36-617: Applied Linear Regression

Introduction to Multi-level Models, I

Brian Junker
132E Baker Hall
brian@stat.cmu.edu

Announcements

- Homework & Quizzes
  - HW08 due tonight
  - HW09 out; due next Weds
- Projects:
  - I am grading Project 01
  - Project 02 out soon (last project for the class)
- Reading (Sheather):
  - Please read all of 10.1 for this week (but not 10.2)
  - Monday’s quiz will be on this.

Outline

- Introduction, Terminology, Multi-level Models
- London Schools Data
  - Plotting clusters (groups, clumps, …)
- Minnesota Radon Data
- The Random-Intercept Model
- Different ways to write the model:
  - Mixed Effects, Variance Components, Multilevel Model
- Modeling the intercept as a function of a group-level covariate

Introduction

- *Most common*: linear regression and generalized linear regression (logistic regression) models
- *Next most common*: hierarchical and multilevel models (hierarchical linear models are a special case!)
- Situations…
  - Clustered sampling
  - Grouped experimental trials
    - multicenter clinical trials in medicine
    - group-randomized trials in education
  - Growth curves and random coefficient models

A Note on Terminology

All of the following refer to approximately the same class of models:

- These models emphasize connections with linear regression and generalized least squares (GLS):
  - Mixed Models
  - Mixed Effects Models
  - Variance Components Models
- These models emphasize the data generation process (& they are almost Bayesian):
  - Multilevel Models
  - Hierarchical Linear Models

Multilevel Models…

- Useful when information comes to us in clumps of observations that are more like each other within a clump than between clumps
  - Classrooms within schools or schools within a city
  - States or geographic areas within a nation
  - Election precincts within a larger election
  - Answers given by the same student on a test
- Useful when a different linear regression should be fitted within each clump, but there is not enough information to separately estimate all clumps
  - Deducing state opinions from a national opinion survey
  - Fitting separate regressions to rank schools in London – some schools are represented by only 1 or 2 students!

More on Multilevel Models…

- Traditional linear regression can either
  - Ignore the clumps completely and fit a single model to all the data
  - Treat each clump completely separately but fail to share information across clumps when some clumps "need help"
  - *Both of these are examples of "Fixed Effects"*
- Multilevel models allow
  - treating clumps separately, ***and***
  - sharing information across clumps to make better estimates
  - *These are examples of "Random Effects"*
- Most MLM's have both fixed and random effects – "Mixed Effects" models

Example: The London Schools Data

Goldstein et al. (1993) present an analysis of examination results from inner London schools. They use hierarchical or multilevel models to study the between-school variation, and calculate school-level residuals in an attempt to differentiate between “good” and “bad” schools.

The variables are described on the next slide

Example: The London Schools Data

Y = end-of-year exam scores for each pupil (1..1978)

school = school each pupil is in (1..38)

LRT = London reading test score

VR.1 = 1 for highest verbal-reasoning pupils, else 0

VR.2 = 1 for medium verbal-reasoning pupils, else 0

Gender : 0 = female, 1 = male (I think!)

School.gender.1 = 1 for all-girl schools

School.gender.2 = 1 for all-boy schools

School.denom.1 = 1 for Roman Catholic schools, 0 else

School.denom.2 = 1 for Church of England schools, 0 else

- The LRT and VR assessments are made at the beginning of the year.
- Goldstein's goal was to rank the schools in some way

Thinking About London Schools…

- Consider the three models; Are any of them useful for ranking?

```
mean.lm <- lm(Y ~ school - 1, data=school.frame)
adj.1.lm <- lm(Y ~ school + LRT - 1, data=school.frame)
adj.2.lm <- lm(Y ~ school*LRT - 1 - LRT, data=school.frame)
```

- Easy to compare with F-test, but how can we understand what they say?

```
anova(mean.lm,adj.1.lm,adj.2.lm)
Analysis of Variance Table

Model 1: Y ~ school - 1
Model 2: Y ~ school + LRT - 1
Model 3: Y ~ school * LRT - 1 - LRT
  Res.Df    RSS Df Sum of Sq        F    Pr(>F)    
1   1940 1801.8                                    
2   1939 1218.9  1    582.91 940.8659 < 2.2e-16 ***
3   1906 1180.9 33     38.03   1.8602  0.002189 ** 
```

How could we rank London Schools?

- We can illustrate, with ggplot facet graphs, how these (and similar models) are representing the relationship between
  - Y (end of year score)
  - LRT (beginning of year score)

