# Data Science-Ish: The Use of Mixed Effects (or Multi-Level) Models for Analyzing Complex Data Sources

Joshua Rosenberg
2019-06-20 (updated: 2019-06-21)

# #whoami

- Joshua Rosenberg, Ph.D.
- Assistant Professor, STEM Education, University of Tennessee, Knoxville
- Dad (16 month toddler!)
- Primary areas of interest:
  - Data science in education
  - Data science education

## Following along with the presentation

- Code: https://github.com/jrosen48/data-science-ish
- Presentation: https://jrosen48.github.io/data-science-ish

# Background/Motivation

# Rationale and Aim

- Mixed effects (or multilevel) models are extensions of linear models (regressions)
- Commonly used in experimental and observational research and policy and evaluation contexts
- Are less widely-used by data scientists
- One piece of evidence: Cross Validated has 1,392 questions tagged #mixed-model that are unanswere (of 3,337 total questions)
- I am trying to introduce mixed effects model as a data science(-*ish*) statistical method that can be useful and is easy to estimate and interpret.

# Example: Rail-trails!

## Springfield Greenway

*Image from kz440gal on TraiLink.com*

# The Rails-to-Trails Conservancy

- Rail trails are railroad tracks that have been converted into pathways/greenways
- The Rails-to-Trails Conservancy is an organization "dedicated to creating a nationwide network of trails from former rail lines and connecting corridors to build healthier places for healthier people"
- TrailLink is their website that lists most of the rail-trails in the United States (~4,000 total)
- Let's take a look at their website

# TrailLink.com

- Which rail-trail is best?
- In one state
- Across the nation
- This can be a (surprisingly) difficult question to answer
- View trails in TN here: https://www.traillink.com/trailsearch/?mmloc=tn

**What do you think? Which is (or, which are) the best?**

Discuss your answer with an elbow partner!

# Building up to Mixed Effects Models

# One data-based solution

## A regression model!

$$\mathrm{y} = \alpha + \beta_1(\mathrm{var1}) + \beta_2(\mathrm{var2}) + \epsilon$$

# Accessing rail-trails data

- Use the `railtrails` R package
- Install via `install.packages('railtrails')`)

```
library(railtrails)

railtrails
```

```
## # A tibble: 3,846 x 11
##    state name  distance surface category mean_review description n_reviews
##    <chr> <chr>    <dbl> <chr>   <chr>          <int> <chr>           <int>
##  1 AK    Chas…     14   Dirt, … Rail-Tr…           4 "Though cl…         1
##  2 AK    Tony…     11   Asphalt Rail-Tr…           5 "The Tony …         5
##  3 AK    Bird…     13   Asphalt Rail-Tr…           5 "The Bird …         3
##  4 AK    Camp…      7.5 Asphal… Greenwa…           5 "The Campb…         3
##  5 AK    Goos…      1.5 Asphal… Greenwa…           0 "The sceni…         0
##  6 AK    Home…      4   Asphalt Greenwa…           5 "On the so…         1
##  7 AK    Lani…      3.9 Asphal… Greenwa…           3 "The Lanie…         1
##  8 AK    Palm…      6.1 Gravel  Rail-Tr…           0 "As its na…         0
##  9 AK    Ship…      2.6 Asphalt Rail-Tr…           4 "Ship Cree…         1
## 10 AL    Chie…     33   Asphalt Rail-Tr…           5 "In northe…        77
## # … with 3,836 more rows, and 3 more variables: raw_reviews <list>,
## #   lat <dbl>, lng <dbl>
```

