APSTA-GE 2123 Assignment 1 Answer Key

Ben Goodrich

1 Oregon Medicaid Experiment

```r
J <- 50000 # number of households
dataset <- data.frame(household_ID = as.factor(unlist(lapply(1:J, FUN = function(j) {
  rep(j, each = sample(1:3, size = 1, prob = c(0.5, 0.3, 0.2)))
}))))
selection <- rbinom(nrow(dataset), size = 1, prob = 0.2)
dataset$lottery <- ave(selection, dataset$household_ID, FUN = any)
dataset$numhh <- as.factor(ave(dataset$lottery, dataset$household_ID, FUN = length))
```

1.1 Actual Prior Predictive Distribution

```r
rstan::expose_stan_functions("quantile_functions.stan")
source("GLD_helpers.R")
```

1.2 Index Variable Approach

If you use index variables, then

$$\mu = \alpha_{numhh} + \beta \times LOTTERY$$

The distribution of income at the low end is irregular because about 24% of people have no (reported) income, according to table A14.

```r
a_s_nhh <- GLD_solver_bounded(bounds = c(0, 35000), median = 19000, IQR = 5000) # warnings OK
```

```
## Warning in GLD_solver_bounded(bounds = c(0, 35000), median = 19000, IQR = 5000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 0.00907276484213071 and 34999.9919725692
```

```r
a_s_trt <- GLD_solver(lower_quartile = -100, median = 250, upper_quartile = 500,
                      other_quantile = 1000, alpha = 0.9)
a_s_sig <- GLD_solver(lower_quartile = 8000, median = 10000, upper_quartile = 12000,
                      other_quantile = 0, alpha = 0) # warning OK
```

```
## Warning in GLD_solver(lower_quartile = 8000, median = 10000, upper_quartile =
## 12000, : solution implies a bounded upper tail at 20000.8906460174
```

You could certainly use different values for the quantiles, although α_{numhh} and σ should definitely have a lower bound of zero.

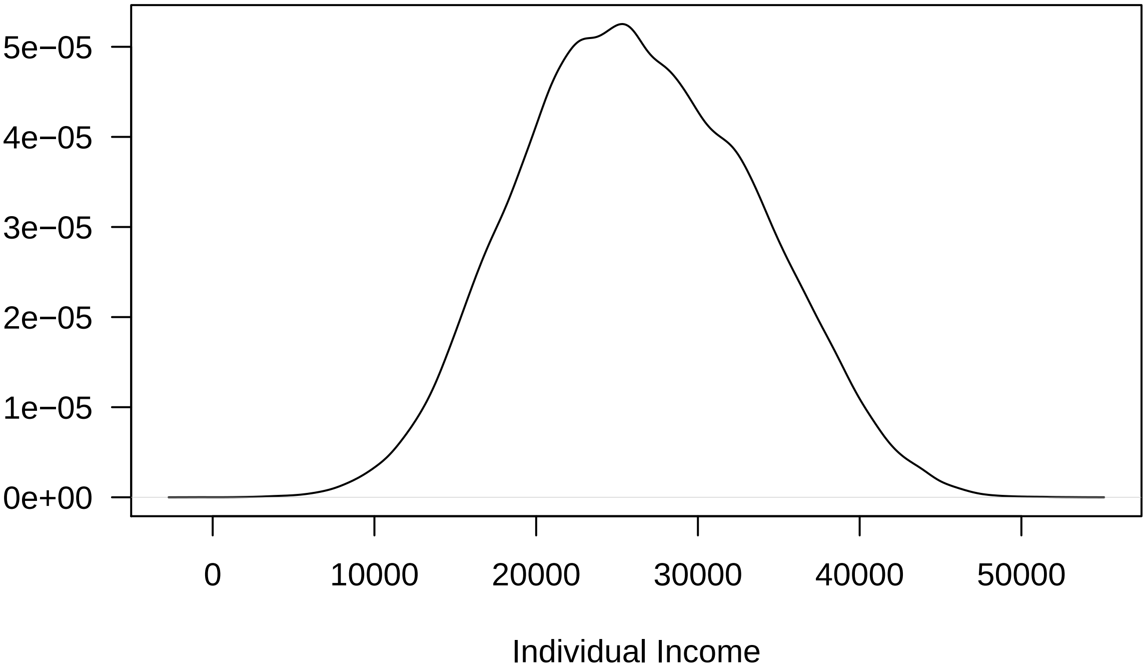
We can check whether our prior on each element of α_{numhh} has the right expectation by doing

```r
Q <- Vectorize(GLD_icdf, vectorize.args = "p")
integrate(Q, lower = 0, upper = 1, median = 19000, IQR = 5000,
          asymmetry = a_s_nhh[1], steepness = a_s_nhh[2])$value * .76
```

```
## [1] 14413.21
```

