00	0.03	-0.28	-0.24	-0.22	-0.20	-0.16	4089	1
## beta[4]	1.48	0.00	0.16	1.18	1.38	1.48	1.59	1.81	3988	1
## beta[5]	1.33	0.00	0.16	1.01	1.21	1.33	1.44	1.64	3920	1
## beta[6]	1.98	0.00	0.27	1.46	1.79	1.98	2.16	2.50	4933	1
## beta[7]	0.68	0.00	0.31	0.07	0.46	0.68	0.88	1.29	4328	1

```

## beta[8]      0.66    0.01 0.51 -0.32  0.32  0.66  1.00  1.65  5423     1
##
## Samples were drawn using NUTS(diag_e) at Tue May 17 15:26:06 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).

```

When the alpha values vary, the coefficient on `cnum_mt2` is inconclusive. The 95% credible interval is from -0.54 to 0.15 and the mean is -0.19. Considering that the outcome is in log units, this indicates a broad range of effects and a high level of uncertainty about the impact of any one covariate.

```

loo_mlm <- loo(post_mlm)
loo_mlm

##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo   -8878.0 20.0
## p_loo       38.6  0.8
## looic     17756.0 40.1
## -----
## 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.

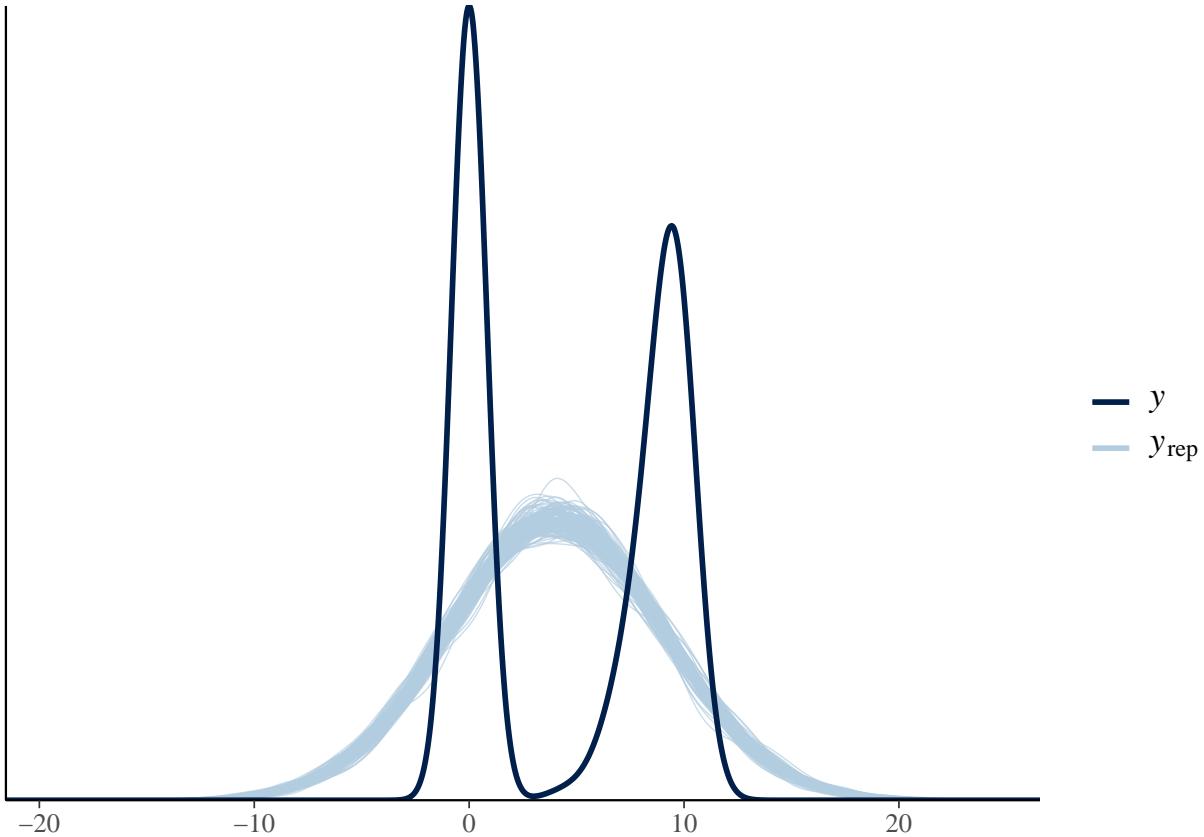
```

All Pareto k estimates are less than 0.5, which indicates that importance sampling is able to estimate the `elpd_loo` accurately and there are few observed values that have an outsized influence. The `elpd_loo` will be useful for comparing to other models.

```

## visualize density
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
)

```



```

# comparison of predictions
yrep <- rstan::extract(post_mlm, "yrep")[[1]]
low <- apply(yrep, MARGIN = 2, FUN = quantile, probs = 1 / 3)
high <- apply(yrep, MARGIN = 2, FUN = quantile, probs = 2 / 3)

y <- stan_data$y

c(too_low = mean(y < low),
  just_right = mean(y > low & y < high),
  too_high = mean(y > high))

##      too_low just_right    too_high
##  0.4103333  0.1000000  0.4896667

```

Since the observed values of y appear to be bimodal, the posterior predictive distribution, which is approximately normal, has trouble predicting values at both ends.

