APSTA-GE 2123 Assignment 4

Yuyue Hua

1 Oregon Medicaid Experiment

```
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

```
setwd("D:\\GRAD-NYU\\1ST YEAR\\NYUclass\\2020 spring\\Bayesian\\NYU2020\\Assignments\\Assignment1")
rstan::expose_stan_functions("quantile_functions.stan")
source("GLD_helpers.R")

# create two dummy variables
dataset$numhh2 <- ifelse(dataset$numhh == 2, 1, 0)
dataset$numhh3 <- ifelse(dataset$numhh == 3, 1, 0)

# Set Priors
a_s_alpha <- GLD_solver_bounded(bounds = c(0, 26), median = 14, IQR = 6)
```

```
## Warning in GLD_solver_bounded(bounds = c(0, 26), median = 14, IQR = 6): no
## asymmetry and steepness values achieve the bounds exactly; actual bounds
## are 1.49297155011439e-05 and 25.9999497413936
```

```
a_s_beta1 <- GLD_solver_bounded(bounds = c(-2, 2), median = -0.2, IQR = 1)
```

```
## Warning in GLD_solver_bounded(bounds = c(-2, 2), median = -0.2, IQR = 1):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds
## are -1.99999669220003 and 1.99999489577722
```

```
a_s_beta2 <- GLD_solver_bounded(bounds = c(-1, 1), median = -0.4, IQR = 1)
```

```
## Warning in GLD_solver_bounded(bounds = c(-1, 1), median = -0.4, IQR = 1):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds
## are -1.16939663122674 and 1.02839326669462
```

```
a_s_beta3 <- GLD_solver_bounded(bounds = c(-1, 1), median = 0.5, IQR = 1)
```

```
## Warning in GLD_solver_bounded(bounds = c(-1, 1), median = 0.5, IQR = 1):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds
## are -1.03444511281768 and 1.25339634906509
```

```
a_s_sigma <- GLD_solver_bounded(bounds = c(0, 5), median = 2, IQR = 1.5)
```

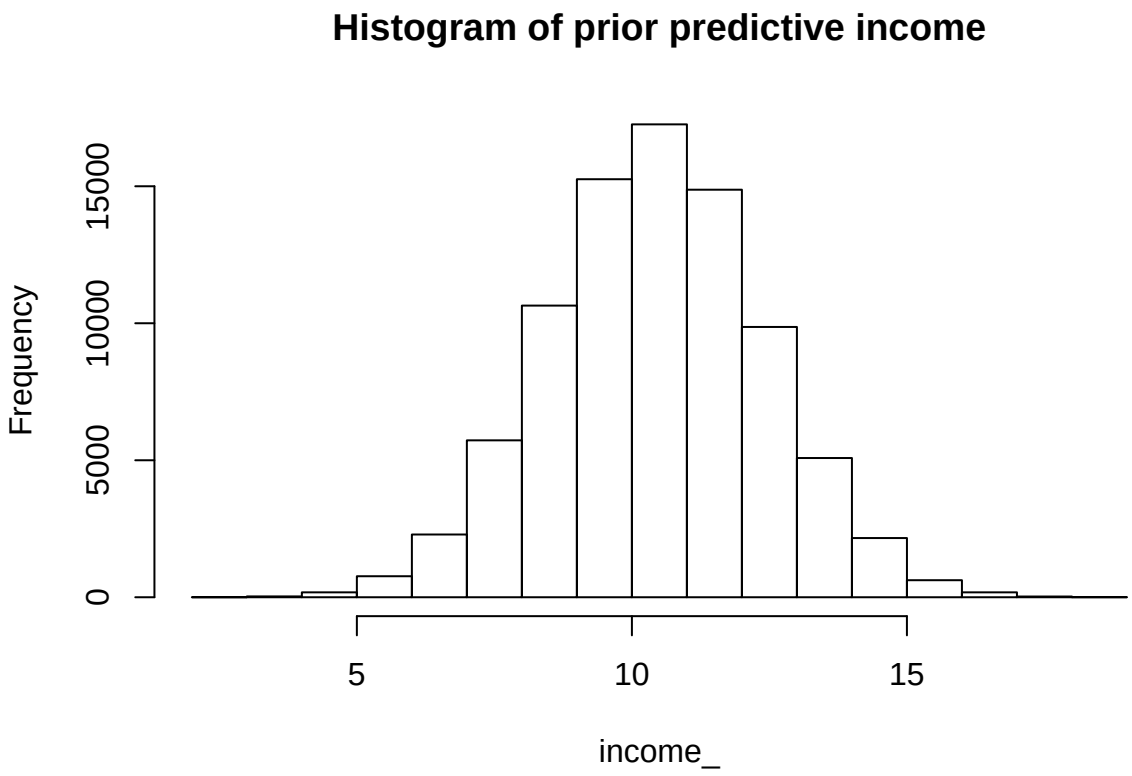
```
## Warning in GLD_solver_bounded(bounds = c(0, 5), median = 2, IQR = 1.5): no
## asymmetry and steepness values achieve the bounds exactly; actual bounds
## are 6.85054220988862e-07 and 4.99999964478035
```