# Let's focus in on Tennessee

```
library(dplyr)

d ← railtrails %>%
  filter(state == "TN")

d
```

```
## # A tibble: 55 x 11
##    state name  distance surface category mean_review description n_reviews
##    <chr> <chr>    <dbl> <chr>   <chr>           <int> <chr>           <int>
##  1 TN    Tenn…     4.73 Asphal… Rail-Tr…            5 "The Tenne…         9
##  2 TN    Bets…     2    Asphalt Rail-Tr…            0 "The Betsy…         0
##  3 TN    Alta…     0.7  Asphal… Greenwa…            0 "Alta Lake…         0
##  4 TN    Bear…     2.7  Asphalt Greenwa…            0 "From Thir…         0
##  5 TN    Big …     1    Concre… Rail-Tr…            5 "Big River…         2
##  6 TN    Bria…     1.1  Asphalt Rail-Tr…            0 "The Brian…         0
##  7 TN    Broo…     0.4  Asphalt Greenwa…            3 "Brookmead…         5
##  8 TN    Cave…     0.95 Asphalt Greenwa…            0 "Cavet Sta…         0
##  9 TN    Chat…    13.1  Concre… Greenwa…            5 "The Chatt…        17
## 10 TN    Clar…     4.6  Asphalt Rail-Tr…            5 "The 4.6-m…        21
## # … with 45 more rows, and 3 more variables: raw_reviews <list>,
## #   lat <dbl>, lng <dbl>
```

# Let's fit a linear model

## Unnesting the reviews

```
d ← d %>%
  unnest(raw_reviews) %>%
  rename(raw_review = raw_reviews)
```

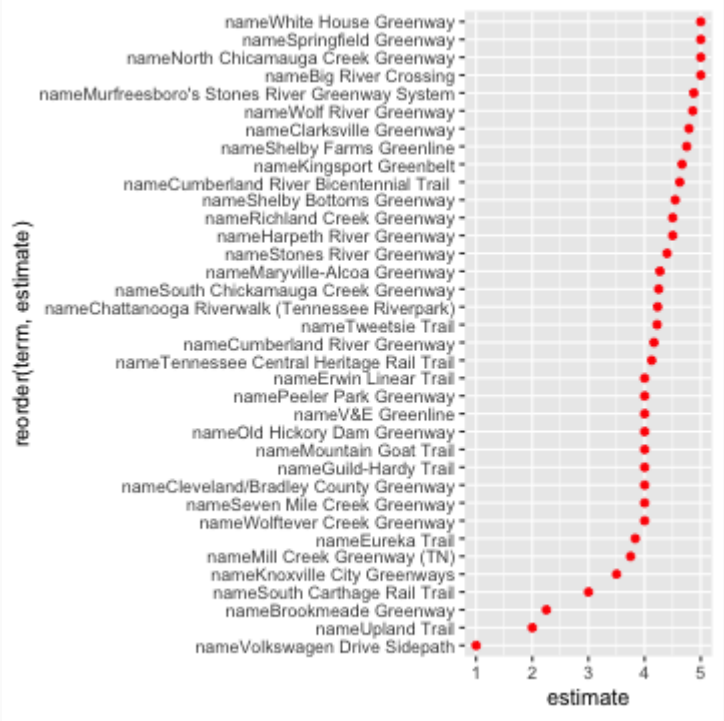
- `lm()` will automatically recognize that `name` is a categorical variable

```
m1 ← lm(raw_review ~ -1 + name, data = d)
```

- The output is difficult to read (because of the number of trails/`name`s)!

