Roadmap

- Models in ecology
- A toolbox of deterministic models
- Stochasticity
- Probability models
- Likelihood
- Understanding Bayes theorem
- Classical Bayes
- Monte Carlo Markov chain
- Programming MCMC
- Hierarchical models
- JAGS
- Bayesian regression and multilevel models (random effects)
- Zero-inflation, occupancy, capture mark recapture
- State space models
- Model evaluation and selection
- Last lecture: April 25
- April 26 - May 17 individual projects

Individual projects

- Application of Bayesian methods to an ecological problem of your choice
  - your data
  - simulated data
  - data I give you
- Please discuss your problem with me before you start work.
- Write up formatted as article for submission to Ecology.
- Due at end of finals week.

Bayesian Regression

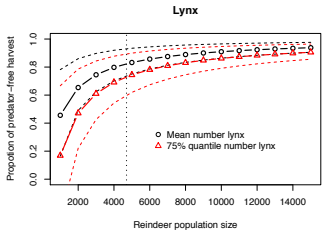
Simple components of complex models
ESS 575 April 9, 2013

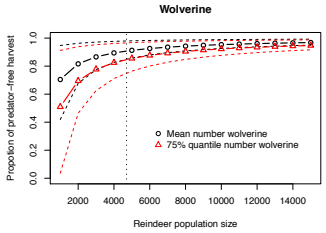
Gelman, A., and J. Hill. 2009. Data analysis using regression and multilevel / hierarchical models. Cambridge University Press, Cambridge, UK.

Reindeer model

$$\lambda_{i,t} = e^{r+\beta_0 N_{i,t-1}+\beta_1 lynx_{i,t}+\beta_2 wolverine_{i,t}+\beta_3 gradient_i+\beta_4 NAO_t}$$

$$N_{i,t} = \lambda_{i,t} N_{i,t-1} - H_{i,t}$$





Data (the y's)	Regression type
Continuous, real valued	normal
Discrete, strictly positive	Poisson
0 or 1	logistic
0 to 1	beta
0 to a	modified logistic
Continuous, strictly positive	lognormal or gamma (or normal)

Normal, data continuous and real valued

$$g(b_0, b_1, x_i) = b_0 + b_1 x_i$$

$$P(b_0, b_1, \tau \mid \mathbf{y}, \mathbf{x}) \propto \prod_{i=1}^{n} \text{normal}(y_i \mid g(b_0, b_1, x_i)_i, \tau) \times$$
$$\text{normal}(b_0 \mid 0.0001)\,\text{normal}(b_1 \mid 0, .0001)\,\text{gamma}(\tau \mid .00001 .00001)$$

```
model{
b0 ~ dnorm(0,.0001)
b1 ~ dnorm(0,.0001)
tau ~ dgamma(.0001,.0001)
sigma<-1/sqrt(tau)
for(i in 1:length(y)){
   mu[i] <- b0 + b1*x[i]
   y[i] ~ dnorm(mu[i],tau)
   }
}
```

Poisson, data discrete and positive

$$g(b_0, b_1, x_i) = e^{b_0 + b_1 x_i}$$

$$P(b_0, b_1 \mid \mathbf{y}, \mathbf{x}) \propto \prod_{i=1}^{n} \text{Poisson}(y_i \mid g(b_0, b_1, x_i)) \times$$
$$\text{normal}(b_0 \mid 0.0001)\,\text{normal}(b_1 \mid 0, .0001)$$

Where did σ go?

```
b0 ~ dnorm(0,.000001)
b1 ~ dnorm(0,.000001)
for(i in 1:length(y)){
  log(mu[i]) <- b0 + b1*x[i]
  y[i] ~ dpois(mu[i])
  }
}
```

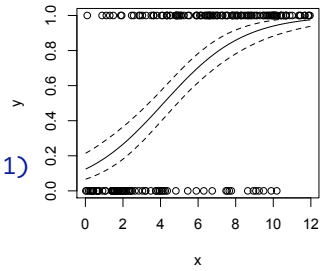
Bernoulli, data 0 or 1(aka logistic)

$$g(b_0,b_1,x_i)=\frac{1}{1+e^{-(b_0+b_1x_1)}}$$

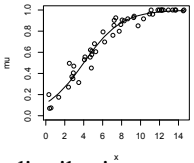
$$P(b_0,b_1\mid \mathbf{y},\mathbf{x})\propto \prod_{i=1}^{n}\text{Bernoulli}(y_i\mid g(b_0,b_1,x_i))\times$$

$$\text{normal}(b_0\mid 0.0001)\text{normal}(b_1\mid 0,.0001)$$

```
model{
b0 ~ dnorm(0,.0000001)
b1 ~ dnorm(0,.0000001)
for(i in 1:length(y)){
 logit(p[i]) <- b0 + b1*x[i]
 y[i] ~ dbern(p[i]) #same as dbin(p[i],1)
 }
}
```



