

Correlated Data and Mixed models

FW8051 Statistics for Ecologists

Department of Fisheries, Wildlife and Conservation Biology



Learning objectives: Correlated Data / Mixed models

Overview

- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])

Learning objectives: Correlated Data / Mixed models

Overview

- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])
- Be able to identify when to use a mixed model

Learning objectives: Correlated Data / Mixed models

Overview

- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])
- Be able to identify when to use a mixed model
- Learn how to implement mixed models in R/JAGs

Learning objectives: Correlated Data / Mixed models

Overview

- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])
- Be able to identify when to use a mixed model
- Learn how to implement mixed models in R/JAGs
 - When the response is Normally distributed (linear mixed effect models, lme)

Learning objectives: Correlated Data / Mixed models

Overview

- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])
- Be able to identify when to use a mixed model
- Learn how to implement mixed models in R/JAGs
 - When the response is Normally distributed (linear mixed effect models, lme)
 - For count, presence absence data (generalized linear mixed effect models, glmms)

Learning objectives: Correlated Data / Mixed models

Overview

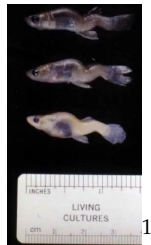
- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])
- Be able to identify when to use a mixed model
- Learn how to implement mixed models in R/JAGs
 - When the response is Normally distributed (linear mixed effect models, lme)
 - For count, presence absence data (generalized linear mixed effect models, glmms)
 - Understand why generalized linear mixed effects can be difficult to fit

Learning objectives: Correlated Data / Mixed models

Overview

- Understand some relatively simple ways to deal with correlated data (bootstrap, Generalized Estimating Equations [later])
- Be able to identify when to use a mixed model
- Learn how to implement mixed models in R/JAGs
 - When the response is Normally distributed (linear mixed effect models, lme)
 - For count, presence absence data (generalized linear mixed effect models, glmms)
 - Understand why generalized linear mixed effects can be difficult to fit
- Be able to describe models and their assumptions using equations and text and match parameters in these equations to estimates in computer output.

Selenium and Fish

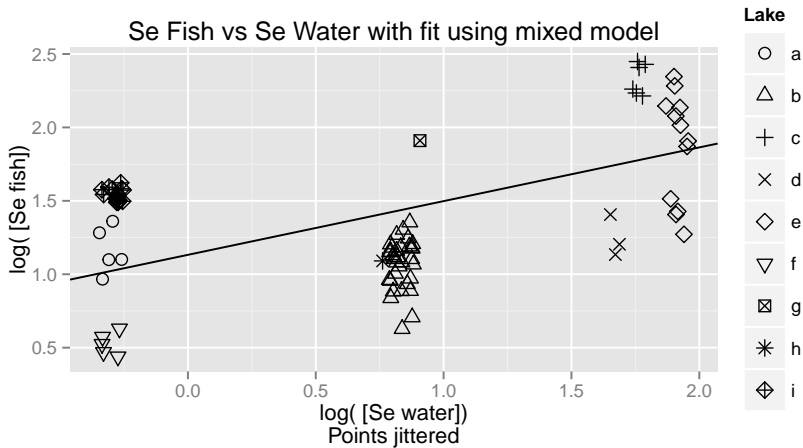


Selenium, Se, a bi-product of burning coal is measured in...

- A set of 9 lakes
- 1 to 34 fish in each lake (total of 83 observations)

Goal: determine the relationship between mean (log) Se in lake and mean (log) Se in fish.

¹<http://appvoices.org/tag/appalachian-voices/page/7/>



Selenium Example

What are the consequences of ignoring the fact that we have multiple observations from each lake?

Selenium Example

What are the consequences of ignoring the fact that we have multiple observations from each lake?

- If we use linear regression (assuming independence), our SE's will be too small (observations from the same lake are not independent)

Selenium Example

What are the consequences of ignoring the fact that we have multiple observations from each lake?

- If we use linear regression (assuming independence), our SE's will be too small (observations from the same lake are not independent)

What strategies might we use to analyze these data?

Selenium Example

What are the consequences of ignoring the fact that we have multiple observations from each lake?

- If we use linear regression (assuming independence), our SE's will be too small (observations from the same lake are not independent)

What strategies might we use to analyze these data?

Note: our main question involves a predictor-response relationship in which the predictor is constant within each cluster or sample unit

Selenium Example

Strategies:

- Fit a linear regression model, but use a cluster-level bootstrap for inference

Selenium Example

Strategies:

- Fit a linear regression model, but use a cluster-level bootstrap for inference
- Calculate averages of Y for each cluster, then fit linear regression models to these averages (will have 1 observation per cluster)

Selenium Example

Strategies:

- Fit a linear regression model, but use a cluster-level bootstrap for inference
- Calculate averages of Y for each cluster, then fit linear regression models to these averages (will have 1 observation per cluster)
- Use a mixed model with a random intercept for each cluster (i.e., lake)

Selenium Example

Strategies:

- Fit a linear regression model, but use a cluster-level bootstrap for inference
- Calculate averages of Y for each cluster, then fit linear regression models to these averages (will have 1 observation per cluster)
- Use a mixed model with a random intercept for each cluster (i.e., lake)

Lets do this! See `Se-lake.R`

Correlated Data: Folklore Theorem: regression parameter estimators will be unbiased, but standard errors will be too small (if we assume data are independent)

Correlated Data: Folklore Theorem: regression parameter estimators will be unbiased, but standard errors will be too small (if we assume data are independent)

- Can use cluster-level bootstrap for inference
- Generalized estimating equations (that treat clusters as independent sample units... more later)

Correlated Data: Folklore Theorem: regression parameter estimators will be unbiased, but standard errors will be too small (if we assume data are independent)

- Can use cluster-level bootstrap for inference
- Generalized estimating equations (that treat clusters as independent sample units... more later)

These above methods are most appropriate when:

- The degree of missingness (or amount of data for each cluster) is 'completely random'

Correlated Data: Folklore Theorem: regression parameter estimators will be unbiased, but standard errors will be too small (if we assume data are independent)

- Can use cluster-level bootstrap for inference
- Generalized estimating equations (that treat clusters as independent sample units... more later)

These above methods are most appropriate when:

- The degree of missingness (or amount of data for each cluster) is 'completely random' (we don't have more data from lakes where Se seems to be having a bigger impact on fish)

Correlated Data: Folklore Theorem: regression parameter estimators will be unbiased, but standard errors will be too small (if we assume data are independent)

- Can use cluster-level bootstrap for inference
- Generalized estimating equations (that treat clusters as independent sample units... more later)

These above methods are most appropriate when:

- The degree of missingness (or amount of data for each cluster) is 'completely random' (we don't have more data from lakes where Se seems to be having a bigger impact on fish)
- We are interested in the effects of covariates to do not vary within a cluster (rather than relationships within particular lakes)

Correlated Data: Folklore Theorem: regression parameter estimators will be unbiased, but standard errors will be too small (if we assume data are independent)

- Can use cluster-level bootstrap for inference
- Generalized estimating equations (that treat clusters as independent sample units... more later)

These above methods are most appropriate when:

- The degree of missingness (or amount of data for each cluster) is 'completely random' (we don't have more data from lakes where Se seems to be having a bigger impact on fish)
- We are interested in the effects of covariates to do not vary within a cluster (rather than relationships within particular lakes)
- Correlation is just a nuisance (we don't care about within-lake variability in Se measurements)

Mixed models

Models fit using lmer in lme4 package

Model Selection/Model Building Strategies

When to use a mixed model

When you have more than one measurement on the same observational unit

When to use a mixed model

When you have more than one measurement on the same observational unit

- Multiple observations per lake, animal, study site, etc.

When to use a mixed model

When you have more than one measurement on the same observational unit

- Multiple observations per lake, animal, study site, etc.

Experiments or surveys with multiple sizes of sample units

When to use a mixed model

When you have more than one measurement on the same observational unit

- Multiple observations per lake, animal, study site, etc.

Experiments or surveys with multiple sizes of sample units

- Split-plot designs (treatments applied to whole plots and subplots)
- Cluster samples (samples of households, individuals within households)

When to use a mixed model

When you have more than one measurement on the same observational unit

- Multiple observations per lake, animal, study site, etc.

Experiments or surveys with multiple sizes of sample units

- Split-plot designs (treatments applied to whole plots and subplots)
- Cluster samples (samples of households, individuals within households)

When you want to generalize to a larger population of sample units

When to use a mixed model

When you have more than one measurement on the same observational unit

- Multiple observations per lake, animal, study site, etc.

Experiments or surveys with multiple sizes of sample units

- Split-plot designs (treatments applied to whole plots and subplots)
- Cluster samples (samples of households, individuals within households)

When you want to generalize to a larger population of sample units

- **Fixed effects**: allow inference to only the sample units in the data set
- **Random effects**: allow us to generalize to a population of sample units by assuming regression parameters have a distribution

Multi-level, Mixed Effects, or Hierarchical models

Key features:

- Regression parameters vary by cluster (e.g., population, individual animal, etc.)

