## §13.1
Varying Intercepts & Slopes

$$
\begin{aligned}
y_i &\sim N(\alpha_{j[i]} + \beta_{j[i]} x_i, \sigma_y^2), \ i = 1, \ldots, n \\
\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} &\sim N\left( \begin{bmatrix} \mu_\alpha \\ \mu_\beta \end{bmatrix}, \begin{bmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{bmatrix} \right), \ j = 1, \ldots, J
\end{aligned}
$$

OR

$$\mathbf{y}_j = \mathbf{X}_j \boldsymbol{\beta} + \mathbf{Z}_j \mathbf{b}_j + \epsilon_j; , \epsilon_j \sim N(\mathbf{0}, \sigma^2 \mathbf{I}), \quad \text{indep of}$$

$$\mathbf{b}_j \sim iidN(\mathbf{0}, \Psi), \ j = 1, \ldots, J \quad \Psi = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix}$$

$\boldsymbol{\beta} = (\mu_\alpha \ \ \mu_\beta)^T$
$\text{Var}(\mathbf{y}_j) = \sigma^2 \mathbf{I} + \mathbf{Z}_j \Psi \mathbf{Z}_j^T$



## Radon Data

$x_i$ is 0 for basement, 1 for 1st floor.

$\mu_\alpha$ is ________________________________

$\mu_\beta$ is ________________________________

$\alpha_j$ is ________________________________

$\beta_j$ is ________________________________

How do elements of $\mathbf{b}_j$ relate to the above quantities?



## Extracting from lmer fit

```
radon.lmerfit <- lmer(log.radon ~ 1 + floor + (1 + floor |county), data = minn)
print(radon.lmerfit,corr=FALSE)

## Linear mixed model fit by REML ['lmerMod']
## Formula: log.radon ~ 1 + floor + (1 + floor | county)
##    Data: minn
## REML criterion at convergence: 2168
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  county   (Intercept) 0.349
##           floor       0.344    -0.34
##  Residual             0.746
## Number of obs: 919, groups:  county, 85
## Fixed Effects:
## (Intercept)        floor
##       1.463       -0.681
```



## Methods for lmer fit
Fixed and Random effects

```
fixef(radon.lmerfit)

## (Intercept)       floor
##       1.463      -0.681

head(ranef(radon.lmerfit)$county)

##                         (Intercept)    floor
## AITKIN                     -0.31824  0.14049
## ANOKA                      -0.52939 -0.08972
## BECKER                      0.00892  0.01222
## BELTRAMI                    0.07267 -0.07146
## BENTON                     -0.03573  0.06043
## BIG STONE                   0.01989 -0.00661

apply(ranef(radon.lmerfit)$county,2,sum)

## (Intercept)       floor
##    7.99e-14   -3.33e-14
```

# More methods for lmer fit

## Variance, Covariance, Correlation

```
vcov(radon.lmerfit)

## 2 x 2 Matrix of class "dpoMatrix"
##             (Intercept)    floor
## (Intercept)      0.0029 -0.00180
## floor           -0.0018  0.00767

VarCorr(radon.lmerfit)

##  Groups   Name        Std.Dev. Corr
##  county   (Intercept) 0.349
##           floor       0.344    -0.34
##  Residual             0.746
```



# Add Uranium

## Measured at county level

Now $\alpha_j = \gamma_0^\alpha + \gamma_1^\alpha u_j$ and $\beta_j = \gamma_0^\beta + \gamma_1^\beta u_j$
OR: $\mathbf{X}_j$ now has columns: $\mathbf{1}$, $\mathbf{x}_j$, $u_j\mathbf{1}$, and $u_j\mathbf{x}_j$ and $\boldsymbol{\beta}$ has intercept, overall floor effect, uranium effect, and an adjustment to uranium effect for first floor.

```
radon.fit2 <-  lmer(log.radon ~ 1 + floor*u + (1 + floor |county), data = minn
fixef(radon.fit2)

## (Intercept)       floor           u      floor:u
##       1.469      -0.671       0.808       -0.420

head(ranef(radon.fit2)$county)

##                         (Intercept)   floor
## AITKIN                     -0.00993  0.0241
## ANOKA                       0.02705 -0.2180
## BECKER                      0.00834  0.0243
## BELTRAMI                    0.05896  0.0720
## BENTON                      0.01063  0.0477
## BIG STONE                  -0.02049 -0.0208
```



# Plot Effects Code

`coef` adds fixed and random terms together.
How to get SE's?

