KU Leuven Summer School
Segment 2A
First Look at Latents - Missing Data

Paul Gustafson

September 15, 2022

Missing data

```
require(mice) ### just want data ex. from this package
```

```
summary(nhanes2)
```

```
##     age           bmi          hyp           chl
##  20-39:12   Min.   :20.4   no  :13   Min.   :113
##  40-59: 7   1st Qu.:22.6   yes : 4   1st Qu.:185
##  60-99: 6   Median :26.8   NA's: 8   Median :187
##             Mean   :26.6             Mean   :191
##             3rd Qu.:28.9             3rd Qu.:212
##             Max.   :35.3             Max.   :284
##             NA's   :9                NA's   :10
```

Say interested in regressing *chl* (Y) on *age* (X_1), *hyp* (X_2), and *bmi* (X_3).

Toward a generative model (1 of 3)

$$f(\theta) \prod_{i=1}^{n} f(x_{1i}|\theta)\ f(x_{2i}|x_{1i},\theta)\ f(x_{3i}|x_{1i},x_{2i},\theta)\ f(y_i|x_{1i},x_{2i},x_{3i},\theta)$$

Toward a generative model (2 of 3)

```
statmod.string <-"
  for (i in 1:n) {
    x2[i] ~ dbern(pr.x2[i])
      logit(pr.x2[i]) <- alpha0 + alpha1a*x1a[i] +
                         alpha1b*x1b[i]

    x3[i] ~ dnorm(mn.x3[i], prec.x3)
      mn.x3[i] <- kappa0 + kappa1a*x1a[i] +
                  kappa1b*x1b[i]+kappa2*x2[i]

    y[i] ~ dnorm(mn.y[i], prec.y)
      mn.y[i] <- beta0 + beta1a*x1a[i] + beta1b*x1b[i] +
                 beta2*x2[i] + beta3*x3[i]
  }
"
```

Toward generative model (3 of 3)

```
prior.string <- "
  alpha0 ~ dnorm(0, 0.1)
  alpha1a ~ dnorm(0, 0.1)
  alpha1b ~ dnorm(0, 0.1)

  kappa0 ~ dnorm(0, 0.01)
  kappa1a ~ dnorm(0, 0.01)
  kappa1b ~ dnorm(0, 0.01)
  kappa2 ~ dnorm(0, 0.01)
  prec.x3 ~ dgamma(0.1, 0.1)

  beta0 ~ dnorm(0, 0.01)
  beta1a ~ dnorm(0, 0.01)
  beta1b ~ dnorm(0, 0.01)
  beta2 ~ dnorm(0, 0.01)
  beta3 ~ dnorm(0, 0.01)
  prec.y ~ dgamma(0.5, 0.5)
  sig.y <- sqrt(1/prec.y)
"
```

Housekeeping

```
genmod.string <- paste(
"model {",prior.string, statmod.string,"}")
```

Pause to comment on this prior and stat model specification

Almost have supplied a joint distribution of everything

Turn the JAGS crank

```
### generative model, data go in
mod <- jags.model(textConnection(genmod.string),
         data=list(x1a=as.numeric(nhanes2$age=="40-59"),
                   x1b=as.numeric(nhanes2$age=="60-99"),
                    x2=as.numeric(nhanes2$hyp)-1,
                    x3=nhanes2$bmi,
                     y=nhanes2$chl,
                     n=dim(nhanes2)[1]),
         n.chains=4)

update(mod, 2000)   ### burn-in

###  MC output comes out
opt1.JAGS <- coda.samples(mod, n.iter=10000,
  variable.names=c("beta1a","beta1b","beta2","beta3","sig.y",
    "x2[6]","x3[6]","y[6]"))
```

JAGS, continued

```
summary(opt1.JAGS)
```

```
##
## Iterations = 3001:13000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean    SD Naive SE Time-series SE
## beta1a    5.36 9.325  0.04662        0.05306
## beta1b    7.81 9.824  0.04912        0.05631
## beta2     4.61 9.493  0.04747        0.05035
## beta3     6.89 0.590  0.00295        0.00500
## sig.y    43.05 9.236  0.04618        0.05678
## x2[6]     0.43 0.495  0.00248        0.00337
## x3[6]    24.79 3.979  0.01989        0.02550
## y[6]    184.00 0.000  0.00000        0.00000
##
## 2. Quantiles for each variable:
```

