

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

Jennah Gosciak

May 16, 2022

Contents

Background	1
Bayesian Analysis	5
Running with <code>rstanarm</code> for initial analysis	5
Linear model with untransformed outcome	10
Linear model with log-transformed outcome	17
Graphical models	24
IV with Heckman Two-Step Process	24
Re-run with weeks worked outcome	32
Re-run two-stage model for college subgroup	33
Frequentist Heckman example (for reference)	35

Background

In 1998, Joshua Angrist and William Evans published an article called **Children and Their Parents' Labor Supply: Evidence from Exogenous Variation in Family Size** about the effect an additional child on labor supply. In general, outside the context of an experiment, it's hard to determine the true effect of children on adults' labor supply since fertility is endogenous. The authors note that many economists believe fertility and labor supply are "jointly determined." Varying research studies assess the effect of children on wages and vice versa. This study, conducted using a frequentist framework, using the "sibling-sex composition" as an instrumental variable (IV). The authors argue that an indicator variable for whether the first two children have the same sex is as if randomly assigned. The study finds that children lead to a reduction in labor supply for women—an outcome that remains significant even among the IV estimates. It also leads to lower wages on average and fewer weeks worked. The study identified small and null effects on college educated women with high wage husbands and on male labor supply outcomes, based on the sample of 'husbands' in the data.

The causal effect of children on women's labor supply is important for many reasons: a reduction in women's labor supply could have positive effects on children's development, if women devote more time to caring for their children, or, if one values women's contributions to the labor market, small and null impacts may indicate that children do not pose an obstacle to women's career trajectories. Large negative impacts may provide some explanation for the persistent gender wage gap. In 2020, women earned 84% of what men did

(Pew Research Center, 2021). Given evidence in recent years of delayed family formation, particularly in large cities and urban areas, the impact of children on women's labor supply may be a motivating factor (Bui & Miller, 2018). The effect is both interesting in terms of causal research and the application of IV, but it's also meaningful for reasons Angrist and Evans don't even mention in their article.

In their paper, Angrist and Evans use data from the 1980 and 1990 Census Public Use Micro Samples (PUMS). They use a variety of restrictions to generate a sample of women, ages 21-35 whose oldest child was less than 18 years of age and who have at least two children. While Angrist and Evans run their analysis on a second sample of married women, for this project I focus on the larger sample of all women—regardless of marital status. Additionally, I focus on women's earnings, not the binary outcome of whether they are in the labor force or the number of weeks worked. I previously replicated the findings in this paper using the frequentist two stage least squares (TSLS) approach to IV. Using the detailed sample restrictions that Angrist and Evans outline in their paper, I was able to replicate Tables 3, 6, and 7. The OLS and TSLS estimates that I replicated are below.

	All mothers			Married mothers			Married fathers		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Estimation method	OLS	2SLS	2SLS	OLS	2SLS	2SLS	OLS	2SLS	2SLS
Instrument for <i>More than 2 children</i>	-	<i>Same sex</i>	<i>Two boys, Two girls</i>	-	<i>Same sex</i>	<i>Two boys, Two girls</i>	-	<i>Same sex</i>	<i>Two boys, Two girls</i>
Dependent variable									
<i>Worked for pay</i>	-0.173 (0.002)	-0.129 (0.026)	-0.122 (0.025)	-0.162 (0.002)	-0.124 (0.028)	-0.118 (0.028)	-0.008 (0.001)	0.006 (0.009)	0.003 (0.009)
<i>Weeks worked</i>	-8.768 (0.071)	-6.205 (1.133)	-5.886 (1.126)	-7.827 (0.089)	-5.833 (1.224)	-5.664 (1.216)	-0.850 (0.046)	0.604 (0.610)	0.452 (0.606)
<i>Hours/week</i>	-6.538 (0.062)	-4.777 (0.972)	-4.508 (0.966)	-5.854 (0.076)	-5.024 (1.033)	-4.837 (1.026)	0.217 (0.054)	0.803 (0.711)	0.727 (0.707)
<i>Labor income</i>	-3639.61 (33.55)	-2169.72 (536.41)	-2097.92 (533.66)	-3048.90 (39.71)	-1736.58 (555.74)	-1749.73 (552.43)	-1792.17 (102.26)	-355.51 (1356.42)	-414.19 (1348.97)
<i>Ln(Family income)</i>	-0.131 (0.005)	-0.046 (0.070)	-0.055 (0.069)	-0.132 (0.005)	-0.055 (0.060)	-0.058 (0.060)	-	-	-
<i>Ln(Non-wife income)</i>	-	-	-	-	-0.055 (0.006)	0.043 (0.072)	0.031 (0.071)	-	-

Figure 1: table 7

I only replicated this analysis with data from the 1980 Census, which is dated. Additionally, given time and processing constraints, I randomly sampled only 3,000 records from the total dataset, which is around 400,000 records. A more precise estimate of the causal estimate would use more data. I hope that in replicating this analysis using Bayesian inference I will either strengthen (or contradict) the claims made by Angrist and Evans and provide a working example for updating this analysis with more recent data and larger datasets.

For the Bayesian analysis, I use the following methods: * a simple linear model with a normal PDF as the likelihood and `incwage` as the outcome * a simple linear model with a normal PDF as the likelihood and `log(incwage)` as the outcome * a two-stage model with a Probit in the first stage for the decision function of having an additional child and a normal likelihood in the second stage for the impact of children on `log(incwage)` * This approach is based off of the likelihood function in 2.3 of the sampleSelection vignette

```

## load data
df <- read_dta("../00_data/sample1.dta")
df <- df %%
  # create indicator for some college
  mutate(coll = if_else(str_sub(educus, 1, 1) == "8", 1, 0))

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)))

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

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

# distribution of states
df_samp %>%
  group_by(stateus) %>%
  summarize(n = n()) %>%
  arrange(desc(n))

## # A tibble: 51 x 2
##   stateus     n
##   <dbl+lbl> <int>
## 1 California 257
## 2 Texas      214
## 3 New York   210
## 4 Michigan   165
## 5 Illinois   162
## 6 Pennsylvania 148
## 7 Ohio       137
## 8 Indiana    103
## 9 Florida    99
## 10 Missouri  80
## # ... with 41 more rows

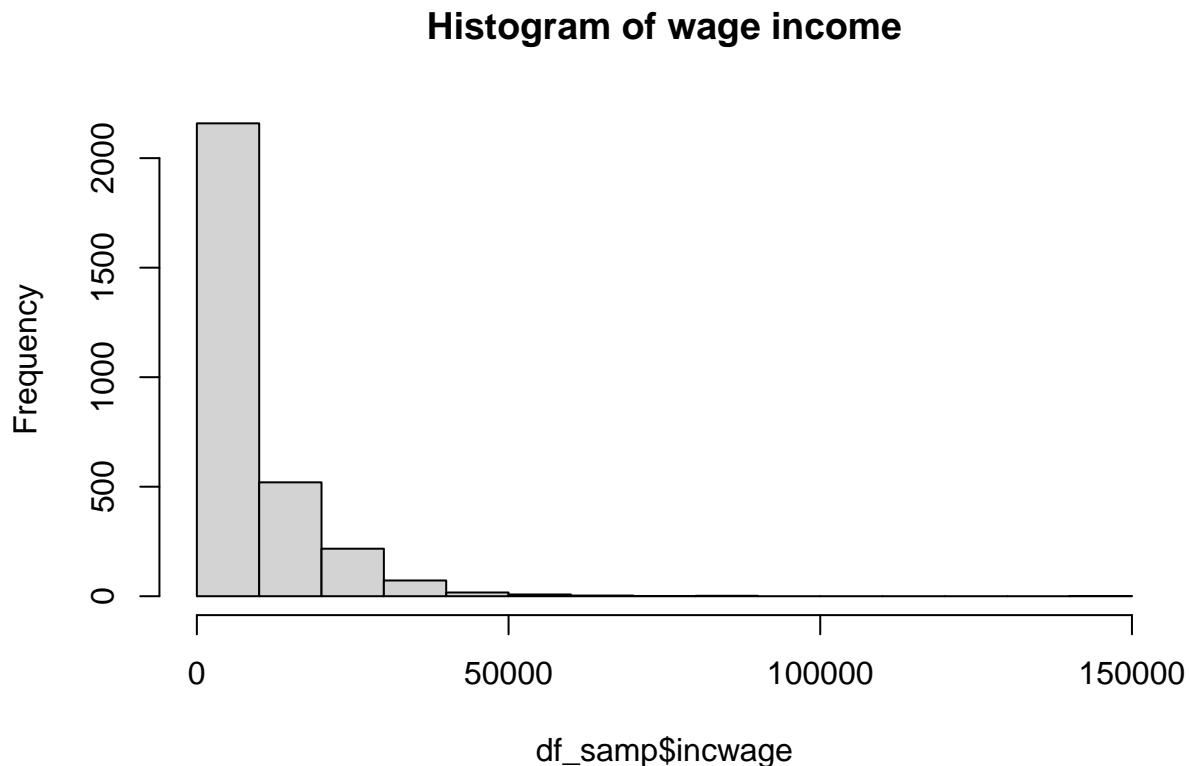
## distribution of education
df_samp %>%
  group_by(coll) %>%
  summarize(n = n())

## # A tibble: 2 x 2
##   coll     n

```

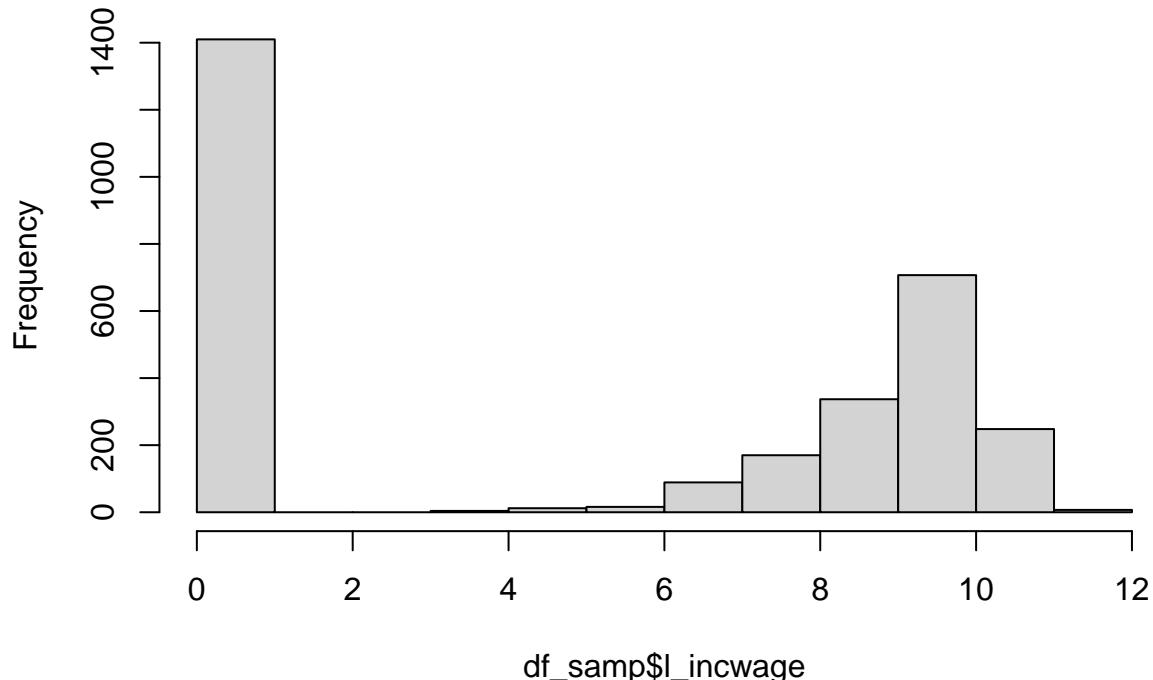
```
##      <dbl> <int>
## 1      0   2202
## 2      1    798

hist(df_samp$incwage, main = "Histogram of wage income")
```



```
hist(df_samp$l_incwage, main = "Histogram of log wage")
```

