The Effect of Children on Women's Labor Supply: A Bayesian Replication Analysis

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
```

Run outcome 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 + 2] m;                        // prior medians
##   vector<lower = 0>[K + 2] scale;             // prior IQRs
##   real r;
```

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

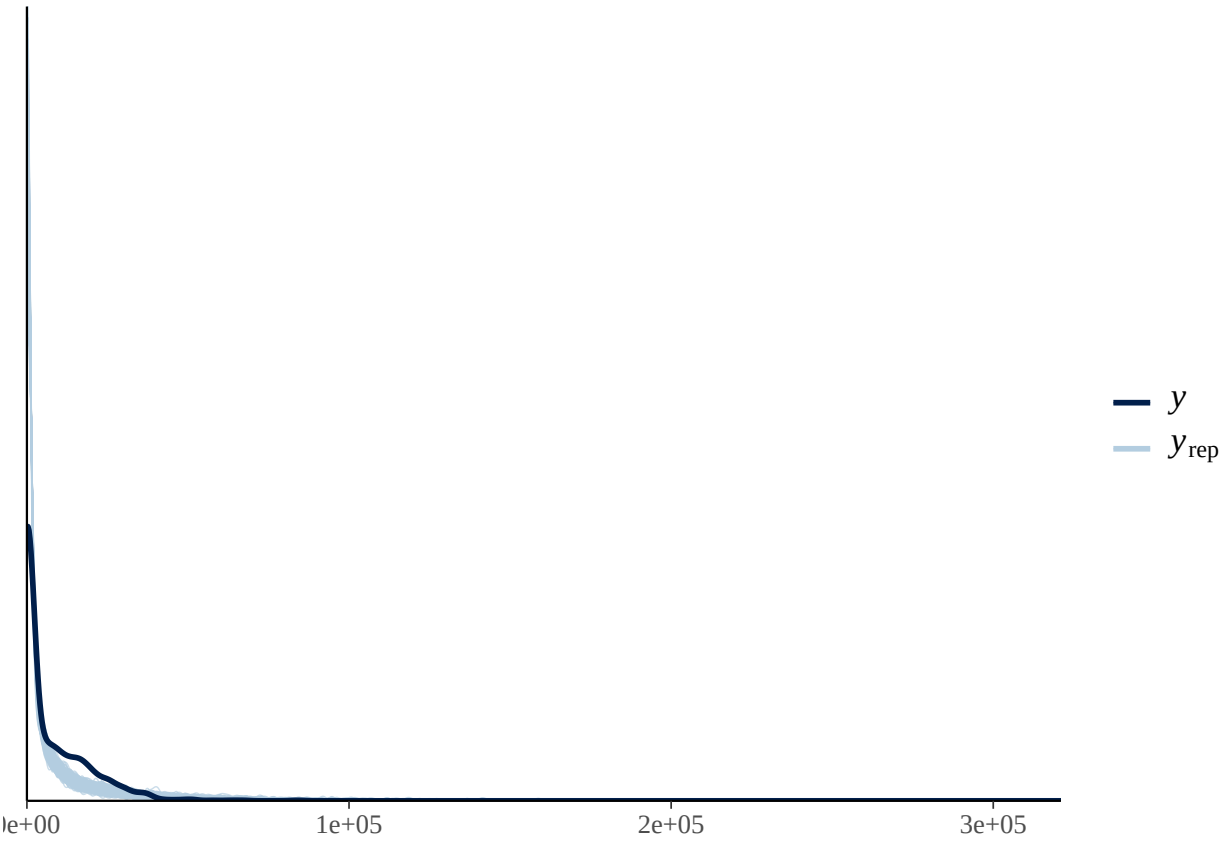
```
m <- rep(-0.1, 10)
s<- rep(1, 10)

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 = FALSE, m = m,
                                        scale = s, r = 0.5)
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  2528    1
## beta[1] 0.00       0 0.00 -0.01  0.00 0.00 0.00  0.01  3268    1
## beta[2] 0.00       0 0.00  0.00  0.00 0.00 0.00  0.00  4119    1
## beta[3] 0.00       0 0.00  0.00  0.00 0.00 0.00  0.00  4196    1
## beta[4] 0.00       0 0.00 -0.01  0.00 0.00 0.00  0.01  3383    1
## beta[5] 0.00       0 0.00 -0.01  0.00 0.00 0.00  0.01  3162    1
```

```
## beta[6] 0.00       0 0.01 -0.01  0.00 0.00 0.01  0.01  3601    1
## beta[7] 0.00       0 0.01 -0.02 -0.01 0.00 0.01  0.02  2787    1
## beta[8] 0.00       0 0.01 -0.03 -0.01 0.00 0.01  0.03  3029    1
## shape   0.18       0 0.00  0.18  0.18 0.18 0.18  0.18  4836    1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 08:52:30 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).
```

```
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(post_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.
```