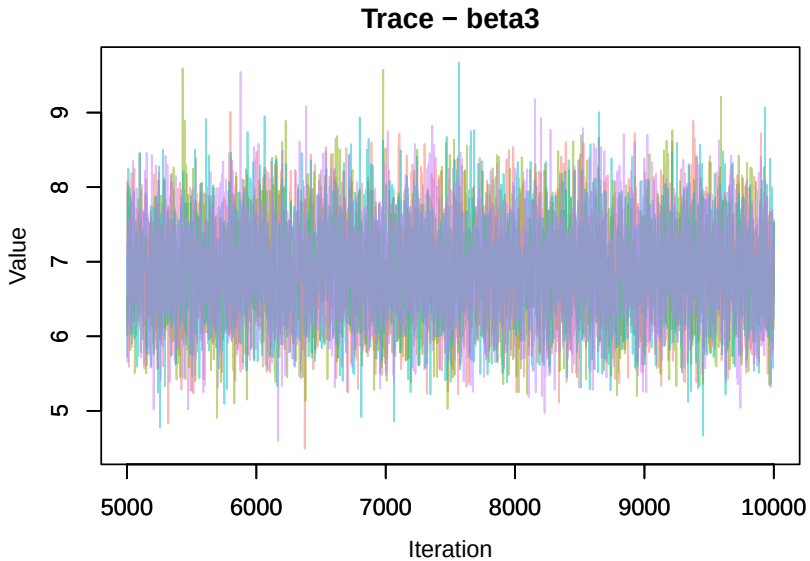
Or

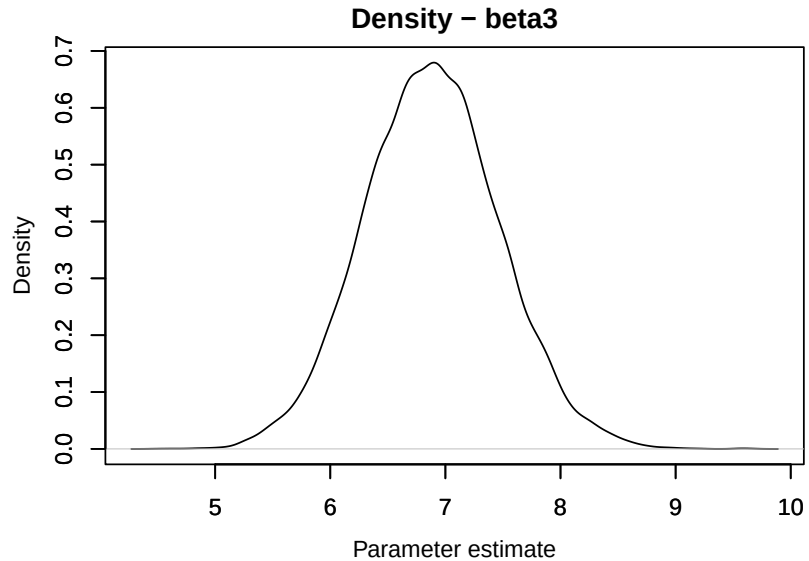
```
require(MCMCvis)
MCMCsummary(opt1.JAGS)
```