Histogram of log wage



Bayesian Analysis

Running with `rstanarm` for initial analysis

```
# Running with rstanarm
post <- stan_glm(
  incwage ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth,
  data = df_samp,
  family = gaussian(),
  prior = cauchy(),
  prior_intercept = cauchy(),
  seed = 12345
)
post

## stan_glm
## family:      gaussian [identity]
## formula:     incwage ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black +
##               hisp + r_oth
## observations: 3000
## predictors:   9
## -----
```

```

##           Median MAD_SD
## (Intercept) 7667.8   258.5
## cnum_mt2    -3148.3   410.9
## age         537.0   57.1
## age_fbirth -403.8   72.2
## f_boy       -0.1    3.7
## s_boy        0.0    3.4
## r_black     3113.0   609.1
## hisp        0.1    3.5
## r_oth        0.0    3.6
##
## Auxiliary parameter(s):
##           Median MAD_SD
## sigma 10058.6   129.5
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg

post_log <-
stan_glm(
  l_incwage ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth,
  data = df_samp,
  family = gaussian(),
  prior = cauchy(),
  prior_intercept = cauchy(),
  seed = 12345
)
post_log

## stan_glm
## family: gaussian [identity]
## formula: l_incwage ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black +
##           hisp + r_oth
## observations: 3000
## predictors: 9
## -----
##           Median MAD_SD
## (Intercept) 5.3    0.2
## cnum_mt2   -1.6    0.2
## age        0.2    0.0
## age_fbirth -0.3    0.0
## f_boy      0.0    0.2
## s_boy     -0.1    0.2
## r_black    1.3    0.3
## hisp      0.0    0.3
## r_oth      0.0    0.5
##
## Auxiliary parameter(s):
##           Median MAD_SD
## sigma 4.4    0.1
##
## -----
## * For help interpreting the printed output see ?print.stanreg

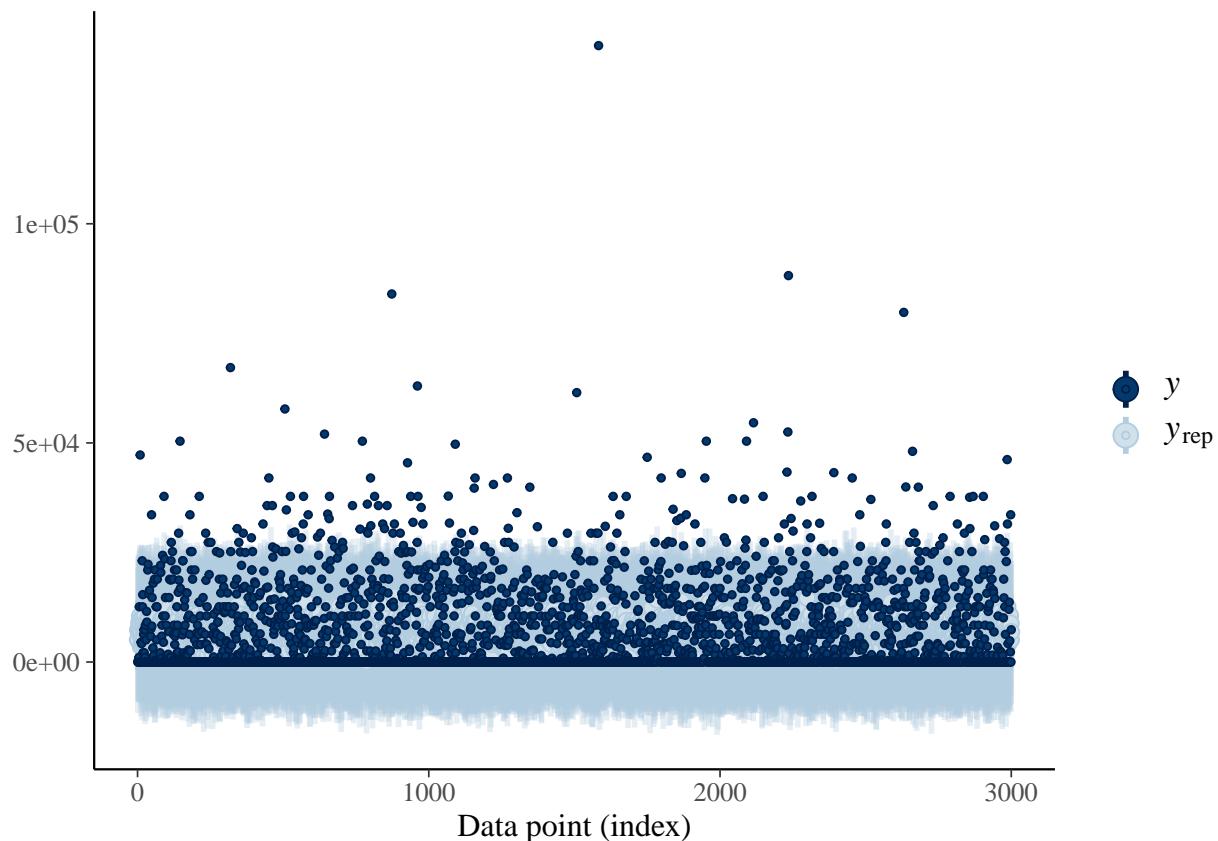
```

```
## * For info on the priors used see ?prior_summary.stanreg
```

```
pp_check(post, plotfun = "loo_intervals")
```

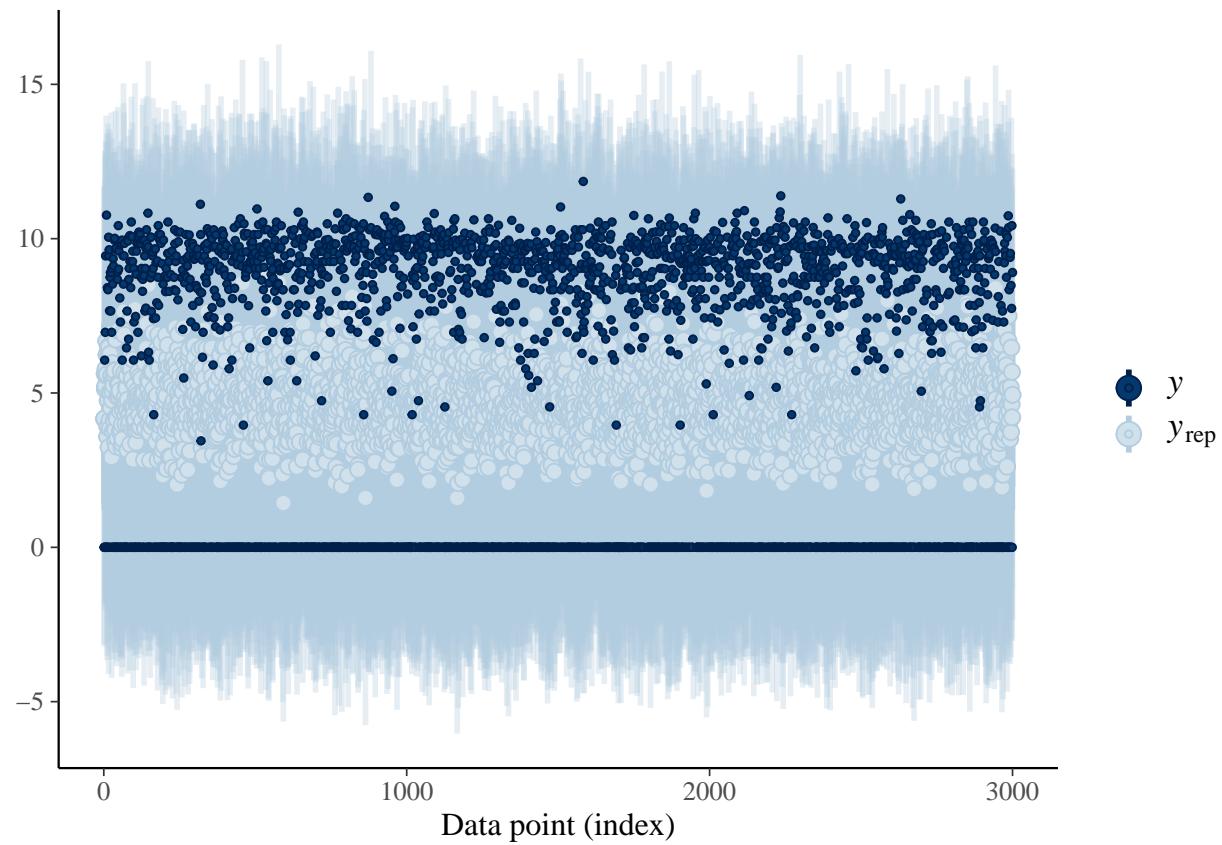
```
## Running PSIS to compute weights...
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

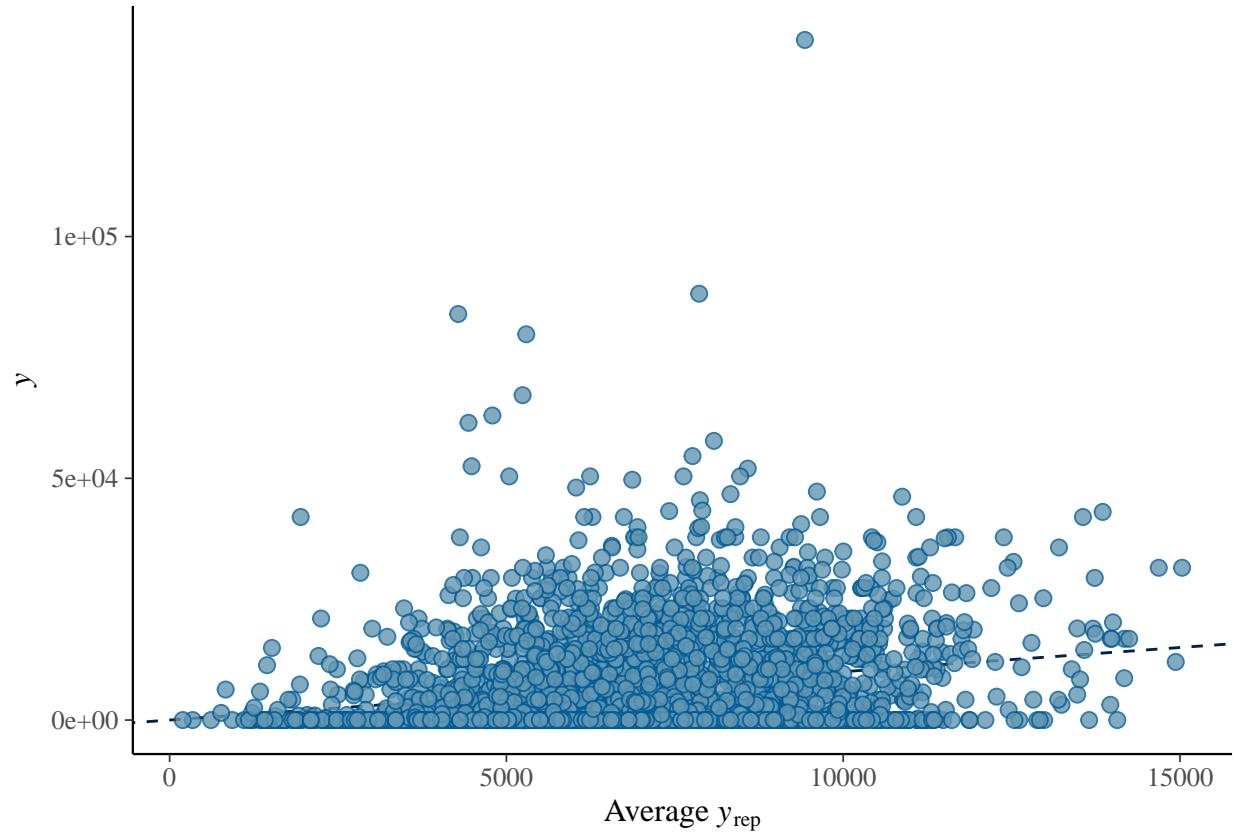


```
pp_check(post_log, plotfun = "loo_intervals")
```

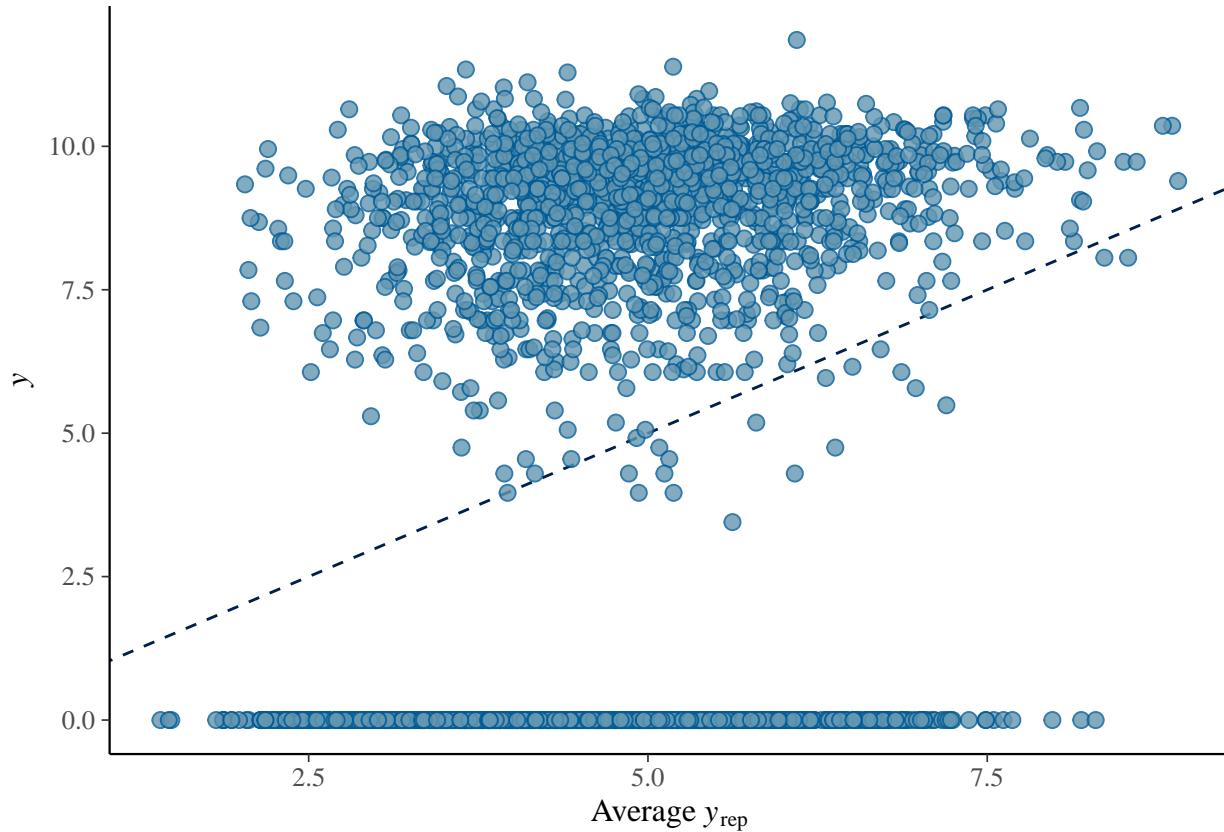
```
## Running PSIS to compute weights...
```



```
pp_check(post, plotfun = "scatter_avg")
```



```
pp_check(post_log, plotfun = "scatter_avg")
```



Linear model with untransformed outcome

```
# display stan code
writeLines(readLines("linear.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 scale values
##   real<lower = 0> r;
## }
## parameters {
##   vector[K] beta;
##   real alpha;
##   real<lower = 0> sigma;
## }
## model { // log likelihood, equivalent to target += normal_lpdf(y | alpha + X * beta, sigma)
##   if (!prior_only) target += normal_id_glm_lpdf(y | X, alpha, beta, sigma);
## }
```

```

##  target += normal_lpdf(alpha | m[1], scale[1]);
##  target += normal_lpdf(beta | m[2:K + 1], scale[2:K + 1]);
##  target += exponential_lpdf(sigma | r);
## }

## generated quantities {
##   vector[N] log_lik;
##   vector[N] yrep;
##   {
##     vector[N] mu = alpha + X * beta;
##     for (n in 1:N) {
##       log_lik[n] = normal_lpdf(y[n] | mu[n], sigma);
##       yrep[n] = normal_rng(mu[n], sigma);
##     }
##   }
## }

# use normal priors
m <- c(8000, -2000, 0, -300, -40, -40, 2000, 2000, 2000)
s<- c(500, 3000, 600, 400, 140, 140, 3000, 3000, 3000)