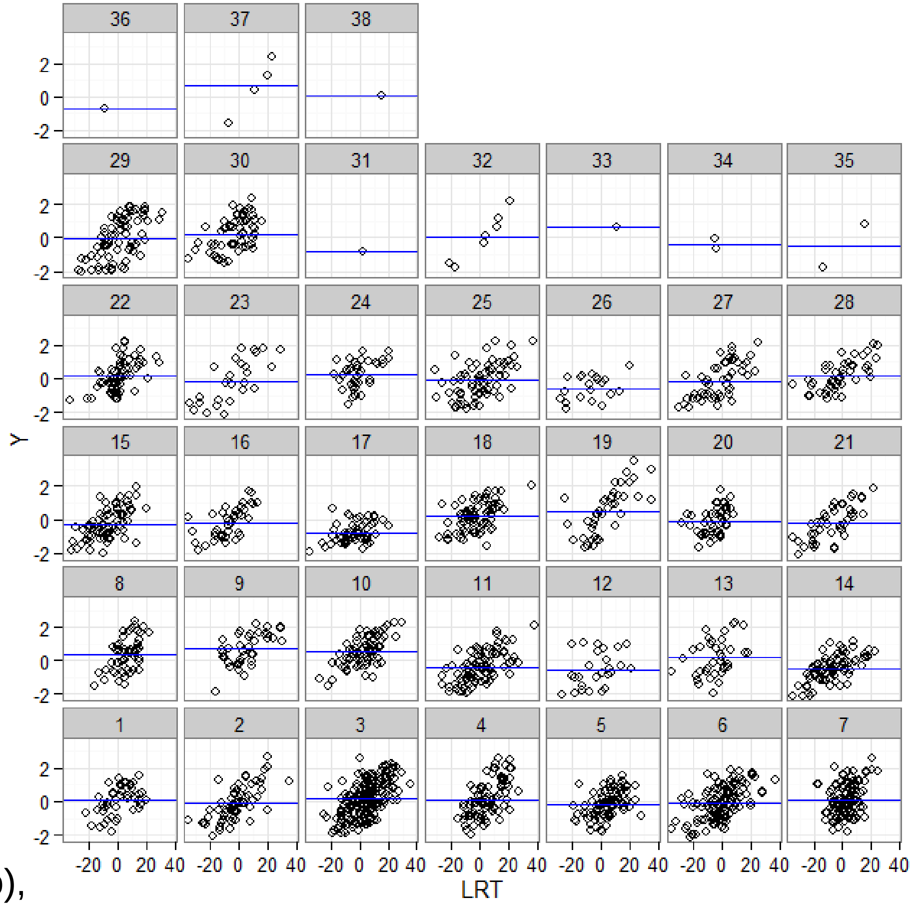
  in the data
- ggplot facet graphs are very useful for this!
  - Code in
    20 - ggplot-for-grouped-clustered-data-london.r

London Schools: Ignore LRT and only look at mean(y) in each school

```
g <- ggplot(school.frame,
  aes(x=LRT,y=Y)) +
  facet_wrap( ~ school,
  as.table=F) +
  geom_point(pch=1)

coef <- lm(Y ~ school - 1,
data = school.frame)$coef
slo <- int <- rep(NA,J)
for (j in 1:38) {
    int[j] <- coef[j]
    slo[j] <- 0}
par <- ddply(school.frame,
"school", summarize,
 int <- int[school[1]],
 slo <- slo[school[1]])
names(par) <-
c("school","int","slo")

g + geom_abline(data=par,
aes(intercept=int,slope=slo),
color="blue")
```



We would rank schools by mean(y) in this case. This ignores the status of students at the beginning of the school year.

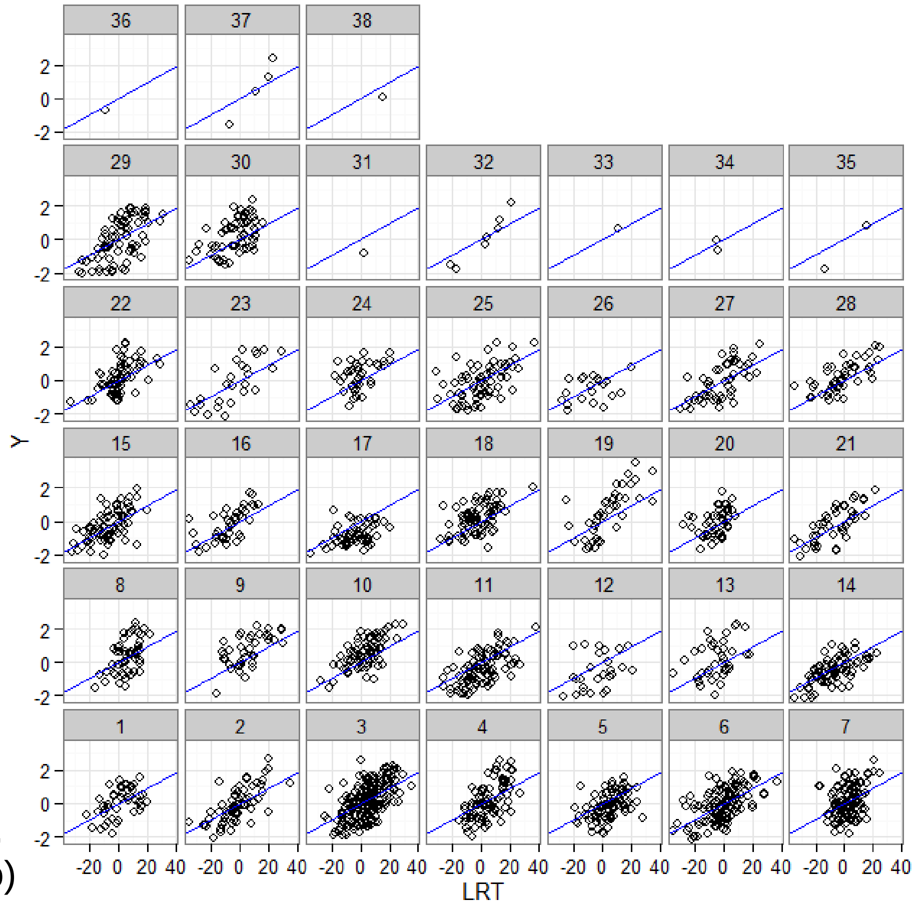
(This code requires library(ggplot2) and library(plyr))

London Schools: Ignore Schools and fit a single linear regression Y ~ LRT

```
g <- ggplot(school.frame,
  aes(x=LRT,y=Y)) +
  facet_wrap( ~ school,
  as.table=F) +
  geom_point(pch=1)

coef <- lm(Y ~ LRT,
data = school.frame)$coef
slo <- int <- rep(NA,J)
for (j in 1:38) {
    int[j] <- coef[1]
    slo[j] <- coef[2]}
par <- ddply(school.frame,
"school", summarize,
 int <- int[school[1]],
 slo <- slo[school[1]])
names(par) <-
c("school","int","slo")

g + geom_abline(data=par,
aes(intercept=int,slope=slo)
, color="blue")
```



We really don't have anything to rank schools with here…

Everyone has the same slope and the same intercept.