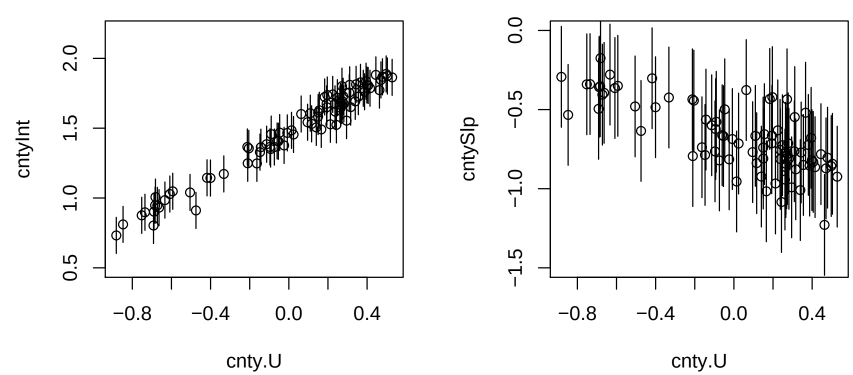
```
head(coef(radon.fit2)$county)

##                         (Intercept)  floor     u floor:u
## AITKIN                         1.46 -0.647 0.808   -0.42
## ANOKA                          1.50 -0.889 0.808   -0.42
## BECKER                         1.48 -0.647 0.808   -0.42
## BELTRAMI                       1.53 -0.599 0.808   -0.42
## BENTON                         1.48 -0.623 0.808   -0.42
## BIG STONE                      1.45 -0.692 0.808   -0.42

cnty.U <- with(minn, tapply(u, county, mean))
cntyInt <- coef(radon.fit2)$county[,1] + .808*cnty.U
cntySlp <- coef(radon.fit2)$county[,2] -0.4195*cnty.U
cntyIntSE <- sqrt(vcov(radon.fit2)[1,1] + VarCorr(radon.fit2)$county[1])
cntySlpSE <- sqrt(vcov(radon.fit2)[2,2] + VarCorr(radon.fit2)$county[4])
par(mfrow=c(1,2))
plot(cnty.U, cntyInt, ylim=c(.5,2.2))
segments(cnty.U,  cntyInt + cntyIntSE, cnty.U,  cntyInt - cntyIntSE)
plot(cnty.U, cntySlp, ylim=c(-1.5,0))
segments(cnty.U,  cntySlp + cntySlpSE, cnty.U,  cntySlp - cntySlpSE)
```



# Plots




They say they use posterior medians and SD's.

## Typo p 284

I'll skip over models with varying slope, only one intercept, but there is a typo in equation (13.5):

$$y_i \sim N(\alpha + \beta x_i + \theta_{1,j[i]} T_i + \theta_{2,j[i]} x_i T_i, \sigma^2)$$



## §13.3 Inverse-$\chi^2$ distribution

Start with a single measurement, $y_i \sim iid\ N(\mu, \sigma^2),\ i = 1, \dots, n$
Standardize and sum to get a $\chi^2$:
$\sum (y_i - \mu)^2/\sigma^2 \sim \chi^2_n$ and
$\frac{s^2}{\sigma^2} \sim \chi^2_{n-1}$
Bayesians consider $s^2$ fixed by the data, and focus on $\sigma^2$. Recall: $\chi^2$ distributions are part of the $\Gamma$ family.
A conjugate family:
Assume we know $\mu$ and are interested in $\sigma^2$. If we take inverse-gamma prior $p(\sigma^2) \propto (\sigma^2)^{\alpha+1} e^{-\beta/\sigma^2}$ or $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$, then the posterior is

$$\sigma^2 | y \sim \text{Inv-}\chi^2 \left( \nu_0 + n, \frac{\nu_0 + n\nu}{\nu_0 + n} \right);\ \nu = \frac{1}{n} \sum (y_i - \mu)^2$$