Multi-level, Mixed Effects, or Hierarchical models

Key features:

- Regression parameters vary by cluster (e.g., population, individual animal, etc.)
- Regression parameters are assumed to follow a probability distribution

Multi-level, Mixed Effects, or Hierarchical models

Key features:

- Regression parameters vary by cluster (e.g., population, individual animal, etc.)
- Regression parameters are assumed to follow a probability distribution
- We estimate the variance of these parameters across clusters

Multi-level, Mixed Effects, or Hierarchical models

Key features:

- Regression parameters vary by cluster (e.g., population, individual animal, etc.)
- Regression parameters are assumed to follow a probability distribution
- We estimate the variance of these parameters across clusters

Why are they so popular:

- Provide a framework for modeling correlated and nested data

Multi-level, Mixed Effects, or Hierarchical models

Key features:

- Regression parameters vary by cluster (e.g., population, individual animal, etc.)
- Regression parameters are assumed to follow a probability distribution
- We estimate the variance of these parameters across clusters

Why are they so popular:

- Provide a framework for modeling correlated and nested data
- Allow estimation of variance components (e.g., variance among individuals, within-individuals)

Multi-level, Mixed Effects, or Hierarchical models

Key features:

- Regression parameters vary by cluster (e.g., population, individual animal, etc.)
- Regression parameters are assumed to follow a probability distribution
- We estimate the variance of these parameters across clusters

Why are they so popular:

- Provide a framework for modeling correlated and nested data
- Allow estimation of variance components (e.g., variance among individuals, within-individuals)
- Many ecological data sets are hierarchically structured data (e.g., wolves in packs, populations)

RIKZ data

Sampling Effort:

- 9 beaches (high, medium, low exposure)
- 5 stations at each beach.

Interest lies in modeling:

- Richness = species richness (number of species counted).

Using macro-fauna and abiotic variables:

- Exposure = low or high exposure to waves, length of surf zone, slope, grain size, and depth of the anaerobic layer
- NAP = height of the sampling station compared to mean tidal level

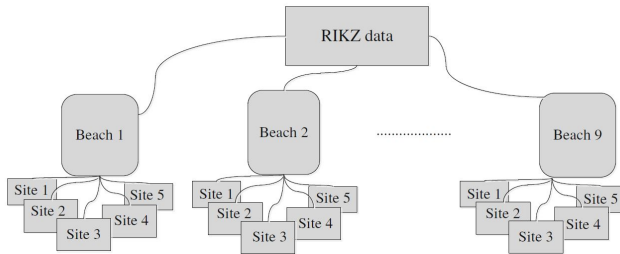


Fig. 5.1 Set up of the RIKZ data. Measurements were taken on 9 beaches, and on each beach 5 sites were sampled. Richness values at sites on the same beach are likely to be more similar to each other than to values from different beaches

Linear regression assumes that observations are independent.
Is that reasonable in this case?

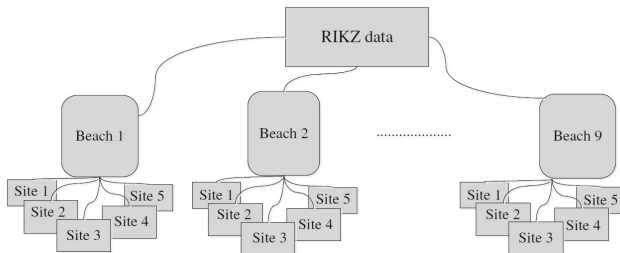


Fig. 5.1 Set up of the RIKZ data. Measurements were taken on 9 beaches, and on each beach 5 sites were sampled. Richness values at sites on the same beach are likely to be more similar to each other than to values from different beaches

Linear regression assumes that observations are independent. Is that reasonable in this case?

- 2 observations from the same beach may be more alike than 2 observations taken from 2 different beaches.

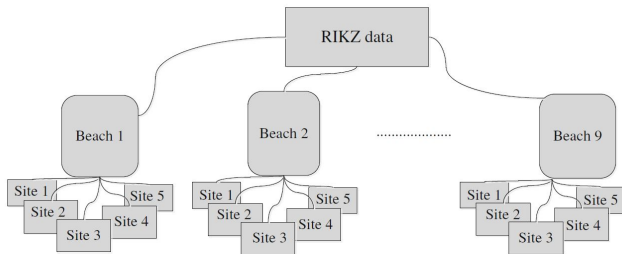


Fig. 5.1 Set up of the RIKZ data. Measurements were taken on 9 beaches, and on each beach 5 sites were sampled. Richness values at sites on the same beach are likely to be more similar to each other than to values from different beaches

Linear regression assumes that observations are independent. Is that reasonable in this case?

- 2 observations from the same beach may be more alike than 2 observations taken from 2 different beaches.
- \Rightarrow observations from the same beach are likely correlated

Multi-level model

Think of models at 2 levels:

- Level 1: model the how individual observations vary within a cluster

Multi-level model

Think of models at 2 levels:

- Level 1: model the how individual observations vary within a cluster
- Level 2: model how (cluster-specific) parameters, in the level-1 model, vary (across clusters)

2-stage multi-level modeling approach

Stage 1 (level 1 model):

- Build a separate model for each cluster (beach)
- Only consider variables that are NOT constant within a cluster

2-stage multi-level modeling approach

Stage 1 (level 1 model):

- Build a separate model for each cluster (beach)
- Only consider variables that are NOT constant within a cluster

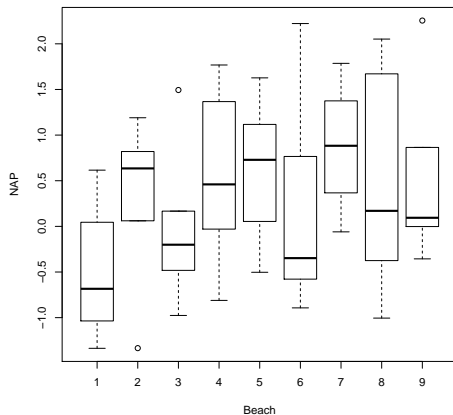
Stage 2 (level 2 model):

- Treat the coefficients from stage 1 as 'data'
- Model the coefficients as a function of variables that are constant within a cluster

Can be useful exploratory approach when you have lots of data for each cluster, but few clusters (e.g., animal telemetry studies)

RIKZ data

NAP is a “level-1” covariate (it varies within each cluster)



RIKZ data

exposure is a “level-2” covariate (it is constant within a cluster)

```
xtabs(~ exposure + Beach, data=RIKZ)
```

	Beach								
exposure	1	2	3	4	5	6	7	8	9
8	0	5	0	0	0	0	0	0	0
10	5	0	0	0	5	0	0	5	5
11	0	0	5	5	0	5	5	0	0

```
# Only 1 beach with lowest exposure level: modify to have 2 categories  
RIKZ$exposure.c<-"High"  
RIKZ$exposure.c[RIKZ$exposure%in%c(8,10)]<-"Low"
```

2-Stage approach

Let R_{ij} = the species richness for the j^{th} sample on the i^{th} beach
(note: we now need two subscripts!)

2-Stage approach

Let R_{ij} = the species richness for the j^{th} sample on the i^{th} beach
(note: we now need two subscripts!)

Level 1 model: model for observations within each cluster (i.e.,
for each beach)

$$R_{ij} = \beta_{0i} + \beta_{1i}NAP_{ij} + \epsilon_{ij}; (j = 1, 2, \dots, 5 \text{ observations for each Beach})$$

Each beach has its own intercept β_{0i} and slope β_{1i}

Modified R code

```
RIKZ$NAPc = RIKZ$NAP - mean(RIKZ$NAP)
Beta <- matrix(NA, 9, 2) # to hold slope and intercepts
Exposure <- matrix(NA, 9, 1) # to hold exposure level for each beach
for(i in 1:9){
  Mi <- lm(Richness~NAPc, data=subset(RIKZ, Beach==i))
  Beta[i,] <- coef(Mi)
  Exposure[i] <- subset(RIKZ, Beach==i)$exposure.c[1]
}
level1dat <- data.frame(beach=unique(RIKZ$Beach), intercepts=Beta[,1],
                        slopes.NAP=Beta[,2], exposure=Exposure)
```

Note: I have centered the NAP variable

- Makes intercept more meaningful = R_{ij} at the mean value of NAP
- Helps avoid numerical problems and identifiability problems due to correlation of $\hat{\beta}_{0i}$ and $\hat{\beta}_{1i}$

This gives us a data frame of coefficients and level-2 predictors for a level-2 model:

```
level1dat
```

	beach	intercepts	slopes.NAP	exposure
1	1	10.692614	-0.3718279	Low
2	2	11.893999	-4.1752712	Low
3	3	2.790385	-1.7553529	High
4	4	2.653600	-1.2485766	High
5	5	9.688335	-8.9001779	Low
6	6	3.841864	-1.3885120	High
7	7	2.992969	-1.5176126	High
8	8	4.293257	-1.8930665	Low
9	9	5.263276	-2.9675304	Low

tidyverse solution

```
library(tidyverse)
library(broom)
library(tidyr)
```

```
tidy(lm(Richness~NAPc, data=RIKZ))
```

```
# A tibble: 2 x 5
```

