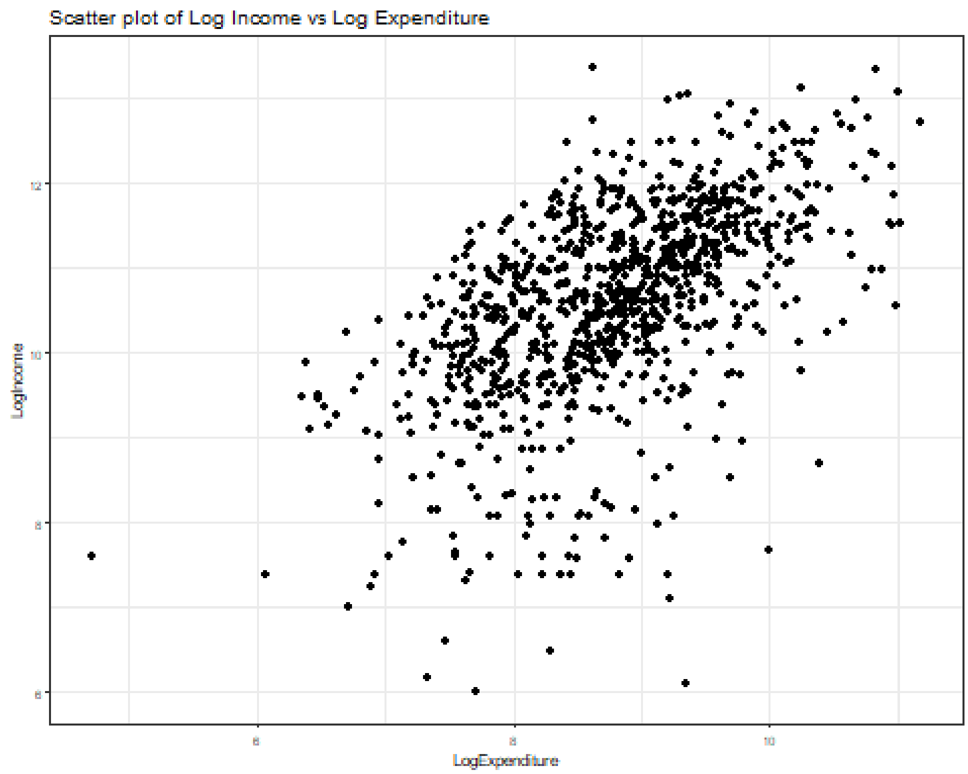
Spring 2020: MATH 301-56 Data Confidentiality

```r
library(ggplot2)
library(runjags)
library(readxl)
library(coda)

CEdata <- read_excel("C:/Users/Ted Xie/Downloads/CEdata.xlsx")

CEdata$LogIncome <- log(CEdata$Income)
CEdata$LogExpenditure <- log(CEdata$Expenditure)

summary(CEdata)

##    UrbanRural        Income             Race        Expenditure
##  Min.   :1.000   Min.   :   400   Min.   :1.000   Min.   :  110.3
##  1st Qu.:1.000   1st Qu.: 21546   1st Qu.:1.000   1st Qu.: 3663.4
##  Median :1.000   Median : 44611   Median :1.000   Median : 6700.3
##  Mean   :1.051   Mean   : 67593   Mean   :1.351   Mean   : 9422.0
##  3rd Qu.:1.000   3rd Qu.: 90038   3rd Qu.:1.000   3rd Qu.:11726.0
##  Max.   :2.000   Max.   :633840   Max.   :6.000   Max.   :71634.6
##    LogIncome      LogExpenditure
##  Min.   : 5.991   Min.   : 4.704
##  1st Qu.: 9.978   1st Qu.: 8.206
##  Median :10.706   Median : 8.810
##  Mean   :10.595   Mean   : 8.784
##  3rd Qu.:11.408   3rd Qu.: 9.370
##  Max.   :13.360   Max.   :11.179

ggplot(CEdata, aes(x = LogExpenditure, y = LogIncome)) + geom_point(size = 1)
+ labs(title = "Scatter plot of Log Income vs Log Expenditure") +
theme_bw(base_size = 6, base_family = "")
```



