APSTA-GE 2123 Assignment 4

Your Name

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

The general functions for predicting income should be

$$Income = \beta_{lottery}Lottery + \beta_{small}Small + \beta_{medium}Medium + \beta_{large}Large + \epsilon$$

```
rstan::expose_stan_functions(file.path('quantile_functions.stan'))
source(file.path('GLD_helpers.R'))
library(dplyr)
#distribution for household size of 1
beta_s_small<- GLD_solver_bounded(bounds=3000:100000,median=14700,IQR=3000)
```

```
## Warning in GLD_solver_bounded(bounds = 3000:1e+05, median = 14700, IQR = 3000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 2805.93247782199 and 16643.9922174065
```

```
#distribution for household size of 2
beta_s_medium<- GLD_solver_bounded(bounds=3000:100000,median=15000,IQR=3000)
```

```
## Warning in GLD_solver_bounded(bounds = 3000:1e+05, median = 15000, IQR = 3000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 2812.93157667387 and 16939.90302135
```

```
#distribution for household size of 3 or above
beta_s_large<- GLD_solver_bounded(bounds=3000:100000,median=16000,IQR=3000)
```

```
## Warning in GLD_solver_bounded(bounds = 3000:1e+05, median = 16000, IQR = 3000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 2827.6460004129 and 17927.6335608598
```

```r
#distribution for winning the lottery
beta_s_lottery<- GLD_solver_bounded(bounds=-1500:2000, median=-20, IQR=150)
```

```
## Warning in GLD_solver_bounded(bounds = -1500:2000, median = -20, IQR = 150):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## -1497.3766058788 and 72.5005761080866
```

```r
#sigma for error
a_s_sigma<- GLD_solver(lower_quartile = 250 ,median=300, upper_quartile = 500, other_quantile = 0, alpha

#coefficient of household of size 1
beta_small<- GLD_rng(median=14700,IQR=3000,asymmetry = beta_s_small[1],steepness = beta_s_small[2])

#coefficient of household with size 2
beta_medium<- GLD_rng(median=15000, IQR=3000, asymmetry = beta_s_medium[1],steepness = beta_s_medium[2]

#coefficient of household with size 3 or above
beta_large<- GLD_rng(median=16000, IQR=3000, asymmetry = beta_s_large[1],steepness=beta_s_large[2])

#coefficient of winning the lottery
beta_lottery<- GLD_rng(median=-20, IQR=150, asymmetry = beta_s_lottery[1],steepness = beta_s_lottery[2]

#sigma for error to the estimation
sigma_<- GLD_rng(median=300, IQR=250, asymmetry = a_s_sigma[1],steepness = a_s_sigma[2])

#vector space for storing coefficient for different household size
gamma<- cbind(beta_small,beta_medium,beta_large)

dataset$income<- beta_lottery*dataset$lottery+gamma[dataset$numhh]+sigma_

#verify prediction on income
winning_lottery<- dataset %>% filter(lottery==1) %>% select(income)
summary(winning_lottery)
```

```
##      income
##  Min.   :16766
##  1st Qu.:16853
##  Median :16853
##  Mean   :16888
##  3rd Qu.:16992
##  Max.   :16992
```

```r
notwinning_lottery<- dataset %>% filter(lottery==0) %>% select(income)
summary(notwinning_lottery)
```

```
##      income
##  Min.   :16842
##  1st Qu.:16842
##  Median :16929
##  Mean   :16947
##  3rd Qu.:17068
##  Max.   :17068
```