Beta, 0 to1 data



$$g(b_0,b_1,x_i)=\frac{1}{1+e^{-(b_0+b_1x_1)}}$$

$$\left.\begin{aligned} a&=f_1\left[g(b_0,b_1,x_i),\sigma\right]\\ b&=f_2\left[g(b_0,b_1,x_i),\sigma\right]\end{aligned}\right\}\text{moment matching functions for beta distribution}$$

$$P(b_0,b_1,\sigma\mid \mathbf{y},\mathbf{x})\propto \prod_{i=1}^{n}\text{beta}(y_i\mid a,b)\times$$

$$\text{normal}(b_0\mid 0,.0001)\text{normal}(b_1\mid 0,.0001)\text{gamma}(\sigma\mid .0001.001)$$

```
model{
 b0 ~ dnorm(0,.000001)
 b1 ~ dnorm(0,.000001)
 sigma ~ dunif(0,.1)
 for(i in 1:length(y)){
  logit(mu[i]) <- b0 + b1*x[i]
  alpha[i] <-(mu[i]^2-mu[i]^3-mu[i]*sigma^2)/sigma^2
  beta[i] <- (mu[i]-2*mu[i]^2+mu[i]^3-sigma^2+mu[i]*sigma^2)/
sigma^2
  y[i] ~ dbeta(alpha[i],beta[i])
  }
 }
```

Link functions

$$g(b_0,b_1,x_i)=b_0+b_1x_i$$

$$P(b_0,b_1\mid \mathbf{y},\mathbf{x})\propto \prod_{i=1}^{n}\text{Poisson}\left(\underbrace{\log(y_i)}_{\text{a log link}}\mid g(b_0,b_1,x_i)\right)\times$$

$$\text{normal}(b_0\mid 0.0001)\text{normal}(b_1\mid 0,.0001)$$

$$g(b_0,b_1,x_i)=b_0+b_1x_i$$

$$P(b_0,b_1\mid \mathbf{y},\mathbf{x})\propto \prod_{i=1}^{n}\text{Bernoulli}\left(\underbrace{\text{logit}(y_i)}_{\text{a logit link}}\mid g(b_0,b_1,x_i)\right)\times$$

$$\text{normal}(b_0\mid 0.0001)\text{normal}(b_1\mid 0,.0001)$$

What caused this error?

Error in jags.model("beta",
data = data, n.chain = 1,
n.adapt = 1000) :
  Error in node y[7]
Invalid parent values

Keeping variables defined

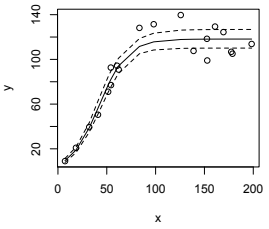
```
model{
b0 ~ dnorm(0,.00001)
b1 ~ dnorm(0,.00001)
sigma ~ dunif(0, 1)
for(i in 1:length(y)){
  logit(mu[i]) <- b0 + b1*x[i]
  a[i] <- max(.0000000001, (mu[i]^2-mu[i]^3-
mu[i]*sigma^2)/sigma^2)
  b[i] <- max(.000000001,(mu[i]-2*mu[i]^2+mu[i]^3-
sigma^2+mu[i]*sigma^2)/sigma^2)
  y[i] ~ dbeta(a[i],b[i])
  }
```

Continuous data, 0 to upper asymptote, *a*

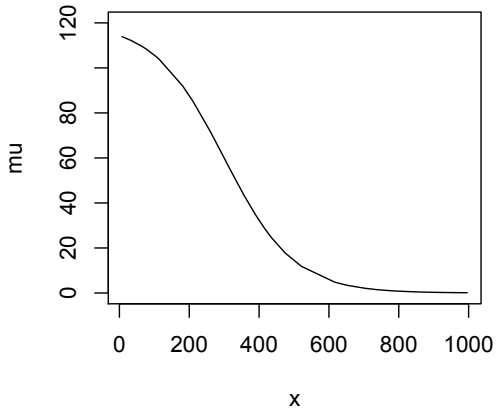
$$g(a,b_0,b_1,x_i)=\frac{a}{1+e^{-(b_0+b_1x_1)}}$$

$$P(a,b_0,b_1,\sigma\,|\,\mathbf{y},\mathbf{x})\propto\prod_{i=1}^{n}\text{lognormal}\left(y_i\,|\,\log\left[g(a,b_0,b_1,x_i)\right],\tau\right)\times$$
$$\text{gamma}(a\,|\,.001,.001)\,\text{normal}\left(b_0\,|\,0,.0001\right)\text{normal}(b_1\,|\,0,.0001)$$
$$\text{gamma}\left(\sigma\,|\,.001,.001\right)$$

```
model{
a~dunif(50,200)
b0 ~ dnorm(0,.0001)
b1 ~ dnorm(0,.0001)
tau ~ dgamma(.0001,.0001)
sigma<-1/sqrt(tau)
for(i in 1:length(y)){
  mu[i] <- a*ilogit(b0+b1*x[i])
  y[i] ~ dlnorm(log(mu[i]),tau)
  }
}
```