Then we can draw once from the prior predictive distribution:

```
beta_nhh <- replicate(3, GLD_rng(median = 19000, IQR = 5000,
                                 a_s_nhh[1], steepness = a_s_nhh[2]))
beta_trt <- GLD_rng(median = 250, IQR = 500 - -100,
                    asymmetry = a_s_trt[1], steepness = a_s_trt[2])
mu_ <- with(dataset, beta_trt * lottery + beta_nhh[as.integer(numhh)])
sigma_ <- GLD_rng(median = 10000, IQR = 12000 - 8000,
                  asymmetry = a_s_sig[1], steepness = a_s_sig[2])
epsilon_ <- rnorm(length(mu_), mean = 0, sd = sigma_)
income_ <- mu_ + epsilon_
plot(density(income_), las = 1, main = "", ylab = "", xlab = "Individual Income")
```



It should not put too much prior probability on negative values, but a tiny bit is OK. This is more of a consequence of using a normal conditional distribution for income, which has to be non-negative.

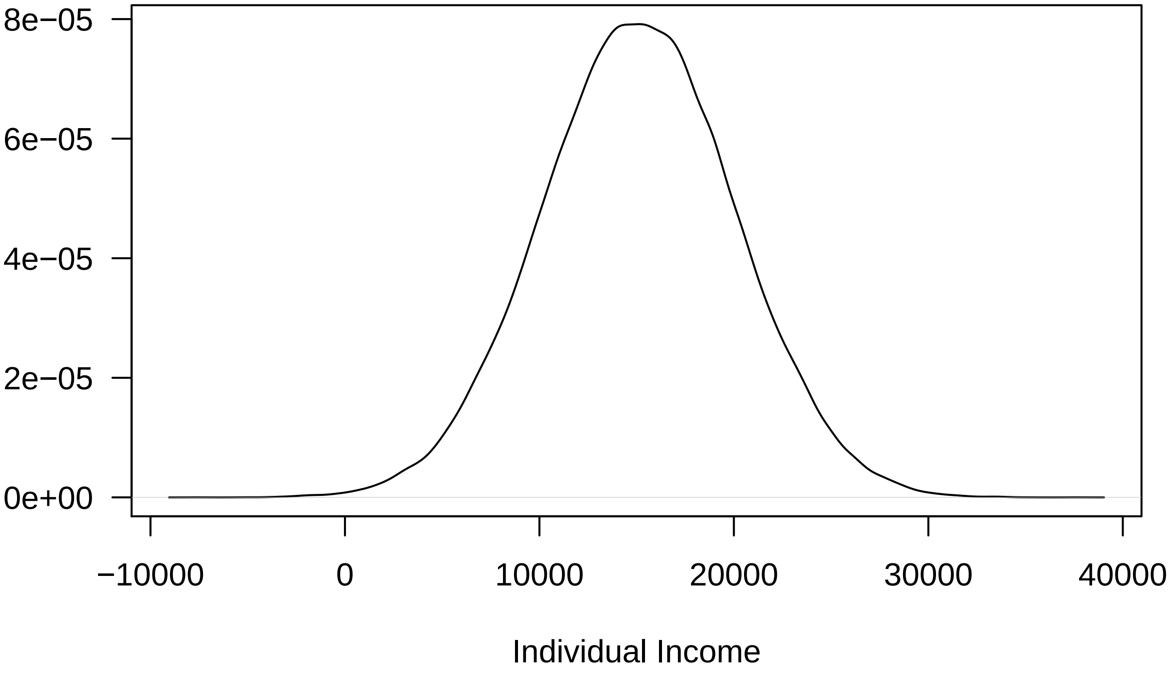
1.2.1 Dummy Variable Approach

If you use dummy variables for the number of adults in the household relative to a reference category,