The model is predicting too many extreme values (in the bottom and top tertile). In terms of prediction, it is slightly better than the previous model with the gamma distribution and a log link. Almost 6% of observed y values fall within the middle tertile.

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))

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

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

# reset covariates, use only 4 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))

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

print(post_iv_mlm, pars = c("beta1", "beta2", "alpha0", "alpha1"))

## 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]    0.08     0 0.03  0.03  0.06  0.08  0.10  0.14  6774    1
## beta1[2]   -0.03     0 0.04 -0.10 -0.05 -0.03  0.00  0.04  7012    1
## beta1[3]    1.70     0 0.15  1.41  1.60  1.70  1.80  1.99  6702    1
## beta1[4]    1.50     0 0.15  1.21  1.41  1.51  1.61  1.79  6864    1
## beta1[5]    1.03     0 0.21  0.61  0.89  1.03  1.18  1.46  8833    1
## beta1[6]    0.17     0 0.24 -0.29  0.01  0.16  0.34  0.63  8388    1
## beta1[7]   -0.01     0 0.27 -0.56 -0.19 -0.01  0.17  0.53 10639    1

```

## beta2	4.23	0	0.20	3.83	4.09	4.23	4.36	4.62	4660	1
## alpha0[1]	-0.07	0	0.14	-0.35	-0.17	-0.07	0.02	0.19	10991	1
## alpha0[2]	-0.11	0	0.24	-0.58	-0.27	-0.11	0.06	0.37	9592	1
## alpha0[3]	0.34	0	0.17	0.00	0.22	0.34	0.46	0.68	11411	1
## alpha0[4]	-0.05	0	0.15	-0.35	-0.15	-0.05	0.05	0.23	9462	1
## alpha0[5]	-0.13	0	0.08	-0.29	-0.18	-0.13	-0.07	0.03	6071	1
## alpha0[6]	-0.16	0	0.17	-0.49	-0.27	-0.15	-0.04	0.15	7937	1
## alpha0[7]	-0.26	0	0.18	-0.61	-0.37	-0.26	-0.14	0.08	10504	1
## alpha0[8]	-0.10	0	0.25	-0.59	-0.27	-0.10	0.07	0.39	11549	1
## alpha0[9]	-0.31	0	0.26	-0.83	-0.49	-0.31	-0.13	0.19	9818	1
## alpha0[10]	-0.04	0	0.11	-0.25	-0.11	-0.04	0.04	0.18	8018	1
## alpha0[11]	0.06	0	0.14	-0.20	-0.03	0.06	0.16	0.33	8333	1
## alpha0[12]	0.05	0	0.21	-0.35	-0.09	0.05	0.20	0.46	9943	1
## alpha0[13]	0.05	0	0.22	-0.39	-0.10	0.05	0.21	0.49	10298	1
## alpha0[14]	-0.04	0	0.09	-0.23	-0.10	-0.04	0.02	0.13	8299	1
## alpha0[15]	0.10	0	0.11	-0.10	0.03	0.10	0.17	0.31	8970	1
## alpha0[16]	0.23	0	0.14	-0.03	0.14	0.23	0.33	0.48	9128	1
## alpha0[17]	0.15	0	0.18	-0.19	0.03	0.15	0.27	0.49	11077	1
## alpha0[18]	-0.17	0	0.14	-0.45	-0.27	-0.18	-0.08	0.10	8692	1
## alpha0[19]	-0.15	0	0.14	-0.43	-0.24	-0.15	-0.05	0.13	10778	1
## alpha0[20]	-0.25	0	0.20	-0.65	-0.38	-0.25	-0.11	0.15	11752	1
## alpha0[21]	-0.10	0	0.16	-0.43	-0.21	-0.10	0.00	0.22	8378	1
## alpha0[22]	0.14	0	0.13	-0.12	0.05	0.14	0.23	0.39	8498	1
## alpha0[23]	0.00	0	0.09	-0.18	-0.06	0.00	0.06	0.18	7625	1
## alpha0[24]	0.12	0	0.13	-0.14	0.03	0.12	0.20	0.37	8734	1
## alpha0[25]	-0.12	0	0.16	-0.45	-0.22	-0.11	-0.01	0.21	11207	1
## alpha0[26]	0.06	0	0.12	-0.18	-0.02	0.06	0.14	0.29	8338	1
## alpha0[27]	-0.10	0	0.21	-0.53	-0.24	-0.09	0.05	0.33	10767	1
## alpha0[28]	0.08	0	0.18	-0.28	-0.05	0.08	0.20	0.43	10020	1
## alpha0[29]	0.10	0	0.22	-0.34	-0.05	0.10	0.25	0.53	11059	1
## alpha0[30]	-0.02	0	0.24	-0.49	-0.18	-0.03	0.14	0.43	10378	1
## alpha0[31]	-0.01	0	0.12	-0.24	-0.09	-0.01	0.07	0.22	9260	1
## alpha0[32]	-0.15	0	0.19	-0.52	-0.29	-0.15	-0.02	0.23	10870	1
## alpha0[33]	-0.10	0	0.08	-0.26	-0.16	-0.11	-0.05	0.06	6860	1
## alpha0[34]	0.00	0	0.13	-0.25	-0.09	0.00	0.09	0.26	8811	1
## alpha0[35]	-0.14	0	0.24	-0.60	-0.29	-0.14	0.02	0.33	10870	1
## alpha0[36]	-0.07	0	0.10	-0.26	-0.14	-0.07	-0.01	0.12	7228	1
## alpha0[37]	-0.09	0	0.15	-0.39	-0.19	-0.09	0.01	0.20	9776	1
## alpha0[38]	-0.03	0	0.16	-0.34	-0.13	-0.03	0.08	0.28	9997	1
## alpha0[39]	-0.07	0	0.10	-0.26	-0.14	-0.07	0.00	0.12	7869	1
## alpha0[40]	0.11	0	0.24	-0.33	-0.05	0.12	0.28	0.58	12146	1
## alpha0[41]	0.02	0	0.16	-0.29	-0.09	0.02	0.13	0.34	9741	1
## alpha0[42]	0.00	0	0.23	-0.45	-0.16	0.00	0.16	0.46	11390	1
## alpha0[43]	-0.28	0	0.13	-0.54	-0.37	-0.28	-0.19	-0.02	11215	1
## alpha0[44]	-0.05	0	0.08	-0.21	-0.11	-0.05	0.00	0.11	6095	1
## alpha0[45]	0.50	0	0.16	0.21	0.40	0.50	0.61	0.81	10938	1
## alpha0[46]	-0.34	0	0.26	-0.87	-0.51	-0.34	-0.17	0.16	11451	1
## alpha0[47]	-0.32	0	0.14	-0.59	-0.41	-0.32	-0.22	-0.05	9366	1
## alpha0[48]	-0.04	0	0.14	-0.32	-0.14	-0.04	0.06	0.23	9778	1
## alpha0[49]	-0.18	0	0.16	-0.49	-0.29	-0.18	-0.07	0.13	9418	1
## alpha0[50]	0.00	0	0.15	-0.29	-0.10	0.00	0.10	0.29	9934	1
## alpha0[51]	0.00	0	0.21	-0.42	-0.14	0.01	0.16	0.41	10042	1
## alpha1[1]	0.13	0	0.04	0.05	0.10	0.13	0.16	0.21	6803	1
## alpha1[2]	0.08	0	0.01	0.07	0.08	0.08	0.08	0.09	7411	1