In a linear model with iid normal errors and vague prior $(\alpha\ \beta)^T = \mathbf{0}$, then

$$\sigma^2 | \mathbf{y} \sim \text{Inv-}\chi^2(n - r, s^2)$$



## §13.3 Inverse-Wishart distribution

If we start with a multivariate normal:

$$\mathbf{y}_i \sim iid\ N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),\ i = 1, \dots, n$$

then our sum of squares matrix (n by n) is

$$\mathbf{S} = \sum_i (\mathbf{y}_i - \widehat{\mathbf{y}}_i)(\mathbf{y}_i - \widehat{\mathbf{y}}_i)^T$$

Start with prior: $\boldsymbol{\Sigma} \sim \text{Inv-Wishart}_{\nu_0}$ and $\boldsymbol{\mu} | \boldsymbol{\Sigma} \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}/\kappa_0)$ and we get posterior:
$\boldsymbol{\Sigma} | y \sim \text{Inv-Wishart}_{\nu_n}(\boldsymbol{\Lambda}_n^{-1})$ and
$\boldsymbol{\mu} | \mathbf{y}, \boldsymbol{\Sigma} \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}/\kappa_n)$
where $\boldsymbol{\mu}_n = \frac{\kappa_0}{\kappa_0 + n} \boldsymbol{\mu}_0 + \frac{n}{\kappa_0 + n} \bar{\mathbf{y}}$, $\kappa_n = \kappa_0 + n$, $\nu_n = \nu_0 + n$, and
$\boldsymbol{\Lambda}_n = \boldsymbol{\Lambda}_0 + \mathbf{S} + \frac{\kappa_0 n}{\kappa_0 + n} (\bar{\mathbf{y}} - \boldsymbol{\mu}_0)(\bar{\mathbf{y}} - \boldsymbol{\mu}_0)^T$.



## Inverse-Wishart for random vectors

When we have random intercept and slope, $\mathbf{b}_j = \begin{pmatrix} b_{0j} \\ b_{1j} \end{pmatrix}$, we need to use Inv-Wishart prior and posterior for its variance-covariance matrix.

## §13.4 Correlations in $\mathbf{b}_j$

If $x$ is not centered, then changes in slope and intercept are like a teeter-totter, typically negatively correlated, so we should center. After centering, intercept is mean($y$) at mean($x$). Positive correlation in, say growth curves, means those with larger mean are also growing faster.



## §13.5 Non-nested Random Effects

In the skater/judges data, pair 1 is the same for each judge, and judge A is the same for each skater, so the effects are crossed.

```
skate.fit <- lmer(tech ~ 1 + jmatch + (1+pair) + (1+judge), data = skaters)
```

In the dogs example (`Pixel`), side was nested within dog, there was no real meaning to "left-side" or "right-side" in general, sides within dog just tend to vary a bit.

```
dogFit <- lmer( pixel ~ 1 + day + day^2 + (1|Dog/Side), data=Pixel)
```



## §13.6 Model Selection

The excuse for the detour into class notes

We saw many methods for selecting variables to be in the model.

G& H suggest we can use multilevel models to get round this decision, for example reducing 87 coefficients down to 36 with extra random variation at the food level.

For an elections example, they standardize each of 5 predictors, then take a weighted average (why is there a double sum in (13.13)?) where the weights are $N(1, \sigma_\gamma^2)$ (times 1/5). If $\sigma_\gamma = 0$, it's a simple average, if $\sigma_\gamma \to \infty$, then estimate each individually, and in between we do partial pooling.



## Factor Analysis and §13.7

A topic for multivariate class.
Assume we have $p$ possible measurements on each of $n$ subjects, but each of the measurements is determined by $q$ ($q < p$) unobserved latent variables (factors). We try to estimate "loadings": linear combinations of the $p$ variables which describe the unknown factors.

More complex models p 297.

- Varying variances
- Interactions between fixed and random are random
- Multivariate response (skaters)
- Correlated data in time & space
- networks