# define covariates
cov <- c("cnum_mt2", "age", "age_fbirth", "f_boy", "s_boy", "r_black", "hisp", "r_oth")

# generate stan data
stan_data <- list(N = nrow(df_samp), K = 8, y = df_samp$incwage,
                    X = df_samp[, cov],
                    prior_only = TRUE, m = m,
                    scale = s,
                    r = 1)

# call stan for prior predictive distribution checks
pre <- stan("linear.stan", data = stan_data, seed = 12345)
# print output
print(pre, pars = c("alpha", "beta", "sigma"))

## Inference for Stan model: linear.
## 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%
## alpha    7996.38    5.81 488.37 7030.90 7664.50 8003.21 8330.94 8925.42
## beta[1] -2017.94   32.63 2977.93 -7862.70 -4000.97 -2049.15 -18.79 3763.79
## beta[2]    7.21    6.36 577.61 -1133.33 -380.62  12.72 393.62 1121.94
## beta[3]  -305.60    4.62 391.34 -1083.30 -565.93 -306.14 -38.00 443.63
## beta[4]   -40.20    1.76 143.18 -307.95 -138.38 -40.02  57.07 239.75
## beta[5]   -40.35    1.58 141.06 -315.41 -135.64 -40.99  53.78 233.61
## beta[6]  1984.00   33.23 2991.23 -4025.77 -43.42 1991.59 4011.57 7806.12
## beta[7]  2023.87   34.37 2983.72 -3784.26    9.21 2016.28 4012.96 7767.03
## beta[8]  2062.00   32.01 2941.47 -3614.49   44.11 2041.59 4080.40 7729.61
## sigma     1.01    0.01  1.05   0.02    0.27    0.67   1.40   3.86
##           n_eff Rhat
## alpha      7066     1

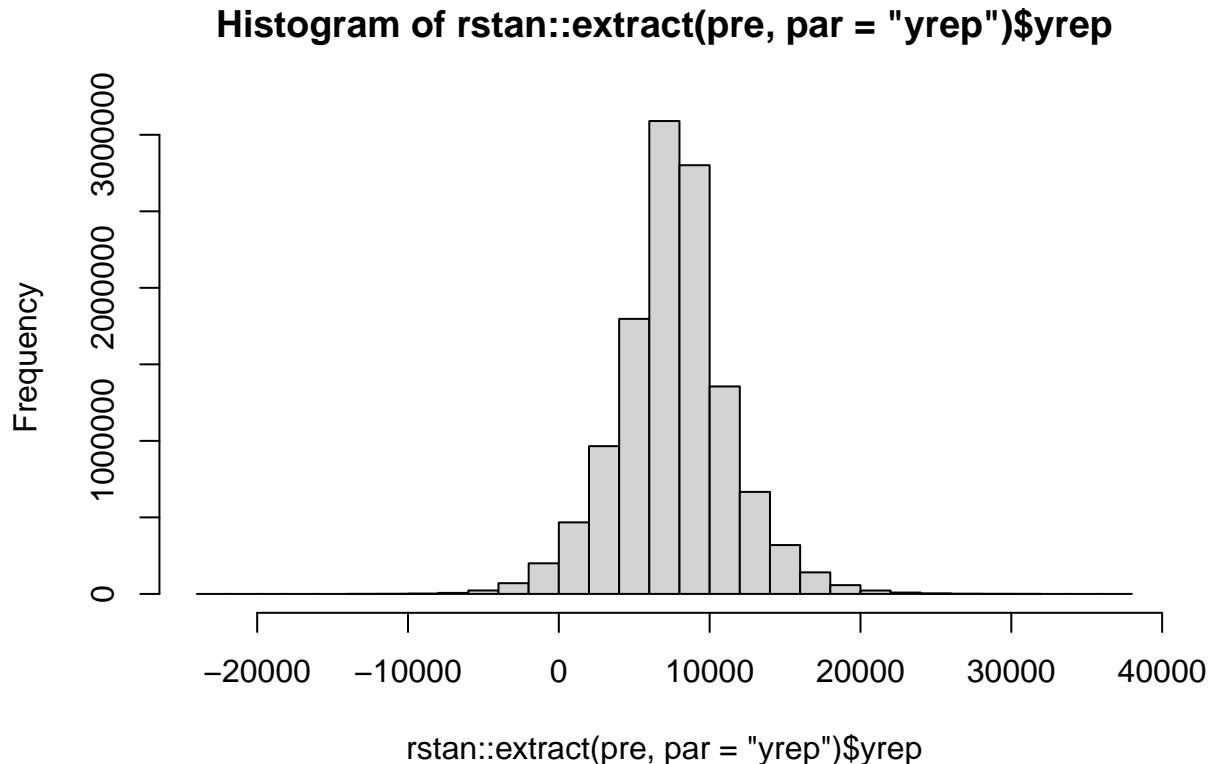
```

```

## beta[1] 8328 1
## beta[2] 8244 1
## beta[3] 7187 1
## beta[4] 6626 1
## beta[5] 7943 1
## beta[6] 8104 1
## beta[7] 7534 1
## beta[8] 8447 1
## sigma    5857 1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 01:14:35 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, par = "yrep")$yrep)

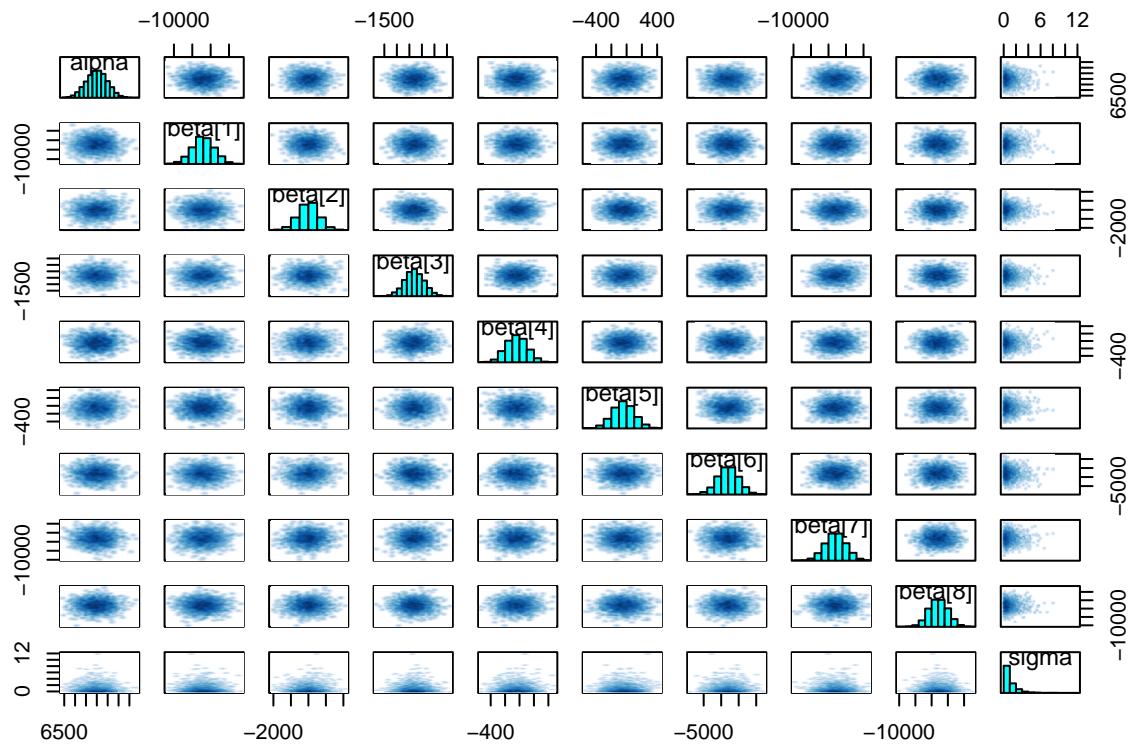
```



```

pairs(pre, pars = c("alpha", "beta", "sigma"))

```



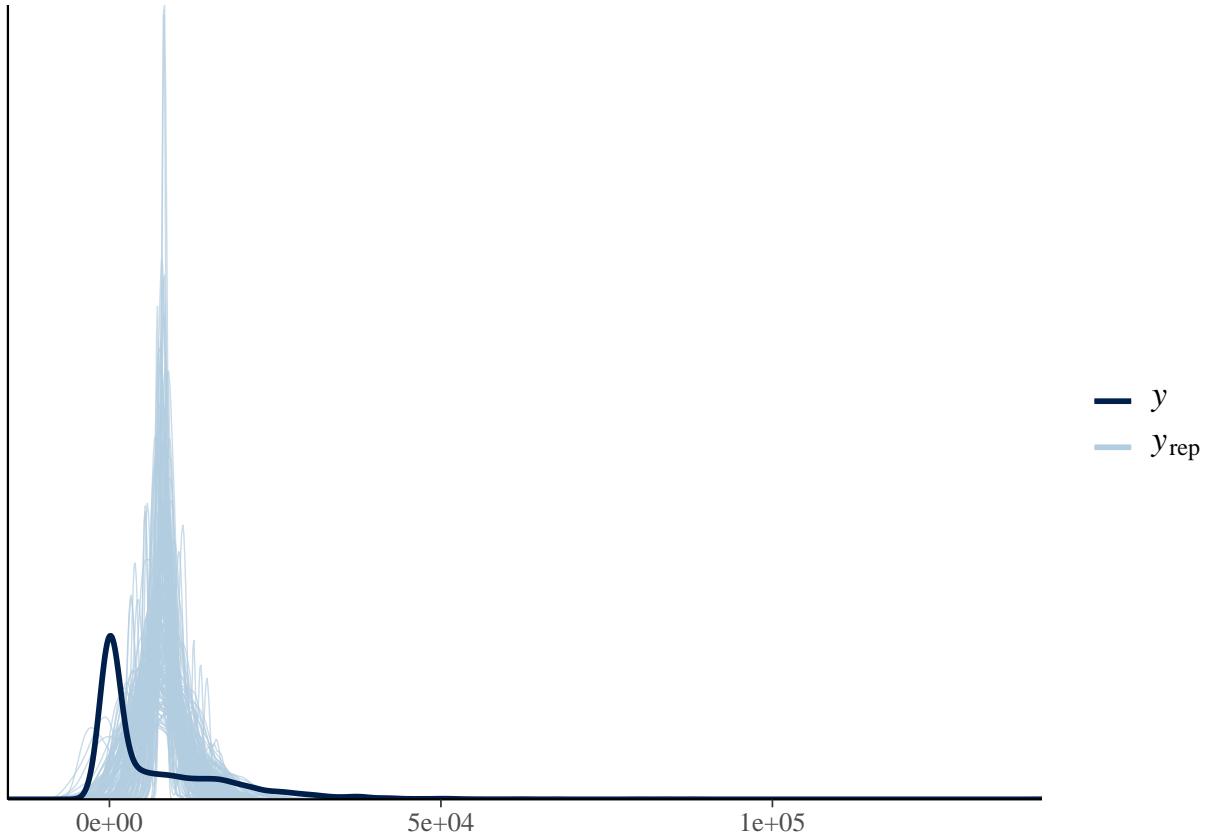
```
loo(pre)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate       SE
## elpd_loo -7.653539e+17 5.026072e+16
## p_loo     7.653539e+17 5.026072e+16
## looic    1.530708e+18 1.005214e+17
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                               Count Pct.   Min. n_eff
## (-Inf, 0.5]   (good)      0  0.0% <NA>
## (0.5, 0.7]   (ok)        0  0.0% <NA>
## (0.7, 1]     (bad)       0  0.0% <NA>
## (1, Inf)    (very bad) 3000 100.0% 0
## See help('pareto-k-diagnostic') for details.
```

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

```
    ppc_dens_overlay
)
```



```
stan_data$prior_only <- FALSE
post <- stan("linear.stan", data = stan_data, seed = 12345)
# print output
print(post, pars = c("alpha", "beta", "sigma"))
```

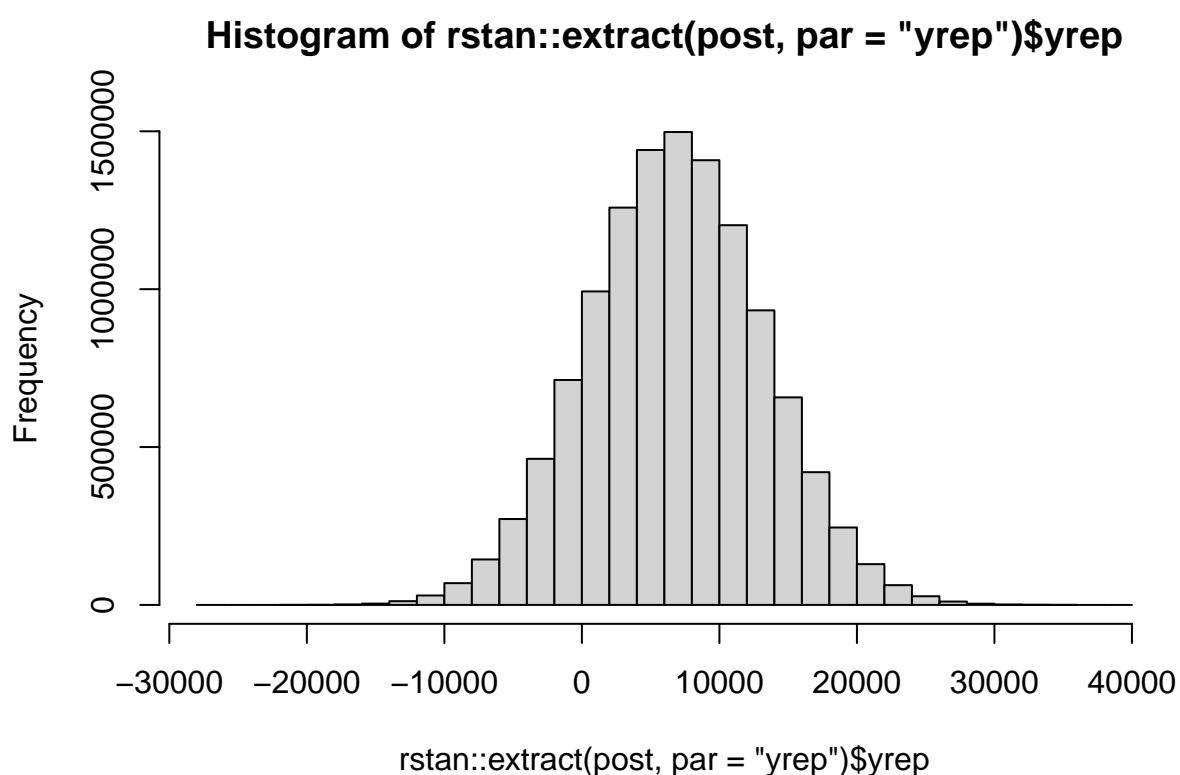
```
## Inference for Stan model: linear.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean   se_mean     sd      2.5%      25%      50%      75%     97.5%
## alpha    7714.59    2.83 161.96  7408.76  7606.55  7714.60  7823.96  8030.16
## beta[1] -3479.04    3.42 220.90 -3915.05 -3629.53 -3478.32 -3327.64 -3054.51
## beta[2]   573.89    0.50 33.35   508.30   551.16   575.05   596.50   636.26
## beta[3]  -446.38    0.65 39.39  -522.25  -472.32  -446.29  -419.61  -369.30
## beta[4]   -33.94    1.77 120.61  -267.13  -115.77  -36.34   49.57   202.58
## beta[5]   -27.33    1.67 113.09  -255.07  -104.30  -28.27   49.70   193.38
## beta[6]   3555.77    4.88 339.75  2893.42  3326.30  3554.06  3782.12  4251.18
## beta[7]   524.76    5.54 388.10  -248.31   264.36   522.88   787.92  1293.15
## beta[8]  1902.38   10.32 708.34   475.61  1437.43  1892.31  2373.83  3294.42
## sigma    5851.55    0.47 38.25  5776.61  5826.41  5851.82  5877.08  5924.98
##           n_eff Rhat
## alpha     3265     1
```