Can also have negative slope



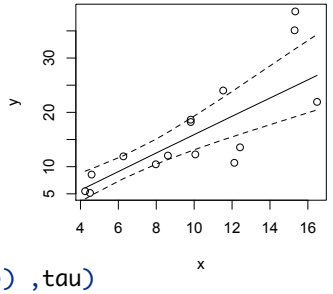
lognormal, data continuous and > 0

$$g(b_0,b_1,x_i)=b_0+b_1x_i(\ )$$

$$P(b_0,b_1,\tau\,|\,\mathbf{y},\mathbf{x})\propto\prod_{i=1}^{n}\text{lognormal}\left(y_i\,|\,\log\left[g(b_0,b_1,x_i)\right],\tau\right)\times$$
$$\text{normal}\left(b_0\,|\,0,.0001\right)\text{normal}(b_1\,|\,0,.0001)\,\text{gamma}\left(\tau\,|\,.00001.00001\right)$$

```
#in R:
data=list(log.y=log(y),x=x)

model{
b0 ~ dnorm(0,.0001)
b1 ~ dnorm(0,.0001)
tau ~ dgamma(.0001,.0001)
sigma<-1/sqrt(tau)
for(i in 1:length(y)){
  mu[i] <- b0 + b1*x[i]
  y[i] ~ dlnorm(log(max(.000001,mu[i])) ,tau)
  }
}
```



Talk about interpretation of τ

you will also see

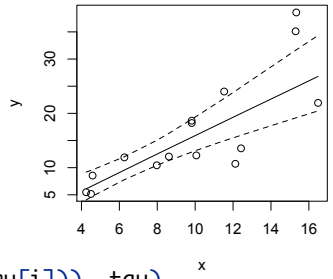
$$g(b_0, b_1, x_i) = b_0 + b_1 x_i$$

$$P(b_0, b_1, \tau \mid \mathbf{y}, \mathbf{x}) \propto \prod_{i=1}^{n} \text{normal}\left(\log(y) \mid \log\left[g(b_0, b_1, x_i)\right], \tau\right) \times$$

$$\text{normal}(b_0 \mid 0.0001)\,\text{normal}(b_1 \mid 0, .0001)\,\text{gamma}(\tau \mid .00001 .00001)$$

```
#in R:
data=list(log.y=log(y),x=x)

model{
b0 ~ dnorm(0,.0001)
b1 ~ dnorm(0,.0001)
tau ~ dgamma(.0001,.0001)
sigma<-1/sqrt(tau)
for(i in 1:length(y)){
  mu[i] <- b0 + b1*x[i]
  log(y[i]) ~ dlnorm(log(max(.000001,mu[i])) ,tau)
  }
}
```



All of the examples above apply to non-linear models as well as the linear forms.

Homework: write out the posterior and the joint distribution for a gamma regression.

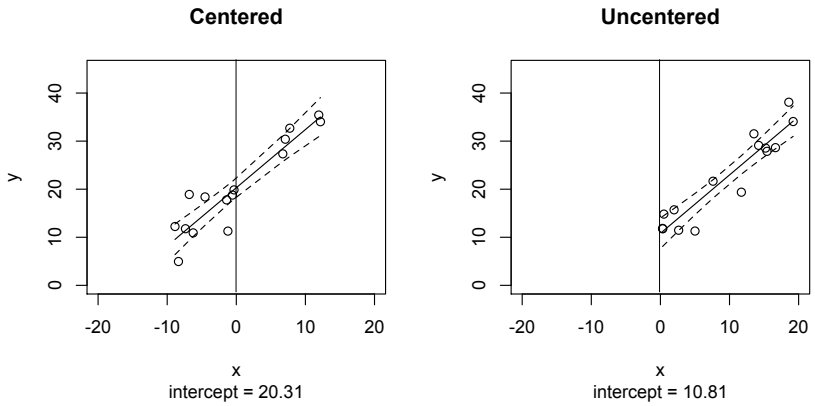
The remainder of the slides (centering and multilevel models) apply to *all* of the models we covered in Bayesian regression, but I will use a simple linear model as example.

Centering data

$$y_i = b_0 + b_1\left(x_i - \bar{x}\right)$$

Why ?

