Bayesian Synthesis

Reese Guo

2/10/2020

2

```
library(ggplot2)
Data <- read.csv("CEdata.csv")
Rural <- Data$UrbanRural
Income <- Data$Income
Race <- Data$Race
Expenditure <- Data$Expenditure
```

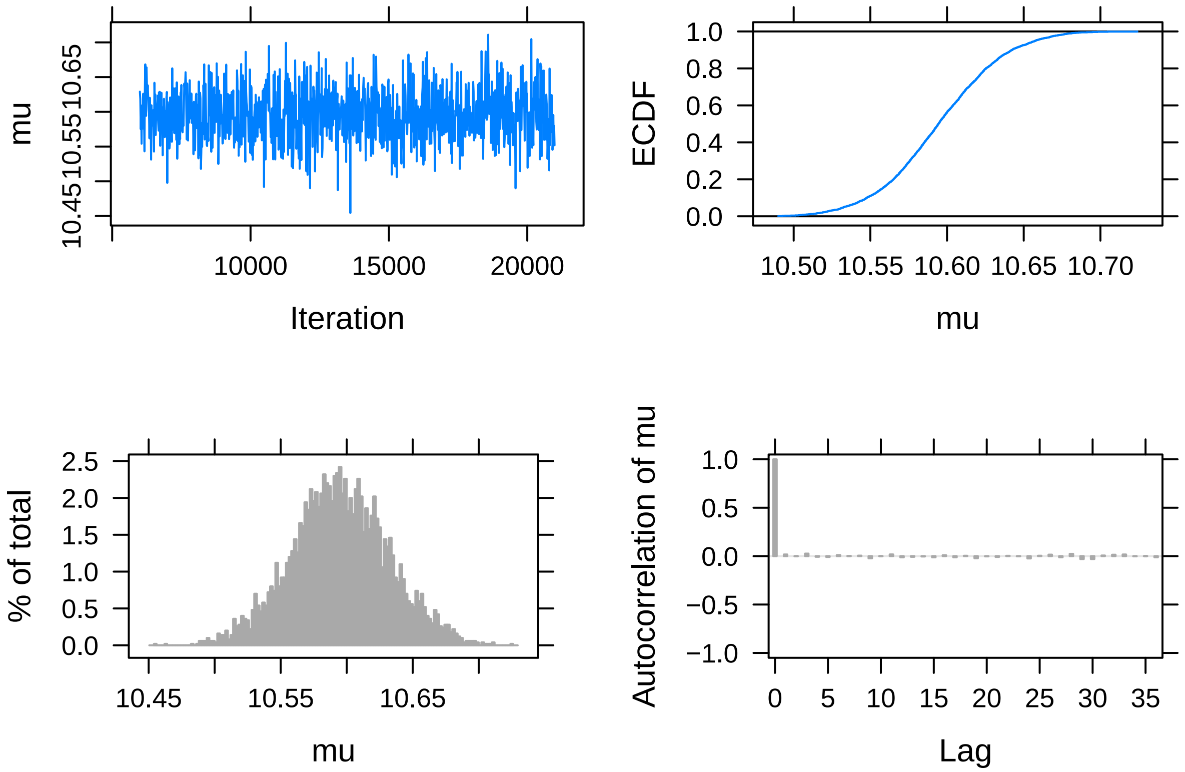
```
require(runjags)
require(coda)

modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dlnorm(mu, invsigma2)
}
## priors
mu ~ dnorm(mu_0, invtau2)
invsigma2 ~ dgamma(a, b)
invtau2 ~ dgamma(c, d)
sigma <- sqrt(pow(invsigma2, -1))
tau <- sqrt(pow(invtau2, -1))
}"

N <- length(Income)

the_data <- list("y" = Income, "N" = N,
"mu_0" = 0,
"a" = 1, "b" = 1,
"c" = 1, "d" = 1)

initsfunction <- function(chain){
.RNG.seed <- c(1,2)[chain]
.RNG.name <- c("base::Super-Duper",
"base::Wichmann-Hill")[chain]
return(list(.RNG.seed=.RNG.seed,
.RNG.name=.RNG.name))
}

posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("mu", "sigma", "tau"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 3,
                      inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Tue Feb 11 16:03:09 2020
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 994
##    Unobserved stochastic nodes: 3
##    Total graph size: 1009
## . Reading parameter file inits1.txt
## . Initializing model
## . Adapting 1000
## -------------------------------------------------| 1000
## ++++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 15000
## -------------------------------------------------| 15000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Finished running the simulation
```