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	5.69	0.620	9.17	1.11e-11
2	NAPc	-2.87	0.631	-4.55	4.42e- 5

tidyverse solution

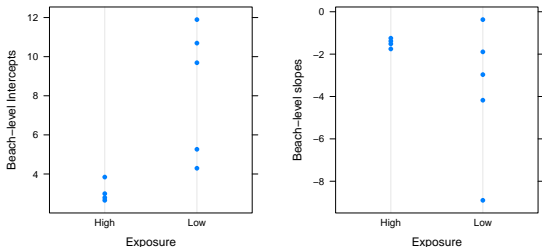
```
levelldat.tidy <- RIKZ %>% group_by(Beach) %>%  
  do(tidy(lm(Richness~NAPc, data=))) %>%  
  pivot_wider(id_cols=Beach, names_from=term, values_from=estimate)  
levelldat.tidy
```

```
# A tibble: 9 x 3  
# Groups:   Beach [9]  
  Beach `(Intercept)`    NAPc  
  <int>      <dbl>    <dbl>  
1     1      10.7  -0.372  
2     2      11.9  -4.18  
3     3       2.79 -1.76  
4     4       2.65 -1.25  
5     5       9.69 -8.90  
6     6       3.84 -1.39  
7     7       2.99 -1.52  
8     8       4.29 -1.89  
9     9       5.26 -2.97
```

```

library(gridExtra)
library(mosaic)
par(mfrow=c(1,2))
d1<-dotplot(intercepts~Exposure, data=level1dat, xlab="Exposure",
            ylab="Beach-level Intercepts")
d2<-dotplot(slopes.NAP~Exposure, data=level1dat, xlab="Exposure",
            ylab="Beach-level slopes")
grid.arrange(d1, d2, ncol=2)

```



Level-2 model

Model for the slope and intercept parameters (analyze the summary statistics, $\hat{\beta}_{0i}, \hat{\beta}_{1i}$) using level-2 predictors (ones that are constant within a cluster)

- $\hat{\beta}_{0i} = \beta_0 + \gamma_0 Exposure_i + b_{0i}$
- $\hat{\beta}_{1i} = \beta_1 + \gamma_1 Exposure_i + b_{1i}$

For now, ignore the fact that the variability of b_{0i}, b_{1i} seems to depend on exposure level (“low”, “high”).

Level-2 Model: Intercepts

```
summary(lm(intercepts~Exposure, data=level1dat))
```

Call:

```
lm(formula = intercepts ~ Exposure, data = level1dat)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-4.0730	-0.4161	-0.0767	1.3220	3.5277

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.070	1.291	2.378	0.0491 *
ExposureLow	5.297	1.732	3.058	0.0184 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.582 on 7 degrees of freedom

Multiple R-squared: 0.5719, Adjusted R-squared: 0.5107

F-statistic: 9.349 on 1 and 7 DF, p-value: 0.01838

Level-2 Model: Slopes

```
summary(lm(slopes.NAP~Exposure, data=level1dat))
```

Call:

```
lm(formula = slopes.NAP ~ Exposure, data = level1dat)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-5.2386	-0.2778	0.0890	0.6940	3.2897

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.478	1.229	-1.202	0.268
ExposureLow	-2.184	1.649	-1.325	0.227

Residual standard error: 2.458 on 7 degrees of freedom

Multiple R-squared: 0.2005, Adjusted R-squared: 0.08625

F-statistic: 1.755 on 1 and 7 DF, p-value: 0.2268

Putting things together: Composite Equation

Level-1 Model:

- $R_{ij} = \beta_{0i} + \beta_{1i}NAP_{ij} + \epsilon_{ij}$

Level-2 Model:

- $\beta_{0i} = \beta_0 + \gamma_0 Exposure_i + b_{0i}$
- $\beta_{1i} = \beta_1 + b_{1i}$

Putting things together: Composite Equation

Level-1 Model:

- $R_{ij} = \beta_{0i} + \beta_{1i}NAP_{ij} + \epsilon_{ij}$

Level-2 Model:

- $\beta_{0i} = \beta_0 + \gamma_0 Exposure_i + b_{0i}$

- $\beta_{1i} = \beta_1 + b_{1i}$

Substitute into level-1 equation to get the *composite equation*

$$R_{ij} = (\beta_0 + \gamma_0 Exposure_i + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \epsilon_{ij}$$

Putting things together: Composite Equation

Level-1 Model:

- $R_{ij} = \beta_{0i} + \beta_{1i}NAP_{ij} + \epsilon_{ij}$

Level-2 Model:

- $\beta_{0i} = \beta_0 + \gamma_0 Exposure_i + b_{0i}$
- $\beta_{1i} = \beta_1 + b_{1i}$

Substitute into level-1 equation to get the *composite equation*

$$R_{ij} = (\beta_0 + \gamma_0 Exposure_i + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \epsilon_{ij}$$

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \gamma_0 Exposure_i + \epsilon_{ij}$$

\Rightarrow *random intercepts and slopes model (or random coefficients model)*

Mixed Models

Rather than use a 2-stage approach, we could just posit a model for the data using random and fixed effects.

Random Intercepts Model:

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$b_{0i} \sim N(0, \tau^2)$$

Mixed Models

Rather than use a 2-stage approach, we could just posit a model for the data using random and fixed effects.

Random Intercepts Model:

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$b_{0i} \sim N(0, \tau^2)$$

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i}) NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

Mixed Models

Rather than use a 2-stage approach, we could just posit a model for the data using random and fixed effects.

Random Intercepts Model:

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$b_{0i} \sim N(0, \tau^2)$$

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i}) NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

Can think of b_{0i} and b_{1i} as deviations from the average intercept (β_0) and slope (β_1), respectively.

Mixed Models

Rather than use a 2-stage approach, we could just posit a model for the data using random and fixed effects.

Random Intercepts Model:

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$b_{0i} \sim N(0, \tau^2)$$

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i}) NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

Can think of b_{0i} and b_{1i} as deviations from the average intercept (β_0) and slope (β_1), respectively.

Or, think in terms of beach-level intercepts and slopes: $\beta_{0i} = \beta_0 + b_{0i}$ and $\beta_{1i} = \beta_1 + b_{1i}$, with $(\beta_{0i}, \beta_{1i}) \sim MVN(\beta, D)$

Random intercepts versus random coefficient models

Although random intercepts models are common. . .

Schielzeth and Forstmeier (2009) suggest random slopes are usually appropriate for level-1 predictors (i.e., when x varies within a subject).

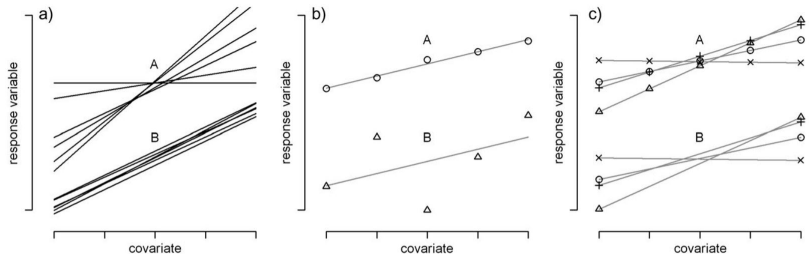


Figure 1

Schematic illustrations of more (A) and less (B) problematic cases for the estimation of fixed-effect covariates in random-intercept models. (a) Regression lines for several individuals with high (A) and low (B) between-individual variation in slopes (σ_β). (b) Two individual regression slopes with low (A) and high (B) scatter around the regression line (σ_β). (c) Regression lines with (A) many and (B) few measurements per individual (independent of the number of levels of the covariate).

See *Readings, Linear Mixed Effects Page* for a copy of Schielzeth and Forstmeier (2009)

Fitting Mixed Effects Models in R

Two popular packages: `nlme` and `lme4`:

`nlme` (older)

- More flexibility for modeling within-cluster correlation and heterogeneity (e.g., time series data, spatial data); see e.g., Ch. 4 in Zuur et al.

Fitting Mixed Effects Models in R

Two popular packages: `nlme` and `lme4`:

`nlme` (older)

- More flexibility for modeling within-cluster correlation and heterogeneity (e.g., time series data, spatial data); see e.g., Ch. 4 in Zuur et al.
- Responses must be Normally distributed

Fitting Mixed Effects Models in R

Two popular packages: `nlme` and `lme4`:

`nlme` (older)

- More flexibility for modeling within-cluster correlation and heterogeneity (e.g., time series data, spatial data); see e.g., Ch. 4 in Zuur et al.
- Responses must be Normally distributed

`lme4` (newer)

- Better options for fitting non-normal data: generalized linear mixed effects models [GLMMS] for count or binary data

Fitting Mixed Effects Models in R

Two popular packages: `nlme` and `lme4`:

`nlme` (older)

- More flexibility for modeling within-cluster correlation and heterogeneity (e.g., time series data, spatial data); see e.g., Ch. 4 in Zuur et al.
- Responses must be Normally distributed

`lme4` (newer)

- Better options for fitting non-normal data: generalized linear mixed effects models [GLMMS] for count or binary data
- Easier to fit non-nested or 'crossed' random effects (e.g., `year` and `Beach` if we had many years of data).