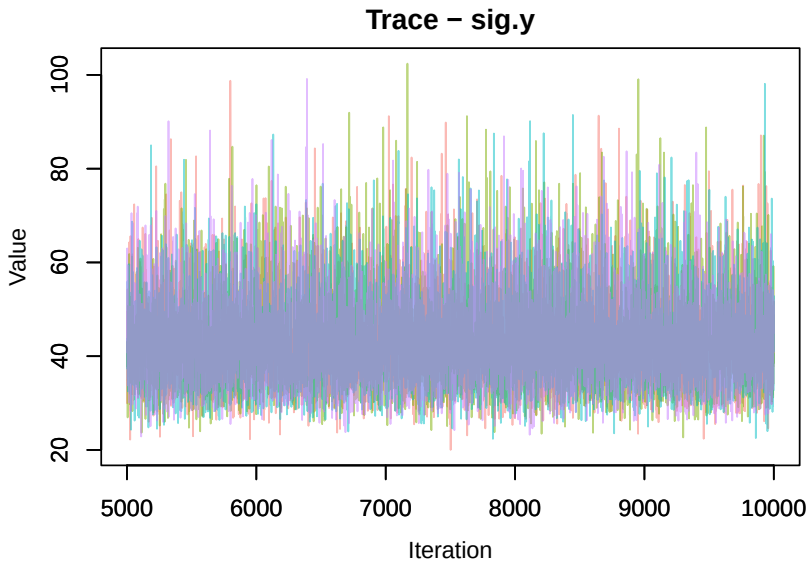
```
##           mean    sd   2.5%    50%  97.5% Rhat n.eff
## beta1a    5.36 9.325 -12.96   5.46  23.57    1 30919
## beta1b    7.81 9.824 -11.44   7.84  27.08    1 30503
## beta2     4.61 9.493 -13.98   4.63  23.33    1 35557
## beta3     6.89 0.590   5.75   6.88   8.06    1 13940
## sig.y    43.05 9.236  29.00  41.72  64.90    1 26495
## x2[6]     0.43 0.495   0.00   0.00   1.00    1 21675
## x3[6]    24.79 3.979  17.04  24.77  32.79    1 24423
## y[6]    184.00 0.000 184.00 184.00 184.00  NaN     0
```

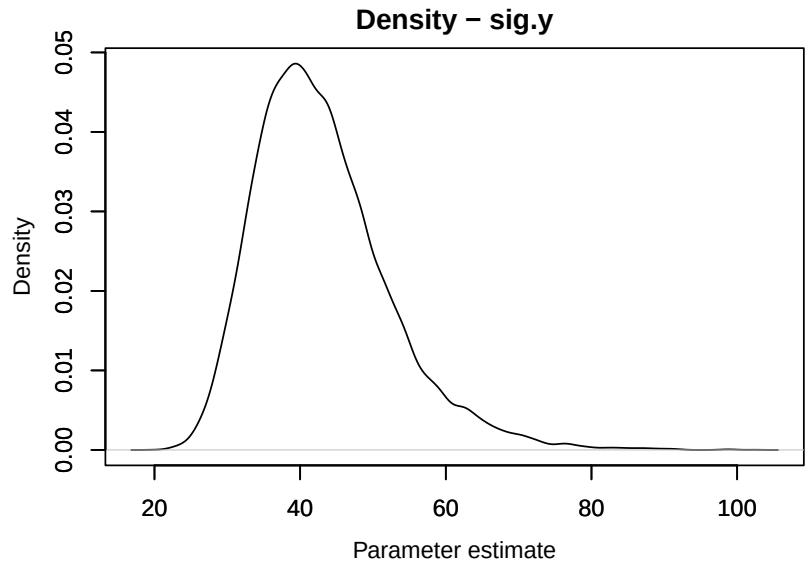
Some due diligence on our computational work

```
MCMCtrace(opt1.JAGS, params=c("beta3","sig.y"), pdf=F)
```









Thoughts: Under-the-hood, exactly what distribution are the Monte Carlo draws (approximately) coming from?

Thoughts: Hall-pass that freed me from having a model for X_1?

Or did I have a "hall pass" to cheat a little?

Under the hood JAGS produced Monte Carlo draws from joint?
What about from marginal instead?

Thoughts: In concept at least could I have done the computing in the "collapsed" frame of reference?

$f(\theta) \prod_{i=i}^{n} f(observed_i|\theta)$

Collapsed, for instance:

```
nhanes2[c(3,6),]
```

```
##     age bmi  hyp chl
## 3 20-39  NA   no 187
## 6 60-99  NA <NA> 184
```

Collapsed versus Augmented: Two different strategies **for computing the same thing**

Collapsed:

$$f(\theta|\text{observed}) \quad \propto \quad f(\text{observed}|\theta)f(\theta)$$

Augmented:

$$f(\theta|\text{observed}) = \int f(\theta, \text{latent}|\text{observed})d\text{latent}$$

Any more thoughts?