$$\mu = \alpha + \beta \times LOTTERY + \gamma \times \mathcal{I}\{NUMHH == 2\} + \lambda \times \mathcal{I}\{NUMHH == 3\}$$

We can then go through the same steps to draw from the prior predictive distribution:

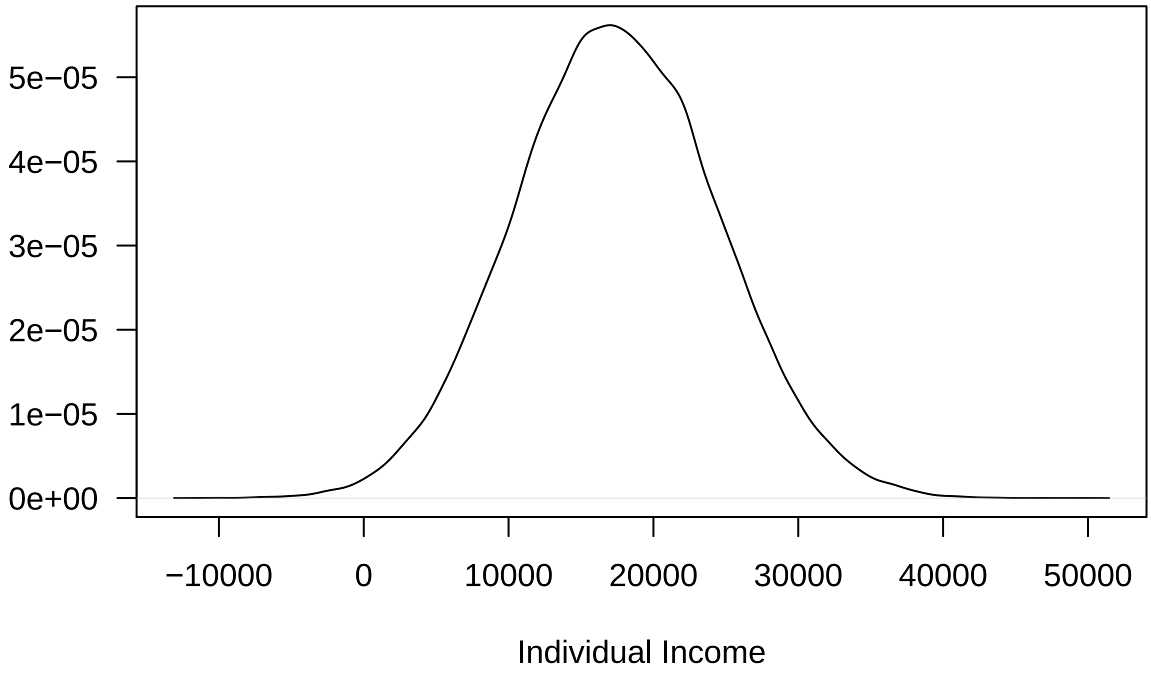
```
a_s_alpha <- a_s_nhh
a_s_gamma <- GLD_solver(lower_quartile = -250, median = 0, upper_quartile = 250,
                        other_quantile = 600, alpha = 0.9)

alpha_ <- GLD_rng(median = 19000, IQR = 5000,
                  asymmetry = a_s_alpha[1], steepness = a_s_alpha[2])
beta_trt <- GLD_rng(median = 250, IQR = 500 - -100,
                    asymmetry = a_s_trt[1], steepness = a_s_trt[2])
gamma_ <- replicate(2, GLD_rng(median = 0, IQR = 250 - -250,
                               asymmetry = a_s_gamma[1], steepness = a_s_gamma[2]))
mu_ <- with(dataset, alpha_ + beta_trt * lottery +
              gamma_[1] * (numhh == 2) + gamma_[2] * (numhh == 3))
epsilon_ <- rnorm(length(mu_), mean = 0, sd = sigma_)
income_ <- mu_ + epsilon_
plot(density(income_), las = 1, main = "", ylab = "", xlab = "Individual Income")
```



