

framingham health data

```
library(runjags)
library(coda)
library(ggplot2)
data2<- read.csv("frmgham2.csv")
data2$sexd= fastDummies::dummy_cols(data2$SEX)[,names(fastDummies::dummy_cols(data2$SEX))== ".data_1"]
data2$educs= fastDummies::dummy_cols(data2$educ)[,names(fastDummies::dummy_cols(data2$educ))== ".data_1"]
data2$educsc= fastDummies::dummy_cols(data2$educ)[,names(fastDummies::dummy_cols(data2$educ))== ".data_1"]
data2$educcc= fastDummies::dummy_cols(data2$educ)[,names(fastDummies::dummy_cols(data2$educ))== ".data_4"]
data2$cigd= fastDummies::dummy_cols(data2$CURSMOKE)[,names(fastDummies::dummy_cols(data2$CURSMOKE))== ".data_1"]
data2<- na.omit(data2)

modelString <-"
model {
  for (i in 1:N){
    y[i] ~ dnorm(beta0 + beta1*x_hr[i] + beta2*x_igpd[i] + beta3*x_age[i], invsigma2)
  }
  beta0 ~ dnorm(mu0, g0)
  beta1 ~ dnorm(mu1, g1)
  beta2 ~ dnorm(mu2, g2)
  beta3 ~ dnorm(mu3, g3)
  invsigma2 ~ dgamma(a, b)
  sigma <- sqrt(pow(invsigma2, -1))
}
"

y = as.vector(data2$BMI)
x_hr = as.vector(data2$HEARTRTE)
x_igpd = as.vector(data2$CIGPDAY)
x_age = as.vector(data2$AGE)
N = length(y)

the_data2<- list("y" = y, "x_hr"= x_hr,
                "x_igpd"= x_igpd, "x_age"= x_age,
                "N" = N,
                "mu0" = 0, "g0" = 1, "mu1" = 0, "g1" = 1,
                "mu2" = 0, "g2" = 1, "mu3" = 0, "g3" = 1,
                "a" = 1, "b" = 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))
}

posterior2<- run.jags(modelString,
                      n.chains = 1,
                      data = the_data2,
                      monitor = c("beta0", "beta1", "beta2",
                                "beta3", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
```

```

        sample = 5000,
        thin = 1,
        inits = initsfunction)

