Hierarchical and Mixed Effects Models in R

Marcus Becker

February 15, 2020

Chapter 1: Overview and Introduction to Hierarchical and Mixed Models

- lme4 package

```r
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(lmerTest)
```

```
## Warning: package 'lmerTest' was built under R version 3.6.3
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```r
library(broom.mixed)
```

```
## Warning: package 'broom.mixed' was built under R version 3.6.3
```

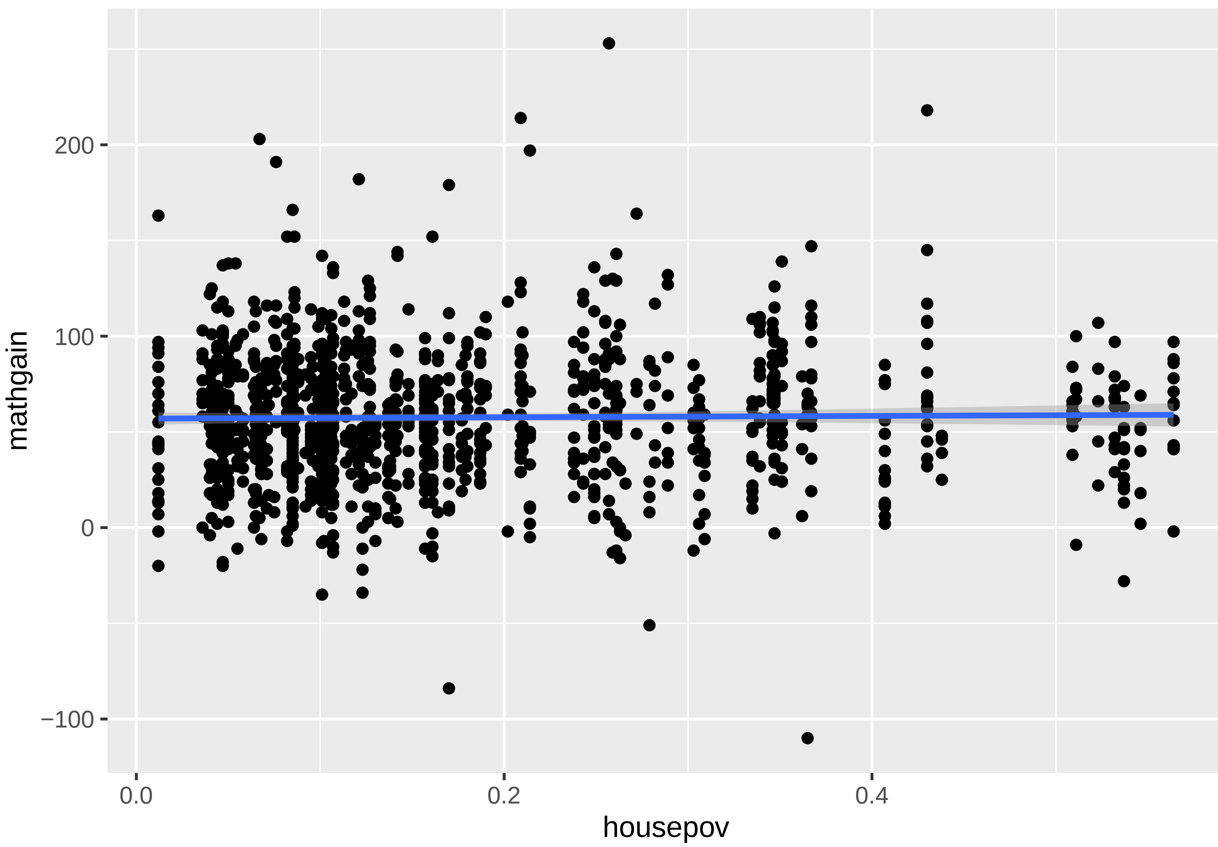
```
## Registered S3 methods overwritten by 'broom.mixed':
##   method         from
##   augment.lme    broom
##   augment.merMod broom
##   glance.lme     broom
##   glance.merMod  broom
##   glance.stanreg broom
##   tidy.brmsfit   broom
##   tidy.gamlss    broom
##   tidy.lme       broom
##   tidy.merMod    broom
```

```
##   tidy.rjags     broom
##   tidy.stanfit   broom
##   tidy.stanreg   broom
```

```r
studentData <- read_csv("https://assets.datacamp.com/production/repositories/1803/datasets/975fe2b01908
```

```
## Parsed with column specification:
## cols(
##   sex = col_double(),
##   minority = col_double(),
##   mathkind = col_double(),
##   mathgain = col_double(),
##   ses = col_double(),
##   yearstea = col_double(),
##   mathknow = col_double(),
##   housepov = col_double(),
##   mathprep = col_double(),
##   classid = col_double(),
##   schoolid = col_double(),
##   childid = col_double()
## )
```

```r
# Visualize the data first
ggplot(data = studentData, aes(x = housepov, y = mathgain)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE)
```