```

## beta[1] 4184 1
## beta[2] 4515 1
## beta[3] 3704 1
## beta[4] 4643 1
## beta[5] 4610 1
## beta[6] 4850 1
## beta[7] 4913 1
## beta[8] 4713 1
## sigma    6672 1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 08:33:25 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(post, par = "yrep")$yrep)

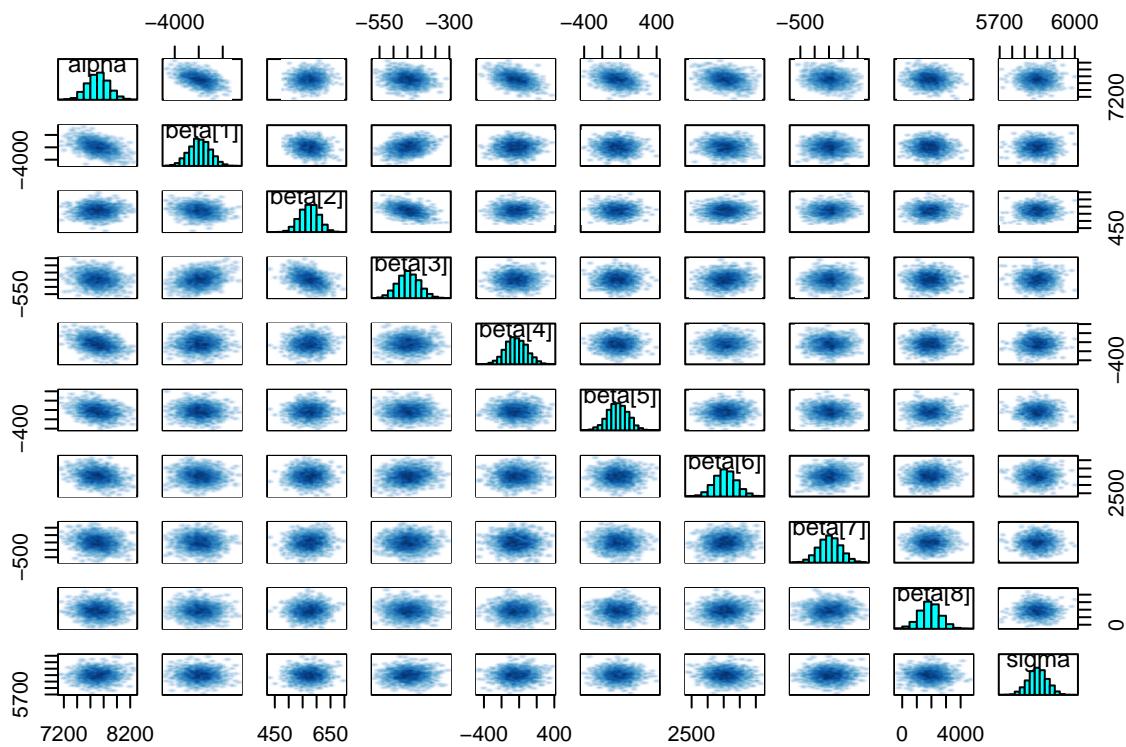
```



```

pairs(post, pars = c("alpha", "beta", "sigma"))

```



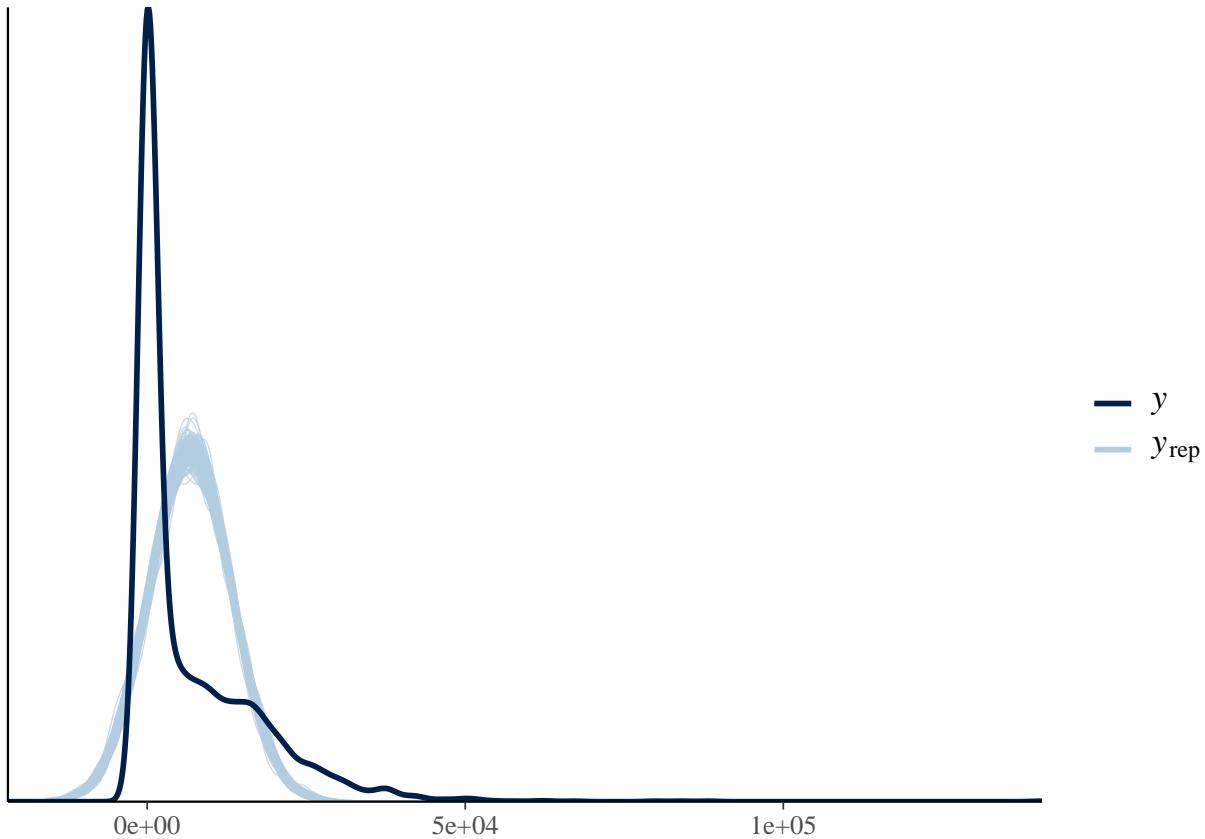
```
loo(post)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate     SE
## elpd_loo -33227.9 345.4
## p_loo      44.9  13.2
## looic     66455.8 690.8
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                               Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)    2996 99.9%  1279
## (0.5, 0.7]   (ok)       2  0.1%  562
## (0.7, 1]     (bad)      1  0.0%   38
## (1, Inf)    (very bad) 1  0.0%    7
## See help('pareto-k-diagnostic') for details.

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

```
    ppc_dens_overlay  
)
```



Linear model with log-transformed outcome

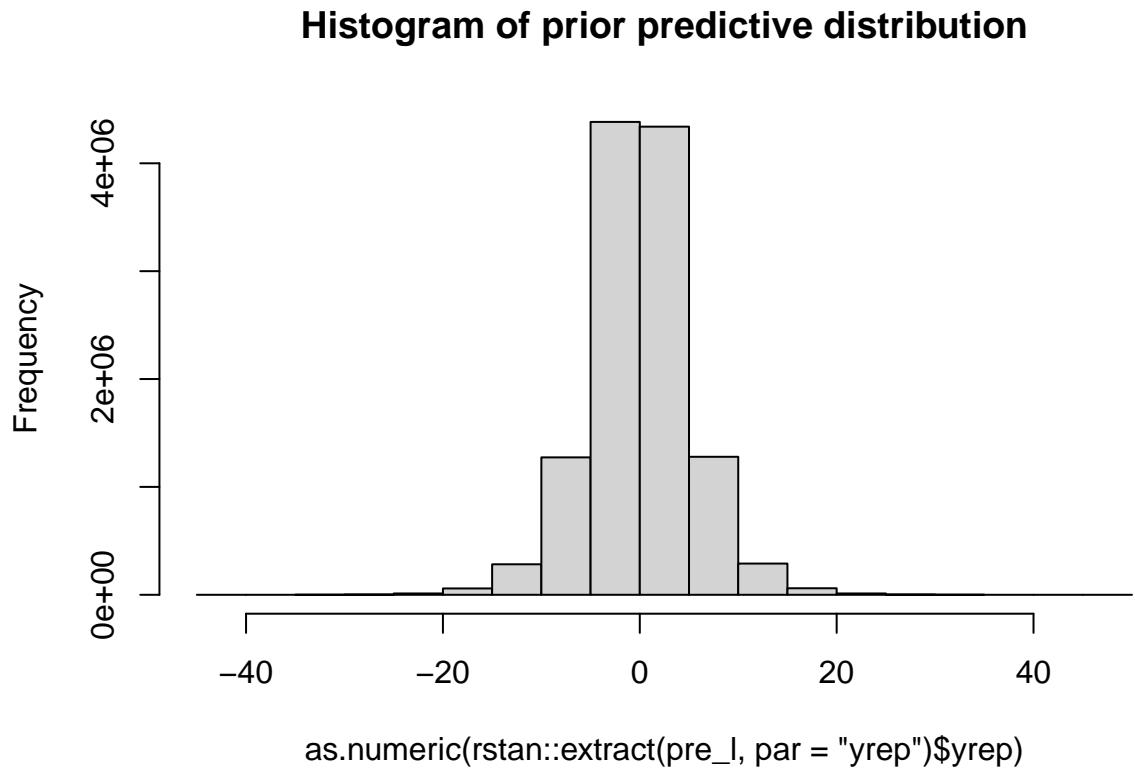
```
# set generic priors  
m <- rep(0, 9)  
s<- rep(1, 9)  
  
stan_data$y <- df_samp$l_incwage  
stan_data$m <- m  
stan_data$scale <- s  
stan_data$prior_only = TRUE  
  
pre_l <- stan("linear.stan", data = stan_data, seed = 12345)  
# print output  
print(pre_l, pars = c("alpha", "beta", "sigma"))  
  
## Inference for Stan model: linear.  
## 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    0.00   0.01 0.97 -1.89 -0.67   0.00 0.68   1.86   6371   1
## beta[1]  0.00   0.01 1.01 -1.97 -0.70   0.00 0.69   1.98   8306   1
## beta[2] -0.01   0.01 0.99 -1.95 -0.67 -0.01 0.64   1.98   7545   1
## beta[3]  0.00   0.01 0.97 -1.88 -0.65 -0.01 0.64   1.89   6362   1
## beta[4]  0.00   0.01 1.01 -1.99 -0.66   0.01 0.70   1.90   7275   1
## beta[5]  0.00   0.01 1.00 -1.95 -0.67   0.01 0.67   2.04   6800   1
## beta[6]  0.02   0.01 1.01 -1.96 -0.67   0.02 0.69   1.99   7851   1
## beta[7]  0.00   0.01 1.00 -1.92 -0.68   0.00 0.67   1.96   7598   1
## beta[8]  0.00   0.01 1.03 -1.98 -0.73   0.00 0.72   2.03   9306   1
## sigma    0.99   0.01 0.99  0.03  0.29   0.68 1.37   3.69   5330   1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 08:43:57 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).

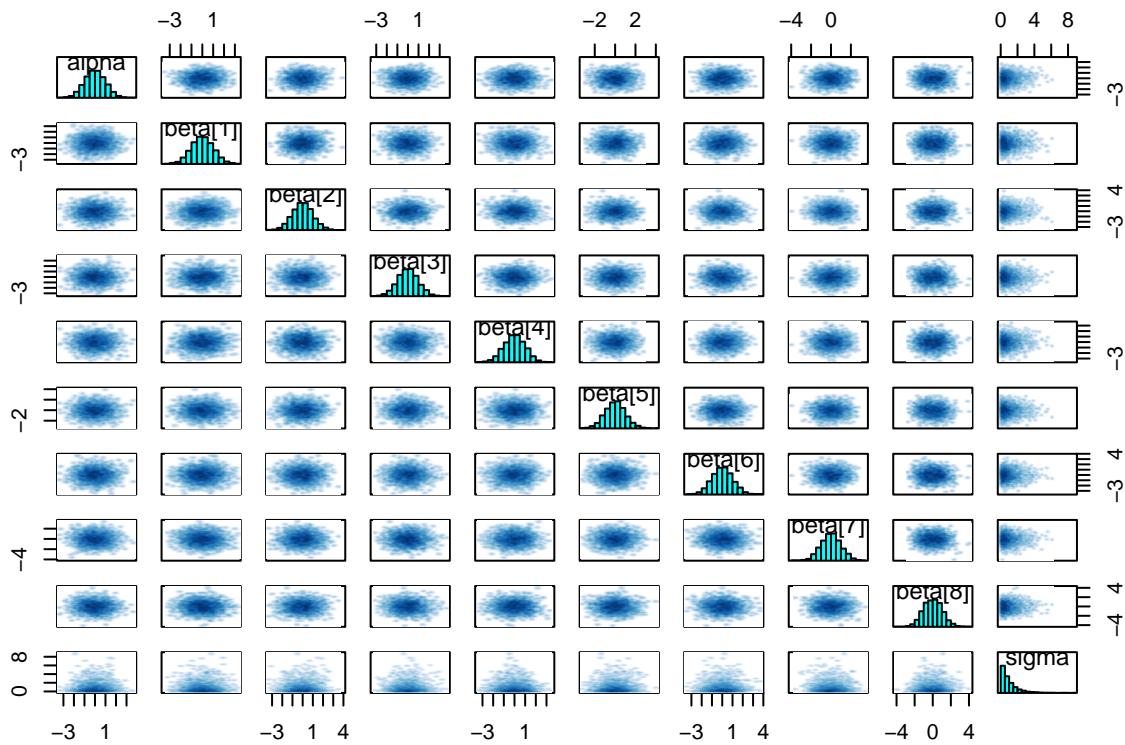
```

```
hist(as.numeric(rstan:::extract(pre_l, par = "yrep")$yrep), main = "Histogram of prior predictive distribution")
```



```
as.numeric(rstan:::extract(pre_l, par = "yrep")$yrep)
```

```
pairs(pre_l, pars = c("alpha", "beta", "sigma"))
```



```
loo_pre <- loo(pre_1)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

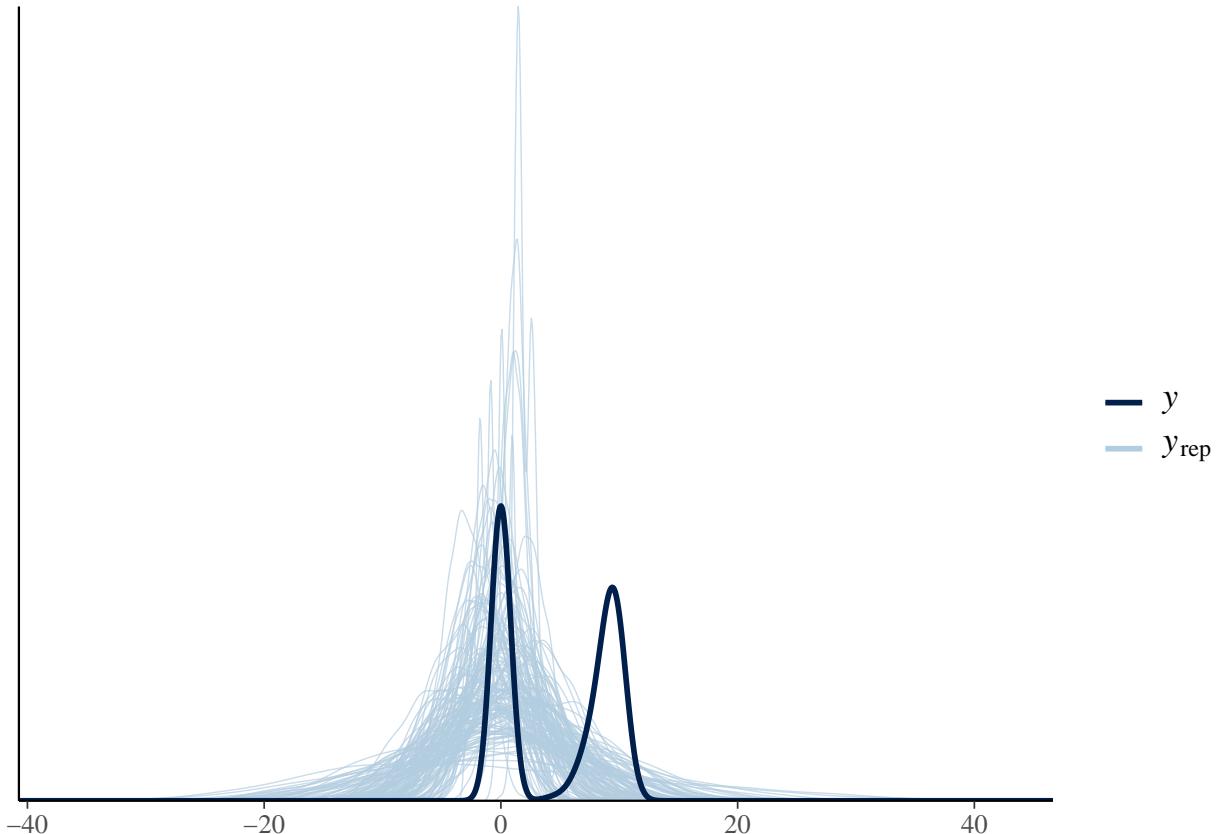
```
loo_pre
```