- To reduce autocorrelation in MCMC chain and speed convergence.
- To make coefficients more easily interpreted.




```
#recover intercept in centered regression
b0.uc <- b0 - b1*x.bar
#slopes will not be the same if there is an interaction term or
quadratic.  In these cases, back transforming is not simple.
```

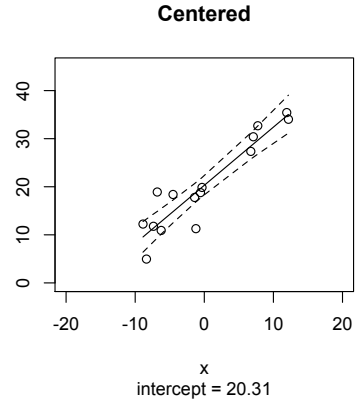
Centering predictor data

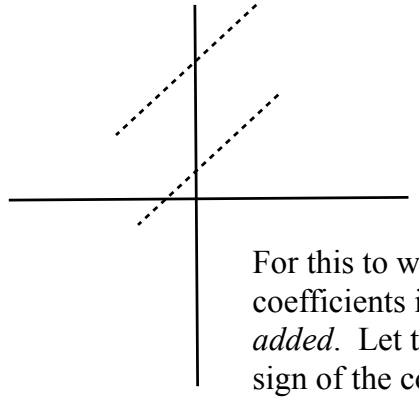
$$g(b_0, b_1, x_i, \bar{x}) = b_0 + b_1\left(x_i - \bar{x}\right)$$

$$P\left(b_0, b_1, \tau \mid \mathbf{y}, \mathbf{x}\right) \propto \prod_{i=1}^{n} \text{normal}\left(y_i \mid g(b_0, b_1, x_i, \bar{x})_i, \tau\right) \times$$

$$\text{normal}\left(b_0 \mid 0.0001\right)\text{normal}(b_1 \mid 0, .0001)\text{gamma}\left(\tau \mid .00001.00001\right)$$

```
model{
b0 ~ dnorm(0,.0001)
b1 ~ dnorm(0,.0001)
tau ~ dgamma(.0001,.0001)
sigma<-1/sqrt(tau)
x.bar <- mean(x)
for(i in 1:length(y)){
  mu[i] <- b0 + b1*(x[i]-x.bar)
  y[i] ~ dnorm(mu[i],tau)
  }
#recover intercept
b0.uc <- b0 - b1*x.bar
}
```





$y_i = b_0 + b_1\left(x_i - \bar{x}\right)$

Collect over x_i

$y_i = \overbrace{b_1}^{B_1} x_i + \overbrace{b_o - b_1\bar{x}}^{B_0}$

Back transform:

$B_0 = b_o - b_1\bar{x}$

$B_1 = b_1$

$y = B_0 + B_1 x$

For this to work properly, all of the coefficients in the model must be *added*. Let the analysis decide the sign of the coefficients. i.e.,

$y_i = b_0 + b_1 x_i$

not

$y_i = b_0 - b_1 x_i$

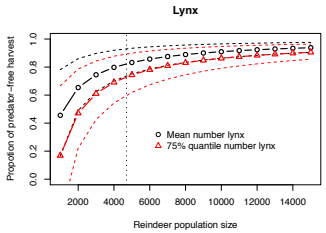
For multiple regression:

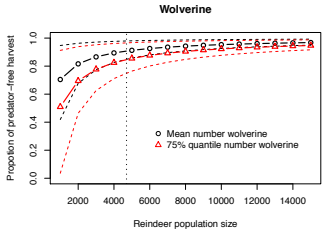
$$b_0^{uncentered} = b_0^{centered} - b_1\bar{x}_1 - b_2\bar{x}_2 \ldots\ldots - b_n\bar{x}_n$$

Reindeer model: example of centering to improve interpretation

$$\lambda_{i,t} = e^{r+\beta_0 N_{t-1}+\beta_1 lynx+\beta_2 wolverine+\beta_3 gradient+\beta_4 NAO}$$

$$N_t = \lambda_{i,t} N_{t-1} - H_t$$





Centering to improve interpretation of the intercept

$$\lambda_{i,t} = e^{r+\beta_0 N_{t-1}+\beta_1 lynx+\beta_2 wolverine+\beta_3 gradient+\beta_4 NAO}$$

$$N_t = \lambda_{i,t} N_{t-1} - H_t$$

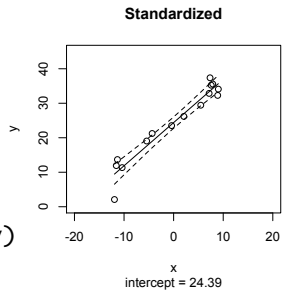
For the un-centered regression, the value of the intercept, *r,* was *enormous*. Why? Hint: consider the N-S gradient