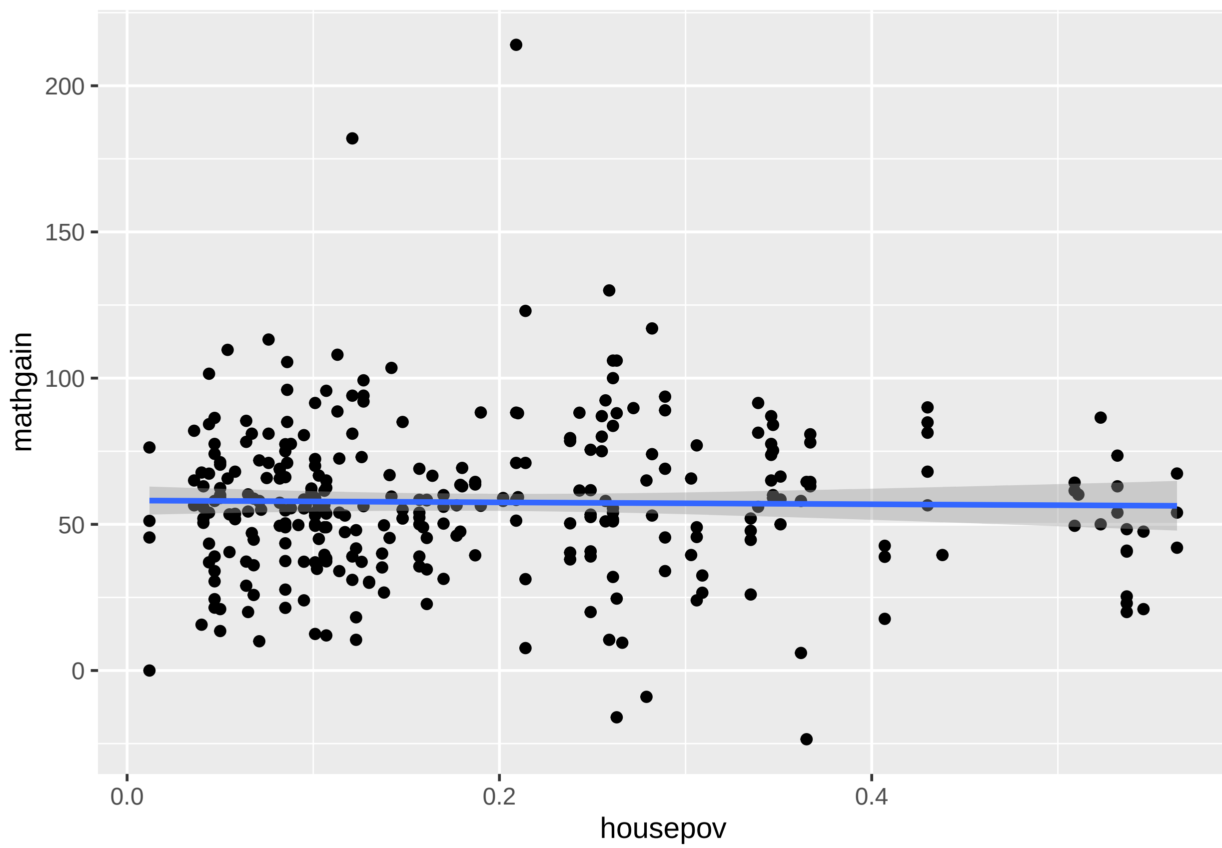


```r
# Fit a linear model
summary(lm(formula = mathgain ~ housepov, data = studentData)) # Not predictive. We haven't accounted f
```

```
##
## Call:
## lm(formula = mathgain ~ housepov, data = studentData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -168.226  -22.222   -1.306   19.763  195.156
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   56.937      1.674   34.02   <2e-16 ***
## housepov       3.531      7.515    0.47    0.639
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.63 on 1188 degrees of freedom
## Multiple R-squared:  0.0001858,  Adjusted R-squared:  -0.0006558
## F-statistic: 0.2208 on 1 and 1188 DF,  p-value: 0.6385
```