## Loading required namespace: rjags

## Compiling rjags model...
## Calling the simulation using the rjags method...
## Note: the model did not require adaptation
## Burning in the model for 5000 iterations...
## Running the model for 5000 iterations...
## Simulation complete
## Calculating summary statistics...

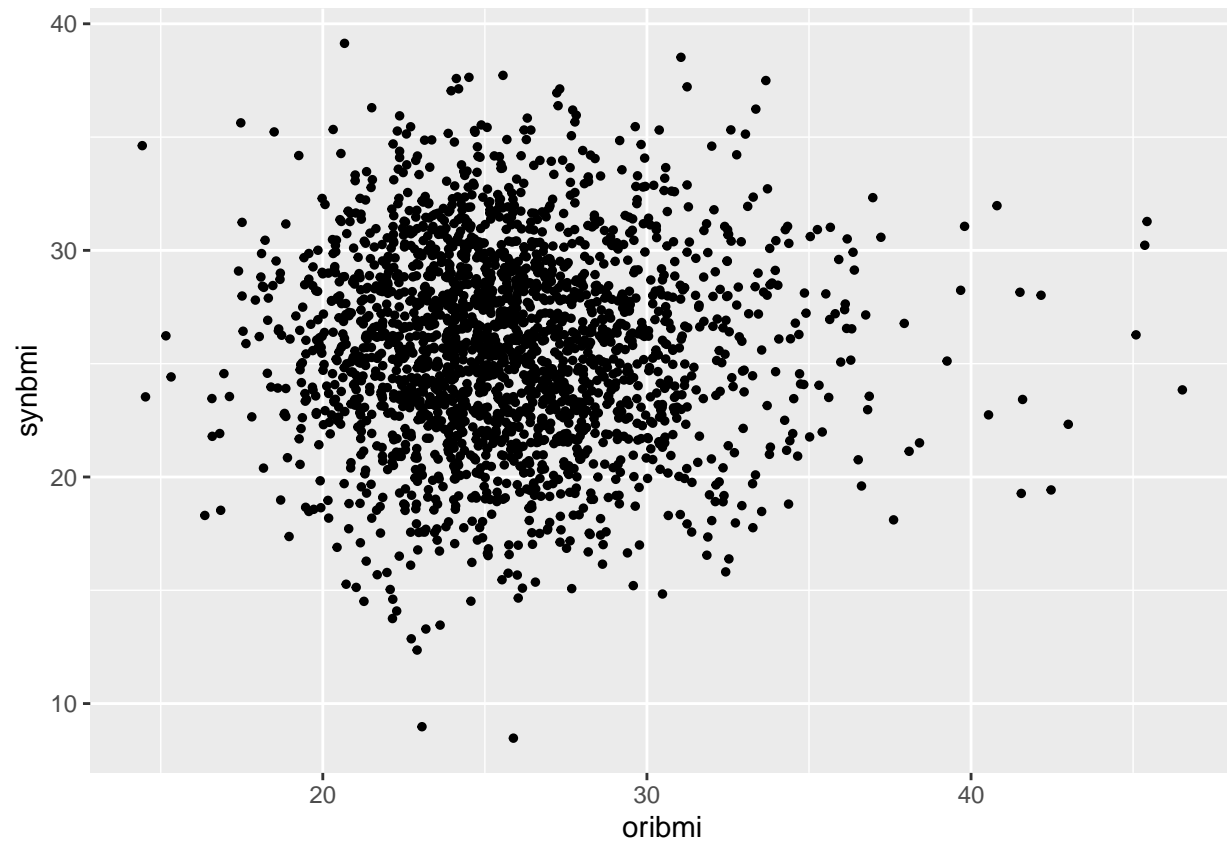
## Warning: Convergence cannot be assessed with only 1 chain

## Finished running the simulation
posterior2

##
## JAGS model summary statistics from 5000 samples (adapt+burnin = 6000):
##
##           Lower95      Median      Upper95      Mean      SD Mode      MCerr
## beta0      14.815      15.982      17.281      16.006      0.61447 --      0.087995
## beta1      0.057035    0.069132    0.081745    0.068968    0.0063925 --      0.0007446
## beta2     -0.030298   -0.014954   -0.00088427 -0.014963    0.0075358 --      0.00020452
## beta3      0.05541     0.073097     0.090914     0.072756    0.0092707 --      0.0012111
## sigma      3.8932      4.0208      4.1416      4.021      0.06377 --      0.0024006
##
##           MC%ofSD SSeff      AC.10 psrf
## beta0      14.3      49 0.82378 --
## beta1      11.6      74 0.7485 --
## beta2       2.7    1358 0.02312 --
## beta3      13.1      59 0.80014 --
## sigma       3.8     706 0.070464 --
##
## Total time taken: 5.5 seconds

post2 <- as.mcmc(posterior2)
syn2<- function(X, index, n){
  mean_Y <- post2[index, "beta0"] + X$x_hr * post2[index, "beta1"] + X$x_igpd * post2[index, "beta2"]
  syny<- rnorm(n,mean_Y, post2[index,"sigma"])
  data.frame(X$y, syny)
}
n <- dim(data2)[1]
frame2 <- data.frame(y, x_hr, x_igpd, x_age)
syndata2<- syn2(frame2, 1, n)
names(syndata2)<- c("oribmi", "synbmi")
ggplot(syndata2, aes(x= oribmi, y= synbmi)) + geom_point(size = 1)