```r
# Draw once from prior predictive distribution
alpha_ <- GLD_icdf(runif(1), median = 14, IQR = 6, asymmetry = a_s_alpha[1],
    steepness = a_s_alpha[2])
beta1_ <- GLD_icdf(runif(1), median = -0.2, IQR = 1, asymmetry = a_s_beta1[1],
    steepness = a_s_beta1[2])
beta2_ <- GLD_icdf(runif(1), median = -0.4, IQR = 1, asymmetry = a_s_beta2[1],
    steepness = a_s_beta2[2])
beta3_ <- GLD_icdf(runif(1), median = 0.5, IQR = 1, asymmetry = a_s_beta3[1],
    steepness = a_s_beta3[2])
mu_ <- alpha_ + beta1_ * dataset$numhh2 + beta2_ * dataset$numhh3 + beta3_ *
    dataset$lottery
sigma_ <- GLD_icdf(runif(1), median = 2, IQR = 1.5, asymmetry = a_s_sigma[1],
    steepness = a_s_sigma[2])
epsilon_ <- rnorm(n = length(mu_), mean = 0, sd = sigma_)
y_ <- mu_ + epsilon_

income_prior <- as.data.frame(y_)
summary(income_prior)
```

```
##        y_
##  Min.   : 2.659
##  1st Qu.: 9.118
##  Median :10.442
##  Mean   :10.440
##  3rd Qu.:11.747
##  Max.   :18.394
```

```r
hist(income_prior$y_, main = "Histogram of prior predictive income", xlab = "income_",
    breaks = 20)
```



Beta1 and beta2 are two dummy variables to estimate the effect of number of adults in the household. Beta 3 estimates the effect of being given the opportunity to enroll in Medicaid. Alpha represents the average estimated income for people in a household with one adult and did not win the Medicaid lottery and I assume income was measured in thousand dollars. The average individual income of someone who is poor enough for Medicaid was about 14,700 dollars in 2007. So I chose the median of alpha to be 14,000 dollars and set the lower bound to be 0. From the summary and histogram of prior predictive income, we can see that it is centered at around 14,000 dollars and the minimum is positive.

1.2 Prior Predictive Distribution for a Journal

```
#Adjust Priors for beta3 with median = 0
a_s_betaJ3<- GLD_solver(lower_quartile = -1, median =0, upper_quartile = 0.65, other_quantile = 0.51, a

#Draw from prior distribution of beta3
betaJ3_<-replicate(10000,GLD_rng(median = 0, IQR = 0.65 - -1, asymmetry = a_s_betaJ3[1], steepness = a_

#Compute the percentage of treatment effect greater than 0.5
#0.5 is the assumed median for beta3 in Question 1.1
mean(betaJ3_>0.5)
```

```
## [1] 0.3056
```

2 2018 American Community Survey

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

2.1 Posterior Distribution

```
#options(mc.cores = parallel::detectCores())
require(rstanarm)
post <-stan_lm(log(WAGP) ~ AGEP+I(AGEP^2)+WKHP+WKW,
 data = Idaho, prior = R2(location = 0.5, what = "mode"), adapt_delta = 0.95)
```

```
## Warning: There were 126 divergent transitions after warmup. Increasing adapt_delta above 0.95 may hel
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
# analyze the results here
summary(post, digits = 3)
```