```
##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate          SE
## elpd_loo -5.499785e+12 105568780434.6
## p_loo      5.499785e+12 105568780378.4
## looic     1.099957e+13 211137560869.2
## -----
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##           Count Pct. Min. n_eff
## (-Inf, 0.5] (good)    0 0.0% <NA>
## (0.5, 0.7] (ok)      0 0.0% <NA>
## (0.7, 1] (bad)      0 0.0% <NA>
## (1, Inf) (very bad) 3000 100.0% 0
## See help('pareto-k-diagnostic') for details.
```

```

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

```



```

stan_data$prior_only = FALSE
post_l <- stan("linear.stan", data = stan_data,
               seed = 12345)
# print output
print(post_l, pars = c("alpha", "beta", "sigma"))

## Inference for Stan model: linear.
## 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     5.14     0.00 0.16  4.83  5.03  5.14  5.25  5.45  3068     1
## beta[1] -1.50     0.00 0.17 -1.84 -1.62 -1.50 -1.38 -1.15  4196     1
## beta[2]  0.24     0.00 0.02  0.19  0.22  0.24  0.26  0.29  4394     1
## beta[3] -0.29     0.00 0.03 -0.35 -0.31 -0.29 -0.27 -0.23  4199     1
## beta[4]  0.09     0.00 0.16 -0.23 -0.02  0.09  0.20  0.41  3833     1
## beta[5] -0.04     0.00 0.16 -0.35 -0.14 -0.03  0.07  0.27  4104     1
## beta[6]  1.25     0.00 0.25  0.78  1.08  1.25  1.41  1.74  4042     1
## beta[7]  0.02     0.00 0.28 -0.52 -0.18  0.01  0.22  0.57  4571     1
## beta[8]  0.00     0.01 0.49 -0.95 -0.33  0.01  0.32  0.95  4287     1

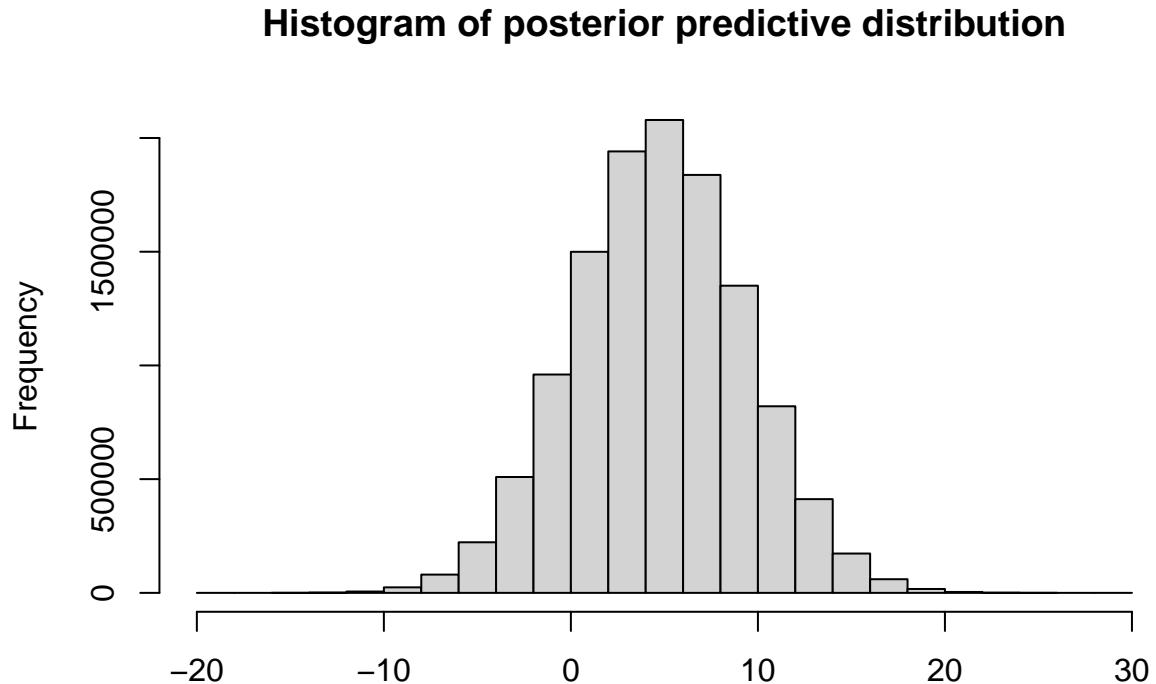
```

```

## sigma      4.42     0.00 0.06   4.31   4.38   4.42   4.46   4.53   4483     1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 08:48:10 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).

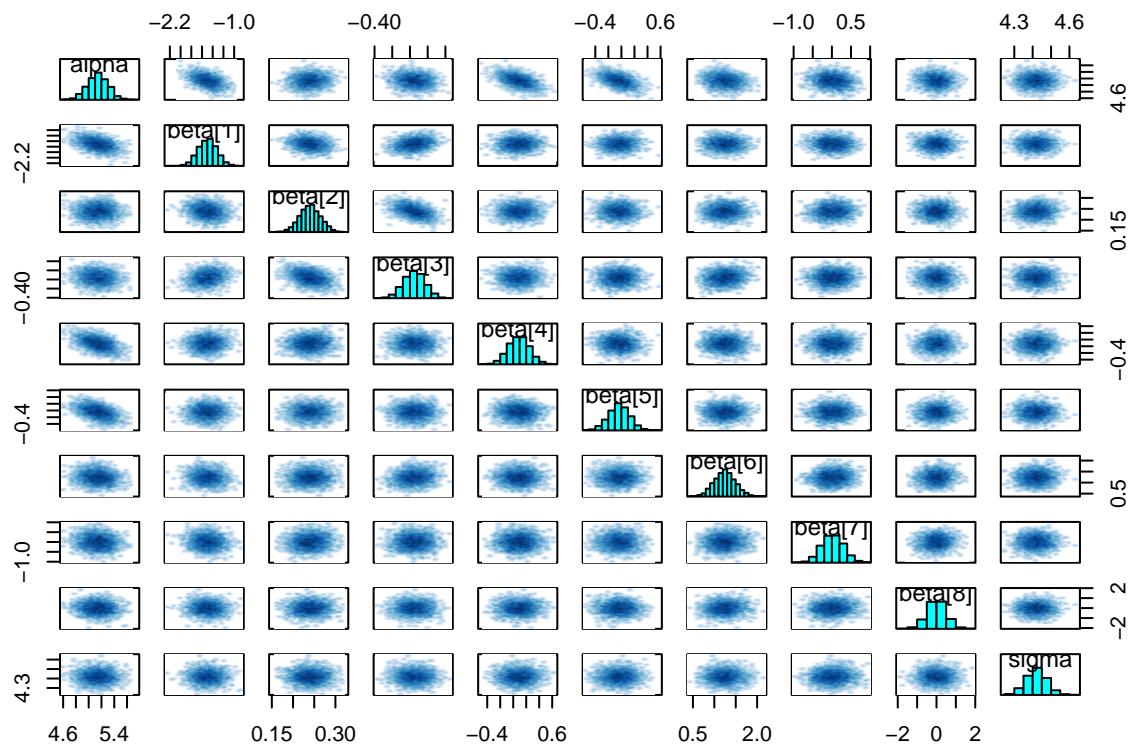
```

```
hist(as.numeric(rstan::extract(post_l, par = "yrep")$yrep), main = "Histogram of posterior predictive distribution")
```



```
as.numeric(rstan::extract(post_l, par = "yrep")$yrep)
```

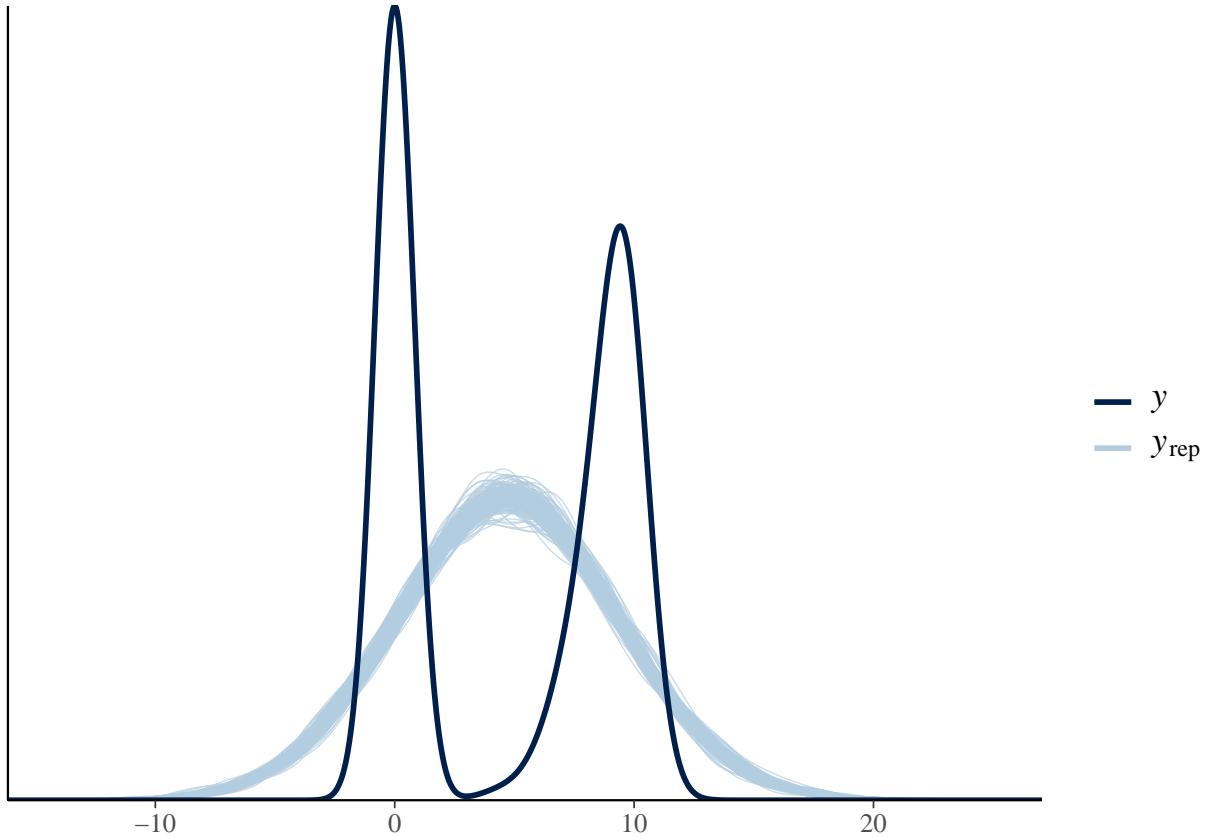
```
pairs(post_l, pars = c("alpha", "beta", "sigma"))
```