Fitting Mixed Effects Models in R

Two popular packages: `nlme` and `lme4`:

`nlme` (older)

- More flexibility for modeling within-cluster correlation and heterogeneity (e.g., time series data, spatial data); see e.g., Ch. 4 in Zuur et al.
- Responses must be Normally distributed

`lme4` (newer)

- Better options for fitting non-normal data: generalized linear mixed effects models [GLMMS] for count or binary data
- Easier to fit non-nested or 'crossed' random effects (e.g., `year` and `Beach` if we had many years of data).
- Cannot handle within-cluster correlation or heterogeneity

Other Packages

Many others too... see:

<http://glmm.wikidot.com/pkg-comparison>

We may also consider:

- glmmADMB
- glmmTMB
- GLMMadaptive

```
library(nlme)
lme.fit<-lme(Richness~NAPc+exposure.c, random=~1|Beach, data=RIKZ)
```

fixed effects: Richness~NAPc+exposure.c

$$\beta_0 \cdot 1 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$


```
library(nlme)
lme.fit<-lme(Richness~NAPc+exposure.c, random=~1|Beach, data=RIKZ)
```

fixed effects: Richness~NAPc+exposure.c

$$\beta_0 \cdot 1 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

random effects: random=~1|Beach

- 1 is R's notation for the intercept
- |Beach means that the intercepts are random for each Beach.
- Includes a b_{0i} for each Beach (to capture deviations from the fixed, population-level intercept, β_0)

```
library(nlme)
lme.fit<-lme(Richness~NAPc+exposure.c, random=~1|Beach, data=RIKZ)
```

fixed effects: Richness~NAPc+exposure.c

$$\beta_0 \cdot 1 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

random effects: random=~1|Beach

- 1 is R's notation for the intercept
- |Beach means that the intercepts are random for each Beach.
- Includes a b_{0i} for each Beach (to capture deviations from the fixed, population-level intercept, β_0)

Default assumption: $b_{0i} \sim N(0, \tau^2)$

- We estimate τ^2 , not the individual b_{0i}

```
library(nlme)
lme.fit<-lme(Richness~NAPc+exposure.c, random=~1|Beach, data=RIKZ)
```

fixed effects: `Richness~NAPc+exposure.c`

$$\beta_0 \cdot 1 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

random effects: `random=~1|Beach`

- 1 is R's notation for the intercept
- | Beach means that the intercepts are random for each Beach.
- Includes a b_{0i} for each Beach (to capture deviations from the fixed, population-level intercept, β_0)

Default assumption: $b_{0i} \sim N(0, \tau^2)$

- We estimate τ^2 , not the individual b_{0i}
- Since the b_{0i} are assumed to be “random”, we “predict” them (similar to “estimating” errors, ϵ_i using residuals)

```
library(nlme)
lme.fit<-lme(Richness~NAPc+exposure.c, random=~1|Beach, data=RIKZ)
```

fixed effects: `Richness~NAPc+exposure.c`

$$\beta_0 \cdot 1 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

random effects: `random=~1|Beach`

- 1 is R's notation for the intercept
- | Beach means that the intercepts are random for each Beach.
- Includes a b_{0i} for each Beach (to capture deviations from the fixed, population-level intercept, β_0)

Default assumption: $b_{0i} \sim N(0, \tau^2)$

- We estimate τ^2 , not the individual b_{0i}
- Since the b_{0i} are assumed to be “random”, we “predict” them (similar to “estimating” errors, ϵ_i using residuals)
- BLUPS = best linear unbiased predictions.

```
library(nlme)
lme.fit<-lme(Richness~NAPc+exposure.c, random=~1|Beach, data=RIKZ)
```

fixed effects: `Richness~NAPc+exposure.c`

$$\beta_0 \cdot 1 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

random effects: `random=~1|Beach`

- 1 is R's notation for the intercept
- | Beach means that the intercepts are random for each Beach.
- Includes a b_{0i} for each Beach (to capture deviations from the fixed, population-level intercept, β_0)

Default assumption: $b_{0i} \sim N(0, \tau^2)$

- We estimate τ^2 , not the individual b_{0i}
- Since the b_{0i} are assumed to be “random”, we “predict” them (similar to “estimating” errors, ϵ_i using residuals)
- BLUPS = best linear unbiased predictions.

... unless you are a Bayesian.

```
summary(lme.fit)
```

Linear mixed-effects model fit by REML

Data: RIKZ

	AIC	BIC	logLik
	240.5538	249.2422	-115.2769

Random effects:

Formula: ~1 | Beach

(Intercept) Residual

StdDev: 1.907175 3.059089

Fixed effects: Richness ~ NAPc + exposure.c

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.170680	1.1739988	35	2.700752	0.0106
NAPc	-2.581708	0.4883901	35	-5.286160	0.0000
exposure.cLow	4.532777	1.5755612	7	2.876928	0.0238

Correlation:

(Intr) NAPc

NAPc -0.028

exposure.cLow -0.746 0.037

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-1.5163203	-0.4815106	-0.1218701	0.2922855	3.8777562

Number of Observations: 45

Number of Groups: 9

```
summary(lme.fit)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.170680	1.1739988	35	2.700752	1.058924e-02
NAPc	-2.581708	0.4883901	35	-5.286160	6.745464e-06
exposure.cLow	4.532777	1.5755612	7	2.876928	2.375560e-02

fixed effects: Richness~NAPc+exposure.c

$$\beta_0 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

```
summary(lme.fit)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.170680	1.1739988	35	2.700752	1.058924e-02
NAPc	-2.581708	0.4883901	35	-5.286160	6.745464e-06
exposure.cLow	4.532777	1.5755612	7	2.876928	2.375560e-02

fixed effects: Richness~NAPc+exposure.c

$$\beta_0 + \beta_1 NAP_{ij} + \beta_2 exposure.c$$

Degrees of Freedom (differ for level-1 and level-2 predictors):

- NAPc = 35
- exposure.cLow = 7

Degrees of Freedom

Level-1: within-subjects degrees of freedom calculated as the number of observations minus the number of groups minus the number of level-1 regressors in the model.

```
nrow(RIKZ) - length(unique(RIKZ$Beach)) - 1
```

```
[1] 35
```

Degrees of Freedom

Level-1: within-subjects degrees of freedom calculated as the number of observations minus the number of groups minus the number of level-1 regressors in the model.

```
nrow(RIKZ) - length(unique(RIKZ$Beach)) - 1
```

```
[1] 35
```

Level-2: among-subjects degrees of freedom calculated as the number of groups minus the number of level-2 regressors in the model - 1 for the intercept.

```
length(unique(RIKZ$Beach)) - 1 - 1
```

```
[1] 7
```

Degrees of Freedom

The formula are not important. . . . what is:

Degrees of Freedom

The formula are not important...what is:

- we have more information about the effect of NAP on species richness than exposure since NAP varies between and within beaches.

Degrees of Freedom

The formula are not important. . . . what is:

- we have more information about the effect of NAP on species richness than `exposure` since NAP varies between and within beaches.
- `lme` accounts for the data structure when carrying out statistical tests.

Degrees of Freedom: More accurately

Note: `lme`'s df are essentially correct for **balanced data** (all clusters have an equal number of observations). For unbalanced data, the tests (and df) are only approximate.

Degrees of Freedom: More accurately

Note: `lme`'s df are essentially correct for **balanced data** (all clusters have an equal number of observations). For unbalanced data, the tests (and df) are only approximate.

- thus, a decision was made to NOT report p-values for models fit with `lmer` in `lme4`

Degrees of Freedom: More accurately

Note: `lme`'s df are essentially correct for **balanced data** (all clusters have an equal number of observations). For unbalanced data, the tests (and df) are only approximate.

- thus, a decision was made to NOT report p-values for models fit with `lmer` in `lme4`
- there are “better” degrees of freedom approximations for unbalanced data (see, e.g., *lmerTest* package).

Variance Components

```
VarCorr(lme.fit)
```

```
Beach = pdLogChol(1)
```

	Variance	StdDev
(Intercept)	3.637317	1.907175
Residual	9.358027	3.059089

$$\epsilon_i \sim N(0, \sigma^2)$$

$$b_{0i} \sim N(0, \tau^2)$$

Variance Components

```
VarCorr(lme.fit)
```

```
Beach = pdLogChol(1)
          Variance StdDev
(Intercept) 3.637317 1.907175
Residual    9.358027 3.059089
```

$$\epsilon_i \sim N(0, \sigma^2)$$

$$b_{0i} \sim N(0, \tau^2)$$

- $var(\epsilon_{ij}) = \sigma^2 = 9.36$ (variance within a Beach)
- $var(b_{0i}) = \tau^2 = 3.637$ (variance among beaches)

Induced correlation: random intercepts model

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$

$$\text{Variance of } R_{ij} = \text{var}(b_{0i} + \epsilon_{ij}) = \text{var}(b_{0i}) + \text{var}(\epsilon_{ij}) = \tau^2 + \sigma^2$$

Induced correlation: random intercepts model

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$

Variance of $R_{ij} = var(b_{0i} + \epsilon_{ij}) = var(b_{0i}) + var(\epsilon_{ij}) = \tau^2 + \sigma^2$

Covariance $(Y_{ij}, Y_{ij'}) = \tau^2$ (2 observations, same cluster [beach] since they share b_{0i})

Induced correlation: random intercepts model

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$

Variance of $R_{ij} = var(b_{0i} + \epsilon_{ij}) = var(b_{0i}) + var(\epsilon_{ij}) = \tau^2 + \sigma^2$

Covariance ($Y_{ij}, Y_{ij'}$) = τ^2 (2 observations, same cluster [beach] since they share b_{0i})

Covariance ($Y_{ij}, Y_{i'j}$) = 0 (2 observations taken from 2 different clusters [beaches])

Induced correlation: random intercepts model

$$R_{ij} = \beta_0 + b_{0i} + \beta_1 NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$

Variance of $R_{ij} = var(b_{0i} + \epsilon_{ij}) = var(b_{0i}) + var(\epsilon_{ij}) = \tau^2 + \sigma^2$

Covariance ($Y_{ij}, Y_{ij'}$) = τ^2 (2 observations, same cluster [beach] since they share b_{0i})

Covariance ($Y_{ij}, Y_{i'j}$) = 0 (2 observations taken from 2 different clusters [beaches])

Intraclass correlation = $Cor(Y_{ij}, Y_{ij'}) = \frac{\tau^2}{\tau^2 + \sigma^2} = 0.28$, correlation among observations taken from the same cluster.