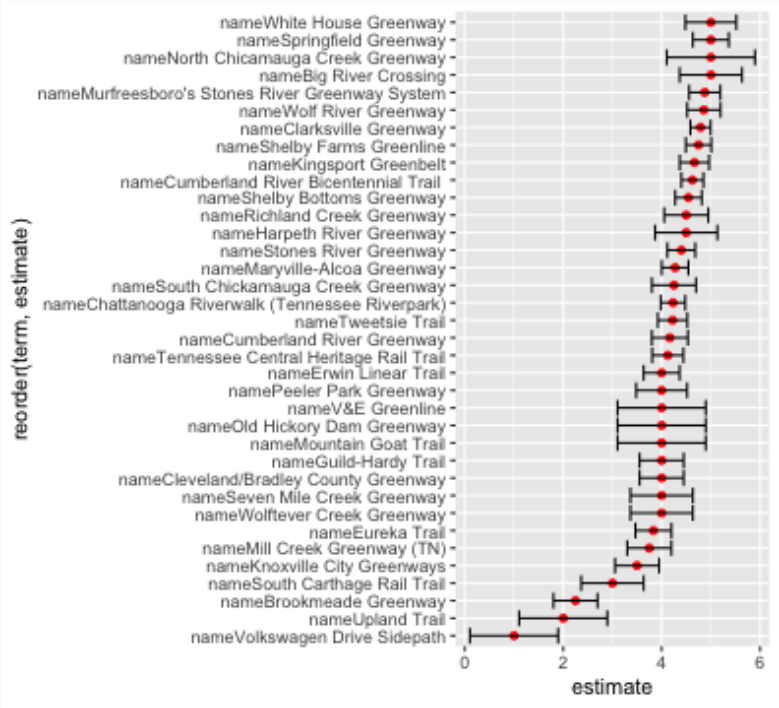
# Plotting the results

```
library(ggplot2)
library(broom)
tidy(m1) %>%
  ggplot(aes(x = reorder(term, estimate), y = estimate)) +
  geom_point(color = "red") +
  coord_flip()
```



# Plotting the results with SEs

```
tidy(m1) %>%
  ggplot(aes(x = reorder(term, estimate), y = estimate)) +
  geom_point(color = "red") +
  geom_errorbar(aes(ymin = estimate - std.error,
                    ymax = estimate + std.error)) +
  coord_flip()
```




**Which trail is best?**

- White House Greenway?
- Springfield Greenway?
- North Chicamauga Creek Greenway?

# Interpreting the output

- Gelman and Hill (2007) consider this model to be the model with *no pooling*:
- The estimates for the trail means are not changed based on 'pooling' together the estimates (or in any other way)
- The problem is that if we only consider the mean, then we may be ignoring how uncertain the estimates are
- Ignoring this uncertainity may mean that the estimates are not good: they may be biased

$$
\begin{aligned}
\mathbf{raw\_review} = \beta_0(\mathbf{name}_{\text{Big River Crossing}}) + \\
\beta_1(\mathbf{name}_{\text{Brookmeade Greenway}}) + \\
\beta_2(\mathbf{name}_{\text{Chattanooga Riverwalk (Tennessee Riverpark)}}) + \\
\beta_3(\mathbf{name}_{\text{Clarksville Greenway}}) + \\
\ldots \quad \beta_{35}(\mathbf{name}_{\text{Wolftever Creek Greenway}}) + \\
\epsilon
\end{aligned}
$$

# Other approaches

## What if we ignored the trail completely?

```
m2 ← lm(raw_review ~ 1, data = d)

m2
```

```
##
## Call:
## lm(formula = raw_review ~ 1, data = d)
##
## Coefficients:
## (Intercept)
##       4.343
```

Which trail is best?

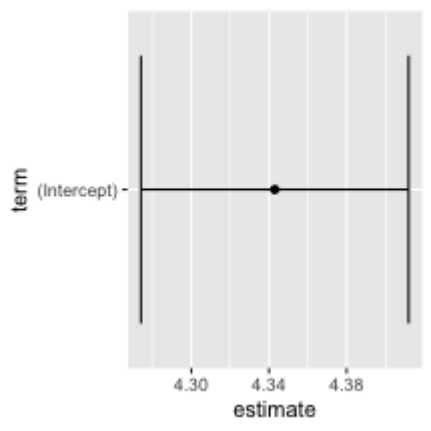
¯_(ツ)_/¯

# Interpreting the output

- This can be considered a model with *complete pooling*
- The estimates are affected by pooling them together

The model is pretty simple: Not sure we need to plot this...

```
tidy(m2) %>%
  ggplot(aes(x = term, y = estimate)) +
  geom_errorbar(aes(ymin = estimate - st
                    ymax = estimate + st
  geom_point() +
  coord_flip()
```




```
extract_eq(m2, use_coef = TRUE)
```

$$\text{raw\_review} = 4.34 + \epsilon$$

# The mixed effects model approach

- Models the reviews as if they are 'grouped' within trails
- The trail (name) is considered to be a 'grouping factor'
- (also known as a 'random effect')

```
library(lme4) # the most common R package for mixed effects models; see also nlme

m3 ← lmer(raw_review ~ 1 +
            (1|name),
          data = d)
```