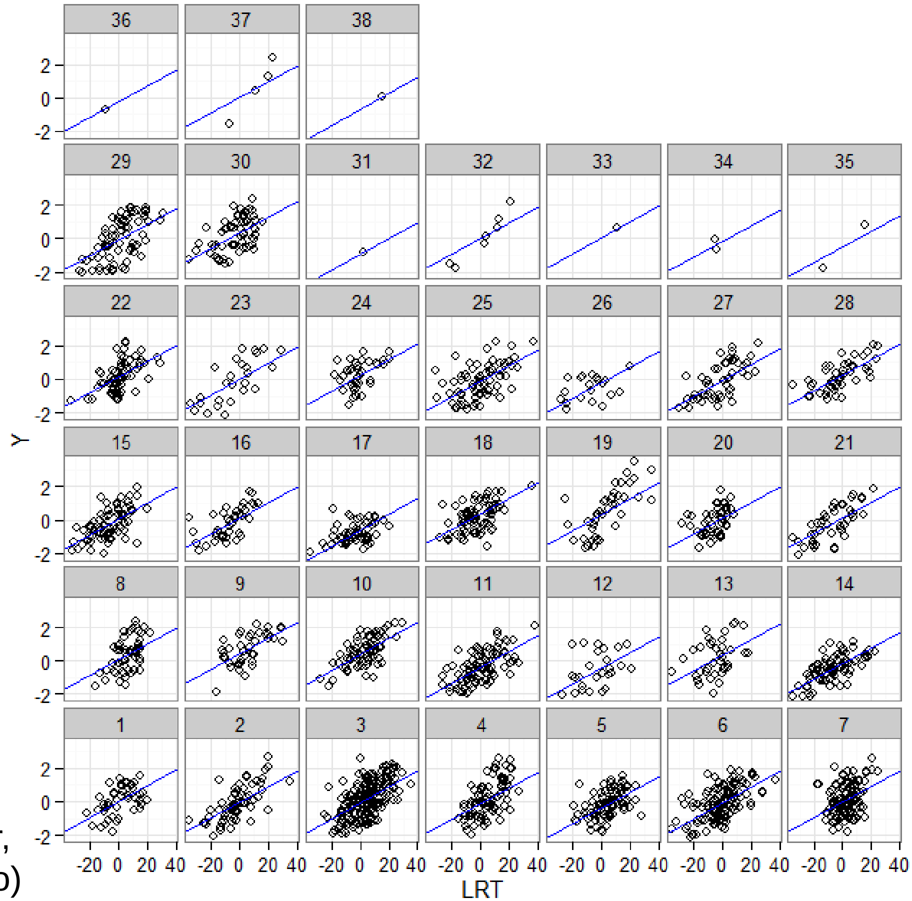
(This code requires library(ggplot2) and library(plyr))

London Schools: Use same slope on LRT for all schools, different intercepts

```
g <- ggplot(school.frame,
  aes(x=LRT,y=Y)) +
  facet_wrap( ~ school,
  as.table=F) +
  geom_point(pch=1)

coef <- lm(
Y ~ school+LRT-1,
data = school.frame)$coef
slo <- int <- rep(NA,J)
for (j in 1:38) {
    int[j] <- coef[j]
    slo[j] <- coef[39]}
par <- ddply(school.frame,
"school", summarize,
 int <- int[school[1]],
 slo <- slo[school[1]])
names(par) <-
c("school","int","slo")

g + geom_abline(data=par,
aes(intercept=int,slope=slo)
, color="blue")
```



We could rank schools based on their intercepts.

However, the model clearly fits some schools better than others!

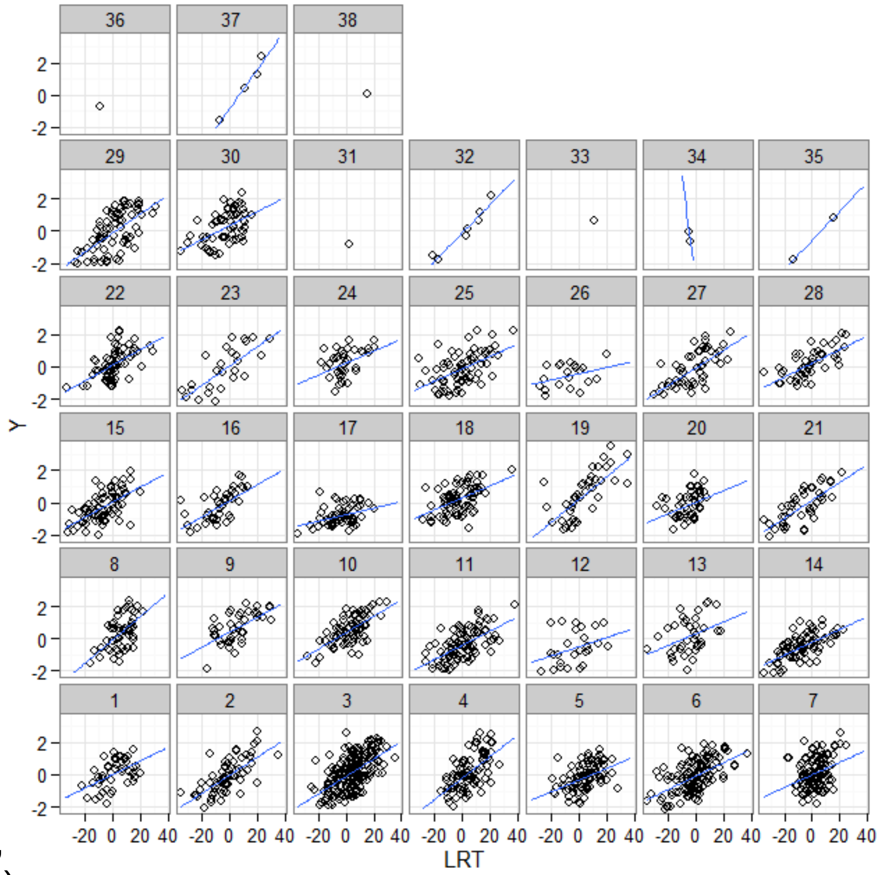
(This code requires library(ggplot2) and library(plyr))