Each beach also has its own intercept. What if we modeled Beach using fixed effects?

Each beach also has its own intercept. What if we modeled Beach using fixed effects?

```
lm.fe <- lm(Richness~factor(Beach)-1+NAPc, data=RIKZ)
summary(lm.fe)
```

Call:

```
lm(formula = Richness ~ factor(Beach) - 1 + NAPc, data = RIKZ)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-4.8518	-1.5188	-0.1376	0.7905	11.8384

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
factor(Beach)1	8.9392	1.4301	6.251	3.61e-07	***
factor(Beach)2	12.0173	1.3690	8.778	2.29e-10	***
factor(Beach)3	2.5343	1.3796	1.837	0.074716	.
factor(Beach)4	2.9063	1.3723	2.118	0.041364	*
factor(Beach)5	8.0409	1.3746	5.850	1.22e-06	***
factor(Beach)6	3.7161	1.3697	2.713	0.010271	*
factor(Beach)7	3.5025	1.3934	2.514	0.016705	*
factor(Beach)8	4.3862	1.3707	3.200	0.002920	**
factor(Beach)9	5.1572	1.3731	3.756	0.000629	***
NAPc	-2.4928	0.5023	-4.963	1.79e-05	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.06 on 35 degrees of freedom

Multiple R-squared: 0.8719, Adjusted R-squared: 0.8353

F-statistic: 23.82 on 10 and 35 DF, p-value: 9.56e-13

Fixed versus random

Fixed effects:

- `lm.fe <- lm(Richness~factor(Beach)-1+NAPc, data=RIKZ)`
- each beach has its own intercept which we estimate

Fixed versus random

Fixed effects:

- `lm.fe <- lm(Richness~factor(Beach)-1+NAPc, data=RIKZ)`
- each beach has its own intercept which we estimate

Random effects:

- `lme.fit<-lme(Richness~NAPc+exposure.c,
random=~1 | Beach, data=RIKZ)`
- each beach has its own intercept
- we further assume $\beta_i \sim N(\beta, \sigma_{b_{oi}}^2)$ or equivalently $b_{oi} \sim N(0, \sigma_{b_{oi}}^2)$
- we estimate the variance of the intercepts and “predict” the beach-level intercepts

Downsides to fixed effects model

- Requires estimation of 8 parameters

Downsides to fixed effects model

- Requires estimation of 8 parameters
- Cannot include `exposure.c` since it is constant for each Beach (and therefore, confounded with the Beach coefficients)

Downsides to fixed effects model

- Requires estimation of 8 parameters
- Cannot include `exposure.c` since it is constant for each Beach (and therefore, confounded with the Beach coefficients)

```
lm.fe2 <- lm(Richness~factor(Beach)-1+NAPc+exposure.c, data=RIKZ)
coef(lm.fe2)
```

factor(Beach)1	factor(Beach)2	factor(Beach)3	factor(Beach)4	factor(Beach)5	factor(Beach)6
8.939200	12.017303	2.534266	2.906323	8.040936	3.716094
factor(Beach)7	factor(Beach)8	factor(Beach)9	NAPc	exposure.cLow	
3.502535	4.386168	5.157177	-2.492836	NA	

Downsides to fixed effects model

- Requires estimation of 8 parameters
- Cannot include `exposure.c` since it is constant for each Beach (and therefore, confounded with the Beach coefficients)

```
lm.fe2 <- lm(Richness~factor(Beach)-1+NAPc+exposure.c, data=RIKZ)
coef(lm.fe2)
```

```
factor(Beach)1  factor(Beach)2  factor(Beach)3  factor(Beach)4  factor(Beach)5  factor(Beach)6
      8.939200      12.017303      2.534266      2.906323      8.040936      3.716094
factor(Beach)7  factor(Beach)8  factor(Beach)9      NAPc  exposure.cLow
      3.502535      4.386168      5.157177      -2.492836      NA
```

- Random coefficients would require interactions between Beach and NAP (another 8 parameters)

Predicted values

Population Average (averages over beaches):

- $E[R|X] = E(E[R|X, b_{0i}])) = \beta_0 + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Predicted values

Population Average (averages over beaches):

- $E[R|X] = E(E[R|X, b_{0i}])) = \beta_0 + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Subject-Specific (lines for a particular beach):

- $E[R|X, b_{0i}] = \beta_0 + b_{0i} + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Predicted values

Population Average (averages over beaches):

- $E[R|X] = E(E[R|X, b_{0i}])) = \beta_0 + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Subject-Specific (lines for a particular beach):

- $E[R|X, b_{0i}] = \beta_0 + b_{0i} + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

```
# beta0 + beta1*NAP + beta2*Exposure
RIKZ$EY.Pop<-fitted(lme.fit, level=0)

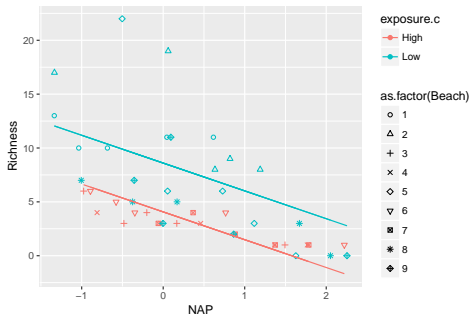
# Subject specific lines
RIKZ$EY.Beach<-fitted(lme.fit, level=1)

head(RIKZ[,13:18],3)
```

	Beach	Richness	exposure.c	NAPc	EY.Pop	EY.Beach
1	1	11	Low	-0.3026889	8.484911	9.252313
2	1	10	Low	-1.3836889	11.275737	12.043140
3	1	13	Low	-1.6836889	12.050250	12.817652

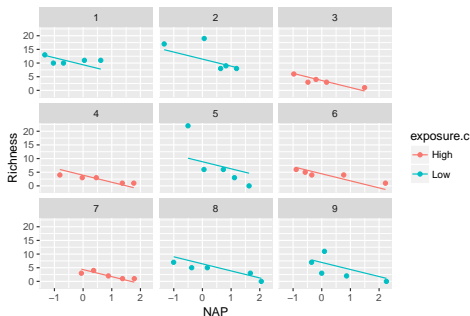
Population Averaged Estimates

```
ggplot(RIKZ, aes(x=NAP, y=Richness, shape=as.factor(Beach),  
  colour=exposure.c)) + scale_shape_manual(values=1:9) +  
  geom_point() +  
  geom_line(aes(y=EY.Pop, x=NAP, color=exposure.c))
```



Subject-specific (Beach-level estimates)

```
ggplot(RIKZ, aes(x=NAP, y=Richness, colour=exposure.c)) +  
  facet_wrap(~Beach) + geom_point() +  
  geom_line(aes(y=EY.Beach, x=NAP, color=exposure.c))
```



Each beach has its own intercept, but the slopes are the same!

Another alternative: Random intercepts and slopes

$$R_{ij} = \beta_0 + b_{0i} + (\beta_1 + b_{1i})NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

Another alternative: Random intercepts and slopes

$$R_{ij} = \beta_0 + b_{0i} + (\beta_1 + b_{1i})NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

```
lme.rc<-lme(Richness~NAPc+exposure.c, random=~1+NAPc|Beach, data=RIKZ)
```

random=~1+NAP | Beach

- Each beach gets its own intercept, $\beta_0 + b_{0i}$
- Each beach gets its own slope parameter for NAP, $\beta_1 + b_{1i}$

```
summary(lme.rc)
```

Linear mixed-effects model fit by REML

Data: RIKZ

	AIC	BIC	logLik
	240.5327	252.6964	-113.2663

Random effects:

Formula: ~1 + NAPc | Beach

Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr
(Intercept)	2.179463	(Intr)
NAPc	1.888822	-0.557
Residual	2.549885	

Fixed effects: Richness ~ NAPc + exposure.c

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.726341	1.1765068	35	3.167292	0.0032
NAPc	-2.808422	0.7596419	35	-3.697035	0.0007
exposure.cLow	3.704915	1.5176687	7	2.441188	0.0447

Correlation:

	(Intr)	NAPc
NAPc		-0.309
exposure.cLow	-0.708	0.024

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-2.02000454	-0.39889890	-0.08147617	0.22318334	2.84753809

```
VarCorr(lme.rc)
```

```
Beach = pdLogChol(1 + NAPc)
```

	Variance	StdDev	Corr
(Intercept)	4.750059	2.179463	(Intr)
NAPc	3.567648	1.888822	-0.557
Residual	6.501916	2.549885	

```
VarCorr(lme.rc)
```

```
Beach = pdLogChol(1 + NAPc)
      Variance StdDev   Corr
(Intercept) 4.750059 2.179463 (Intr)
NAPc        3.567648 1.888822 -0.557
Residual    6.501916 2.549885
```

$$\epsilon_i \sim N(0, \sigma^2) \quad (b_{0i}, b_{1i}) \sim N(0, D);$$
$$D = \begin{bmatrix} \text{var}(b_{0i}) & \text{cov}(b_{0i}, b_{1i}) \\ \text{cov}(b_{0i}, b_{1i}) & \text{var}(b_{1i}) \end{bmatrix}$$