This model can be written as an extension of a linear model, where *j* indexes the different trails.:

$$ {y}=\alpha *{ j }+\epsilon +\ \alpha* { j }=\alpha _{ 00 }+\sigma_j $$

# Checking out the output

```
m3
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: raw_review ~ 1 + (1 | name)
##    Data: d
## REML criterion at convergence: 579.7115
## Random effects:
##  Groups   Name        Std.Dev.
##  name     (Intercept) 0.4236
##  Residual             0.9184
## Number of obs: 207, groups:  name, 36
## Fixed Effects:
## (Intercept)
##       4.234
```

# Let's create a plot of the output

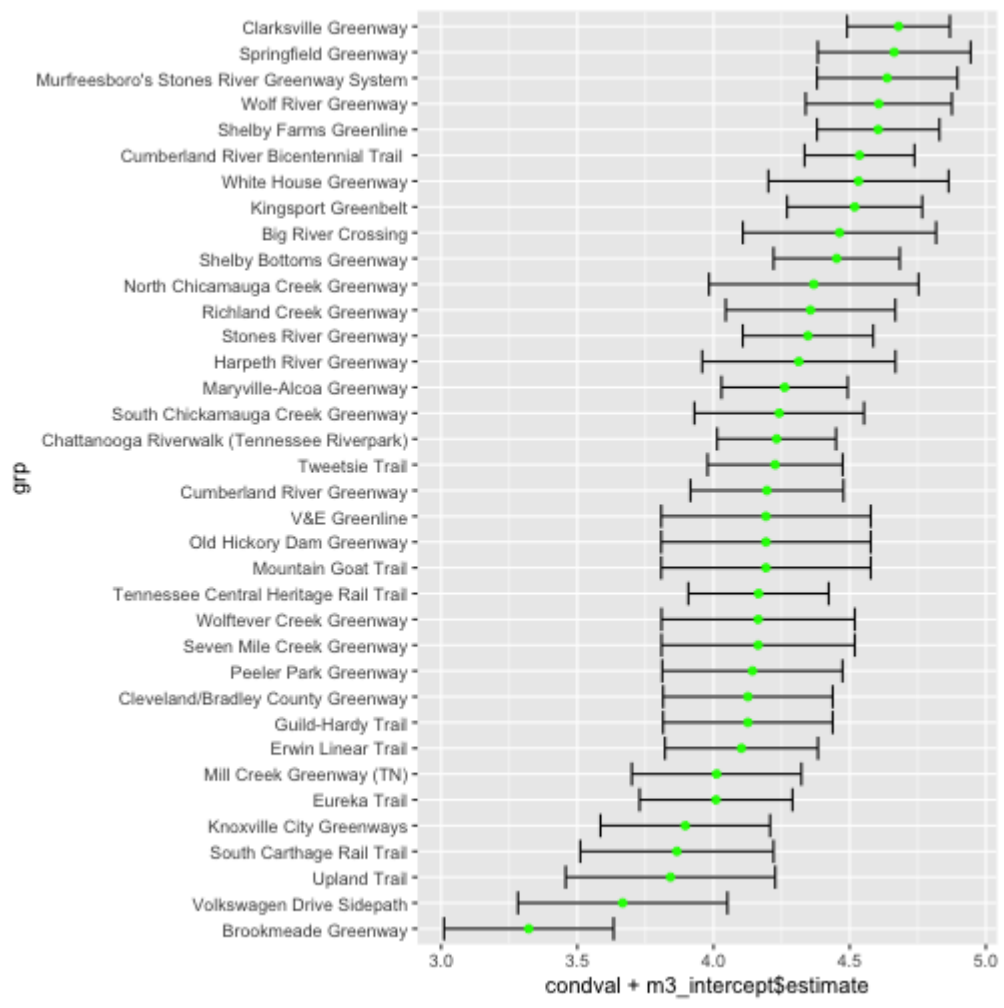
```
m3_intercept ← tidy(m3) %>% filter(term == "(Intercept)")

m3_ranef ← ranef(m3) %>% as_tibble()

p3 ← m3_ranef %>%
  ggplot(aes(x = grp, y = condval + m3_intercept$estimate)) +
  geom_errorbar(aes(ymin = (condval + m3_intercept$estimate) - condsd,
                    ymax = (condval + m3_intercept$estimate) + condsd)) +
  geom_point(color = "green") +
  coord_flip()
```