Standardizing predictor data

$$g(b_0, b_1, x_i, \bar{x}, \sigma_x) = b_0 + b_1\left(\frac{x_i - \bar{x}}{\sigma_x}\right)$$

$$P(b_0, b_1, \tau \mid \mathbf{y}, \mathbf{x}) \propto \prod_{i=1}^{n} \text{normal}\left(y_i \mid g(b_0, b_1, x_i, \bar{x}, \sigma_x), \tau\right) \times$$
$$\text{normal}(b_0 \mid 0.0001)\,\text{normal}(b_1 \mid 0, .0001)\,\text{gamma}(\tau \mid .00001.00001)$$

```
 model{
b0 ~ dnorm(0,.0001)
b1 ~ dnorm(0,.0001)
tau ~ dgamma(.0001,.0001)
sigma<-1/sqrt(tau)
x.bar <- mean(x)
sdev <- sd(x)
for(i in 1:length(y)){
  mu.z[i] <- b0 + b1*((x[i]-x.bar)/sdev)
  y[i] ~ dnorm(mu.z[i],tau)
  }
}
```



Back transforming slopes and intercepts

$$y_i = b_0 + b_1\left(\frac{x_i - \bar{x}}{\sigma}\right)$$

Collect over x_i

$$y_i = \frac{b_1}{\sigma} x_i + b_o - \frac{b_1 \bar{x}}{\sigma}$$

Back transform:

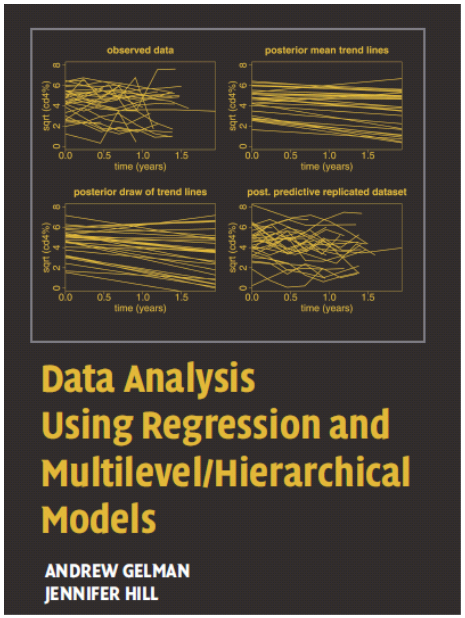
$$B_0 = b_o - \frac{b_1 \bar{x}}{\sigma}$$

$$B_1 = \frac{b_1}{\sigma}$$

This only works if there are not squared values or interactions

Interpreting standardized coefficients (slopes)

If there are no interactions, each coefficient gives estimate the change in y that occurs per change in 1 standard deviation of the x. Therefore, we can compare the coefficients to estimate their "relative importance" in the model.



Multi-level regression models

Gelman, A., and J. Hill. 2009. Data analysis using regression and multilievel / hierarchical models. Cambridge University Press, Cambridge, UK.

Multi-level regression models

Group 1 | Group 2 | Group 3 | ..... | Group n

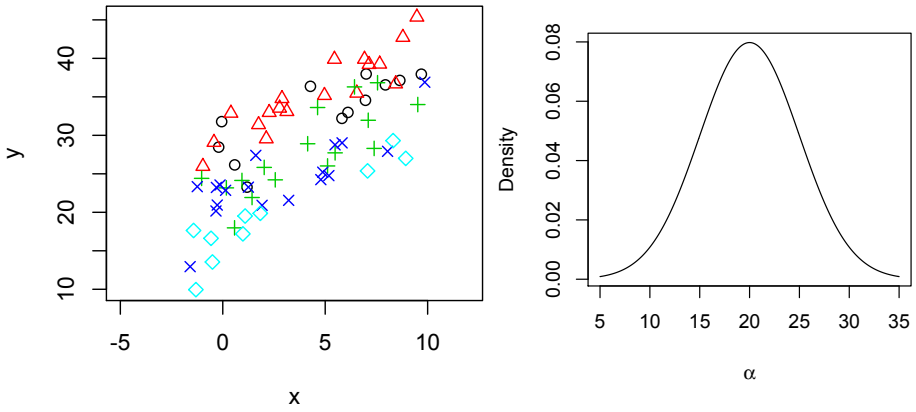
$\mathbf{y}_1, \mathbf{x}_1$ | $\mathbf{y}_2, \mathbf{x}_2$ | $\mathbf{y}_3, \mathbf{x}_3$ | ..... | $\mathbf{y}_n, \mathbf{x}_n$

The **y**'s are vectors of responses and the **x**'s are vectors or matrices of predictors (covariates). We want to predict the ith response.

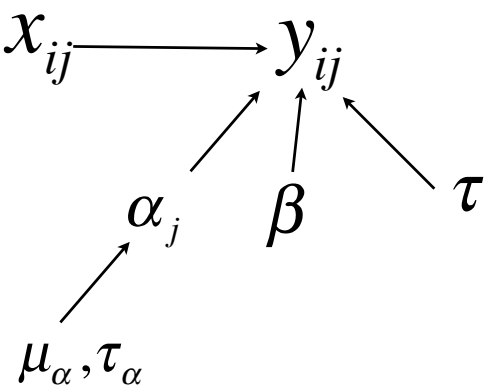
How can we model the relationships between the x's and the y's? What are some alternatives?