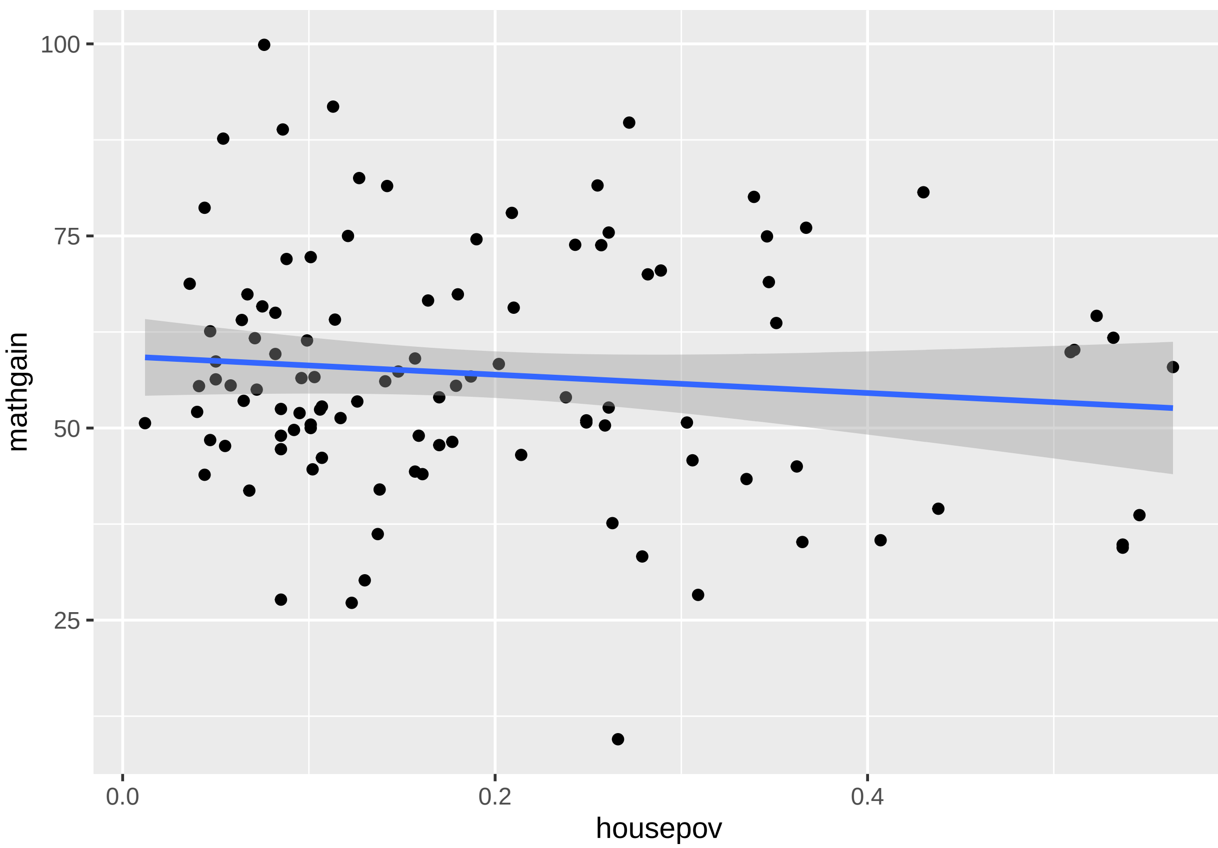
```r
# Class
classData <- studentData %>%
  group_by(classid) %>%
  summarise(mathgain = mean(mathgain),
            housepov = mean(housepov))

ggplot(data = classData, aes(x = housepov, y = mathgain)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE)
```




```r
# School
schoolData <- studentData %>%
  group_by(schoolid) %>%
  summarise(mathgain = mean(mathgain),
            housepov = mean(housepov))

ggplot(data = schoolData, aes(x = housepov, y = mathgain)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE)
```



Multiple regression in R tips:

- lm(y ~ x -1) - the -1 estimates an intercept for each x (group), rather than relative to the first group.
- numeric vs factors - R automatically assumes a numeric predictor is a slope
- scaling parameters and slopes
- shortcut for interaction is x1*x2

Without other coefficients, a single intercept is the global mean of the data. Multiple intercepts allow you to estimate the mean for each group as long as other coefficients are not estimated (when the groups are discrete…).