# Looking at the plot

```
p3
```



# Interpreting the output

## Which trail is best?

- Clarksville Greenway?
- Springfield Greenway?
- Murfeesboro's Stones River Greenway System?

## What is the model 'like'?

- This model represents the *partial pooling*
- Grouping factors/random effects:
- A *variable* with coefficient estimates that differ by group and are modeled with a probability distribution
- The estimate for the coefficient depends on *how different* the observations associated with the group are and *how systematic* the differences are (relative to all of the observations)
    - The largest coefficient estimates will be associated with groups that are systematically different
    - The smallest coefficient estimates will be associated with groups that are either not very different or are not systematically so

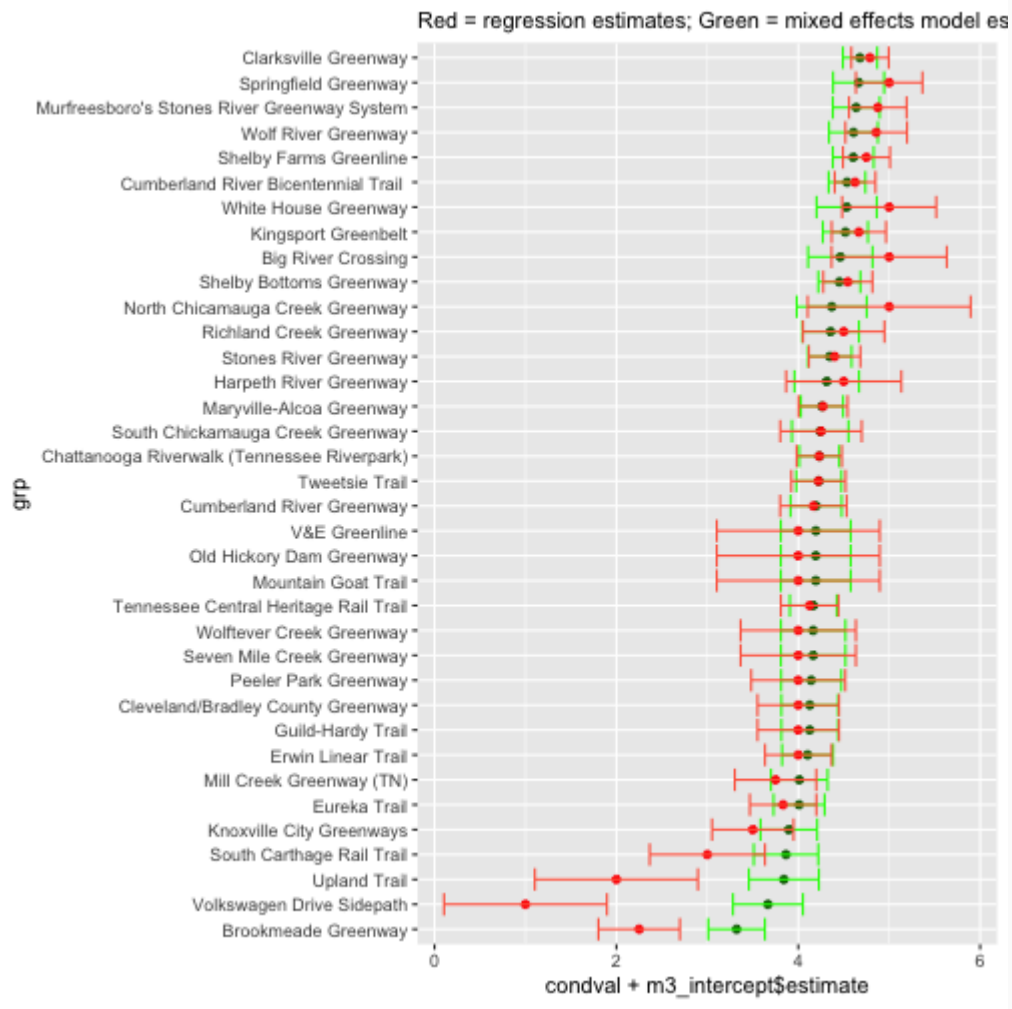
# Let's create a graph to compare

```
dt ← tidy(m1) %>% mutate(term = str_replace(term, "name", ""))

pc ← m3_ranef %>%
  ggplot(aes(x = grp, y = condval + m3_intercept$estimate)) +
  geom_point(color = "darkgreen") +
  geom_errorbar(aes(ymin = (condval + m3_intercept$estimate) - condsd,
                    ymax = (condval + m3_intercept$estimate) + condsd),
                color = 'green') +
  geom_point(data = dt, aes(x = reorder(term, estimate), y = estimate), color = "red",
  geom_errorbar(data = dt, aes(x = reorder(term, estimate),
                               ymin = estimate - std.error,
                               ymax = estimate + std.error), color = 'tomato', inherit
  coord_flip() +
  labs(subtitle = "Red = regression estimates; Green = mixed effects model estimates")
```