London Schools: Different slope and intercept for each school

```
g <- ggplot(school.frame,
  aes(x=LRT,y=Y)) +
  facet_wrap( ~ school,
  as.table=F) +
  geom_point(pch=1)

coef <- lm(
Y ~ school*LRT-1-LRT,
data = school.frame)$coef
slo <- int <- rep(NA,J)
for (j in 1:38) {
    int[j] <- coef[j]
    slo[j] <- coef[j+38]}
par <- ddply(school.frame,
"school", summarize,
 int <- int[school[1]],
 slo <- slo[school[1]])
names(par) <-
c("school","int","slo")

g + geom_abline(data=par,
aes(intercept=int,slope=slo),
color="blue")
```



Now, we let slopes and intercepts vary from school to school, to get the best fit. We would still like to rank based on intercepts.

However some schools have crazy regressions or cannot be fitted (too small a sample in that school!)

This is a problem with fixed effects models, and it is something MLM's are good at fixing!

(This code requires library(ggplot2) and library(plyr))

Example: Radon Levels in Minnesota

- Each individual unit in the data set is a house

  Individual-level (house-level) variables:
  - radon, log(radon)
  - floor = 0 if measurement was made in basement;
    = 1 if measurement on first floor
- Houses are grouped into counties

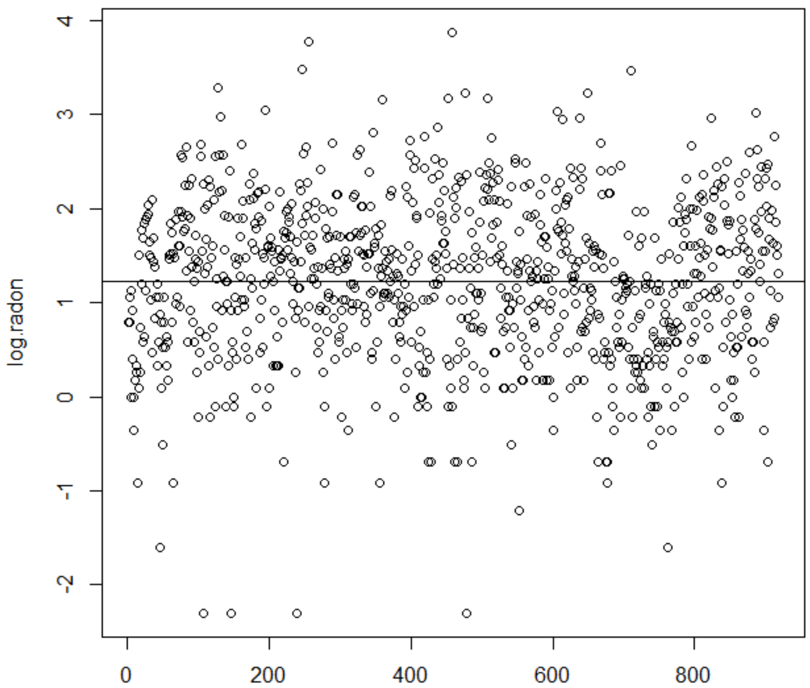
  Group-level (county-level) variables:
  - county.name & county number
  - uranium & log(uranium) – measurement of uranium in the soil in each county
- We want to predict radon levels from the other variables

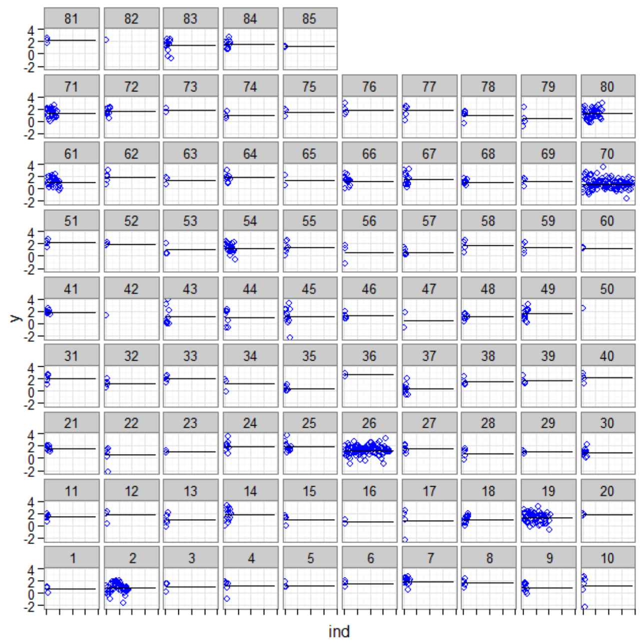
Many ways to view this data

For example…

1. ***Pooled regression***: examine radon as a function of uranium [ignoring county]
2. ***Unpooled, means (intercepts) only***: look at radon levels within each county [ignoring uranium]
3. ***Hierarchical "simple" regression***: Take model #2 and build a second regression predicting mean level of radon in each county from uranium levels in that county.
4. ***Unpooled regression***: examining radon ~ floor within each county