(i)

```
plot(posterior, vars = "mu")
```

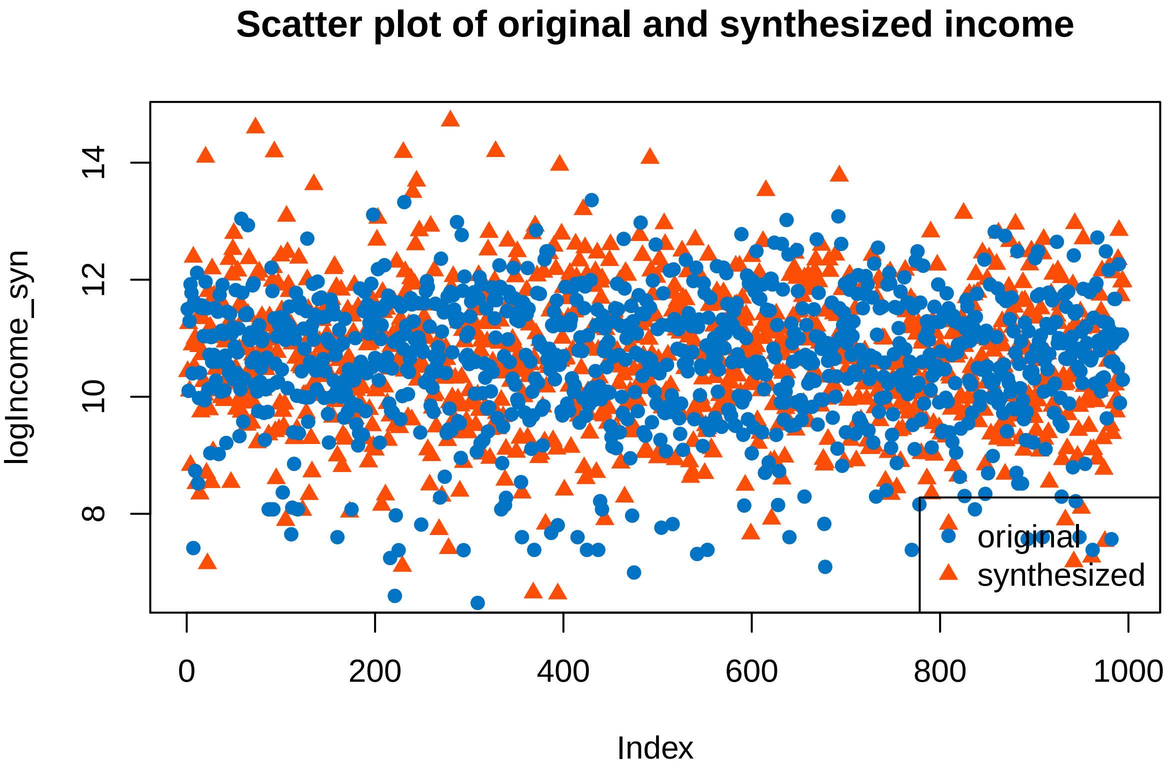
```
## Generating plots...
```



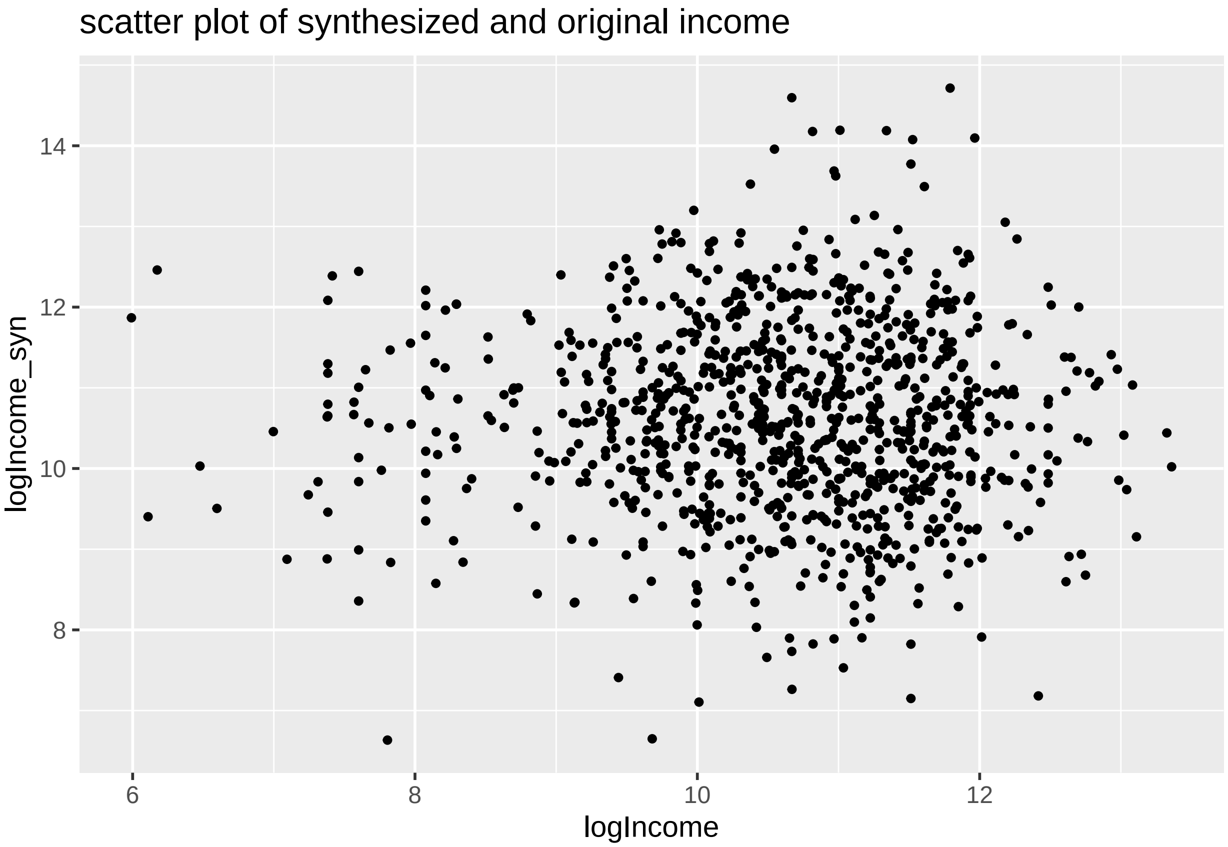
```
post <- as.mcmc(posterior)

logIncome <- log(Income)
n <- length(Income)
Income_syn <- rlnorm(n, post[5000, "mu"], post[5000, "sigma"])
logIncome_syn <- log(Income_syn)


plot(logIncome_syn, col = "#FC4E07", pch = 17)
points(logIncome, pch = 16, col = "#0073C2FF") + title("Scatter plot of original and synthesized income
```

```
## integer(0)
```

```
legend("bottomright", legend = c("original", "synthesized"),
       col = c("#0073C2FF", "#FC4E07"), pch = c(16, 17) )
```



```
ggplot(NULL, aes(x = logIncome, y = logIncome_syn)) +
  geom_point(size = 1) +
  labs(title="scatter plot of synthesized and original income")
```



As shown in the scatter plot above, the distribution of logged synthesized income is similar to that of logged original income. The difference between two set of data is that original data tend to have more low logged income values, whereas, the synthesized data tend to have more high logged income values.

(ii)

```r
mean(logIncome)
```

```
## [1] 10.59507
```

```r
mean(logIncome_syn)
```

```
## [1] 10.62788
```

```r
median(logIncome)
```

```
## [1] 10.70574
```

```r
median(logIncome_syn)
```

```
## [1] 10.60851
```

We can see that the mean of the logged synthesized income is larger than the mean of the logged original income data. However, medians of the two datasets are similar.

###(iii)

```r
logExpenditure <- log(Expenditure)
linearMod <- lm(logExpenditure ~ logIncome)
summary(linearMod)
```

```