```
##
## Model Info:
##  function:     stan_lm
##  family:       gaussian [identity]
##  formula:      log(WAGP) ~ AGEP + I(AGEP^2) + WKHP + WKW
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 7804
##  predictors:   5
##
## Estimates:
##                 mean   sd     10%    50%    90%
## (Intercept)    7.655  0.076  7.558  7.656  7.753
## AGEP           0.077  0.004  0.073  0.077  0.082
## I(AGEP^2)     -0.001  0.000 -0.001 -0.001 -0.001
## WKHP           0.034  0.001  0.033  0.034  0.035
## WKW           -0.366  0.006 -0.374 -0.366 -0.359
## sigma          0.800  0.007  0.791  0.800  0.808
## log-fit_ratio  0.000  0.006 -0.008  0.000  0.008
## R2             0.632  0.005  0.625  0.632  0.639
##
## Fit Diagnostics:
##            mean   sd     10%    50%    90%
## mean_PPD 10.013  0.012  9.998 10.013 10.030
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
```

```
##                mcse  Rhat  n_eff
## (Intercept)   0.002 1.001 2103
## AGEP          0.000 1.001 2420
## I(AGEP^2)     0.000 1.001 2461
## WKHP          0.000 1.001 2765
## WKW           0.000 1.000 2673
## sigma         0.000 1.005  420
## log-fit_ratio 0.000 1.003 1017
## R2            0.000 1.001 1490
## mean_PPD      0.000 1.000 4061
## log-posterior 0.077 1.003  799
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

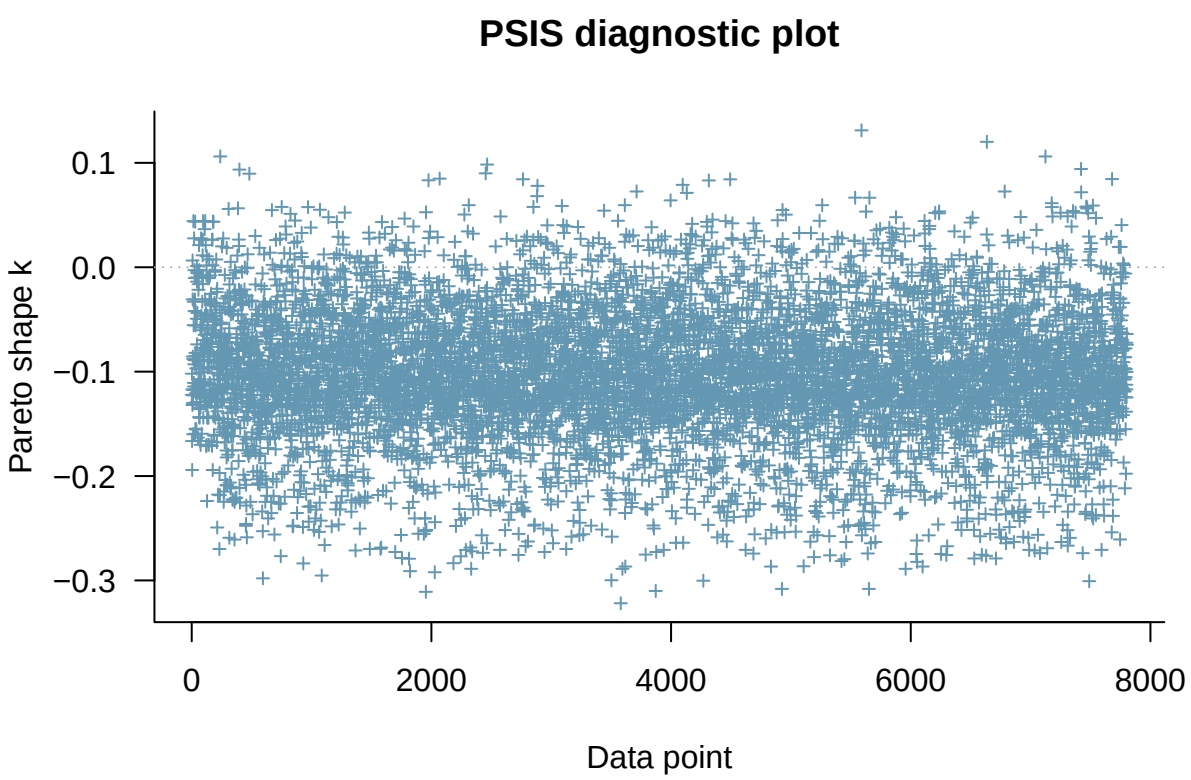
From the above output, we can see that coefficients of AGEP(Age of a person) and WKHP(Usual hours worked per week past 12 months) are positive with a very high posterior probability. The 10th percentile for the coefficient of AGEP is 0.072 and the 10th percentile for the coefficient of WKHP is 0.033. So both of the two coefficients have at least 90% values that are larger than zero.

