Indexing

Data Structures

- So far, the data have all been balanced
  - rectangular structure
- BUGS can handle more flexible structures
- e.g. observations from three individuals:
  - Person 1: 13.2
  - Person 2: 12.3 , 14.1
  - Person 3: 11.0, 9.7, 10.3, 9.6
- Called Ragged Array

How to Handle I: ignore the problem

- Fill the array with missing data:
-
```
y=structure(.Data=
c(13.2,   NA,   NA, NA,
  12.3, 14.1,   NA, NA,
  11.0,  9.7, 10.3, 9.6),.Dim=c(4,3))
```
- BUGS has to estimate the missing data

```
for (i in 1:3) {
   mu[i] ~ dunif(0, 100)
   for (j in 1:4) { y[i,j]~dnorm(mu[i],1)}
}
```

How to Handle II: Indexing

- Create an index that says which group the datum is in
- **People=c(13.2, 12.3, 14.1, 11.0, 9.7, 10.3, 9.6),**
  Ind=c(1, 2, 2, 3, 3, 3, 3)
- Then use the index to give the right value:

```
for(i in 1:3) { mu[i] ~ dunif(0, 100)}

for(k in 1:7) { y[k]~dnorm(mu[Ind[k]], 1)}
```

-

More tricks with Indexing

- The indexing trick can even be used if the index in unknown
  - treat it as missing data
- `People=c(13.2, 12.3, 14.1, 11.0, 9.7, 10.3, 9.6), Ind=c(1, 2, 2, NA, 3, 3, 3)`
- Then use the index to give the right value:

```
for(k in 1:7) {
   y[k]~dnorm(mu[Ind[k]], 1)
   Ind[k] ~ dcat(p[])
}
for(j in 1:3) { p[j] <- 1/3  }
```

dcat()

- dcat defines a categorical distribution
  - Index[i] ~ dcat(p[])
- Index[i] takes values 1,2,3,4...
  - up to the length of *p*
- p[] gives the probabilities of being in each class
  - Must add up to 1, hence: theta[1] <-p; theta[2] <-1-p
- Can easily generalise to several groups