```

## alpha1[3] -0.13      0 0.01 -0.15 -0.14 -0.13 -0.12 -0.11 7038 1
## alpha1[4] -0.29      0 0.04 -0.37 -0.31 -0.28 -0.26 -0.20 6374 1
## alpha1[5] -0.20      0 0.04 -0.28 -0.23 -0.20 -0.17 -0.11 5772 1
## alpha1[6]  0.24      0 0.07  0.11  0.19  0.24  0.28  0.37 7867 1
## alpha1[7]  0.26      0 0.08  0.10  0.20  0.26  0.31  0.42 6901 1
## alpha1[8]  0.33      0 0.13  0.07  0.24  0.33  0.41  0.58 9887 1
##
## Samples were drawn using NUTS(diag_e) at Tue May 17 15:32:09 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).

```

Based on the value of `beta2`, the output indicates that the impact of `cnum_mt2` is likely positive when including an instrument to isolate the causal effect and allowing variation in intercepts. The average estimated value of `beta2` is 4.13 and the 95% credible interval is from 3.74 to 4.52. Similar to the non-hierarchical models, there may be some limitations to the interpretation of a true effect from this model:

1. it only uses 3,000 observations from the full sample
2. it does not account for the fact that income is conditional on labor force participation
3. the model does not have good predictive capabilities compared to the simpler model