```

pp_check(as.numeric(stan_data$y),
  rstan::extract(post_1, par = "yrep")$yrep[sample(1:length(stan_data$y), size = 150), ],
  ppc_dens_overlay
)

```



```
loo_post <- loo(post_1)
loo_post
```

```
##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo   -8721.1 15.1
## p_loo        8.5  0.2
## looic     17442.3 30.1
## -----
## 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.
```

```
loo_compare(loo_pre, loo_post)
```

```
##           elpd_diff    se_diff
## model2  0.000000e+00  0.000000e+00
## model1 -5.499785e+12  1.055688e+11
```

Graphical models

```
library(dagitty)

## Warning: package 'dagitty' was built under R version 4.1.3

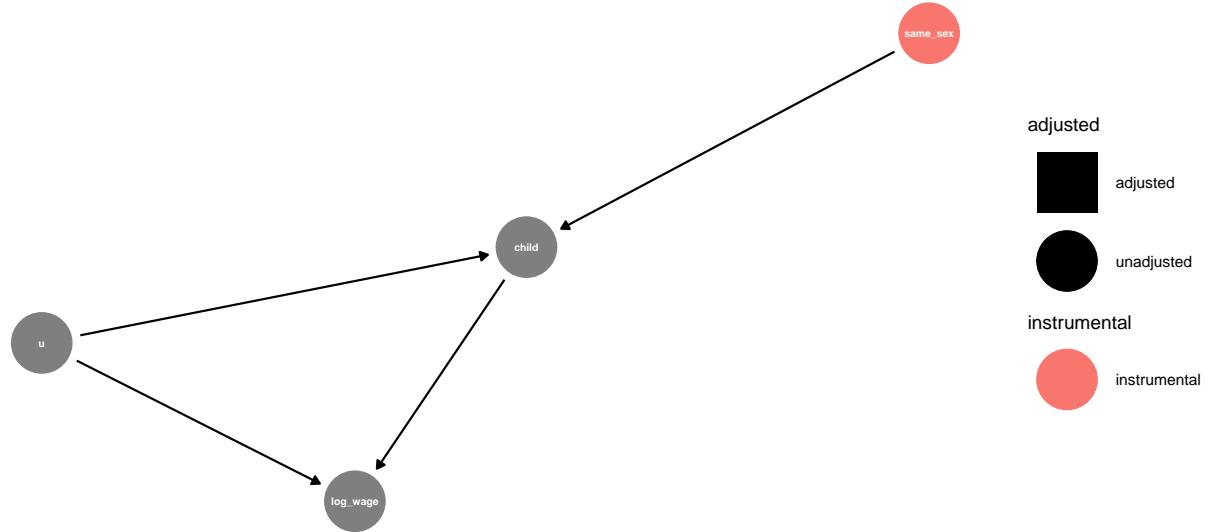
library(ggdag)

## Warning: package 'ggdag' was built under R version 4.1.3

## 
## Attaching package: 'ggdag'

## The following object is masked from 'package:stats':
## 
##     filter

# visualizing the instrumental variables problem
dagify(log_wage ~ child + u, child ~ same_sex + u, exposure = "child", outcome = "log_wage", latent = "same_sex")
```



IV with Heckman Two-Step Process

```
source(file.path("GLD_helpers.R"))
# set prior for rho with gld solver bounded
a_s <- GLD_solver_boundeds(bounds = -0.9:1, median = 0.3, IQR = 0.6)

## Warning in GLD_solver_boundeds(bounds = -0.9:1, median = 0.3, IQR = 0.6): no asymmetry and steepness ...
## effective bounds are -0.939028894871395 and 0.722768561999757
```

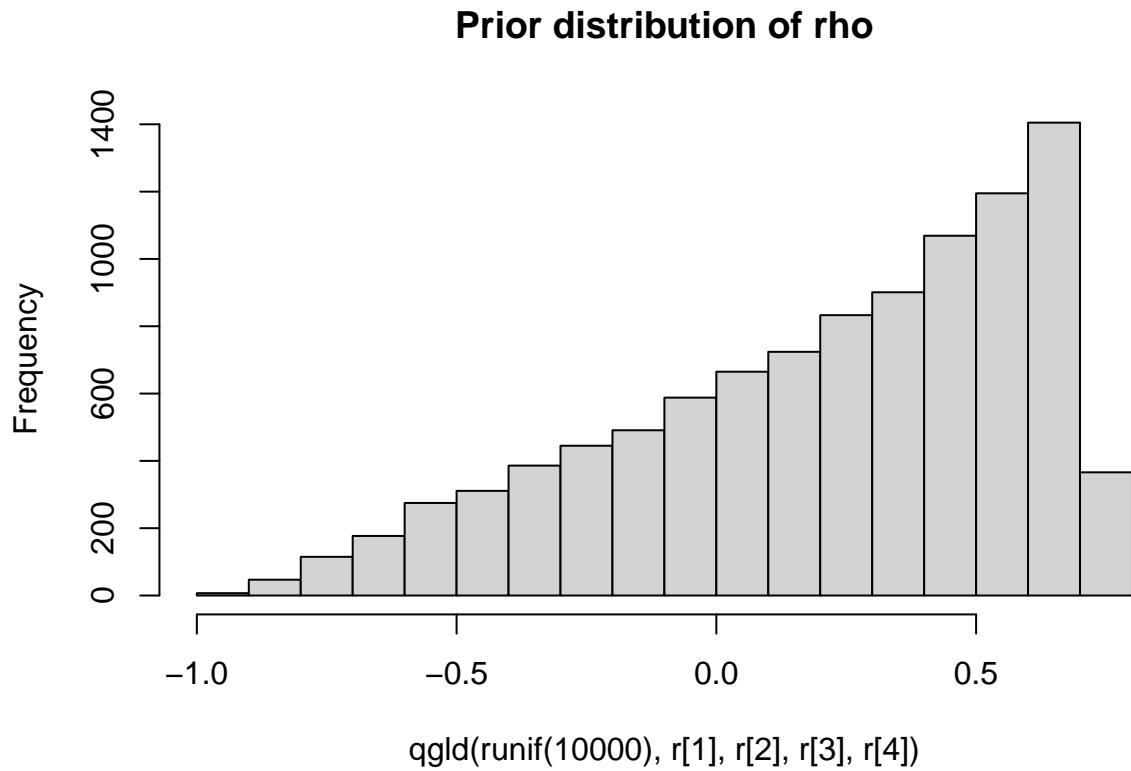
```

r <- c(0.3, 0.6, a_s[1], a_s[2])
r

##                                asymmetry    steepness
## 0.30000000 0.60000000 -0.87521436 0.03133668

hist(qgld(runif(10000), r[1], r[2], r[3], r[4]), main = "Prior distribution of rho")

```



```

# display stan code
writeLines(readLines("iv_bin.stan"))

## #include quantile_functions.stan
## data {
##   int<lower = 0> N;
##   int<lower = 0> N_child;
##   int<lower = 0> N_nochild;
##   int<lower = 0> K; // number of predictors
##   matrix[N_child, K + 1] X_child_s;
##   matrix[N_nochild, K + 1] X_nochild_s;
##   matrix[N_child, K] X_child;
##   matrix[N_nochild, K] X_nochild;
##   vector[N_child] y_child;
##   vector[N_nochild] y_nochild;
##   int<lower = 0, upper = 1> prior_only;

```

```

##  vector[5] m;
##  vector<lower = 0>[5] scale;
##  vector[4] r;
## }
##
## parameters {
##   real<lower = 0> sigma;
##   real<lower = 0, upper = 1> p; // CDF for rho
##   real alpha0;
##   vector[K + 1] alpha1;
##   real beta0;
##   vector[K] beta1;
##   real beta2;
## }
##
## transformed parameters {
##   real<lower = -1, upper = 1> rho = gld_qf(p, r[1], r[2], r[3], r[4]);
##   vector[N_child] mu_child = beta0 + X_child*beta1 + beta2;
##   vector[N_nochild] mu_nochild = beta0 + X_nochild*beta1;
##   vector[N_child] eta_child = alpha0 + X_child_s * alpha1 +
##     rho / sigma * (y_child - mu_child);
##   vector[N_nochild] eta_nochild = alpha0 + X_nochild_s * alpha1 +
##     rho / sigma * (y_nochild - mu_nochild);
##   real conditional_sd = sqrt(1 - square(rho));
## }
##
## model {
##   if (!prior_only) {
##     target += normal_lpdf(y_child | mu_child, sigma);
##     target += normal_lpdf(y_nochild | mu_nochild, sigma);
##     target += normal_lcdf(0 | eta_child, conditional_sd);
##     target += normal_lcdf(0 | -eta_nochild, conditional_sd);
##   }
##   target += normal_lpdf(alpha0 | m[1], scale[1]);
##   target += normal_lpdf(alpha1 | m[2], scale[2]);
##   target += normal_lpdf(beta0 | m[3], scale[3]);
##   target += normal_lpdf(beta1 | m[4], scale[4]);
##   target += normal_lpdf(beta2 | m[5], scale[5]);
##   target += gamma_lpdf(sigma | 2, 2);
## } // implicit: p ~ uniform(0, 1)
##
## generated quantities {
##   vector[N] log_lik;
##   int moves_rep[N];
##   vector[N] yrep;
##   real mu;
##   {
##     for (n in 1:N_child) {
##       log_lik[n] = normal_lpdf(y_child[n] | mu_child[n], sigma) +
##                   normal_lcdf(0 | eta_child[n], conditional_sd);
##       mu = mu_child[n];
##     }
##   }
## }
```

```

##      /* intermediate outcome: if they have an additional child */
##      moves_rep[n] = normal_rng(eta_child[n], conditional_sd) > 0;
##      if (moves_rep[n] == 0) mu -= beta2; // subtract beta if not
##      yrep[n] = normal_rng(mu, sigma); // sample outcome from new normal distribution
##  }
## 
##      for (n in 1:N_nochild) {
##          log_lik[N_child + n] = normal_lpdf(y_nochild[n] | mu_nochild[n], sigma) +
##              normal_lcdf(0 | -eta_nochild[n], conditional_sd);
##          mu = mu_nochild[n];
## 
##          /* intermediate outcome: if they have an additional child */
##          moves_rep[N_child + n] = normal_rng(eta_nochild[n], conditional_sd) > 0;
##          if (moves_rep[N_child + n] == 1) mu += beta2; // add beta if yes
##          yrep[N_child + n] = normal_rng(mu, sigma); // sample outcome from new normal distribution
##      }
##  }
## }

# 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,
                      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 = TRUE,
                      m = rep(-0.1, 5),
                      scale = rep(0.3, 5), r = r)

# call program without data for prior predictive checks
pre_iv <- stan("iv_bin.stan", data = stan_data_iv, seed = 1234)
print(pre_iv, pars = c("alpha0", "alpha1", "beta0", "beta1", "beta2", "sigma", "rho"))