1.3 Prior Predictive Distribution for a Journal

If we were forced to use a prior median of zero for the treatment effect, we would need to artificially inflate the prior uncertainty in order to put a reasonable chance on values of the treatment effect that we think are plausible:

```
a_s_trt <- GLD_solver(lower_quartile = -325, median = 0, upper_quartile = 350,
                      other_quantile = 250, alpha = 0.7)
beta_nhh <- replicate(3, GLD_rng(median = 19000, IQR = 5000,
                                 a_s_nhh[1], steepness = a_s_nhh[2]))
beta_trt <- GLD_rng(median = 0, IQR = 350 - -325,
                    asymmetry = a_s_trt[1], steepness = a_s_trt[2])
mu_ <- with(dataset, beta_trt * lottery + beta_nhh[as.integer(numhh)])
sigma_ <- GLD_rng(median = 10000, IQR = 12000 - 8000,
                  asymmetry = a_s_sig[1], steepness = a_s_sig[2])
epsilon_ <- rnorm(length(mu_), mean = 0, sd = sigma_)
income_ <- mu_ + epsilon_
plot(density(income_), las = 1, main = "", ylab = "", xlab = "Individual Income")
```



Sometimes, putting a prior with a median of zero in order to be "neutral" can result in a prior predictive distribution that does not make as much sense.

2 2018 American Community Survey

First, we load the data, in this case from New York, although everyone had a different state:

```
dataset <- readr::read_csv(dir(pattern = "csv$"))
dataset <- dataset[ , !startsWith(colnames(dataset), prefix = "PWG")]
dataset <- dataset[ , !startsWith(colnames(dataset), prefix = "F")]
dataset <- dataset[!is.na(dataset$WAGP) & dataset$WAGP > 0, ]
```

2.1 Posterior Distribution

You may need to do some recoding of the predictors you chose,

```
dataset$SCHL <- as.integer(dataset$SCHL)
dataset$SEX <- as.integer(dataset$SEX) - 1L
dataset$AGEP <- dataset$AGEP / 10
library(rstanarm)
```

```
## Loading required package: Rcpp

## rstanarm (Version 2.18.2, packaged: 2018-11-08 22:19:38 UTC)

## - Do not expect the default priors to remain the same in future rstanarm versions.

## Thus, R scripts should specify priors explicitly, even if they are just the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

## options(mc.cores = parallel::detectCores())

## - Plotting theme set to bayesplot::theme_default().
```

```
options(mc.cores = parallel::detectCores())
```

before drawing from the implied posterior distribution:

```r
post <- stan_lm(log(WAGP) ~ SCHL + SEX + AGEP + I(AGEP ^ 2), data = dataset,
                prior = R2(0.25, what = "mode"), adapt_delta = 0.999, seed = 123)
```