We could treat the α (and/or the β) as a *random effect* (aka group level effect):

Each group has its own α



We could treat the α (and/or the β) as a *random effect*:



The *[]* subscripting notation used by Gelman

$$y_i = \alpha_{j[i]} + \beta x_i$$

instead of

$$y_{ij} = \alpha_j + \beta x_{ij}$$

We could treat the α (and/or the β) as a *random effect*:

$y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i \}$ note the notation, observation i within group j

$\epsilon_i \sim normal(0, \sigma)$

$\tau = \frac{1}{\sigma^2}$

$$P(\alpha_j, \beta, \tau_p, \mu_{alpha}, \tau_\alpha, | \mathbf{y}, \mathbf{x}) \propto \prod_{i=1}^{I} \text{normal}(y_i | \alpha_{j[i]} + \beta x_i, \tau_p) \prod_{j=1}^{J} (\alpha_j | \mu_\alpha, \tau_p) \times$$
$$\text{normal}(\beta | 0, .000001) \text{normal}(\mu_\alpha | 0, .00001) \times$$
$$\text{gamma}(\tau_p | .001, .001) \text{gamma}(\tau_\alpha | .001, .001)$$

We could treat the α (and/or the β) as a *random effect*:

$$y_{ij} = \alpha_j + \beta x_{ij} + \epsilon_{ij}\Big\} \text{ more conventional notation}$$

$$\epsilon_{ij} \sim normal(0,\sigma)$$

$$\tau = \frac{1}{\sigma^2}$$

$$P(\alpha_j,\beta,\tau_p,\mu_{alpha},\tau_\alpha,|\,\mathbf{y},\mathbf{x}) \propto \prod_{j=1}^{J}\prod_{i=1}^{I}\left[\text{normal}(y_{ij}\,|\,\alpha_j+\beta x_{ij},\tau_p)\right]\prod_{j=1}^{J}(\alpha_j\,|\,\mu_\alpha,\tau_p)\times$$
$$\text{normal}(\beta\,|\,0,.000001)\text{normal}(\mu_\alpha\,|\,0,.00001)\times$$
$$\text{gamma}(\tau_p\,|\,.001,.001)\text{gamma}(\tau_\alpha\,|\,.001,.001)$$

The data set-up

```
x.hat=seq(-2,15,.5)
data=list(y=y[,4],x=y[,3],group=y[,1], n.group=n.group,
x.hat=x.hat)
```

```
> head(y[,1:4])
     group i     x[i]     y[i]
[1,]     1 1 9.217824 32.15281
[2,]     1 2 7.791492 32.66733
[3,]     1 3 4.227976 21.65790
[4,]     1 4 1.801340 19.29774
[5,]     1 5 1.647698 26.27173
[6,]     1 6 4.174580 24.74811
> tail(y[,1:4])
      group  i       x[i]      y[i]
[69,]     5 69  6.1766202 14.915067
[70,]     5 70  0.2885956 11.574517
[71,]     5 71 -0.4279072 10.943233
[72,]     5 72 -1.8753530  7.558408
[73,]     5 73 -0.8184513 17.404322
[74,]     5 74  3.8724734 18.656089
```

Some cool tricks

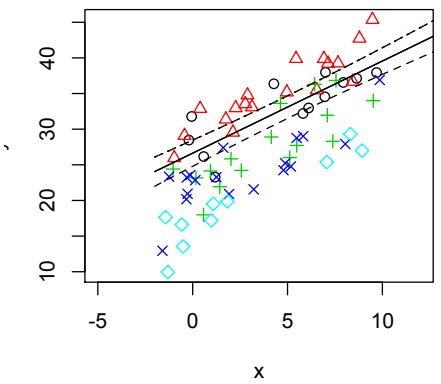
```
model{
beta ~ dnorm(0,.000001)
tau.p ~ dgamma(.0001,.0001)
sigma <- 1/sqrt(tau)
mu.alpha ~ dnorm(0,.00001)
tau.alpha ~ dgamma(.001,.001)
 for (i in 1:length(y)){
  mu[i] <- alpha[group[i]]+ beta*x[i]
  y[i] ~ dnorm(mu[i],tau.p)
  }
  for(j in 1:n.group){
  alpha[j]~dnorm(mu.alpha,tau.alpha)
    }
  #combined predictions
  for(i in 1:length(x.hat)){
    y.hat.all[i] <- mu.alpha + beta*x.hat[i]
    }
  #individual group predictions
  for(j in 1:n.group){
    for(i in 1:length(x.hat)){
    y.hat[i,j] <- alpha[j] + beta*x.hat[i]
    }
  }
}# end of model
```