##
## Call:
## lm(formula = logExpenditure ~ logIncome)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.81953 -0.51688  0.04823  0.44126  2.46195
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.32193    0.21218   20.37   <2e-16 ***
## logIncome    0.42115    0.01991   21.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7237 on 992 degrees of freedom
## Multiple R-squared:  0.3109, Adjusted R-squared:  0.3102
## F-statistic: 447.5 on 1 and 992 DF,  p-value: < 2.2e-16
```

```r
linearMod_one <- lm(logExpenditure ~ logIncome_syn)
summary(linearMod_one)
```

```
##
## Call:
## lm(formula = logExpenditure ~ logIncome_syn)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -4.0911 -0.6067  0.0281  0.5883  2.3591
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.01165    0.24790  36.352   <2e-16 ***
## logIncome_syn -0.02142    0.02318  -0.924    0.356
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8714 on 992 degrees of freedom
## Multiple R-squared:  0.0008599,  Adjusted R-squared:  -0.0001473
## F-statistic: 0.8538 on 1 and 992 DF,  p-value: 0.3557
```

As we can see from the result of the linear regression above, the regression coefficient between logged original income and logged expenditure is very different from the regression coefficient of between logged synthesized income and logged expenditure. The cause of this difference is probably that in my synthesis model, I didn't preserve the relationship of income to other variables and I just synthesized income data itself. Thus, the newly synthesized data is unlikely to have the same relationship with other variables.