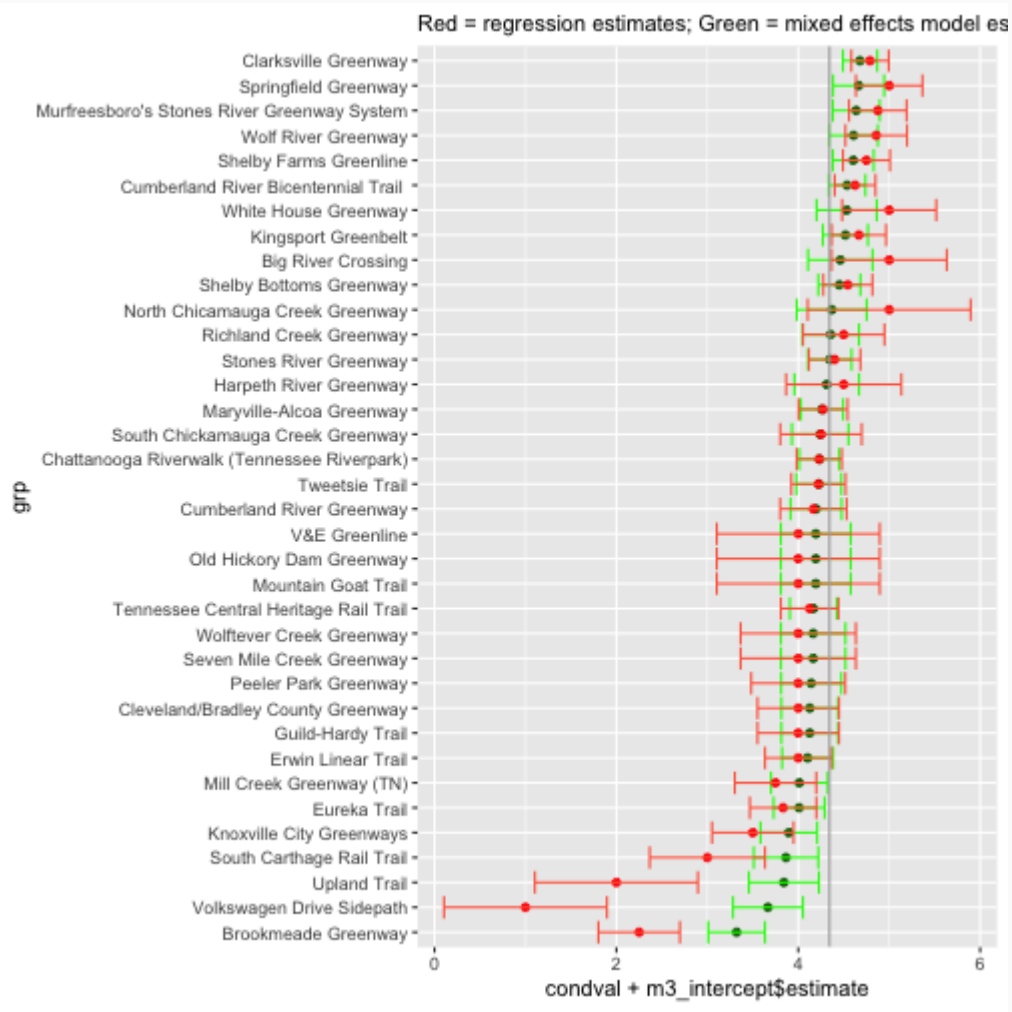
# Let's plot the comparisons

```
pc
```