What about continuous predictor variables? -> slopes.

```
# Mixed effects model (i.e. has both fixed and random effects)
me_mod <- lmer(formula = mathgain ~ sex + mathprep + mathknow + (1|classid) + (1|schoolid),
               data = studentData,
               na.action = "na.omit",
               REML = TRUE)

summary(me_mod)
```

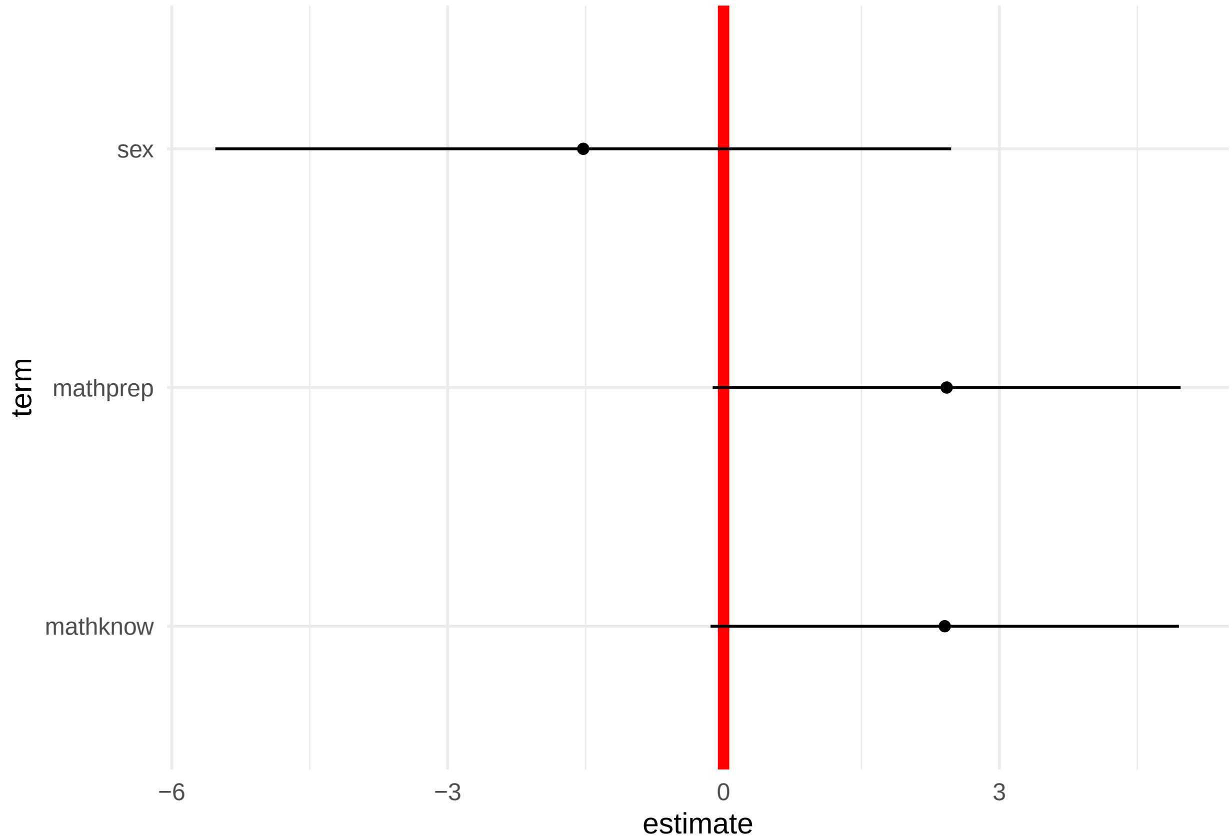
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mathgain ~ sex + mathprep + mathknow + (1 | classid) + (1 | schoolid)
##    Data: studentData
##
## REML criterion at convergence: 10677.9
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3203 -0.6146 -0.0294  0.5467  5.5331
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  classid  (Intercept)  103.57  10.177
##  schoolid (Intercept)   85.44   9.244
##  Residual             1019.47  31.929
## Number of obs: 1081, groups:  classid, 285; schoolid, 105
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   52.250      3.838  233.946  13.613   <2e-16 ***
## sex           -1.526      2.041 1030.557  -0.747   0.4550
## mathprep       2.426      1.298  181.813   1.869   0.0632 .
## mathknow       2.405      1.299  206.425   1.851   0.0656 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) sex    mthprp
## sex      -0.268
## mathprep -0.878  0.001
## mathknow -0.003  0.011  0.005
```

```
# Extract out the coefficients
modelOutPlot <- broom.mixed::tidy(me_mod, conf.int = TRUE) # note: have to use broom.extra for me model

modelOutPlot <- modelOutPlot[modelOutPlot$effect == "fixed" &
                             modelOutPlot$term != "(Intercept)", ]

# Plot the coefficients of interest
ggplot(data = modelOutPlot, aes(x = term, y = estimate, ymin = conf.low, ymax = conf.high)) +
  theme_minimal() +
  geom_hline(yintercept = 0.0, color = "red", size = 2) +
  geom_point() +
  geom_linerange() +
  coord_flip()
```