Totally pooled (#1) vs totally unpooled(#2) log(radon) intercept-only models





$$y_i = \alpha_0 + \epsilon_i$$

$i =$ house,

no attention paid to county

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

$i =$ house,

$j[i] =$ county that house i is in

Looking at the coefficients from fitting separate (unpooled) models

```
> cties <- as.factor(county)
> contrasts(cties) <- contr.sum(85)
> summary(lm.0 <- lm(y ~ 1))
              Est     SE      t value Pr(>|t|)
(Intercept)  1.22     0.03    43.51   <2e-16 ***

> summary(lm.unpooled.contrast.from.grand.mean
+   <- lm(y ~ cties))
              Est     SE     t value Pr(>|t|)
(Intercept)  1.34     0.04   32.01  < 2e-16 ***
cties1      -0.68     0.40   -1.72  0.09 .
cties2      -0.51     0.12   -4.36  1.49e-05 ***
cties3      -0.30     0.46   -0.65  0.52
cties4      -0.20     0.30   -0.67  0.50
cties5      -0.09     0.39   -0.23  0.82
cties6       0.17     0.46    0.37  0.71
cties7       0.57     0.21    2.63  0.01 **
cties8       0.29     0.40    0.72  0.47
cties9      -0.41     0.25   -1.63  0.10
cties10     -0.14     0.32   -0.43  0.67
cties11      0.06     0.36    0.16  0.87
cties12      0.39     0.40    0.98  0.33
cties13     -0.30     0.32   -0.94  0.35
cties14      0.44     0.21    2.04  0.04 *
cties15     -0.37     0.40   -0.92  0.36
cties16     -0.68     0.56   -1.21  0.23
   .           .        .       .     .
   .           .        .       .     .
   .           .        .       .     .
cties68     -0.25     0.28   -0.90  0.37
cties69     -0.10     0.40   -0.26  0.80
cties70     -0.58     0.08   -6.82  1.80e-11 ***
cties71      0.03     0.16    0.20  0.84
cties72      0.24     0.25    0.93  0.35
cties73      0.45     0.56    0.80  0.42
cties74     -0.36     0.40   -0.90  0.37
cties75      0.14     0.46    0.31  0.76
cties76      0.48     0.40    1.22  0.22
cties77      0.38     0.30    1.25  0.21
cties78     -0.35     0.36   -0.97  0.33
cties79     -0.91     0.40   -2.29  0.02 *
cties80     -0.09     0.12   -0.75  0.45
cties81      0.89     0.46    1.94  0.05 .
cties82      0.89     0.79    1.12  0.26
cties83      0.11     0.22    0.51  0.61
cties84      0.25     0.22    1.11  0.27
```

Problems with totally-pooled vs totally-unpooled

- Totally-pooled: It looks like there is some pattern to the county means, so this “over-smooths” (forces all the counties to be the same)
- Totally-unpooled: Although the counties have some variation in means, there may not be very much!

Having different means is better than totally-pooled model…

,…but very few county means are different from overall mean!

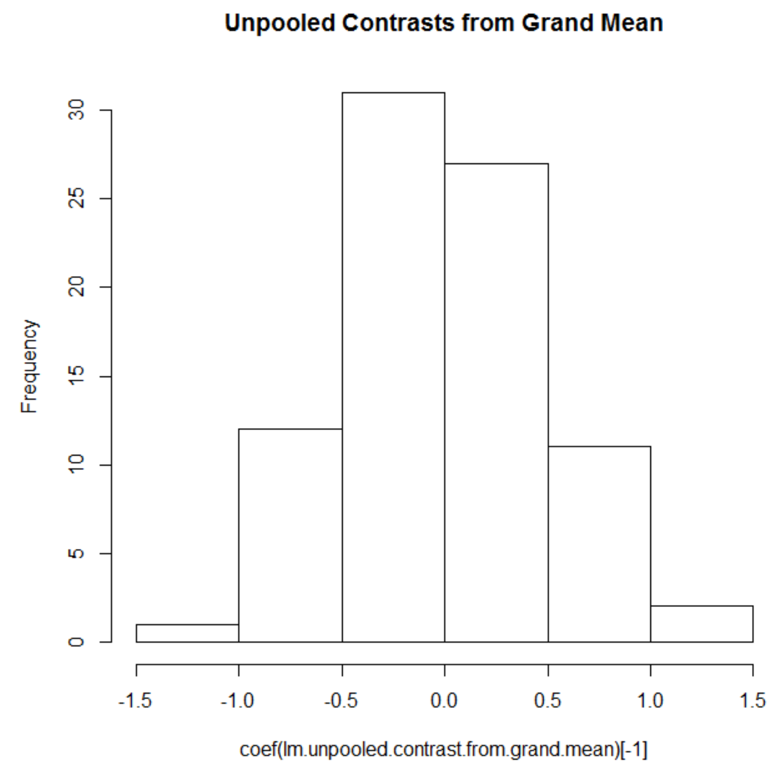
```
cties <- as.factor(county)
contrasts(cties) <- contr.sum(85)
lm.unpooled.contrast.from.grand.mean <- lm(y ~ cties)
anova(lm.unpooled.contrast.from.grand.mean)
#            Df Sum Sq Mean Sq F value    Pr(>F)
# cties      84 136.89 1.62960  2.5567 1.736e-11 ***
# Residuals 834 531.57 0.63738
```

```
length(unique(county))
# [1] 85
sum(coef(summary(lm.unpooled.con-
    trast.from.grand.mean))[,4]<0.05)
# [1] 15
15/85
# [1] 0.1764706
```