```

loo_iv_mlm <- loo(post_iv_mlm)
loo_iv_mlm

```

```

##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo -10727.4 26.3
## p_loo     55.8   1.5
## looic    21454.8 52.7
## -----
## 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, loo_gamma)

```

```

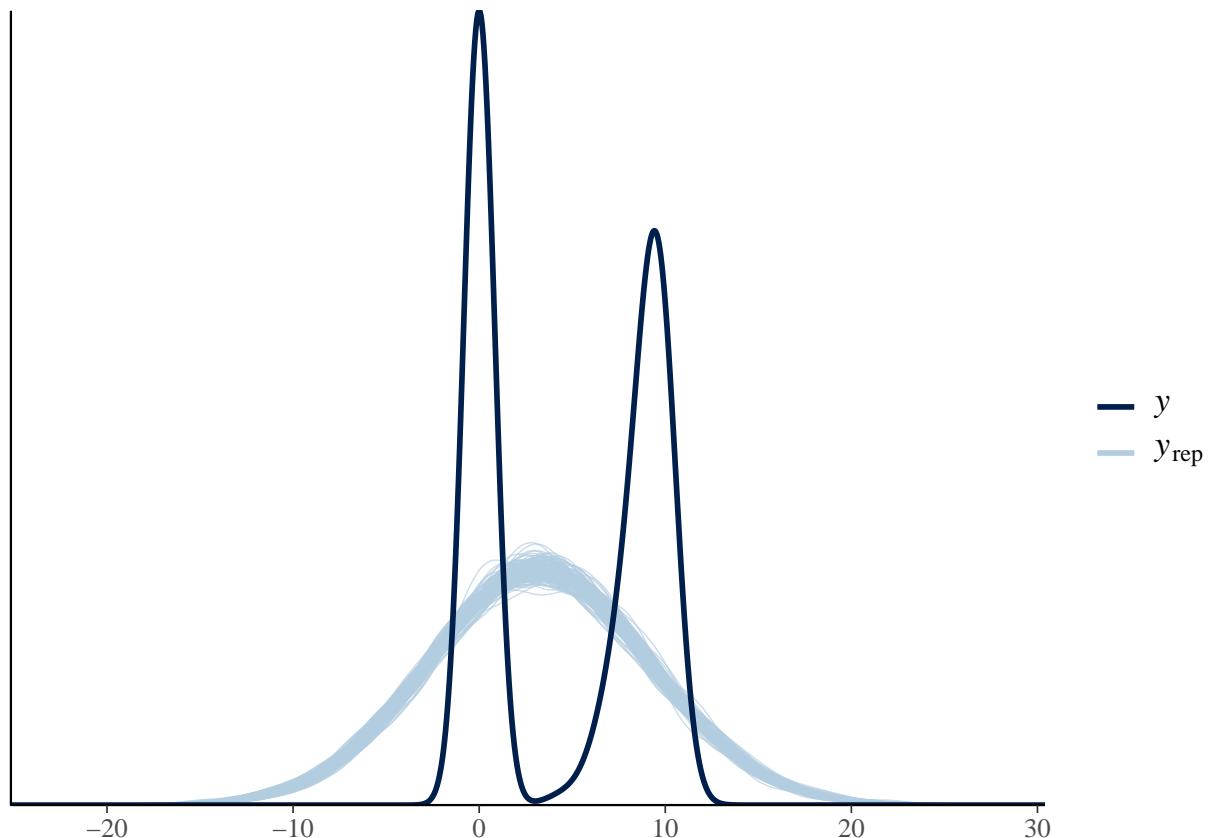
## Warning: Not all models have the same y variable. ('yhash' attributes do not
## match)

##           elpd_diff se_diff
## model2          0.0     0.0
## model1       -1849.4    32.6
## post_gamma_rs -13932.6   209.3

```

The simpler linear model appears to be the better model in terms of prediction accuracy based on cross-validation. The `elpd_diff` values is much larger than the SE.

```
# visualize density
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
)
```



```
# comparison of predictions
yrep <- rstan::extract(post_mlm, "yrep")[[1]]
low <- apply(yrep, MARGIN = 2, FUN = quantile, probs = 1 / 3)
high <- apply(yrep, MARGIN = 2, FUN = quantile, probs = 2 / 3)

y <- stan_data$y

c(too_low = mean(y < low),
  just_right = mean(y > low & y < high),
  too_high = mean(y > high))

##    too_low just_right    too_high
##  0.4103333  0.1000000  0.4896667
```

As with the other linear model, the approximately normal posterior predictive distribution does not fully align with the bimodal distribution of observed y values.

Conclusion