In general, you should try to overcome any divergent transition warnings, but it is OK if you did not do so on this assignment.

```r
print(post, digits = 3) # you can actually estimate 3 decimal places when N = 100K
```

```
## stan_lm
##  family:       gaussian [identity]
##  formula:      log(WAGP) ~ SCHL + SEX + AGEP + I(AGEP^2)
##  observations: 98064
##  predictors:   5
## ------
##             Median MAD_SD
## (Intercept)  3.938  0.032
## SCHL         0.109  0.001
## SEX         -0.365  0.007
## AGEP         2.061  0.013
## I(AGEP^2)   -0.208  0.001
##
## Auxiliary parameter(s):
##               Median MAD_SD
## R2            0.332  0.002
## log-fit_ratio 0.000  0.002
## sigma         1.076  0.002
##
## Sample avg. posterior predictive distribution of y:
##          Median MAD_SD
## mean_PPD 10.335  0.005
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

To evaluate a hypothesis on the direction of a coefficient, we can do:

```r
draws <- as.matrix(post)
colMeans(draws[ , c("SCHL", "SEX", "AGEP", "I(AGEP^2)")] > 0)
```

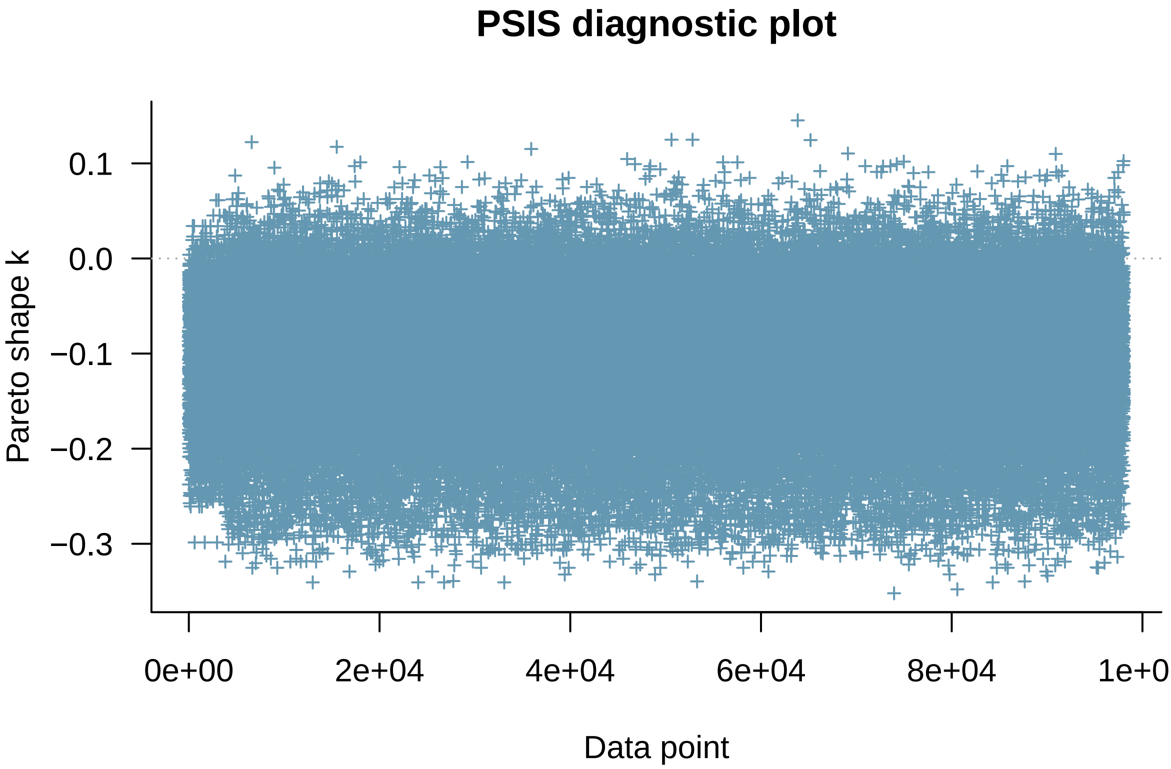
```
##      SCHL       SEX      AGEP I(AGEP^2)
##         1         0         1         0
```

Both “years” of schooling and (the linear term of) age have positive coefficients with near certainty according to the posterior distribution from this model.

2.2 Influential Observations

According to