## Inference for Stan model: iv_bin.
## 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

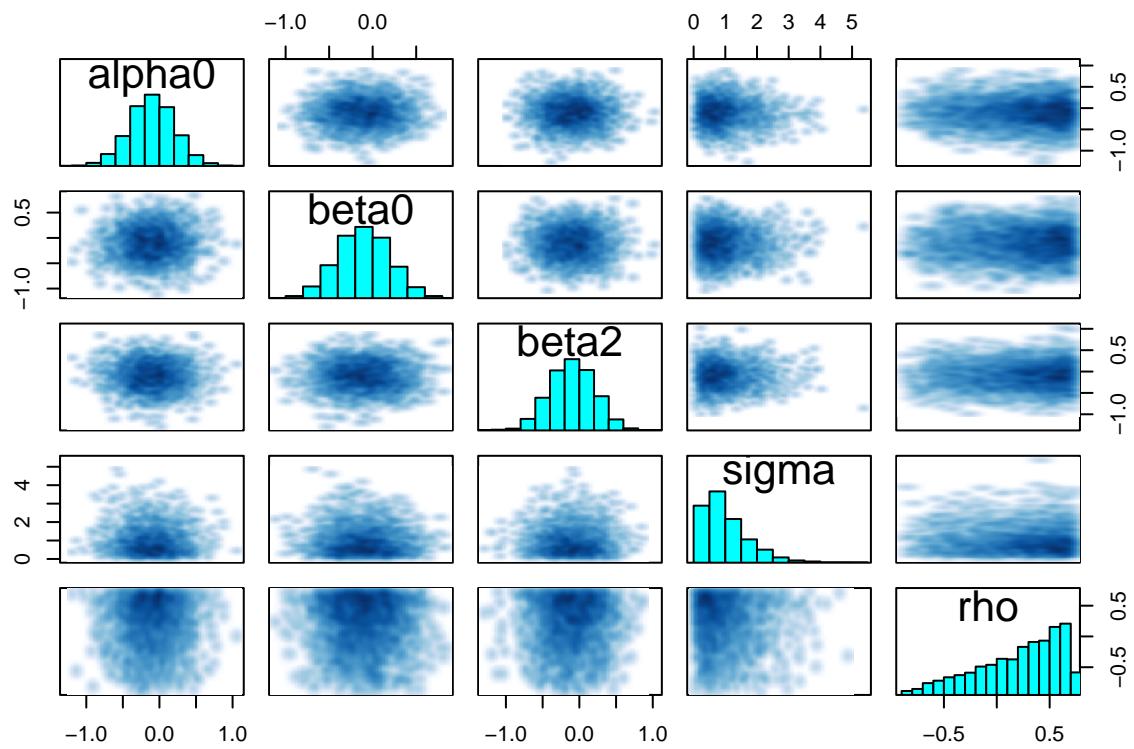
```

```

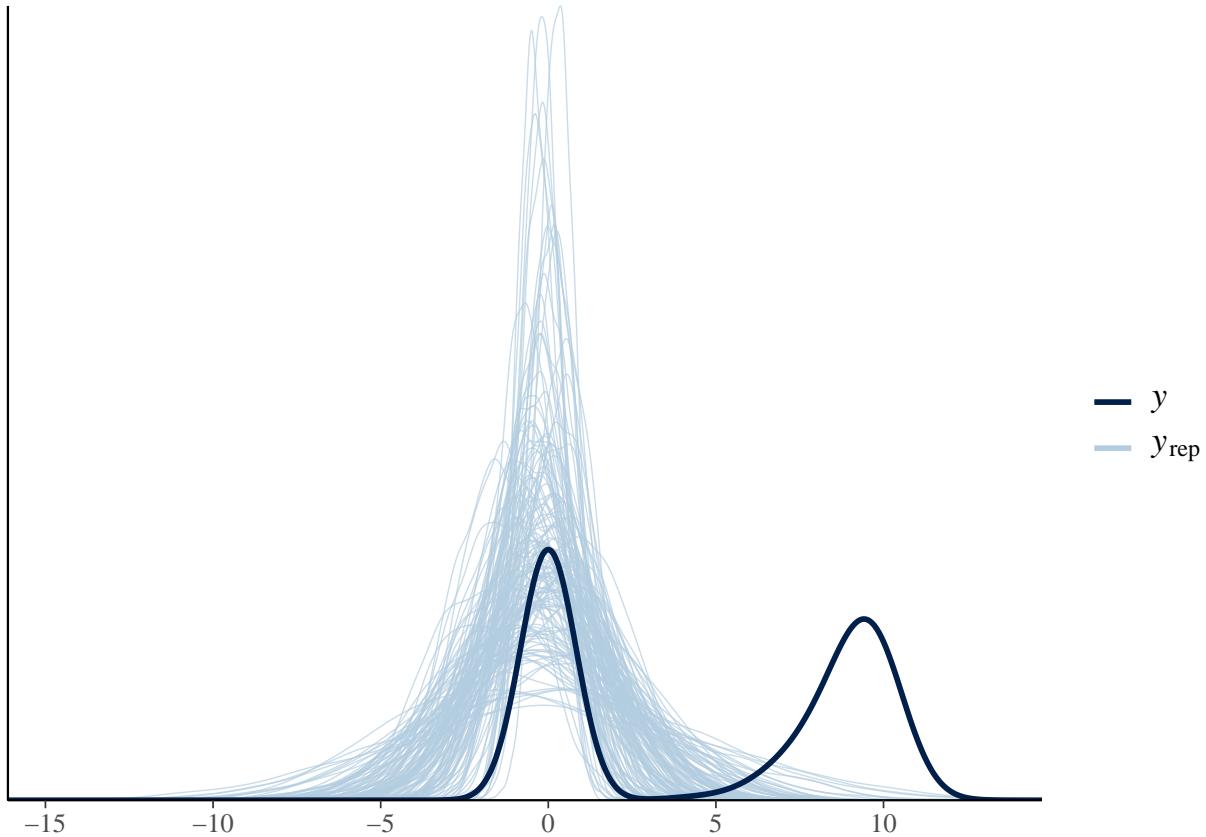
## alpha0      -0.10   0.00 0.31 -0.72 -0.30 -0.10 0.10   0.50   8300   1
## alpha1[1]   -0.10   0.00 0.30 -0.66 -0.30 -0.10 0.10   0.47   8472   1
## alpha1[2]   -0.10   0.00 0.29 -0.66 -0.28 -0.10 0.09   0.48   9568   1
## alpha1[3]   -0.10   0.00 0.30 -0.69 -0.30 -0.10 0.11   0.51   8857   1
## alpha1[4]   -0.10   0.00 0.30 -0.70 -0.30 -0.10 0.09   0.48  10157   1
## alpha1[5]   -0.10   0.00 0.31 -0.69 -0.30 -0.09 0.11   0.49   7979   1
## alpha1[6]   -0.10   0.00 0.30 -0.71 -0.31 -0.10 0.11   0.48   8589   1
## alpha1[7]   -0.10   0.00 0.29 -0.67 -0.29 -0.10 0.10   0.46   9680   1
## alpha1[8]   -0.10   0.00 0.30 -0.70 -0.30 -0.11 0.10   0.49   7614   1
## beta0       -0.10   0.00 0.30 -0.68 -0.32 -0.11 0.11   0.48   9655   1
## beta1[1]    -0.10   0.00 0.30 -0.71 -0.30 -0.10 0.10   0.49   8619   1
## beta1[2]    -0.10   0.00 0.30 -0.69 -0.30 -0.10 0.09   0.49   8998   1
## beta1[3]    -0.10   0.00 0.30 -0.69 -0.31 -0.10 0.09   0.50   9737   1
## beta1[4]    -0.10   0.00 0.30 -0.69 -0.30 -0.11 0.10   0.50   8970   1
## beta1[5]    -0.10   0.00 0.30 -0.68 -0.29 -0.10 0.10   0.50   7880   1
## beta1[6]    -0.09   0.00 0.29 -0.66 -0.29 -0.09 0.10   0.48   8064   1
## beta1[7]    -0.10   0.00 0.29 -0.66 -0.30 -0.10 0.10   0.47   8828   1
## beta2       -0.10   0.00 0.30 -0.69 -0.31 -0.10 0.11   0.48   8796   1
## sigma       1.01    0.01 0.71  0.13  0.48  0.85 1.35  2.79  7311   1
## rho         0.21    0.01 0.39 -0.67 -0.06  0.29 0.54  0.71  6054   1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 08:55:39 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(pre_iv, pars = c("alpha0", "beta0", "beta2", "sigma", "rho"))
```



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



```

stan_data_iv$prior_only <- FALSE

post_iv <- stan("iv_bin.stan", data = stan_data_iv, seed = 1234)
print(post_iv, pars = c("alpha0", "alpha1", "beta0", "beta1", "beta2", "sigma", "rho"))

## Inference for Stan model: iv_bin.
## 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
## alpha0      0.27     0.05  0.17  0.23  0.27  0.30  0.36  0.4154    1
## alpha1[1]  -0.12     0.05  0.21 -0.21 -0.16 -0.12 -0.09 -0.03  7859    1
## alpha1[2]  -0.08     0.01  0.09 -0.09 -0.08 -0.08 -0.07 -0.07  5752    1
## alpha1[3]   0.13     0.01  0.11  0.13  0.13  0.14  0.15  0.16  6031    1
## alpha1[4]   0.17     0.05  0.08  0.14  0.17  0.20  0.26  0.26  7339    1
## alpha1[5]   0.07     0.05  0.02  0.04  0.07  0.10  0.17  0.17  6263    1
## alpha1[6]  -0.24     0.07  0.38 -0.38 -0.29 -0.24 -0.19 -0.10  7307    1
## alpha1[7]  -0.31     0.08  0.47 -0.47 -0.36 -0.31 -0.26 -0.15  8684    1
## alpha1[8]  -0.41     0.14  0.69 -0.51 -0.41 -0.32 -0.14 -0.14  7735    1
## beta0       3.42     0.16  3.12  3.31  3.42  3.53  3.72  3.807   1
## beta1[1]    0.16     0.03  0.11  0.14  0.16  0.18  0.21  0.21  5052    1
## beta1[2]   -0.16     0.03  0.23 -0.23 -0.19 -0.16 -0.14 -0.10  4697    1
## beta1[3]    0.51     0.14  0.23  0.42  0.52  0.61  0.79  0.89  6930    1
## beta1[4]    0.37     0.14  0.09  0.27  0.37  0.46  0.64  0.67  6774    1
## beta1[5]    0.75     0.20  0.36  0.63  0.75  0.88  1.15  1.15  6467    1

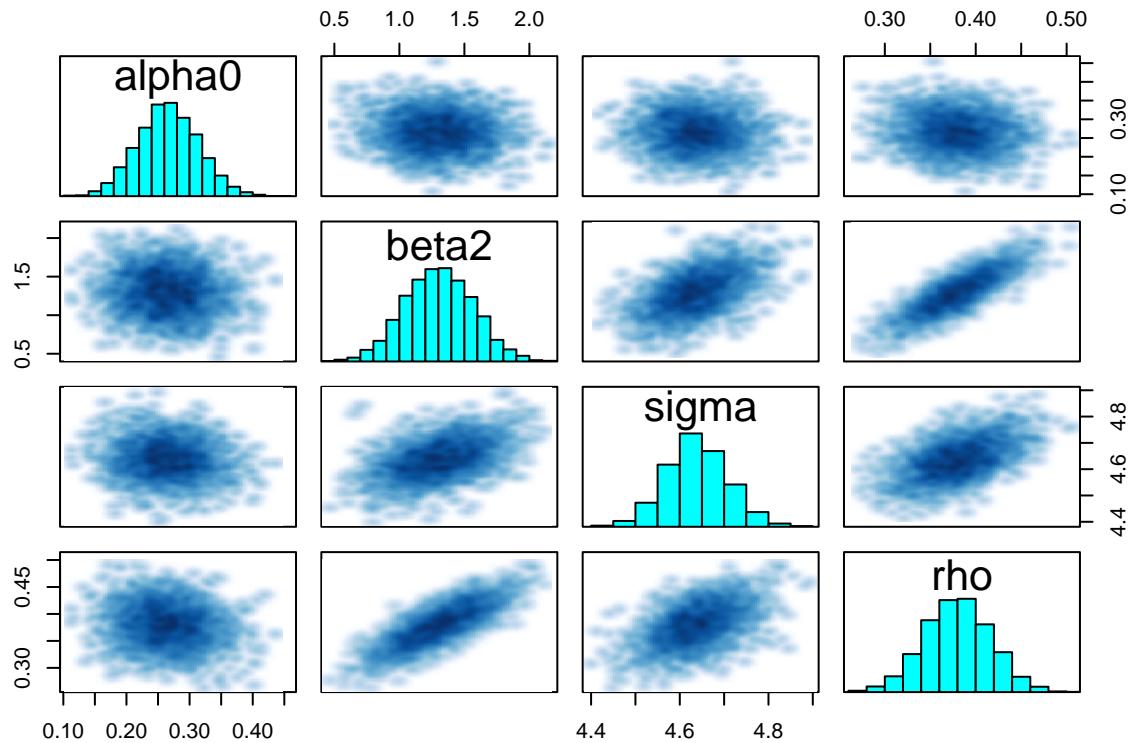
```

```

## beta1[6] -0.08      0 0.21 -0.48 -0.22 -0.08  0.07  0.32  6909   1
## beta1[7] -0.12      0 0.27 -0.65 -0.29 -0.12  0.05  0.42  8064   1
## beta2     1.30       0 0.26  0.79  1.12  1.30  1.49  1.82  3052   1
## sigma     4.64       0 0.07  4.50  4.59  4.64  4.68  4.78  4366   1
## rho       0.38       0 0.04  0.31  0.36  0.38  0.40  0.45  3169   1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 09:01:44 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).


```

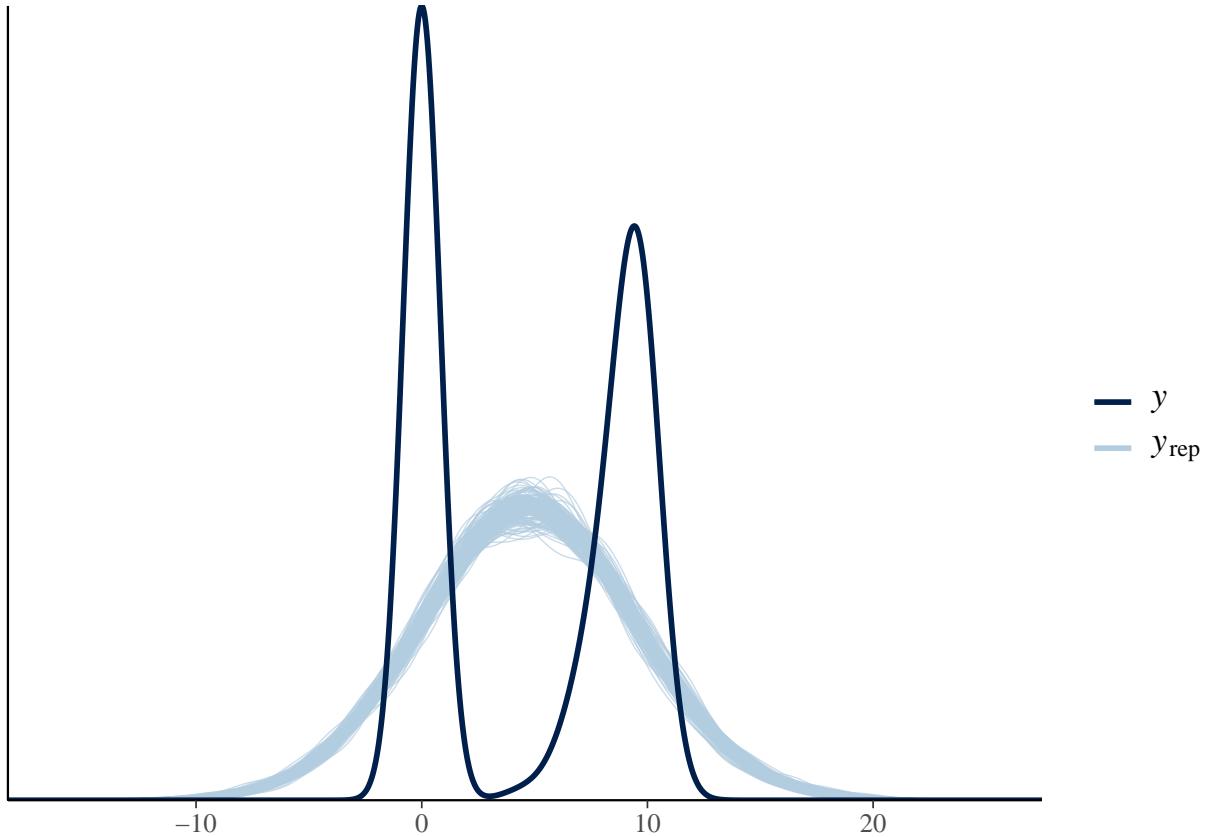
```
pairs(post_iv, pars = c("alpha0", "beta2", "sigma", "rho"))
```