```
VarCorr(lme.rc)
```

```
Beach = pdLogChol(1 + NAPc)
      Variance StdDev   Corr
(Intercept) 4.750059 2.179463 (Intr)
NAPc        3.567648 1.888822 -0.557
Residual    6.501916 2.549885
```

$$\epsilon_i \sim N(0, \sigma^2) \quad (b_{0i}, b_{1i}) \sim N(0, D);$$
$$D = \begin{bmatrix} \text{var}(b_{0i}) & \text{cov}(b_{0i}, b_{1i}) \\ \text{cov}(b_{0i}, b_{1i}) & \text{var}(b_{1i}) \end{bmatrix}$$

- $\text{var}(\epsilon_{ij}) = \sigma^2 = 6.50$ (variance within a Beach)
- $\text{var}(b_{0i}) = 4.750$ (variance among beach intercepts)
- $\text{var}(b_{1i}) = 3.567$ (variance among beach slopes)
- $\text{Cor}(b_{0i}, b_{1i}) = \frac{\text{Cov}(b_{0i}, b_{1i})}{\sqrt{\text{var}(b_{0i})\text{var}(b_{1i})}} = -0.557$

Predicted Values

Population Average (averages over beaches):

- $E[R|X] = E(E[R|X, b_{0i}])) = \beta_0 + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Predicted Values

Population Average (averages over beaches):

- $E[R|X] = E(E[R|X, b_{0i}])) = \beta_0 + \beta_1 NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Subject-Specific (lines for a particular beach):

- $E[R|X, b_{0i}] = \beta_0 + b_{0i} + (\beta_1 + b_{1i})NAP + \beta_2(\text{exposure}=\text{"LOW"})$

Predicted Values

Population Average (averages over beaches):

- $E[R|X] = E(E[R|X, b_{0i}])) = \beta_0 + \beta_1 NAP + \beta_2(\text{exposure} = \text{"LOW"})$

Subject-Specific (lines for a particular beach):

- $E[R|X, b_{0i}] = \beta_0 + b_{0i} + (\beta_1 + b_{1i})NAP + \beta_2(\text{exposure} = \text{"LOW"})$

```
# beta0 + beta1*NAP + beta2*Exposure
RIKZ$EY.Pop2<-fitted(lme.rc, level=0)

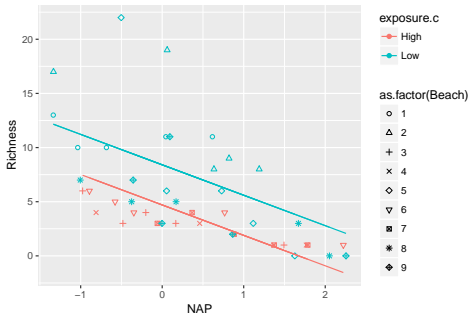
# Subject specific lines
RIKZ$EY.Beach2<-fitted(lme.rc, level=1)

head(RIKZ[,c(13:16, 19:20)], 3)
```

	Beach	Richness	exposure.c	NAPc	EY.Pop2	EY.Beach2
1	1	11	Low	-0.3026889	8.281334	9.225661
2	1	10	Low	-1.3836889	11.317239	11.734620
3	1	13	Low	-1.6836889	12.159766	12.430908

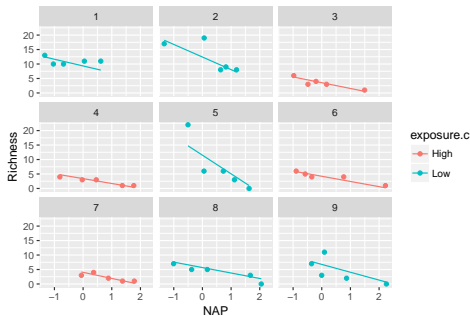
Population Averaged Estimates

```
ggplot(RIKZ, aes(x=NAP, y=Richness, shape=as.factor(Beach), colour=exp  
  scale_shape_manual(values=1:9)+geom_point()+  
  geom_line(aes(y=EY.Pop2,x=NAP, color=exposure.c))
```



Subject-specific (Beach-level estimates)

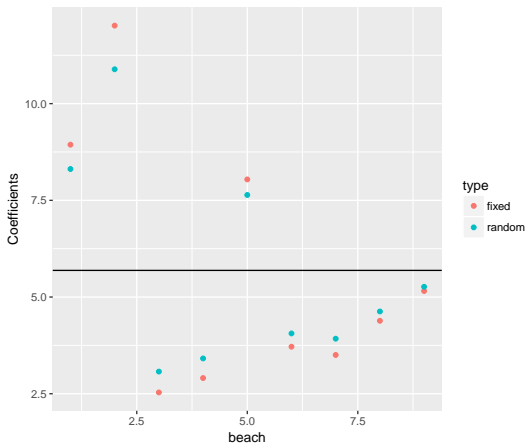
```
ggplot(RIKZ, aes(x=NAP, y=Richness, colour=exposure.c)) +  
  facet_wrap(~Beach) + geom_point() +  
  geom_line(aes(y=EY.Beach2,x=NAP, color=exposure.c))
```



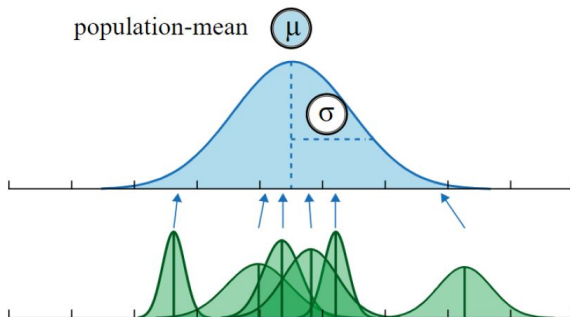
Each beach has its own intercept and slope

Potential Benefit of Mixed Effects Model

Information sharing across beaches: intercepts will be “shrunk” towards the overall population mean:



Shrinkage



https://benediktehinger.de/glm2018/mm_slides.html

Shrinkage depends on:

- how variable the coefficients are across clusters
- the degree of uncertainty associated with individual estimates

Mixed models

Models fit using lmer in lme4 package

Model Selection/Model Building Strategies

```
library(lme4)
lmer.fit<-lmer(Richness~NAPc+exposure.c+(1|Beach), data=RIKZ)
summary(lmer.fit)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Richness ~ NAPc + exposure.c + (1 | Beach)
Data: RIKZ
```

REML criterion at convergence: 230.6

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.5163	-0.4815	-0.1219	0.2923	3.8778

Random effects:

Groups	Name	Variance	Std.Dev.
Beach	(Intercept)	3.637	1.907
Residual		9.358	3.059

Number of obs: 45, groups: Beach, 9

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	3.1707	1.1740	2.701
NAPc	-2.5817	0.4884	-5.286
exposure.cLow	4.5328	1.5756	2.877

Correlation of Fixed Effects:

	(Intr)	NAPc
NAPc	-0.028	
exposur.cLw	-0.746	0.037

```
lmer.rc<-lmer(Richness~NAPc+exposure.c+(1+NAPc|Beach), data=RIKZ)
summary(lmer.rc)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Richness ~ NAPc + exposure.c + (1 + NAPc | Beach)
Data: RIKZ
```

REML criterion at convergence: 226.5

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.02001	-0.39890	-0.08148	0.22318	2.84754

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
Beach	(Intercept)	4.750	2.179	
	NAPc	3.568	1.889	-0.56
Residual		6.502	2.550	

Number of obs: 45, groups: Beach, 9

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	3.7263	1.1765	3.167
NAPc	-2.8084	0.7596	-3.697
exposure.cLow	3.7049	1.5177	2.441

Correlation of Fixed Effects:

	(Intr)	NAPc
NAPc		-0.309
exposur.cLw	-0.708	0.024

Predicted values lme4

Population level predictions:

```
head(predict(lmer.rc, re.form=~0))
```

1	2	3	4	5	6
8.281335	11.317239	12.159766	6.677726	10.328675	5.065691

Beach-level predictions:

```
head(predict(lmer.rc))
```

1	2	3	4	5	6
9.225663	11.734619	12.430907	7.900395	10.917642	7.230622

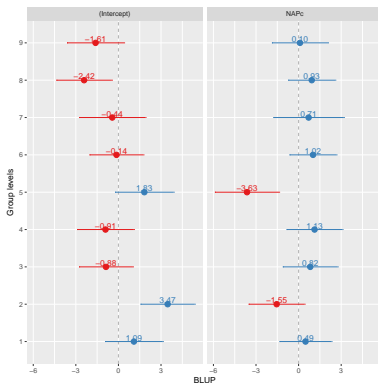
Beach-level predictions (more explicitly:

```
head(predict(lmer.rc, re.form=~(1+NAPc|Beach)))
```

1	2	3	4	5	6
9.225663	11.734619	12.430907	7.900395	10.917642	7.230622

Plots of the random effects