← The index trick

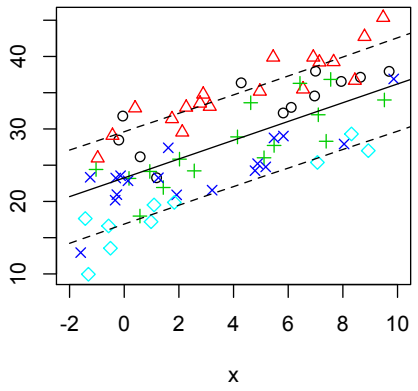
← The hat trick

Estimates

One group (circles)



Across groups



Random effects on intercepts

Alternative notation

$y_i = \alpha + \beta x_i + \gamma_j$

$\gamma_j \sim \text{normal}(0, \sigma)$

is the same as

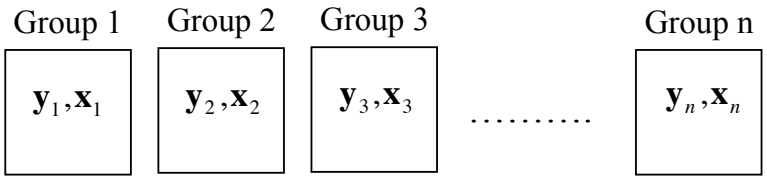
$$\left.\begin{aligned} \mu_i &= \alpha_{j[i]} + \beta x_i \\ y_i &\sim \text{normal}(\mu_i, \sigma) \end{aligned}\right\}$$ This requires one product over all obsevations.

is the same as

$$\left.\begin{aligned} \mu_{ij} &= \alpha_j + \beta x_{ij} \\ y_{ij} &\sim \text{normal}(\mu_{ij}, \sigma) \end{aligned}\right\}$$ This requires a product over i and over j.

Start Thursday lecture here

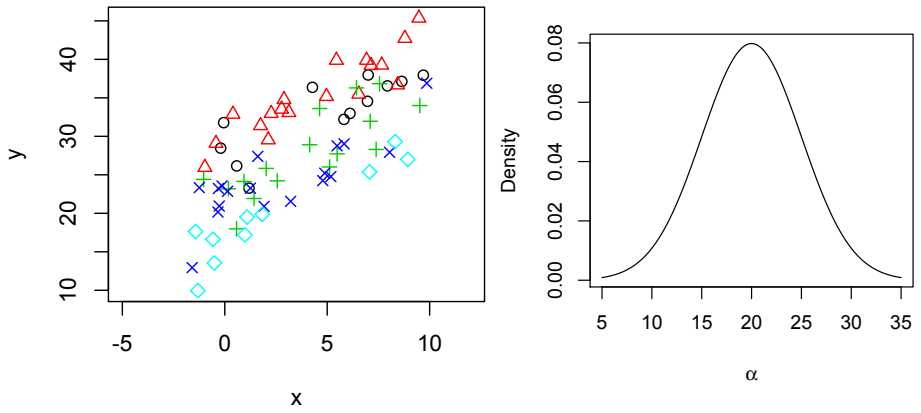
Multi-level regression models



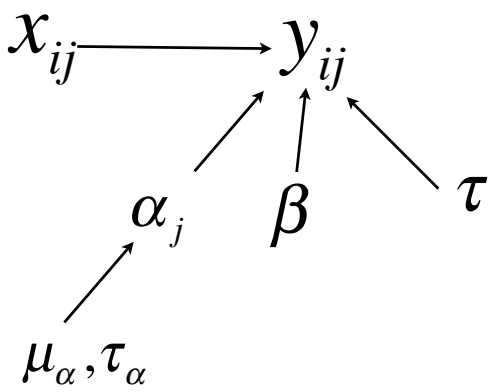
The **y**'s are vectors of responses and the **x**'s are vectors or matrices of predictors (covariates). We want to predict the ith response.

We could treat the α (and/or the β) as a *random effect* (aka group level effect):

Each group has its own α



We could treat the α (and/or the β) as a *random effect*:



The *[]* subscripting notation used by Gelman

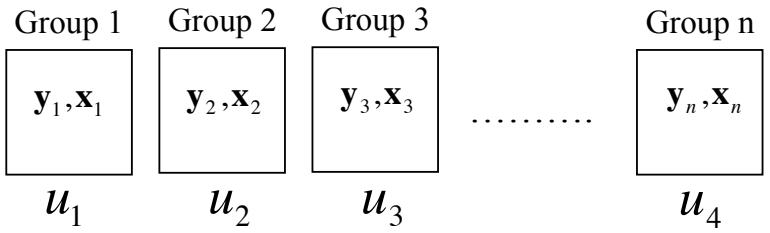
$$y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i$$