```



```

modelString2 <-"
model {
  for (i in 1:N){
    y[i] ~ dnorm(beta0 + beta1*x_age[i] + beta2*x_sex[i] +
      beta3*x_educhs[i] + beta4*x_educsc[i] +
      beta5*x_educc[i], invsigma2)
  }
  beta0 ~ dnorm(mu0, g0)
  beta1 ~ dnorm(mu1, g1)
  beta2 ~ dnorm(mu2, g2)
  beta3 ~ dnorm(mu3, g3)
  beta4 ~ dnorm(mu4, g4)
  beta5 ~ dnorm(mu5, g5)
  invsigma2 ~ dgamma(a, b)
  sigma <- sqrt(pow(invsigma2, -1))
}
"

y= as.vector(data2$BMI)
x_cigd= as.vector(data2$AGE)
x_sex= as.vector(data2$sex)
x_educhs= as.vector(data2$educhs)
x_educsc= as.vector(data2$educsc)
x_educc= as.vector(data2$educc)
N= length(y)

```

```

the_data<- list("y" = y, "x_age"= x_age,
               "x_sex"= x_sex, "x_educhs"= x_educhs,
               "x_educsc"= x_educsc, "x_educc"= x_educc,
               "N" = N,
               "mu0" = 0, "g0" = 1, "mu1" = 0, "g1" = 1,
               "mu2" = 0, "g2" = 1, "mu3" = 0, "g3" = 1,
               "mu4" = 0, "g4" = 1, "mu5" = 0, "g5" = 1,
               "a" = 1, "b" = 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(modelString2,
                    n.chains = 1,
                    data = the_data,
                    monitor = c("beta0", "beta1", "beta2",
                               "beta3", "beta4", "beta5",
                               "sigma"),
                    adapt = 1000,
                    burnin = 5000,
                    sample = 5000,
                    thin = 1,
                    inits = initsfunction)

```

```

## Compiling rjags model...
## Calling the simulation using the rjags method...
## Note: the model did not require adaptation
## Burning in the model for 5000 iterations...
## Running the model for 5000 iterations...
## Simulation complete
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain
## Finished running the simulation

```

```

posterior

##
## JAGS model summary statistics from 5000 samples (adapt+burnin = 6000):
##
##      Lower95  Median  Upper95    Mean      SD Mode    MCerr MC%ofSD
## beta0   17.904   19.145   20.382   19.152   0.63679  --   0.088165   13.8
## beta1  0.082442  0.10161  0.12175  0.10176  0.0099933  --   0.0014264   14.3
## beta2   0.69838    1.008   1.3397   1.0091   0.16627  --   0.0037247    2.2
## beta3  -0.16589  0.21878  0.60871  0.22017  0.19887  --   0.0070986    3.6
## beta4  -1.0781 -0.62646 -0.17261 -0.62898  0.23253  --   0.0052512    2.3
## beta5  -0.90959 -0.40818  0.08918 -0.41333  0.25698  --   0.0052736    2.1
## sigma   3.8519   3.9798   4.1022   3.9817  0.064504  --   0.0027398    4.2
##
##      SSeff      AC.10 psrf
## beta0      52      0.80953  --

```

```
## beta1    49    0.80366  --
## beta2  1993  -0.0085434 --
## beta3   785   0.016964  --
## beta4  1961 -0.00021548  --
## beta5  2375  -0.0061096  --
## sigma   554   0.094295  --
##
## Total time taken: 5.3 seconds

post<- as.mcmc(posterior)
syn<- function(X, index, n){
  mean_Y<- post[index, "beta0"] + X$x_age * post[index, "beta1"] + X$x_sexd * post[index, "beta2"] + X$
  syny<- rnorm(n,mean_Y, post[index,"sigma"])
  data.frame(X$y, syny)
}
n<- dim(data2)[1]
frame<- data.frame(y, x_age, x_sexd, x_educhs, x_educsc, x_educc)
syndata<- syn(frame, 1, n)
names(syndata)<- c("oribmi", "synbmi")

ggplot(syndata, aes(x= oribmi, y= synbmi)) + geom_point(size = 1)
```