# Comparing models with no pooling est.

```
pc + geom_hline(yintercept = tidy(m2) %>% pull(estimate), color = "darkgray")
```



# So, which trail is best?

## Candidate trails from the regression

- White House Greenway
- Springfield Greenway
- North Chicamauga Creek Greenway

## Candidate trails from the mixed effects model

- Clarksville Greenway
- Springfield Greenway
- Murfeesboro's Stones River Greenway System

- The estimates for the groups from the regression tend to me more confident
  - They exhibit *no pooling*
- The estimates for the groups from the mixed effects model are nearer to the mean
  - They exhibit *partial pooling*
  - Especially for trails with few observations (i.e., North Chicamauga Creek and White House)
- What about the *complete pooling* estimates?
  - Not relevant but is often (implicitly) used: Other estimates can be biased

# Extending the mixed effects model

# Adding another grouping factor

- We used a 'varying intercepts' model
- We can add more grouping factors/random effects

```
railtrails ← railtrails %>%
  unnest(raw_reviews) %>%
  rename(raw_review = raw_reviews)

m4 ← lmer(raw_review ~ 1 +
            (1|name) +
            (1|state),
          data = railtrails)
```

# Inspecting the results

```
m4
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: raw_review ~ 1 + (1 | name) + (1 | state)
##    Data: railtrails
## REML criterion at convergence: 44701.29
## Random effects:
##  Groups   Name        Std.Dev.
##  name     (Intercept) 0.5529
##  state    (Intercept) 0.0873
##  Residual             0.9454
## Number of obs: 15562, groups:  name, 2539; state, 51
## Fixed Effects:
## (Intercept)
##       4.017
```

# Examining the ICC

Is there systematic variation at the state level?

The Intra-class Correlation (ICC) is an estimate for the relationship between higher or lower grouping factor/random effect estimates and the outcome (*y*-variable).

```
library(performance)
icc(m4)
```

```
## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.260
##   Conditional ICC: 0.260
```

# Adding a predictor variable - distance

```
m5 ← lmer(raw_review ~ 1 +
             scale(distance) +
             (1|name) +
             (1|state),
           data = railtrails)

m5
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: raw_review ~ 1 + scale(distance) + (1 | name) + (1 | state)
##    Data: railtrails
## REML criterion at convergence: 44696.01
## Random effects:
##  Groups   Name        Std.Dev.
##  name     (Intercept) 0.55044
##  state    (Intercept) 0.08726
##  Residual             0.94551
## Number of obs: 15562, groups:  name, 2539; state, 51
## Fixed Effects:
##     (Intercept)  scale(distance)
##         4.04056          0.08019
```