Some Equations…

```
> hist(coef(lm.unpooled.contrast.from.grand.mean)[-1],
+       main="Unpooled Contrasts from Grand Mean")
```

- The coefficients are nearly normally distributed!
- Suggests that we modify our usual regression model…

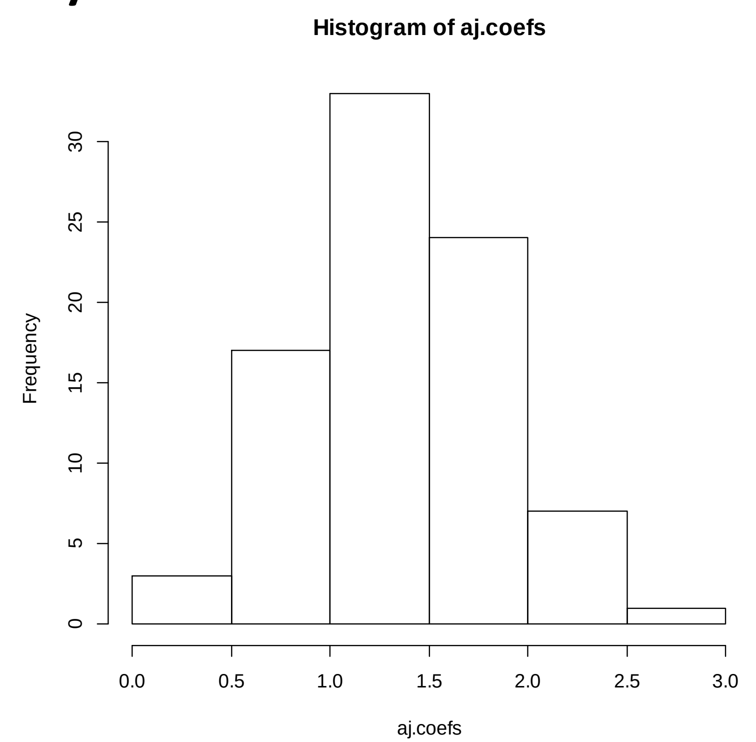


A compromise between totally-pooled and totally-unpooled

- The 85 county means look rather "normal", so why not model them that way?

$$
\begin{aligned}
y_i &= \alpha_{j[i]} + \epsilon_i,\ \ \epsilon_i \overset{iid}{\sim} N(0,\sigma^2) \\
\alpha_j &= \beta_0 + \eta_j,\ \ \eta_j \overset{iid}{\sim} N(0,\tau^2)
\end{aligned}
$$

- Sometimes called a "***random intercept***" model



Fitting the random-intercept model

$$y_i = \alpha_{j[i]} + \epsilon_i, \;\; \epsilon_i \overset{iid}{\sim} N(0, \sigma^2) \qquad (1)$$

$$\alpha_j = \beta_0 + \eta_j, \;\; \eta_j \overset{iid}{\sim} N(0, \tau^2) \qquad (2)$$

Multilevel model (both equations 1 and 2)

```
library(lme4)
lmer.intercept.only <-
   lmer( y ~ 1 + ( 1 | county.name ) )
summary(lmer.intercept.only)
# Random effects:
#  Groups      Name          Var    SD
#  county.name (Intercept)   0.096  0.310
#  Residual                  0.637  0.798
# Numb. of obs: 919, grps: county.name, 85
#
# Fixed effects:
#              Estimate    SE    t value
# (Intercept)     1.31    0.05    26.84
```

$\hat{\sigma}^2$ $\hat{\tau}^2$

$\hat{\beta}_0$

Unpooled fixed effects (equation 1 only)

```
cties <- as.factor(county)
contrasts(cties) <- contr.sum(85)
lm.unpooled.contrast.from.grand.mean <-
   lm(y ~ cties)
summary(lm.unpooled.contrast.from.grand.mean)
# Coefficients:
#                  Est     SE       t Pr(>|t|)
# (Intercept)     1.34    0.04   32.01  < 2e-16 ***
# cties1         -0.68    0.40   -1.72 0.085374 .
# cties2         -0.51    0.11   -4.36 1.49e-05 ***
# cties3         -0.30    0.46   -0.65 0.518720
# […]
# Residual std err:  0.7984 on 834 df
```

Random-intercept model: Where are the intercepts?

```
> summary(lmer.intercept.only)
Random effects:
 Groups      Name        Variance Std.Dev
 county.name (Intercept) 0.095813 0.30954
 Residual                0.636621 0.79789
Numb. of obs: 919, grps: county.name, 85

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.31257    0.04891   26.84
> fixef(lmer.intercept.only)
(Intercept)
   1.312574
> ranef(lmer.intercept.only)
$county.name
                 (Intercept)
AITKIN          -0.245071104
ANOKA           -0.425038053
BECKER          -0.082191868
BELTRAMI        -0.088030506
BENTON          -0.022598796
BIG STONE        0.062346490
BLUE EARTH       0.404629013
[….]
```

```
> summary(lm.unpooled.con-
   trast.from.grand.mean)

Call:
lm(formula = y ~ cties)


Coefficients:
             Estimate Std. Error t value
    Pr(>|t|)
(Intercept)  1.343638   0.041980  32.006
cties1      -0.683231   0.396682  -1.722
cties2      -0.510388   0.117180  -4.356
cties3      -0.295300   0.457408  -0.646
cties4      -0.202652   0.301120  -0.673
cties5      -0.091202   0.396682  -0.230
cties6       0.169372   0.457408   0.370
cties7       0.565589   0.214984   2.631
[…]
```