1.2 Prior Predictive Distribution for a Journal

```r
#distribution for household size of 1
beta_s_small<- GLD_solver_bounded(bounds=3000:100000,median=14700,IQR=3000)
```

```
## Warning in GLD_solver_bounded(bounds = 3000:1e+05, median = 14700, IQR = 3000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 2805.93247782199 and 16643.9922174065
```

```r
#distribution for household size of 2
beta_s_medium<- GLD_solver_bounded(bounds=3000:100000,median=15000,IQR=3000)
```

```
## Warning in GLD_solver_bounded(bounds = 3000:1e+05, median = 15000, IQR = 3000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 2812.93157667387 and 16939.90302135
```

```r
#distribution for household size of 3 or above
beta_s_large<- GLD_solver_bounded(bounds=3000:100000,median=16000,IQR=3000)
```

```
## Warning in GLD_solver_bounded(bounds = 3000:1e+05, median = 16000, IQR = 3000):
## no asymmetry and steepness values achieve the bounds exactly; actual bounds are
## 2827.6460004129 and 17927.6335608598
```

```r
#refit distribution for winning the lottery with the median of 0
beta_s_lottery<- GLD_solver_bounded(bounds=-1480:2020, median=0, IQR=125)
```

```
## Warning in GLD_solver_bounded(bounds = -1480:2020, median = 0, IQR = 125): no
## asymmetry and steepness values achieve the bounds exactly; actual bounds are
## -1494.44456156044 and 76.4900369086214
```

```r
#sigma for error
a_s_sigma<- GLD_solver(lower_quartile = 250 ,median=300, upper_quartile = 500, other_quantile = 0, alpha

#coefficient of household of size 1
beta_small<- GLD_rng(median=14700,IQR=3000,asymmetry = beta_s_small[1],steepness = beta_s_small[2])

#coefficient of household with size 2
beta_medium<- GLD_rng(median=15000, IQR=3000, asymmetry = beta_s_medium[1],steepness = beta_s_medium[2]

#coefficient of household with size 3 or above
beta_large<- GLD_rng(median=16000, IQR=3000, asymmetry = beta_s_large[1],steepness=beta_s_large[2])

#refit coefficient of winning the lottery with the median of 0
beta_lottery<- GLD_rng(median=0, IQR=125, asymmetry = beta_s_lottery[1],steepness = beta_s_lottery[2])

#sigma for error to the estimation
sigma_<- GLD_rng(median=300, IQR=250, asymmetry = a_s_sigma[1],steepness = a_s_sigma[2])

#vector space for storing coefficient for different household size
gamma<- cbind(beta_small,beta_medium,beta_large)
```

```
dataset$income<- beta_lottery*dataset$lottery+gamma[dataset$numhh]+sigma_

#verify prediction on income
winning_lottery<- dataset %>% filter(lottery==1) %>% select(income)
summary(winning_lottery)
```

```
##      income
##  Min.   :11925
##  1st Qu.:11925
##  Median :11974
##  Mean   :13351
##  3rd Qu.:15943
##  Max.   :15943
```

```
notwinning_lottery<- dataset %>% filter(lottery==0) %>% select(income)
summary(notwinning_lottery)
```

```
##      income
##  Min.   :12026
##  1st Qu.:12026
##  Median :12075
##  Mean   :13467
##  3rd Qu.:16044
##  Max.   :16044
```

2 2018 American Community Survey

```
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

The following posterior distribution are performed with dataset on Nebraska.

```
library(rstanarm)
```

```
## Loading required package: Rcpp

## rstanarm (Version 2.19.3, packaged: 2020-02-11 05:16:41 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())

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting
```

```
post_wgap<- stan_lm(log(WAGP)~AGEP+MAR+JWRIP+log(PINCP),data=dataset, prior=R2(location=0.65, what='mod
```

```
## Warning: There were 103 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
```