```r
plot(loo(post), label_points = TRUE)
```

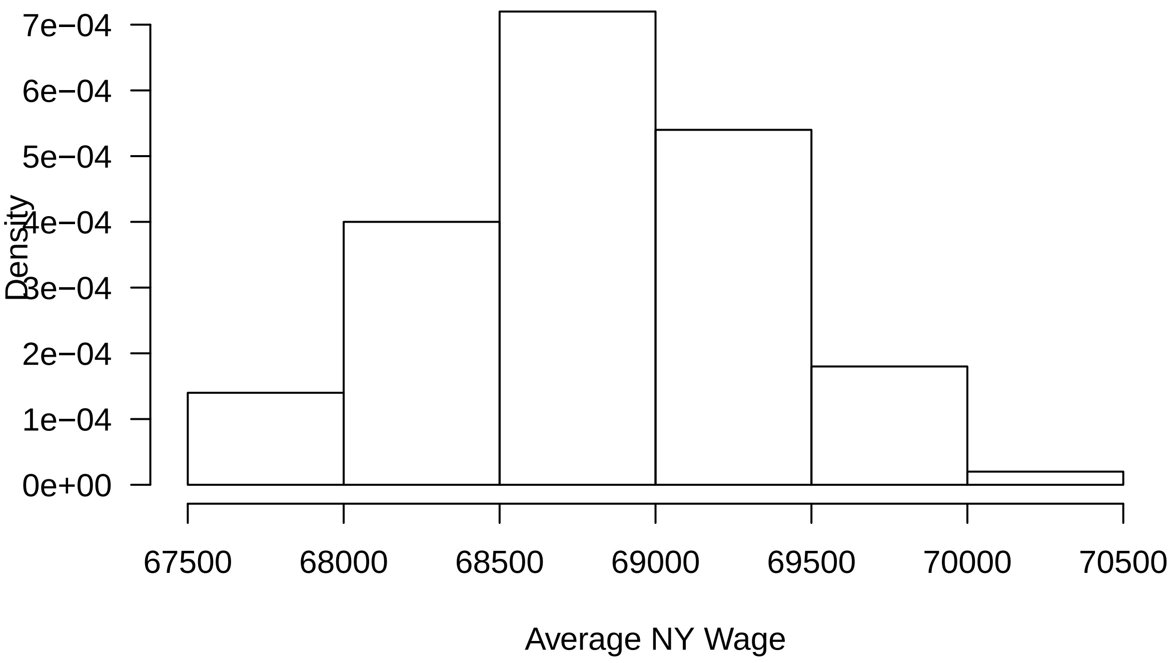


none of the individual observations have an outsized effect on the posterior distribution, which is fortutious for a lot of reasons.

2.3 Posterior Predictions

To describe our posterior beliefs about *average* wages, we could do:

```r
PPD <- posterior_predict(post, draws = 100, fun = exp)
hist(rowMeans(PPD), prob = TRUE, main = "", las = 1, xlab = "Average NY Wage")
```



2.4 Topcoding

If we do something similar for observations that have been topcoded,

```
(topcode <- max(dataset$WAGP))
```

```
## [1] 660000
```

```
too_rich <- which(dataset$WAGP == topcode)
PPD <- posterior_predict(post, fun = exp,
                         newdata = model.frame(post)[too_rich, ])
PPD_ <- PPD
PPD_[PPD_ < topcode] <- NA_real_
head(round(cbind(mean = colMeans(PPD),
           conditional_mean = colMeans(PPD_, na.rm = TRUE))))
```

```
##        mean conditional_mean
## 2336  99392          1018252
## 2448  99786          1060168
## 3745  47019          1002036
## 3822 161309          1077842
## 3970 167667          1111267
## 4030 165284          1119336
```

Here we see that the expectation of the posterior predictive distribution is far below the observed topcoded value. But if we only average over the posterior predictions above the topcoded value, we get much larger values.