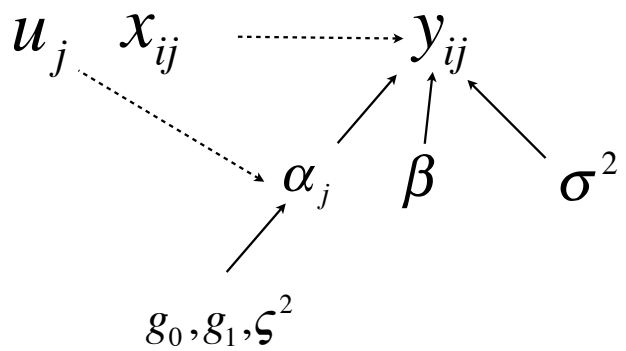
Indexing

ESS 575 Models for Ecological Data

N. Thompson Hobbs

March 27, 2019

An essential coding trick: Indexing groups



$$\left[\boldsymbol{\alpha}, \beta, , \sigma^2, \mathbf{g}, \varsigma^2, |\mathbf{y}\right] \propto \prod_{i=1}^{n_j} \prod_{j=1}^{J} \text{normal}\left(y_{ij} | \alpha_j + \beta x_{ij}, \sigma^2\right)$$
$$\times \text{normal}\left(\alpha_j | g_0 + g_1 u_j, \varsigma^2\right)$$
$$\times \text{normal}\left(\beta | 0, .001\right) \text{normal}\left(g_0 | 0, 1000\right) \times \text{normal}\left(g_1 | 0, 1000\right)$$
$$\times \text{inverse gamma}\left(\sigma^2 | .001, .001\right) \text{uniform}\left(\varsigma | 0, 200\right)$$

Indexing groups

```
> u
[1]  6.215579  8.716296 10.064460 11.292387 14.504154 14.734861
[7] 18.356877 18.910133
```

```
> head(y[,1:4])
     group i        x[i]     y[i]
[1,]     1 1 -0.00266051 13.48934
[2,]     1 2  4.54802848 22.29538
[3,]     1 3  9.86832462 29.03655
[4,]     1 4  0.99869789 18.61136
[5,]     1 5  1.27733200 20.59178
[6,]     1 6  4.32915675 25.37082
> tail(y[,1:4])
       group   i     x[i]     y[i]
[108,]     8 108 4.543959 38.93163
[109,]     8 109 1.287844 34.65796
[110,]     8 110 6.642313 40.62259
[111,]     8 111 7.404183 40.46518
[112,]     8 112 8.252571 41.47995
[113,]     8 113 9.558780 46.14771
```

Indexing groups

```
model{
beta ~ dnorm(0,.0001)
sigma ~ dunif(0,50)
tau.p <- 1/sigma^2
g0 ~ dnorm(0,.0001)
g1 ~ dnorm(0,.0001)
varsigma ~ dunif(0,50)
tau.g <- 1/varsigma^2
 for (i in 1:length(y)){
  mu[i] <- alpha[group[i]]+ beta*x[i]
  y[i] ~ dnorm(mu[i],tau.p)
  }
  for(j in 1:n.group){
  mu.g[j] <- g0 + g1*u[j]
  alpha[j]~dnorm(mu.g[j],tau.g)
    }
```

Creating an index for a group

Assume you have a data frame (or tibble) called `y` with a column for sites containing a character code describing each site, i.e. `y$site`. There are multiple observations within each site. How would you create a sequential index such that each site has its own, unique integer value corresponding to the site code, something like `y$site.index`?[1]

```
library(dplyr) # or library(tidyverse)
y = y %>% mutate(site.index = as.integer(as.factor(site)))
```

or, using base R

```
y$site.index = as.integer(as.factor(site))
```

[1] The `tidyverse` packages, particularly `dplyr` are spectacular.

Creating nested indices ([2])

```
##Make index for site (each unique)
dat = dat %>% mu-
tate(Site.index = as.numeric(as.factor(Site)))
##Make index for strata
dat = dat %>% mu-
tate(Stratum.index = as.numeric(as.factor(Stratum)))
##Make index for sites within strata
dat = dat %>% group_by(Stratum) %>%
mutate(Site.within.strat.index = 1:n()) %>%
ungroup
```

```
> dat
# A tibble: 6 x 6
  Stratum     Site   Count Site.index Stratum.index Site.within.str…
  <fct>       <fct>  <int>      <dbl>         <dbl>             <int>
1 Poudre      Beave…    44          1             2                 1
2 Poudre      LoneP…    23          4             2                 2
3 Poudre      Butes     67          2             2                 3
4 BigThompson HighB…   204          3             1                 1
5 BigThompson Willo…    23          6             1                 2
6 BigThompson Sheep…   123          5             1                 3
>
```

[2] see code MakeIndex.R