```
print(post_wgap,digits=4)
```

```
## stan_lm
##  family:       gaussian [identity]
##  formula:      log(WAGP) ~ AGEP + MAR + JWRIP + log(PINCP)
##  observations: 8206
##  predictors:   5
## ------
##              Median  MAD_SD
## (Intercept)  0.1866  0.0503
## AGEP        -0.0094  0.0003
## MAR         -0.0188  0.0028
## JWRIP       -0.0128  0.0074
## log(PINCP)   1.0162  0.0045
##
## Auxiliary parameter(s):
##                Median MAD_SD
## R2             0.8733 0.0019
## log-fit_ratio 0.0001 0.0037
## sigma          0.3720 0.0028
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
#To check plot and diagnostic use the following code
#launch_shinystan(post_wgap)
```

Based on the information, we can conclude that Log of Personal earning variable has a positive coeffcient with relative high posterior distribution probability.

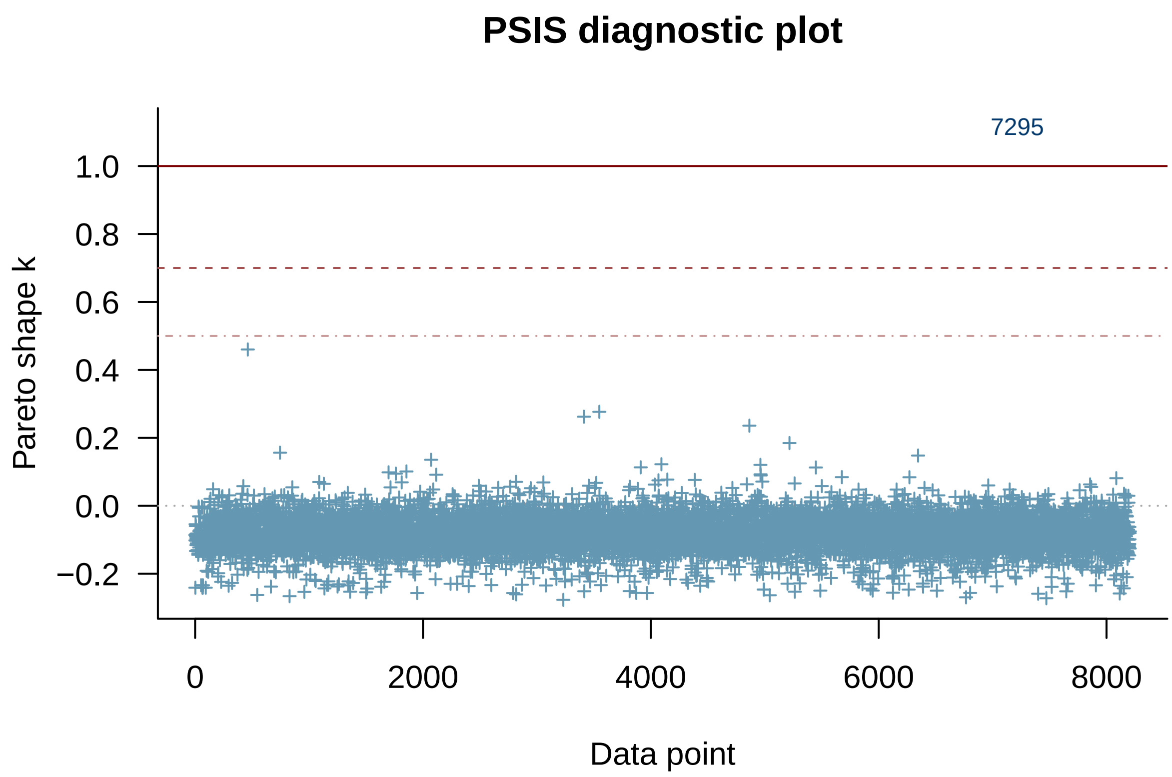
2.2 Influential Observations