```
library(sjPlot)
plot_model(lmer.rc, type="re")
```



Diagnostics

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

What are our assumptions?

Diagnostics

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

What are our assumptions?

1. Linearity:

$$E[Richness|NAP, Exposure] = \beta_0 + \beta_1 NAP + \beta_p Exposure$$

2. Residuals are Normally distributed with constant variance:

$$\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

Diagnostics

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

What are our assumptions?

1. Linearity:

$$E[Richness|NAP, Exposure] = \beta_0 + \beta_1 NAP + \beta_p Exposure$$

2. Residuals are Normally distributed with constant variance:

$$\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

3. Beaches are independent

Diagnostics

Random Intercepts and Slopes Model:

$$R_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})NAP_{ij} + \beta_2 Exposure_i + \epsilon_{ij}$$
$$(b_{0i}, b_{1i}) \sim N(0, D)$$

What are our assumptions?

1. Linearity:

$$E[Richness|NAP, Exposure] = \beta_0 + \beta_1 NAP + \beta_2 Exposure$$

2. Residuals are Normally distributed with constant variance:

$$\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

3. Beaches are independent

4. $(b_{0i}, b_{1i}) \sim MVN(0, \Sigma)$, independent of ϵ_{ij}

Residual versus fitted values

Within-beach residuals (ϵ_{ij}) versus fitted values for each beach (\hat{R}_{ij})

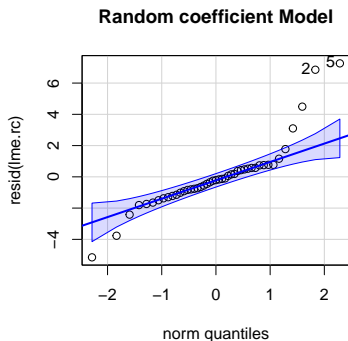
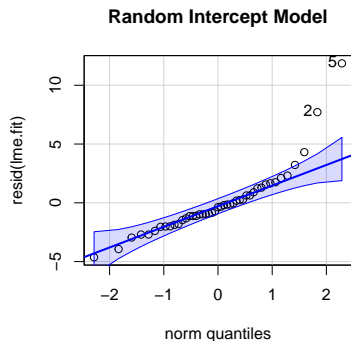
```
ri<-plot(lme.fit, main="Random Intercept Model")
rc<-plot(lme.rc, main="Random coefficient Model")
grid.arrange(ri, rc, ncol=2)
```



Diagnostics

Normality of within-beach errors (ϵ_{ij}):

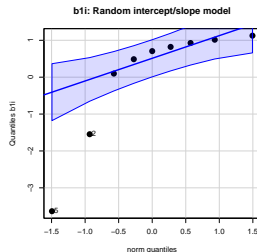
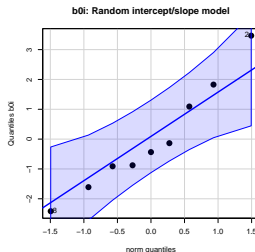
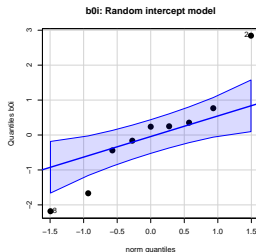
```
library(car)
par(mfrow=c(1,2))
qqPlot(resid(lme.fit), main="Random Intercept Model")
qqPlot(resid(lme.rc), main="Random coefficient Model")
```



Diagnostics

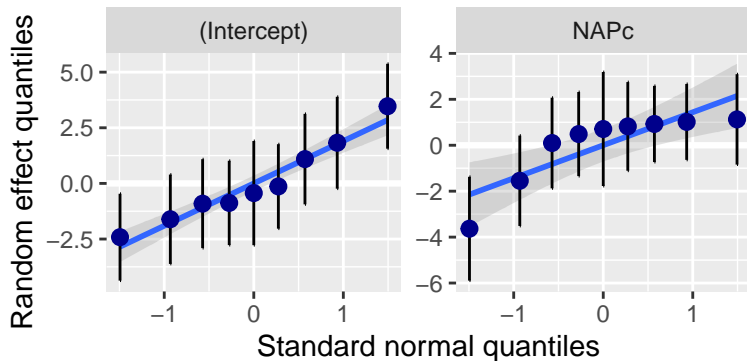
Normality of random effects (b_{0i} , b_{1i})

```
par(mfrow=c(1,3))
qqPlot(ranef(lme.fit)[,1], cex=1.8, pch=16, ylab="Quantiles b0i",
       main="b0i: Random intercept model") # random intercepts
qqPlot(ranef(lme.rc)[,1], cex=1.8, pch=16, ylab="Quantiles b0i",
       main="b0i: Random intercept/slope model") # random intercepts
qqPlot(ranef(lme.rc)[,2], cex=1.8, pch=16, ylab="Quantiles b1i",
       main="b1i: Random intercept/slope model") # random slopes
```



sjPlot library

```
p<- plot_model(lmer.rc, type="diag" ) # gives 4 plots
```



sjPlot library

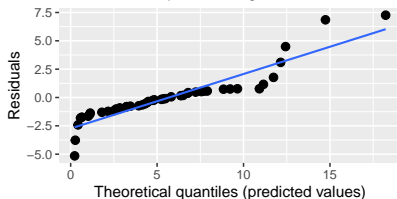
```
p<- plot_model(lmer.rc, type="diag" ) # gives 4 plots
grid.arrange(p[[1]], p[[3]], p[[4]], ncol=2)
```

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

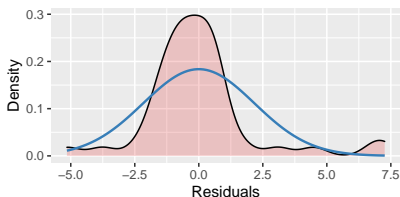
Non-normality of residuals and outliers

Dots should be plotted along the line



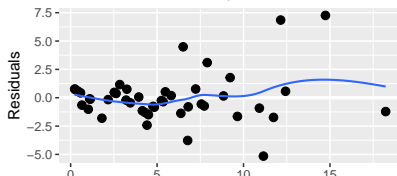
Non-normality of residuals

Distribution should look like normal curve



Homoscedasticity (constant variance of residuals)

Amount and distance of points scattered above/below line is equal or randomly spread



Mixed models

Models fit using lmer in lme4 package

Model Selection/Model Building Strategies

Comparing the 2 Models

```
AIC(lme.fit, lme.rc)
```

	df	AIC
lme.fit	5	240.5538
lme.rc	7	240.5327

This “test” is conservative (tends to overfit) since the variance parameter is “on the boundary” (same goes for Likelihood ratio tests)

See: <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#testing-significance-of-random-effects>

Comparing the 2 Models