instead of

$$y_{ij} = \alpha_j + \beta x_{ij} + \epsilon_{ij}$$

Put notes aside

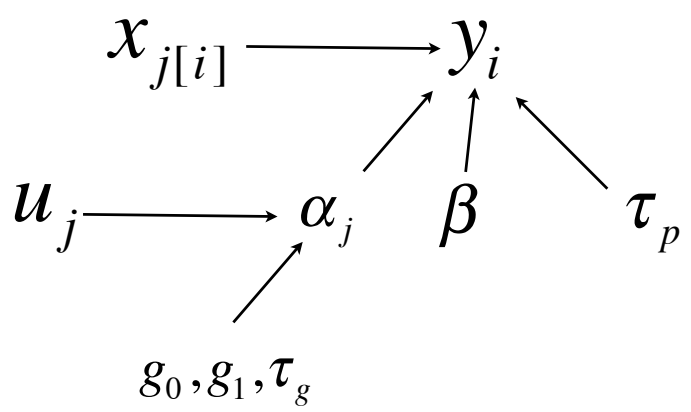
Multi-level regression models



The **y**'s are vectors of responses and the **x**'s are vectors or matrices of predictors (covariates). We want to predict the ith response. We also have a vector of group level measurements (**u**).

We model the intercepts at the group level *as a linear function of the data on the groups*. Diagram the network and write out the joint conditional distribution.

We could model the α (and/or the β) as a function of the group level data:



We could model the α (and/or the β) as a function of the group level data:

$$P(\alpha_j,\beta,g_0,g_1,\tau_p,\tau_g,|\,\mathbf{y},\mathbf{x},\mathbf{u}) \propto \prod_{i=1}^{n.y} \text{normal}(y_i\,|\,\alpha_{j[i]}+\beta x_i,\tau_p)\prod_{j=1}^{n.u}\text{normal}(\alpha_j\,|\,g_0+g_1u_j,\tau_\alpha)\times$$
$$\text{normal}(\beta|0,.000001)\text{normal}(g_0|0,.00001)\times\text{normal}(g_1|0,.00001)$$
$$\text{gamma}(\tau_p|.001,.001)\text{gamma}(\tau_g|.001,.001)$$

The data set-up

```
> 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
```

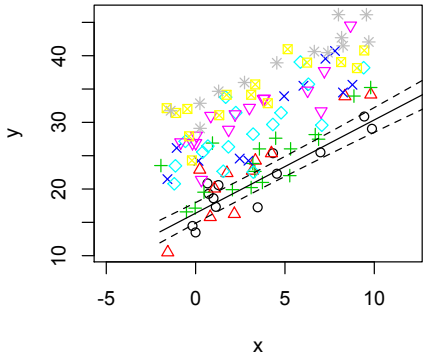
```
data=list(y=y[,4],
x=y[,3],
group=y[,1],
n.group=n.group,
u=u ,
x.hat=x.hat
)
```

```
model{
beta ~ dnorm(0,.000001)
tau.p ~ dgamma(.0001,.0001)
sigma <- 1/sqrt(tau.p)
g0 ~ dnorm(0,.00001)
g1 ~ dnorm(0,.00001)
tau.g ~ dgamma(.001,.001)
 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)
    }
```

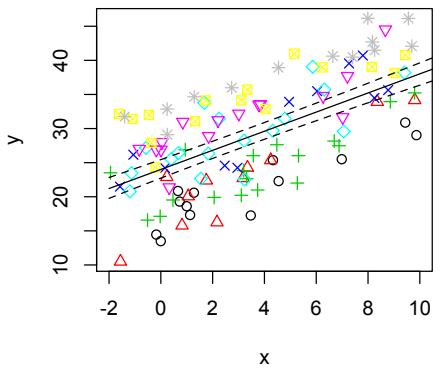
```
#predictions
  #across group
  for(i in 1:length(x.hat)){
    y.hat.all[i] <-
(g0+g1*mean(u)) +
beta*x.hat[i]
    }
  #by groups
  for(j in 1:n.group){
    for(i in 1:length(x.hat)){
    y.hat[i,j] <- alpha[j] +
beta*x.hat[i]
    }
  }
}# end of model
```

Estimates

For a single group



Across groups



Other group level effects

- We can use the same approach to modeling the slope (or any single term in a non-linear model) as a random effect.
- If we want to estimate group level effects for both slope and intercept (or two terms in a non-linear model) then we must consider their covariance. This can be done as outlined by Gelman and Hill 2007: Chapter 17

Random effects for slopes

The usual rule for applying random effects to slopes goes something like this. If the x is deliberately chosen, for example it is a level of an applied treatment, then we treat β as a fixed effect--we do not model it's distribution. If, however, the x is chosen randomly (i.e., a plot in a forest) then we treat β as a random effect.

Be aware that although the use of random effect is fairly consistent, fixed effect is used to mean many different things!

Bayesian regression

- Being able to express observations of all types as a linear function of predictors is an essential tool for ecological analysis.
- Linear models have application in traditional settings (regression, ANOVA) but they also can be components of highly mechanistic process models.
- The Bayesian approach, unlike likelihood, allows us to estimate the posterior distributions of calculated quantities of interest as functions of estimated quantities. This is very useful in estimating effect sizes, creating contrasts, and in making predictions with credible intervals.