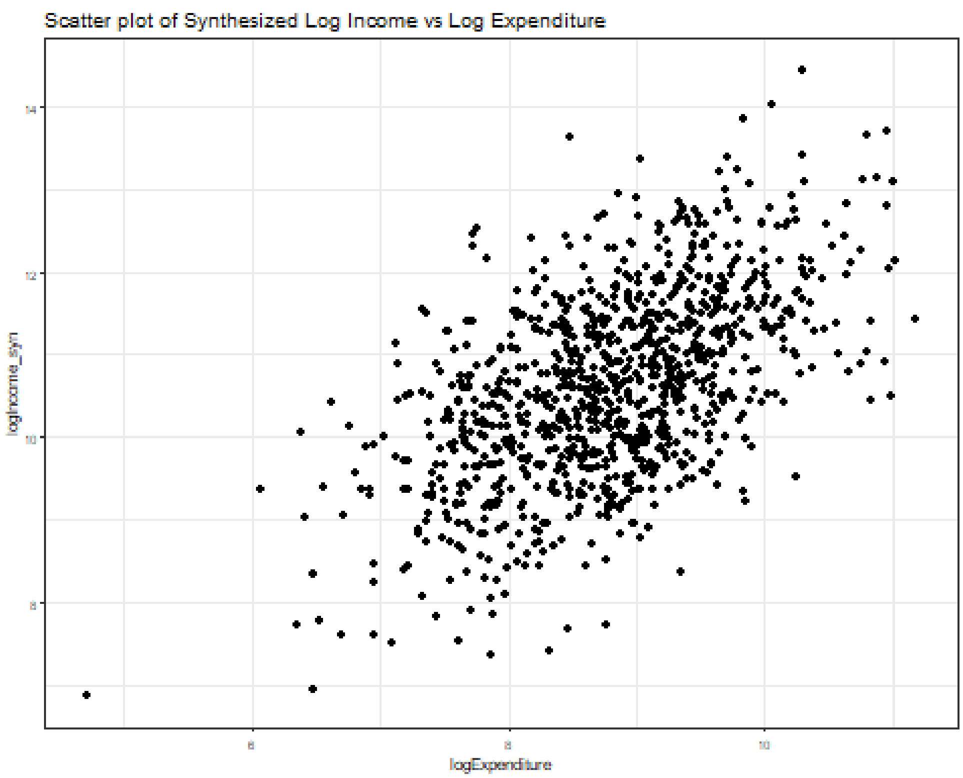

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x[i] +beta2*z[i] + beta3*xx[i], invsigma2)
}

## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
beta2 ~ dbeta(mu2, g2)
beta3 ~ dbeta(1, 1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"

xx <- as.vector(CEdata$Race)
z <- as.vector(CEdata$UrbanRural)
y <- as.vector(CEdata$LogIncome)
x <- as.vector(CEdata$LogExpenditure)
N <- length(y)
the_data <- list("y" = y, "x" = x, "z" = z, "xx" = xx, "N" = N, "mu0" = 0,
"g0" = 0.0001, "mu1" = 0, "g1" = 0.0001, "a" = 1, "b" = 1, "mu2" = 1, "g2"=
10)
initsfunction <- function(chain){
```

```r
  .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("beta0", "beta1", "beta2", "beta3", "sigma"), adapt = 1000, burnin = 5000,
sample = 5000, thin = 50, inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Mon Feb 24 22:54:44 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: 5
##    Total graph size: 5990
## . Reading parameter file inits1.txt
## . Initializing model
## . Adapting 1000
## -------------------------------------------------| 1000
## ++++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . . . Updating 250000
## -------------------------------------------------| 250000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain

## Finished running the simulation
```

```r
post <- as.mcmc(posterior)
synthesize <- function(X, Z, XX, index, n){
  mean_Y <- post[index, "beta0"] + X * post[index, "beta1"] + Z * post[index,
"beta2"] + XX * post[index, "beta3"]
  synthetic_Y <- rnorm(n, mean_Y, post[index, "sigma"])
  data.frame(X, synthetic_Y)
  }
```

```r
n <- dim(CEdata)[1]
synthetic_one <- synthesize(CEdata$LogExpenditure, CEdata$UrbanRural,
CEdata$Race, 1, n)
names(synthetic_one) <- c("logExpenditure", "logIncome_syn")

summary(synthetic_one)