```
AIC(lme.fit, lme.rc)
```

	df	AIC
lme.fit	5	240.5538
lme.rc	7	240.5327

This “test” is conservative (tends to overfit) since the variance parameter is “on the boundary” (same goes for Likelihood ratio tests)

See: <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#testing-significance-of-random-effects>

Number of parameters for calculating AIC also depends on focus (on individual subjects or population)

- See:
<http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#can-i-use-aic-for-mixed-models-how-do-i-count-the-number-of-degrees-of-freedom-for-a-random-effect>

Simulation-based testing

See `LectureMixedMods.Rmd` for an option, or have a look at the `RLRsim` or `pbkrtest` packages for simulation-based alternatives.

Simulation-based testing

See `LectureMixedMods.Rmd` for an option, or have a look at the `RLRsim` or `pbkrtest` packages for simulation-based alternatives.

For nested models, generate a null distribution for likelihood ratio test statistic = $-2(\text{LogL}(\text{model1}) - \text{LogL}(\text{model2}))$.

- Simulate data from the simpler model
- Fit both models to the simulated data
- Calculate the likelihood ratio statistic
- Repeat many times.

Simulation-based testing

See `LectureMixedMods.Rmd` for an option, or have a look at the `RLRsim` or `pbkrtest` packages for simulation-based alternatives.

For nested models, generate a null distribution for likelihood ratio test statistic = $-2(\text{LogL}(\text{model1}) - \text{LogL}(\text{model2}))$.

- Simulate data from the simpler model
- Fit both models to the simulated data
- Calculate the likelihood ratio statistic
- Repeat many times.

p-value = proportion of simulated observations that are as extreme, or more extreme than the likelihood ratio statistic calculated using the observed data.

REML versus ML

REML = Restricted Maximum Likelihood (usual default method)

REML versus ML

REML = Restricted Maximum Likelihood (usual default method)

- Variance components estimated using ML are biased high, REML nearly unbiased

REML versus ML

REML = Restricted Maximum Likelihood (usual default method)

- Variance components estimated using ML are biased high, REML nearly unbiased
- REML maximizes a modified form of the likelihood that depends on the fixed effects components

REML versus ML

REML = Restricted Maximum Likelihood (usual default method)

- Variance components estimated using ML are biased high, REML nearly unbiased
- REML maximizes a modified form of the likelihood that depends on the fixed effects components
- Comparisons of models with different fixed effects are not valid when using REML

REML versus ML

REML = Restricted Maximum Likelihood (usual default method)

- Variance components estimated using ML are biased high, REML nearly unbiased
- REML maximizes a modified form of the likelihood that depends on the fixed effects components
- Comparisons of models with different fixed effects are not valid when using REML

General Recommendation

- Determine random effects structure by comparing models fit using REML (all w/ the same fixed effects)
- Then, test fixed effects structure using models fit using ML (keeping random effects the same)

REML versus ML

REML = Restricted Maximum Likelihood (usual default method)

- Variance components estimated using ML are biased high, REML nearly unbiased
- REML maximizes a modified form of the likelihood that depends on the fixed effects components
- Comparisons of models with different fixed effects are not valid when using REML

General Recommendation

- Determine random effects structure by comparing models fit using REML (all w/ the same fixed effects)
- Then, test fixed effects structure using models fit using ML (keeping random effects the same)

For more, see:

- Zuur et al. 5.6

Zuur's Modeling Strategy

Fixed and random effects can “compete” to explain patterns in your response variable...

Zuur's Modeling Strategy

Fixed and random effects can “compete” to explain patterns in your response variable...

1. Start with as many covariates in the fixed component as possible

Zuur's Modeling Strategy

Fixed and random effects can “compete” to explain patterns in your response variable...

1. Start with as many covariates in the fixed component as possible
2. Compare models with different random effects structures (via AIC, LR tests). Use method = “REML” and keep fixed component constant.

Zuur's Modeling Strategy

Fixed and random effects can “compete” to explain patterns in your response variable...

1. Start with as many covariates in the fixed component as possible
2. Compare models with different random effects structures (via AIC, LR tests). Use method = “REML” and keep fixed component constant.
3. Compare fixed effects models (using AIC, LR tests) using the random structure from step [2]. Use method = “ML” and keep random component constant.

Zuur's Modeling Strategy

Fixed and random effects can “compete” to explain patterns in your response variable...

1. Start with as many covariates in the fixed component as possible
2. Compare models with different random effects structures (via AIC, LR tests). Use method = “REML” and keep fixed component constant.
3. Compare fixed effects models (using AIC, LR tests) using the random structure from step [2]. Use method = “ML” and keep random component constant.
4. Refit the ‘best’ model from step [4] using method = “REML”.

Zuur's Modeling Strategy

Fixed and random effects can “compete” to explain patterns in your response variable...

1. Start with as many covariates in the fixed component as possible
2. Compare models with different random effects structures (via AIC, LR tests). Use method = “REML” and keep fixed component constant.
3. Compare fixed effects models (using AIC, LR tests) using the random structure from step [2]. Use method = “ML” and keep random component constant.
4. Refit the ‘best’ model from step [4] using method = “REML”.
5. Look at diagnostic plots, and modify model as needed

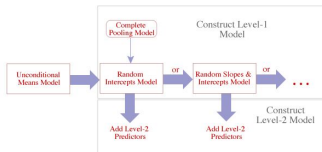


Fig. 1 A general flowchart for fitting multilevel models

Jack Weiss suggests fitting a series of models:

- Pooled model (assuming independence), include level-1 predictors [predictors that vary within clusters] $\text{lm}(y \sim x1)$
- Unconditional means model or variance components model (no predictors, just random intercepts) $\text{lmer}(y \sim 1 + (1|\text{site}))$
- Random intercepts (with level 1 predictors) $\text{lmer}(y \sim x1 + (1|\text{site}))$
- Random intercepts and slopes (with level 1 predictors) $\text{lmer}(y \sim x1 + (1 + x1|\text{site}))$

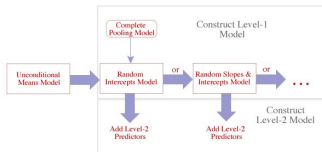


Fig. 1 A general flowchart for fitting multilevel models

Jack Weiss suggests fitting a series of models:

- Pooled model (assuming independence), include level-1 predictors [predictors that vary within clusters] $\text{lme}(y \sim x1)$
- Unconditional means model or variance components model (no predictors, just random intercepts) $\text{lmer}(y \sim 1 + (1|\text{site}))$
- Random intercepts (with level 1 predictors) $\text{lmer}(y \sim x1 + (1|\text{site}))$
- Random intercepts and slopes (with level 1 predictors) $\text{lmer}(y \sim x1 + (1 + x1|\text{site}))$

Pick the best of these, then add level-2 predictors (predictors that are constant within clusters).

Strategy outlined by: Singer, J. D. and Willett, J. B. (2003) *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. (Oxford University Press, Oxford, UK).

Other

Attempt to make inference from a maximal model:

- Include all random slopes that you can for level 1 predictors
- Simplify as needed when encountering convergence problems.

Other

Attempt to make inference from a maximal model:

- Include all random slopes that you can for level 1 predictors
- Simplify as needed when encountering convergence problems.

Lots of debate on how best to approach model building/selection.

Random slopes model will be more conservative

Remember Schielzeth and Forstmeier (2009) suggest random slopes are needed for level-1 predictors (SE increases - see below):

```
# Random intercept  
summary(lme.fit)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.170680	1.1739988	35	2.700752	1.058924e-02
NAPc	-2.581708	0.4883901	35	-5.286160	6.745464e-06
exposure.cLow	4.532777	1.5755612	7	2.876928	2.375560e-02

```
# Random intercept and slope  
summary(lme.rc)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.726341	1.1765068	35	3.167292	0.0031851946
NAPc	-2.808422	0.7596419	35	-3.697035	0.0007425901
exposure.cLow	3.704915	1.5176687	7	2.441188	0.0446809964

Marginal Distribution

$$Y_i = X_i\beta + z_ib + \epsilon_i$$

$$\epsilon_i \sim N(0, \Sigma_i)$$

$$b \sim N(0, D)$$

$$Y_i|b \sim N(X_i\beta + Z_ib, \Sigma_i)$$

Marginal Distribution

$$\begin{aligned}Y_i &= X_i\beta + z_ib + \epsilon_i \\ \epsilon_i &\sim N(0, \Sigma_i) \\ b &\sim N(0, D)\end{aligned}$$

$$Y_i|b \sim N(X_i\beta + Z_ib, \Sigma_i)$$

If we average over (or integrate out) the random effects, we get the **marginal Distribution of Y** .

$$Y_i \sim N(X_i\beta, V_i), V_i = Z_i D Z_i' + \Sigma_i$$

This is actually what R uses to fit the data.

Marginal model is what R is fitting

For random intercepts model:

$$Y_i \sim N(X_i\beta, V_i)$$
$$V_i = \begin{bmatrix} \sigma^2 & \rho & \cdots & \rho \\ \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & \sigma^2 \end{bmatrix} \quad \rho = \frac{\tau^2}{\tau^2 + \sigma^2}$$

Marginal model is what R is fitting

For random intercepts model:

$$Y_i \sim N(X_i\beta, V_i)$$

$$V_i = \begin{bmatrix} \sigma^2 & \rho & \cdots & \rho \\ \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & \sigma^2 \end{bmatrix} \quad \rho = \frac{\tau^2}{\tau^2 + \sigma^2}$$

$$\text{Var/Cov matrix for } Y \text{ (all data)} = \begin{bmatrix} V_i & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & V_i \end{bmatrix}$$

Fitting the marginal model using `gls`

We might have posited this model directly.

Fitting the marginal model using `gls`

We might have posited this model directly.

We can fit it using the `gls` function in the `nlme` library

The `gls` function also allows for:

- A variety of assumptions for capturing within-subject correlation
 - `ar(1)` time series
 - Spatial covariance

Fitting the marginal model using `gls`

We might have posited this model directly.

We can fit it using the `gls` function in the `nlme` library

The `gls` function also allows for:

- A variety of assumptions for capturing within-subject correlation
 - `ar(1)` time series
 - Spatial covariance
- Methods for modeling heterogeneous variance
 - can allow the variance to differ by group (e.g., by exposure level)
 - can allow the variance to depend on a continuous predictor, x (e.g., $\text{var} = \sigma^2 x^{2\theta}$)

Fitting the marginal model using `gls`

We might have posited this model directly.

We can fit it using the `gls` function in the `nlme` library

The `gls` function also allows for:

- A variety of assumptions for capturing within-subject correlation
 - `ar(1)` time series
 - Spatial covariance
- Methods for modeling heterogeneous variance
 - can allow the variance to differ by group (e.g., by exposure level)
 - can allow the variance to depend on a continuous predictor, x (e.g., $\text{var} = \sigma^2 x^{2\theta}$)

See Ch 4 Zuur et al. and the section of the course on `gls` models.

Marginal Model fit using gls

```
gls.fit<-gls(Richness~NAPc+exposure.c, method="REML",  
            correlation=corCompSymm(form=~1|Beach),  
            data=RIKZ)  
summary(gls.fit)
```

Generalized least squares fit by REML

Model: Richness ~ NAPc + exposure.c

Data: RIKZ

	AIC	BIC	logLik
	240.5538	249.2422	-115.2769

Correlation Structure: Compound symmetry

Formula: ~1 | Beach

Parameter estimate(s):

	Rho
	0.2798938

Coefficients:

	Value	Std.Error	t-value	p-value
(Intercept)	3.170680	1.1739987	2.700752	0.0099
NAPc	-2.581708	0.4883901	-5.286160	0.0000
exposure.cLow	4.532777	1.5755610	2.876929	0.0063

Correlation:

	(Intr)	NAPc
NAPc	-0.028	
exposure.cLow	-0.746	0.037

Standardized residuals:

	Min	Q1	Med	Q3	Max
	-1.5551728	-0.6415409	-0.1554932	0.4150315	3.3566242

Residual standard error: 3.604905

Degrees of freedom: 45 total; 42 residual

```
tab_model(gls.fit, lme.fit, show.r2 = FALSE)
```

<i>Predictors</i>	Richness			Richness		
	<i>Estimates</i>	<i>CI</i>	<i>p</i>	<i>Estimates</i>	<i>CI</i>	<i>p</i>
(Intercept)	3.17	0.87 – 5.47	0.010	3.17	0.79 – 5.55	0.011
NAPc	-2.58	-3.54 – -1.62	<0.001	-2.58	-3.57 – -1.59	<0.001
exposure.c [Low]	4.53	1.44 – 7.62	0.006	4.53	0.81 – 8.26	0.024
N	⁹ Beach					
Observations	45					
		45				