2.2 Influential Observations

```
loo1<-loo::loo(post,cores=1)
loo1
```

```
##
## Computed from 4000 by 7804 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -9335.8 105.3
## p_loo        10.3   0.7
## looic     18671.6 210.7
## ------
## 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.
```

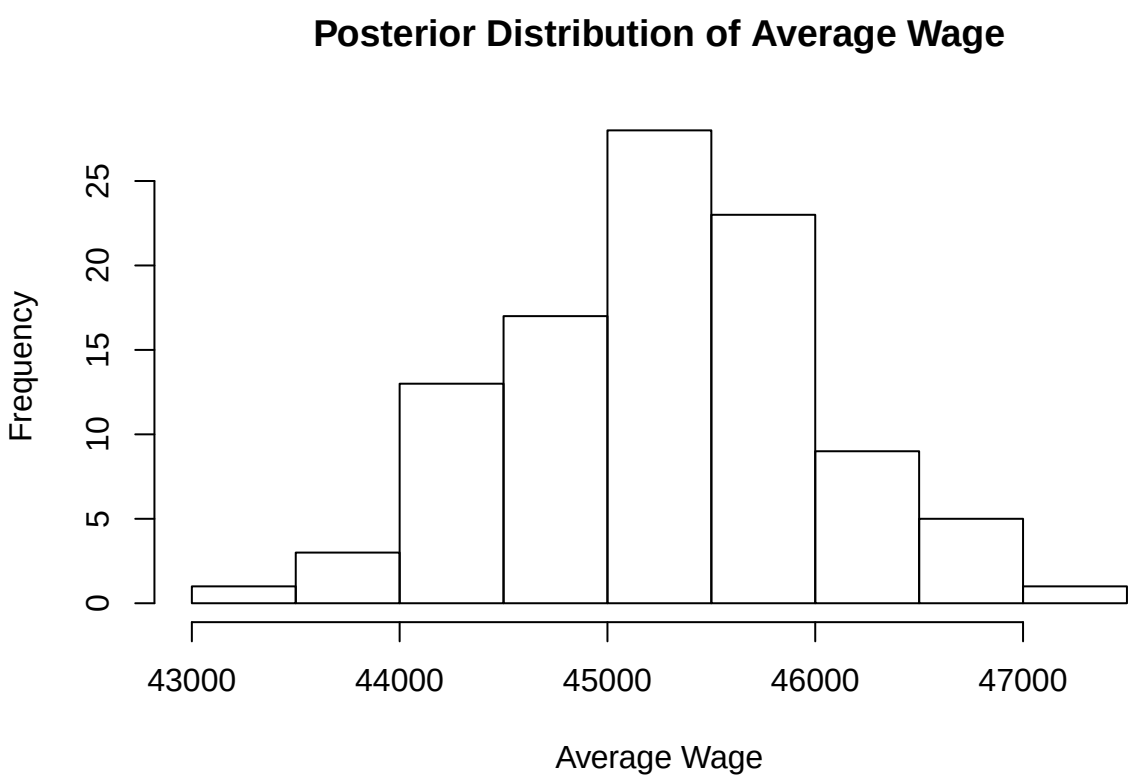
```
plot(loo1,label_points = TRUE)
```



The diagnostic plot shows that all Pareto k estimates are approximately bounded between -0.3 and 0.2 and all of them are less than 0.5, which indicates that the observations do not have outsized influence on the posterior distribution.