##  logExpenditure   logIncome_syn
##  Min.   : 4.704   Min.   : 6.857
##  1st Qu.: 8.206   1st Qu.: 9.837
##  Median : 8.810   Median :10.583
##  Mean   : 8.784   Mean   :10.604
##  3rd Qu.: 9.370   3rd Qu.:11.402
##  Max.   :11.179   Max.   :14.441

ggplot(synthetic_one, aes(x = logExpenditure, y = logIncome_syn)) +
geom_point(size = 1) + labs(title = "Scatter plot of Synthesized Log Income
vs Log Expenditure") + theme_bw(base_size = 6, base_family = "")
```



Propensity Score

```r
df1 <- data.frame(Income = synthetic_one$logIncome_syn, expend =
CEdata$LogExpenditure, syn = 1)
df2 <- data.frame(Income = CEdata$LogIncome, expend = CEdata$LogExpenditure,
syn = 0)
merged <- rbind(df1, df2)
logistic <- glm(syn ~ Income + expend, data = merged, family = "binomial")
#summary(logistic)
```

```
#intercept <- -0.011590
#slope1 <- 0.002742
#slope2 <- 0.001988
#income <- merged[,1]
#expenditure <- merged[,2]

N <- length(merged)
c <- 1/2
#d <- intercept + slope1 * income + slope2 * expenditure
#p_i <- d/(1 + d)
#diff <- (p_i - c)^2
pred <- predict(logistic, data = merged)
probs <- exp(pred)/(1 + exp(pred))
U_p <- sum((probs - c)^2) / N
U_p

## [1] 0.003565892
```

Cluster Analysis Measure

```
clusters <- hclust(dist(merged[,1:2]), method = 'average')
G <- 5
clusterCut <- cutree(clusters, G)
cluster_S <- as.data.frame(cbind(clusterCut,merged$syn))
names(cluster_S) <- c("cluster", "S")
n_gS <- table(cluster_S)[, 1]
n_g <- rowSums(table(cluster_S))
w_g <- n_g / N
U_c <- (1/G) * sum(w_g * (n_gS/n_g - c)^2)
U_c

## [1] 0.125208
```

Emperical CDF Measures

```
S_x <- ecdf(CEdata$LogIncome)
S_y <- ecdf(synthetic_one$logIncome_syn)
#Sdiff <- c()
#for(i in 1:length(CEdata$LogIncome)){
#  Sdiff <- c(Sdiff, (CEdata$LogIncome[i] -
synthetic_one$logIncome_syn[i])^2)
#}
percentile_orig <- S_x(merged[,"Income"])
percentile_syn <- S_y(merged[,"Income"])

ecdf_diff <- percentile_orig - percentile_syn

U_m <- max(abs(ecdf_diff))
U_s <- mean((ecdf_diff)^2)
U_m
```

```
## [1] 0.06136821

U_s

## [1] 0.001139505

m <- 20
synthetic_m <- vector("list", m)

for (j in 1:m){
  synthetic_j <- synthesize(CEdata$LogExpenditure, CEdata$UrbanRural,
CEdata$Race, 1, n)
  names(synthetic_j) <- c("logExpenditure", "logIncome_syn")
  synthetic_m[[j]] <- synthetic_j
}

syn_mean <- vector("list", m)
for (j in 1:m){
  syn_mean[[j]] <- mean(synthetic_m[[j]]$logIncome_syn)
}
syn_mean <- unlist(syn_mean)

q_m_bar <- mean(syn_mean)
b_m <- sum((syn_mean - q_m_bar)^2/(m - 1))
u_m_bar <- var(syn_mean)

T_p <- b_m/m + u_m_bar

q_m_bar

## [1] 10.6063

b_m

## [1] 0.0008310477

u_m_bar

## [1] 0.0008310477

T_p

## [1] 0.0008726

L_o <- quantile(CEdata$LogExpenditure, .05)
U_o <- quantile(CEdata$LogExpenditure, .9)
L_s <- quantile(synthetic_one$logIncome_syn, .05)
U_s <- quantile(synthetic_one$logIncome_syn, .9)
L_i <- max(L_s, L_o)
U_i <- min(U_s, U_o)

I <- (U_i - L_i)/(2 * (U_o - L_o)) + (U_i - L_i)/(2 * (U_s - L_s))
```

```
I

##        90%
## 0.3825151
```