```r
library(rstanarm)
```

```
## Loading required package: Rcpp

## rstanarm (Version 2.19.3, packaged: 2020-02-11 05:16:41 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())

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting
```

```r
plot(loo(post_wgap),label_points=T,)
```

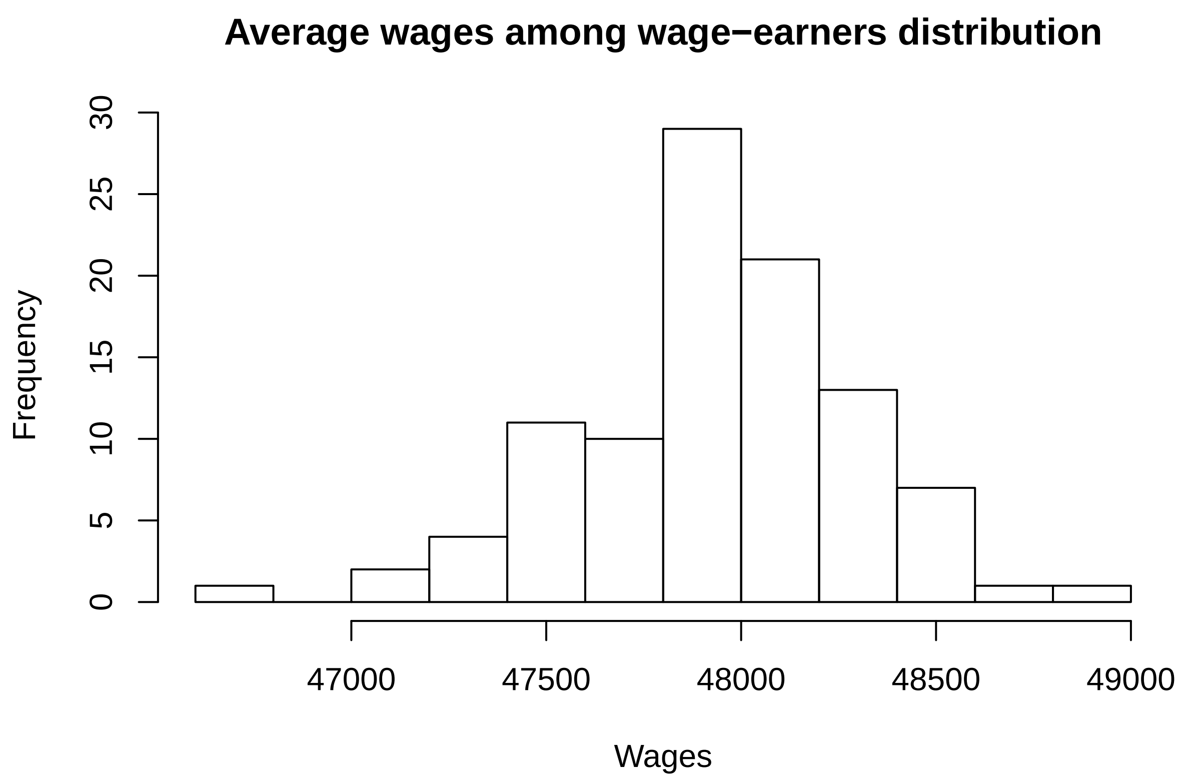
```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argumen
```



Based on the plot, observations 17295 seems to have an outsized influence on the posterior distributions.

2.3 Posterior Predictions

```
# make histogram
Post_pred<- posterior_predict(post_wgap,draws=100,fun = exp)
pred_df<- as.data.frame(Post_pred)
pred_df$mean<- rowMeans(pred_df,na.rm=T)
hist(pred_df$mean,main="Average wages among wage-earners distribution",xlab="Wages",breaks=10)
```



Overall, there exists some uncertainty for people who's average wages are range from 47500 to 47600. As the shape of the distribution has a sudden drop instead of a concave bell-shape.

2.4 Topcoding

```
topcoded_value <- max(dataset$WAGP)
# do the analysis
top_code_df<- dataset %>% filter(WAGP==430000) %>% select(AGEP,MAR,JWRIP,PINCP) %>% na.omit()
post_pred_top_code<- posterior_predict(post_wgap,newdata=top_code_df,draws=100,fun = exp)
exp_df<- as.data.frame(post_pred_top_code)
top_code_df$expectation_income<- colMeans(exp_df)
top_code_df$WAGP<- 430000
top_code_df
```

```
## # A tibble: 79 x 6
```

```
##      AGEP   MAR JWRIP  PINCP expectation_income   WAGP
##     <dbl> <dbl> <dbl>  <dbl>              <dbl>  <dbl>
##   1    21     5     1 430000            490910. 430000
##   2    52     4     3 430000            348336. 430000
##   3    42     1     1 430000            428202. 430000
##   4    58     1     1 731000            633264. 430000
##   5    52     1     1 640000            593656. 430000
##   6    58     1     1 430000            370482. 430000
##   7    59     1     1 440000            386708. 430000
##   8    44     1     1 430000            400520. 430000
##   9    37     3     1 431000            443204. 430000
##  10    51     1     1 430000            395935. 430000
## # ... with 69 more rows
```

The posterior expectation for their actual income are recorded in the expectation_income column in the top_code dataframe.