# Preparing other variables - surface

```
library(forcats)
library(stringr)

railtrails ← mutate(railtrails,
                    surface_rc = case_when(
                      surface == "Asphalt" ~ "Paved",
                      surface == "Asphalt, Concrete" ~ "Paved",
                      surface == "Concrete" ~ "Paved",
                      surface == "Asphalt, Boardwalk" ~ "Paved",
                      str_detect(surface, "Stone") ~ "Crushed Stone",
                      str_detect(surface, "Ballast") ~ "Crushed Stone",
                      str_detect(surface, "Gravel") ~ "Crushed Stone",
                      TRUE ~ "Other"
                    )
)
```

# Adding a predictor variable - surface

```
m6 ← lmer(raw_review ~ 1 +
             scale(distance) +
             surface_rc +
             (1|name) +
             (1|state),
           data = railtrails)

m6
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: raw_review ~ 1 + scale(distance) + surface_rc + (1 | name) +
##     (1 | state)
##    Data: railtrails
## REML criterion at convergence: 44671.95
## Random effects:
##  Groups   Name        Std.Dev.
##  name     (Intercept) 0.54193
##  state    (Intercept) 0.08014
##  Residual             0.94593
## Number of obs: 15562, groups:  name, 2539; state, 51
## Fixed Effects:
##     (Intercept)  scale(distance)  surface_rcOther  surface_rcPaved
##         3.93331          0.11029          0.03028          0.19065
```

# Connection to Bayesian Methods

# (A very tiny introduction to) Bayesian

- Bayesian methods can be distinguished from frequentist methods in two key ways:

    1. Describing estimates with *probability* statements/distributions
    2. Using *prior* information in the estimation process

- How is this different from the models we estimated?

    1. We focused on point estimates (and their standard errors/deviations)
    2. Partial pooling can be considered an instance of using prior information for the grouping factor/random effects estimates

- Also, the estimation process is *usually* different, but this is not a distinguishing feature

# Estimating a model with brms

```
library(brms)
m7 ← brm(raw_review ~ 1 +
           scale(distance) +
           surface_rc +
           (1|name) +
           (1|state),
         iter = 1000, chains = 3, cores = 3,
         data = railtrails)
```

```
## Warning: Rows containing NAs were excluded from the model.
```

# Taking a peak at the model output

```
m7
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: raw_review ~ 1 + scale(distance) + surface_rc + (1 | name) + (1 | state)
##    Data: railtrails (Number of observations: 15562)
## Samples: 3 chains, each with iter = 1000; warmup = 500; thin = 1;
##          total post-warmup samples = 1500
##
## Group-Level Effects:
## ~name (Number of levels: 2539)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.54      0.02     0.51     0.57        494 1.00
##
## ~state (Number of levels: 51)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.08      0.03     0.01     0.13        129 1.03
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept           3.93      0.03     3.88     3.99        592 1.00
## scaledistance       0.11      0.02     0.06     0.16        392 1.00
## surface_rcOther     0.03      0.07    -0.11     0.16        804 1.00
```

# Other Extensions

# Modeling complex data structures

- Additional grouping factors/random effects can easily be added
- Cross-classified grouping factors/random effects can be useful (e.g., when you have time-dependent and location-specific data)
- Can add an arbitrary number of grouping factors
- Random slopes allow for predictor variables to be modeled with slopes that differ based on the group (e.g., if there were a variable for "the weather on the day the trail was reviewed" that could have a different effect depending on the trail)
- Can easily be used to estimate non-linear models (i.e., models with dichotomous or count outcomes)
- Can estimate multivariate models (with brms)

# Wrapping up

# Summary and Resources

- Mixed effects models are extensions of linear models that allow for complex data
- They are easy to estimate and to interpret (and to compare to simpler models)
- Their extensions make them useful for a range of problems
- They are actively being developed and there is an active community of users to provide support
- Resources:
  - Gelman and Hill (2006)
  - West, Welch, and Galecki (2014)
  - Kruschke (2014)

# Get in touch!

- Joshua Rosenberg
- GitHub: https://github.com/jrosen48
- Twitter: @jrosenberg6432
- Code: https://github.com/jrosen48/data-science-ish
- Presentation: https://jrosen48.github.io/data-science-ish

Thank you