Random effects – draws from $N(0,\tau^2)$

Fixed effects – estimates of regression coefficients

Different ways to write the random-intercepts model

- Multi-level Model (emphasize regression)

$$\begin{aligned} y_i &= \alpha_{j[i]} + \epsilon_i, \ \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2) \\ \alpha_j &= \beta_0 + \eta_j, \ \eta_j \stackrel{iid}{\sim} N(0, \tau^2) \end{aligned}$$

- Variance Components Model (substitute for α_j)

$$\begin{aligned} y_i &= \beta_0 + \eta_{j[i]} + \epsilon_i, \qquad \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2) \\ & \qquad \qquad \qquad \qquad \eta_j \stackrel{iid}{\sim} N(0, \tau^2) \end{aligned}$$

- Hierarchical Model (emphasize distributions)

$$\begin{aligned} \text{Level 2:} \quad \alpha_j &\stackrel{iid}{\sim} N(\beta_0, \tau^2) \\ \text{Level 1:} \quad y_i &\stackrel{indep}{\sim} N(\alpha_{j[i]}, \sigma^2) \end{aligned}$$

Multi-level Model (a.k.a. Hierarchical Linear Model)

- Emphasize Regression Structure

$$
\begin{aligned}
y_i &= \alpha_{j[i]} + \epsilon_i,\ \ \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2) \\
\alpha_j &= \beta_0 + \eta_j,\ \ \eta_j \stackrel{iid}{\sim} N(0, \tau^2)
\end{aligned}
$$

- Easy to use intuitions from lm() at each “level” of the model, to build and evaluate models

Variance Components Model

- Emphasize Error Structure

$$y_i = \beta_0 + \eta_{j[i]} + \epsilon_i, \qquad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

$$\eta_j \overset{iid}{\sim} N(0, \tau^2)$$

- Errors from different sources
  - η_j from groups/counties (j); Var(county level) = τ^2
  - ε_i from individual houses (i); Var(arbitrary house) = $\tau^2 + \sigma^2$
  - If $j[i] \neq j[i']$: $\mathrm{Cov}(y_i, y_{i'}) = 0$;
  - If $j[i] = j[i']$: $\mathrm{Cov}(y_i, y_{i'}) = \tau^2$, $\mathrm{Cor}(y_i, y_{i'}) = \tau^2/(\tau^2 + \sigma^2)$

Intra-class correlation (ICC)

- $\mathrm{Var}(\bar{y}_j) = \mathrm{Var}(\beta_0 + \eta_j + \frac{1}{n_j} \sum_{\text{all } i \in \text{ county } j} \epsilon_i) = \tau^2 + \sigma^2/n_j$;
  - The average is a ***reliable*** measure of county levels if σ^2/n_j is much: smaller than τ^2:

$$\frac{\text{Var (county level)}}{\text{Var (average of houses in county)}} = \frac{\tau^2}{\tau^2 + \sigma^2/n_j}$$

reliability

Hierarchical Bayes Model

- Emphasize Distribution Structure

$$\text{Level 2:} \quad \alpha_j \overset{iid}{\sim} N(\beta_0, \tau^2)$$

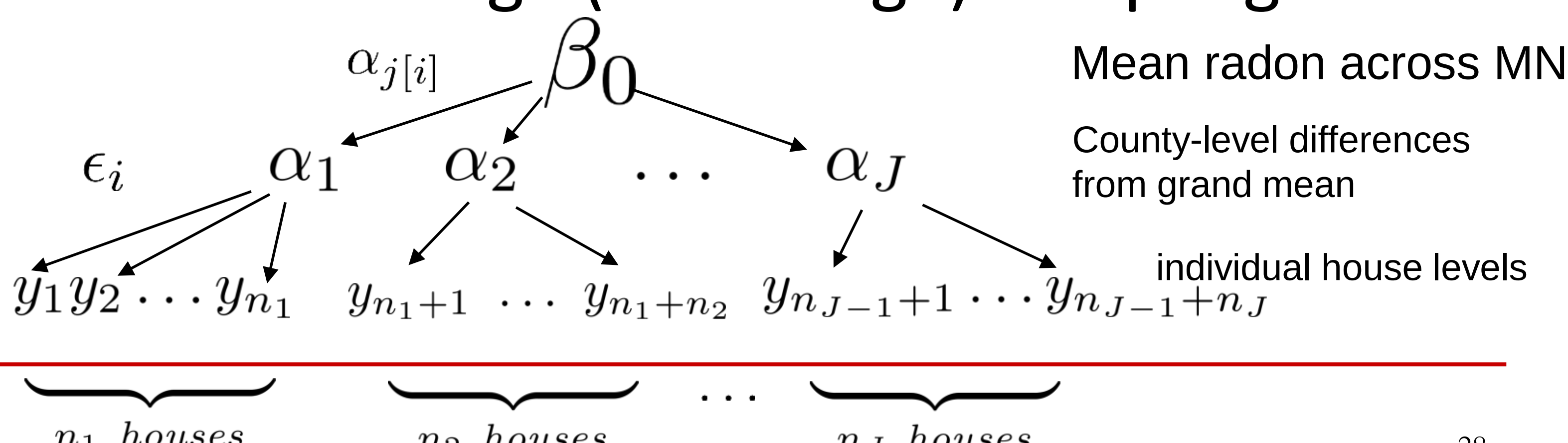
$$\text{Level 1:} \quad y_i \overset{indep}{\sim} N(\alpha_{j[i]}, \sigma^2)$$