```

pp_check(c(as.numeric(stan_data_iv$y_child), as.numeric(stan_data_iv$y_nochild)),
         rstan::extract(post_iv, par = "yrep")$yrep[sample(1:length(stan_data$y), size = 150), ],
         ppc_dens_overlay
)

```



```
loo_post_iv <- loo(post_iv)
loo_post_iv
```

```
##
## Computed from 4000 by 3000 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo -10597.2 23.2
## p_loo      15.0  0.3
## looic     21194.4 46.3
## -----
## 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.
```

Re-run with weeks worked outcome

```
# set stan data
stan_data_iv <- list(N = nrow(df_samp),
                      N_child = nrow(df_child),
                      N_nochild = nrow(df_nochild),
                      K = 7,
```

```

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$wkswork1,
y_nochild = df_nochild$wkswork1,
prior_only = FALSE,
m = rep(-0.1, 5),
scale = rep(0.3, 5), r = r)

post_iv_wk <- stan("iv_bin.stan", data = stan_data_iv, seed = 1234)
print(post_iv_wk, pars = c("alpha0", "alpha1", "beta0", "beta1", "beta2", "sigma", "rho"))

## Inference for Stan model: iv_bin.
## 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
## alpha0     0.15      0 0.05  0.05  0.11  0.15  0.18  0.25  4518     1
## alpha1[1] -0.13      0 0.05 -0.22 -0.16 -0.13 -0.10 -0.05  6730     1
## alpha1[2] -0.08      0 0.01 -0.10 -0.09 -0.08 -0.08 -0.07  6754     1
## alpha1[3]  0.13      0 0.01  0.11  0.13  0.13  0.14  0.15  7150     1
## alpha1[4]  0.15      0 0.05  0.06  0.12  0.15  0.19  0.25  6461     1
## alpha1[5]  0.07      0 0.05 -0.02  0.04  0.07  0.10  0.16  6436     1
## alpha1[6] -0.28      0 0.07 -0.42 -0.33 -0.28 -0.24 -0.15  7619     1
## alpha1[7] -0.31      0 0.08 -0.46 -0.36 -0.31 -0.25 -0.15  7640     1
## alpha1[8] -0.43      0 0.14 -0.71 -0.52 -0.43 -0.34 -0.16  7116     1
## beta0      5.07      0 0.28  4.51  4.89  5.08  5.25  5.61  6271     1
## beta1[1]   0.86      0 0.12  0.63  0.78  0.86  0.94  1.10  6362     1
## beta1[2]  -0.71      0 0.14 -0.98 -0.80 -0.71 -0.62 -0.44  6291     1
## beta1[3]   2.39      0 0.28  1.85  2.20  2.39  2.58  2.93  7952     1
## beta1[4]   2.29      0 0.28  1.74  2.10  2.29  2.48  2.84  6554     1
## beta1[5]   0.78      0 0.29  0.21  0.58  0.78  0.98  1.35  7587     1
## beta1[6]   0.24      0 0.29 -0.34  0.04  0.24  0.44  0.83  7078     1
## beta1[7]   0.01      0 0.29 -0.56 -0.19  0.01  0.21  0.57  7701     1
## beta2      2.30      0 0.29  1.73  2.10  2.30  2.50  2.89  7097     1
## sigma     25.05      0 0.35 24.37 24.82 25.05 25.29 25.76  6963     1
## rho       0.32      0 0.03  0.27  0.30  0.32  0.33  0.37  5098     1
##
## Samples were drawn using NUTS(diag_e) at Mon May 16 09:08:53 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).


```

Re-run two-stage model for college subgroup

```

# subset the data
df_samp2 <- df %>%
  filter(coll == 1) %>%
  sample_n(3000) %>%
  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)))

df_child <- df_samp2 %>%
  filter(cnum_mt2 == 1)

df_nochild <- df_samp2 %>%
  filter(cnum_mt2 == 0)

# set stan data
stan_data_iv <- list(N = nrow(df_samp2),
                      N_child = nrow(df_child),
                      N_nochild = nrow(df_nochild),
                      K = 7,
                      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 = r)

post_iv_coll <- stan("iv_bin.stan", data = stan_data_iv, seed = 1234)
print(post_iv_coll, pars = c("alpha0", "alpha1", "beta0", "beta1", "beta2", "sigma", "rho"))

## Inference for Stan model: iv_bin.
## 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
## alpha0     0.48      0 0.05  0.38  0.45  0.48  0.51  0.57  4234     1
## alpha1[1] -0.17      0 0.05 -0.26 -0.20 -0.17 -0.14 -0.08  7663     1
## alpha1[2] -0.08      0 0.01 -0.10 -0.09 -0.08 -0.08 -0.06  6682     1
## alpha1[3]  0.11      0 0.01  0.09  0.10  0.11  0.11  0.12  6335     1
## alpha1[4]  0.01      0 0.05 -0.09 -0.03  0.00  0.04  0.10  4774     1
## alpha1[5]  0.09      0 0.05  0.00  0.06  0.09  0.12  0.19  6801     1
## alpha1[6]  0.05      0 0.08 -0.10  0.00  0.05  0.11  0.21  7166     1
## alpha1[7] -0.04      0 0.11 -0.25 -0.11 -0.04  0.04  0.18  7578     1
## alpha1[8] -0.13      0 0.11 -0.35 -0.21 -0.13 -0.05  0.09  8248     1
## beta0      3.81      0 0.15  3.51  3.71  3.82  3.92  4.11  3769     1
## beta1[1]   0.16      0 0.03  0.10  0.14  0.16  0.18  0.22  5735     1
## beta1[2]  -0.24      0 0.03 -0.31 -0.27 -0.25 -0.22 -0.18  5271     1
## beta1[3]   0.24      0 0.15 -0.03  0.15  0.24  0.34  0.53  6553     1
## beta1[4]   0.54      0 0.15  0.26  0.44  0.54  0.64  0.82  7020     1
## beta1[5]   1.18      0 0.21  0.76  1.03  1.18  1.33  1.60  7893     1
## beta1[6]   0.24      0 0.24 -0.23  0.07  0.24  0.41  0.72  7284     1
## beta1[7]   0.44      0 0.26 -0.06  0.26  0.44  0.62  0.94  9330     1
## beta2      1.08      0 0.27  0.55  0.90  1.07  1.25  1.61  3584     1
## sigma      4.65      0 0.07  4.51  4.60  4.65  4.69  4.78  4580     1
## rho        0.37      0 0.04  0.30  0.35  0.37  0.40  0.44  3728     1
##

```

```

## Samples were drawn using NUTS(diag_e) at Mon May 16 09:45:49 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).

```

Frequentist Heckman example (for reference)

```

library(sampleSelection)

## Warning: package 'sampleSelection' was built under R version 4.1.3

## Loading required package: maxLik

## Warning: package 'maxLik' was built under R version 4.1.3

## Loading required package: miscTools

## Warning: package 'miscTools' was built under R version 4.1.3

## 
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. C
## 
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum o
## https://r-forge.r-project.org/projects/maxlik/

## two-stage estimator
twostage_fit <- treatReg(
  selection = cnum_mt2 ~ samesex + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth,
  outcome = l_incwage ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth,
  data = df_samp
)
summary(twostage_fit)

## -----
## Tobit treatment model (switching regression model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 3 iterations
## Return code 8: successive function values within relative tolerance limit (reltol)
## Log-Likelihood: -10567.72
## 3000 observations: 1808 non-participants (selection 0) and 1192 participants (selection 1)
## 
## 20 free parameters (df = 2980)
## Probit selection equation:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.339548  0.050289 -6.752 1.75e-11 ***
## samesex      0.146264  0.049109  2.978 0.002921 **
## age         0.081076  0.007319 11.077 < 2e-16 ***
## age_fbirth -0.132303  0.009109 -14.524 < 2e-16 ***

```

```

## f_boy      -0.136818  0.048226 -2.837 0.004585 **
## s_boy      -0.041062  0.048317 -0.850 0.395474
## r_black     0.232856  0.074290  3.134 0.001739 **
## hisp       0.358673  0.087225  4.112 4.03e-05 ***
## r_oth       0.533439  0.161498  3.303 0.000968 ***
## Outcome equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.84718   0.80165  6.047 1.66e-09 ***
## cnum_mt2    -0.52134   1.95110 -0.267 0.789330
## age        0.21123   0.06201  3.406 0.000667 ***
## age_fbirth -0.24230   0.09490 -2.553 0.010721 *
## f_boy       0.06666   0.18468  0.361 0.718169
## s_boy      -0.08369   0.16570 -0.505 0.613539
## r_black     1.19006   0.30784  3.866 0.000113 ***
## hisp       -0.13850   0.38704 -0.358 0.720477
## r_oth      -0.21774   0.67182 -0.324 0.745885
## Error terms:
##             Estimate Std. Error t value Pr(>|t|)
## sigma     4.4436    0.1182  37.603 <2e-16 ***
## rho      -0.1500    0.2640  -0.568     0.57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----
## two-stage estimator
twostage_fit <- treatReg(
  selection = cnum_mt2 ~ samesex + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth,
  outcome = incwage ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth,
  data = df_samp
)
summary(twostage_fit)

## -----
## Tobit treatment model (switching regression model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 14 iterations
## Return code 8: successive function values within relative tolerance limit (reltol)
## Log-Likelihood: -33697.15
## 3000 observations: 1808 non-participants (selection 0) and 1192 participants (selection 1)
##
## 20 free parameters (df = 2980)
## Probit selection equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.297627  0.046299 -6.428 1.5e-10 ***
## samesex      0.115191  0.036460  3.159 0.001597 **
## age         0.071116  0.007091 10.028 < 2e-16 ***
## age_fbirth -0.123429  0.008708 -14.175 < 2e-16 ***
## f_boy       -0.151919  0.046287 -3.282 0.001042 **
## s_boy      -0.054922  0.046272 -1.187 0.235349
## r_black     0.173898  0.073363  2.370 0.017834 *
## hisp       0.321245  0.085154  3.773 0.000165 ***
## r_oth       0.436578  0.160011  2.728 0.006401 **
## Outcome equation:
##             Estimate Std. Error t value Pr(>|t|)
```

```

## (Intercept) 1357.36    494.52   2.745  0.00609 ** 
## cnum_mt2    12102.63   670.16  18.059 < 2e-16 *** 
## age         121.20     71.09   1.705  0.08831 .  
## age_fbirth  272.96     87.12   3.133  0.00175 ** 
## f_boy       705.57     453.91   1.554  0.12019 
## s_boy       283.06     452.86   0.625  0.53198 
## r_black    2221.94     714.83   3.108  0.00190 ** 
## hisp        -1454.92    832.00  -1.749  0.08045 . 
## r_oth      -1014.44    1561.98  -0.649  0.51610 
## Error terms: 
##           Estimate Std. Error t value Pr(>|t|) 
## sigma    1.238e+04  2.421e+02  51.12 <2e-16 *** 
## rho     -7.848e-01  1.758e-02 -44.65 <2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## ----- 

## two-stage estimator 
twostage_fit <- treatReg( 
  selection = cnum_mt2 ~ samesex + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth, 
  outcome = wkswork1 ~ cnum_mt2 + age + age_fbirth + f_boy + s_boy + r_black + hisp + r_oth, 
  data = df_samp 
)
summary(twostage_fit)

## ----- 
## Tobit treatment model (switching regression model) 
## Maximum Likelihood estimation 
## Newton-Raphson maximisation, 5 iterations 
## Return code 8: successive function values within relative tolerance limit (reltol) 
## Log-Likelihood: -15321.01 
## 3000 observations: 1808 non-participants (selection 0) and 1192 participants (selection 1) 
## 
## 20 free parameters (df = 2980) 
## Probit selection equation: 
##           Estimate Std. Error t value Pr(>|t|) 
## (Intercept) -0.341321  0.050105 -6.812 1.16e-11 *** 
## samesex      0.152167  0.047691  3.191  0.00143 ** 
## age          0.080980  0.007329 11.049 < 2e-16 *** 
## age_fbirth  -0.131915  0.009128 -14.451 < 2e-16 *** 
## f_boy        -0.131687  0.048258 -2.729  0.00639 ** 
## s_boy        -0.046461  0.048283 -0.962  0.33600 
## r_black      0.233100  0.074208  3.141  0.00170 ** 
## hisp         0.355926  0.087030  4.090 4.43e-05 *** 
## r_oth        0.526677  0.160625  3.279  0.00105 ** 
## Outcome equation: 
##           Estimate Std. Error t value Pr(>|t|) 
## (Intercept) 25.60241   5.08082   5.039 4.96e-07 *** 
## cnum_mt2   -14.69935  12.45436  -1.180  0.23799 
## age         1.60771   0.38286   4.199 2.76e-05 *** 
## age_fbirth -1.65273   0.59381  -2.783  0.00542 ** 
## f_boy       0.06335   0.97283   0.065  0.94808 
## s_boy      -0.48786   0.82204  -0.593  0.55291 
## r_black     7.48252   1.66266   4.500 7.04e-06 ***

```

```
## hisp          0.34884   2.15017   0.162  0.87113
## r_oth         2.99973   3.62649   0.827  0.40821
##   Error terms:
##       Estimate Std. Error t value Pr(>|t|)
## sigma  21.7586    0.8857  24.568  <2e-16 ***
## rho     0.1910    0.3418   0.559    0.576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----
```