2.3 Posterior Predictions

```
set.seed(2020)
postpred <- as.data.frame(posterior_predict(post, draws = 100, seed = 2344,
    fun = exp))
aveWage <- apply(postpred, FUN = mean, MARGIN = 1)
hist(aveWage, main = "Posterior Distribution of Average Wage", xlab = "Average Wage")
```




```
summary(aveWage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   43471   44697   45213   45254   45767   47084
```

```
sd(aveWage)
```

```
## [1] 742.288
```

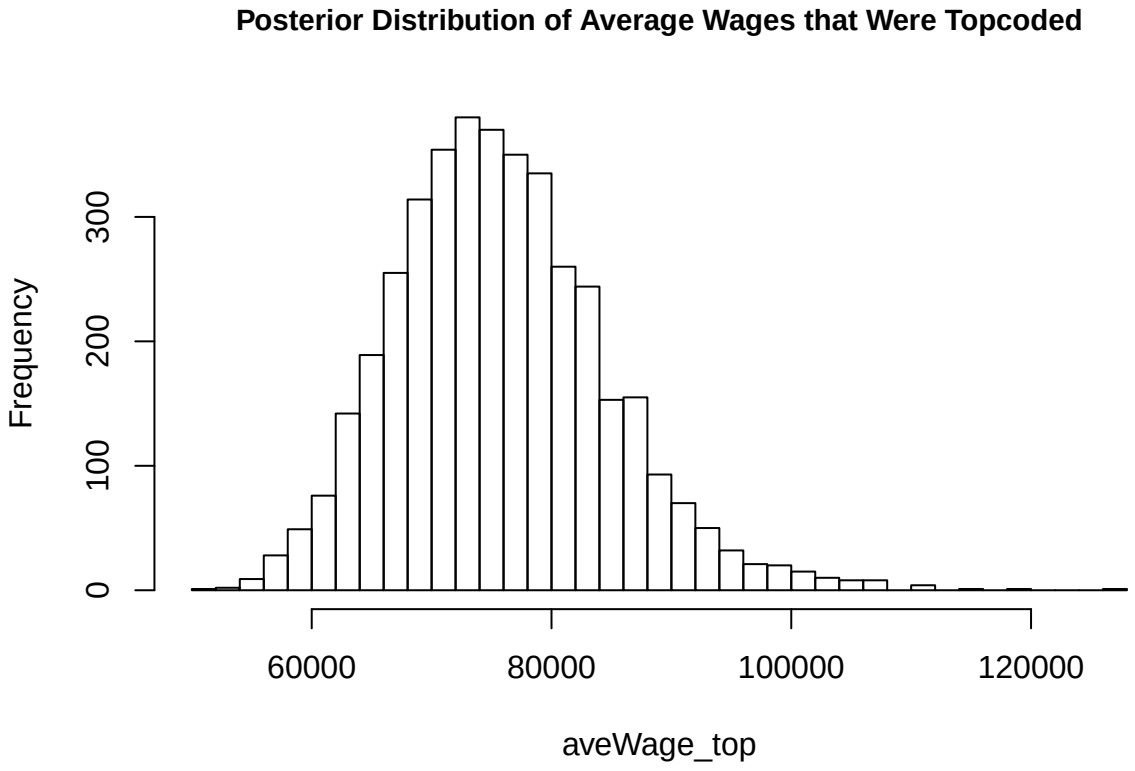
One way to know the variability of average wage is to look at its distribution. The histogram is approximately bell-shaped. The standard deviation of posterior average wage is 903 and the interquartile range is 45957-44772=1185. Both of the two values measure the uncertainty to a certain extent. Compared to the scale of income, we can say that the "true" average wage is pretty close to 45336 dollars if we believe the model is right.

2.4 Topcoding

```
topcoded_value <- max(Idaho$WAGP)

# Create a new dataset
Idaho_tc <- Idaho[Idaho$WAGP == topcoded_value, ]
postpred_top <- as.data.frame(posterior_predict(post, newdata = Idaho_tc, fun = exp))
aveWage_top <- apply(postpred_top, FUN = mean, MARGIN = 1)

hist(aveWage_top, main = "Posterior Distribution of Average Wages that Were Topcoded",
    breaks = 50, cex.main = 0.9)
```




```
summary(aveWage_top)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   51079   69610   75093   75743   81055  127219
```

For people in the dataset whose incomes are topcoded, the posterior distribution for their actual income is centered at around 75055 dollars with mean equaling to 75751 dollars. It is bell-shaped and skewed to the right.