- Emphasize Bayesian point of view

$$\text{Prior:} \quad \alpha_j \overset{iid}{\sim} N(\beta_0, \tau^2)$$

$$\text{Likelihood:} \quad y_i \overset{indep}{\sim} N(\alpha_{j[i]}, \sigma^2)$$

- Emphasize two-stage (multistage) sampling

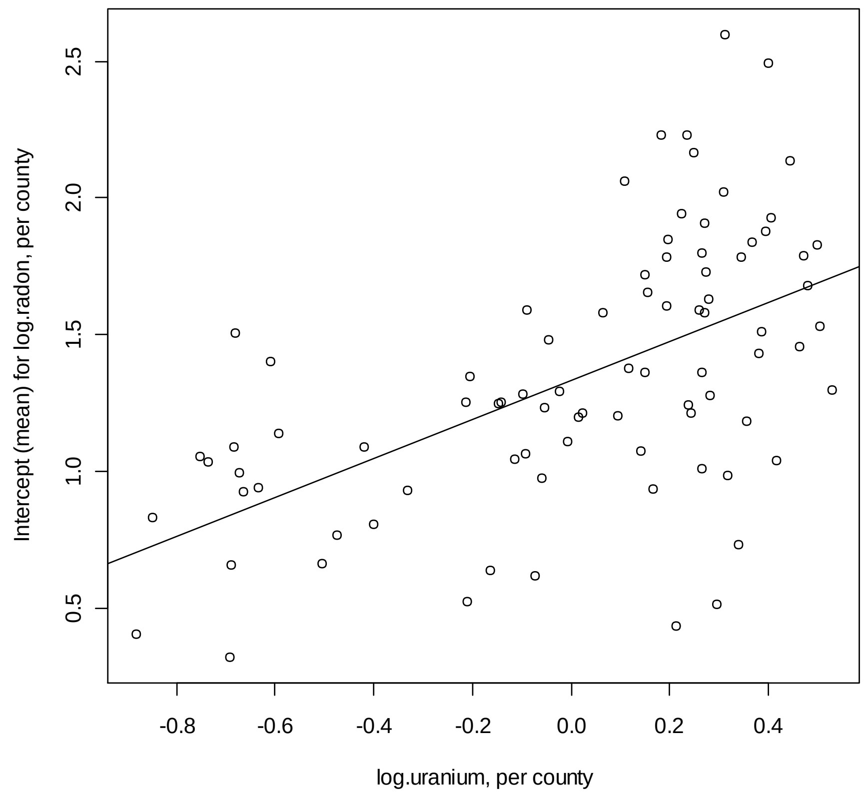


Back to the Radon Example: Plot county means vs log(uranium)…

```
aj.coefs <- NULL
for (cty in
   sort(unique(county))) {
  aj.coefs <- c(aj.coefs,
  coef(lm(y ~ 1,
   subset=(county==cty))))
}

summary(higher.regression <-
   lm(aj.coefs ~ u))

plot(aj.coefs ~ u,
   xlab="log.uranium, per
   county", ylab="Intercept
   (mean) for log.radon, per
   county")
abline(higher.regression)
```



Suggests ways to elaborate the hierarchical linear model…

- Instead of

$$\begin{aligned} y_i &= \alpha_{j[i]} + \epsilon_i,\ \ \epsilon_i \overset{iid}{\sim} N(0,\sigma^2) \\ \alpha_j &= \beta_0 + \eta_j, \eta_j \overset{iid}{\sim} N(0,\tau^2) \end{aligned}$$

we could try to fit

$$\begin{aligned} y_i &= \alpha_{j[i]} + \epsilon_i,\ \ \epsilon_j \overset{iid}{\sim} N(0,\sigma^2) \\ \alpha_j &= \beta_0 + \beta_1 u_j + \eta_j, \eta_j \overset{iid}{\sim} N(0,\tau^2) \end{aligned}$$

$\boxed{U_j = \log(\text{uranium}_j)}$

Fitting this model to the radon data...

```
> summary(lmer.intercepts.depend.on.log.ur-
    anium)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + log.uranium +
    (1 | county.name)

REML criterion at convergence: 2219.794

Random effects:
 Groups      Name        Variance Std.Dev.
 county.name (Intercept) 0.01406  0.1186
 Residual                0.64037  0.8002
Number of obs: 919, groups: county.name, 85

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.33305    0.03397   39.24
log.uranium  0.71912    0.08777    8.19

Correlation of Fixed Effects:
            (Intr)
log.uranium 0.197
```

$\tau^2 = \mathrm{Var}(\eta_j)$

$\sigma^2 = \mathrm{Var}(\varepsilon_i)$

β_0

β_1

```
> fixef(lmer.intercepts.depend.on.log.ur-
    anium)
(Intercept) log.uranium
  1.3330508   0.7191188
> raneef(lmer.intercepts.depend.on.log.ur-
    anium)
$county.name
                (Intercept)
AITKIN        -0.0142971713
ANOKA          0.0583741025
BECKER        -0.0125490841
BELTRAMI       0.0312484900
BENTON         0.0017869830
BIG STONE     -0.0060780289
BLUE EARTH     0.0895241245
BROWN          0.0078003746
CARLTON       -0.0293551573
CARVER        -0.0230826914
CASS           0.0499879229
CHIPPEWA       0.0161734868
CHISAGO        0.0272838175
CLAY           0.0475401692
[…]
```

Estimates of the η_j's themselves

Outline

- Introduction, Terminology, Multi-level Models
- London Schools Data
  - Plotting clusters (groups, clumps, …)
- Minnesota Radon Data
- The Random-Intercept Model
- Different ways to write the model:
  - Mixed Effects, Variance Components, Multilevel Model
- Modeling the intercept as a function of a group-level covariate