```
# ^ Really like this plot!
```

Chapter 2: Linear Mixed-Effect Models

Random-effect syntax (for using lme4):

- (1 | group): random intercept with a fixed mean
- (1 | g1/g2): intercepts vary among g1 and g2 within g2
- (1 | g1) + (1 | g2): random intercepts for two variables
- x + (x | g): correlated random slope and intercept
- x + (x || g): uncorrelated random slope and intercept

```
# Using birth data
countyBirthsData <- read_csv("https://assets.datacamp.com/production/repositories/1803/datasets/eb95cb6
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Year = col_double(),
##   TotalPopulation = col_double(),
##   BirthRate = col_double(),
##   AverageBirthWeight = col_double(),
##   AverageAgeofMother = col_double(),
##   CountyName = col_character(),
##   State = col_character()
## )
```

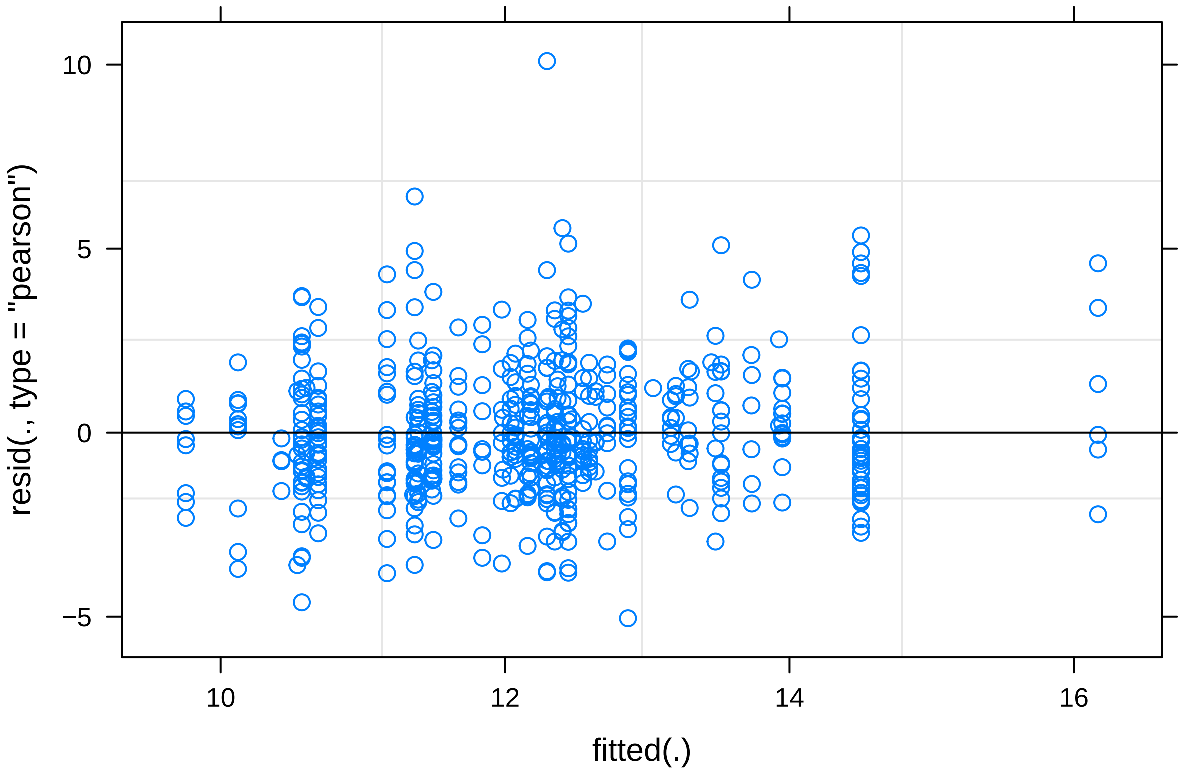
```r
# Counties exist within states, and perhaps states contribute to variability. Hence, the need for rando

# To start, we build a hierarchical model with a global intercept (fixed-effect) and random-effect for
mod1 <- lmer(formula = BirthRate ~ (1 | State),
             data = countyBirthsData)

summary(mod1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: BirthRate ~ (1 | State)
##    Data: countyBirthsData
##
## REML criterion at convergence: 2411
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7957 -0.6056 -0.1063  0.5211  5.5948
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  State    (Intercept) 1.899    1.378
##  Residual             3.256    1.804
## Number of obs: 578, groups:  State, 50
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  12.3362     0.2216 43.3830   55.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# plot the residuals
plot(mod1)
```

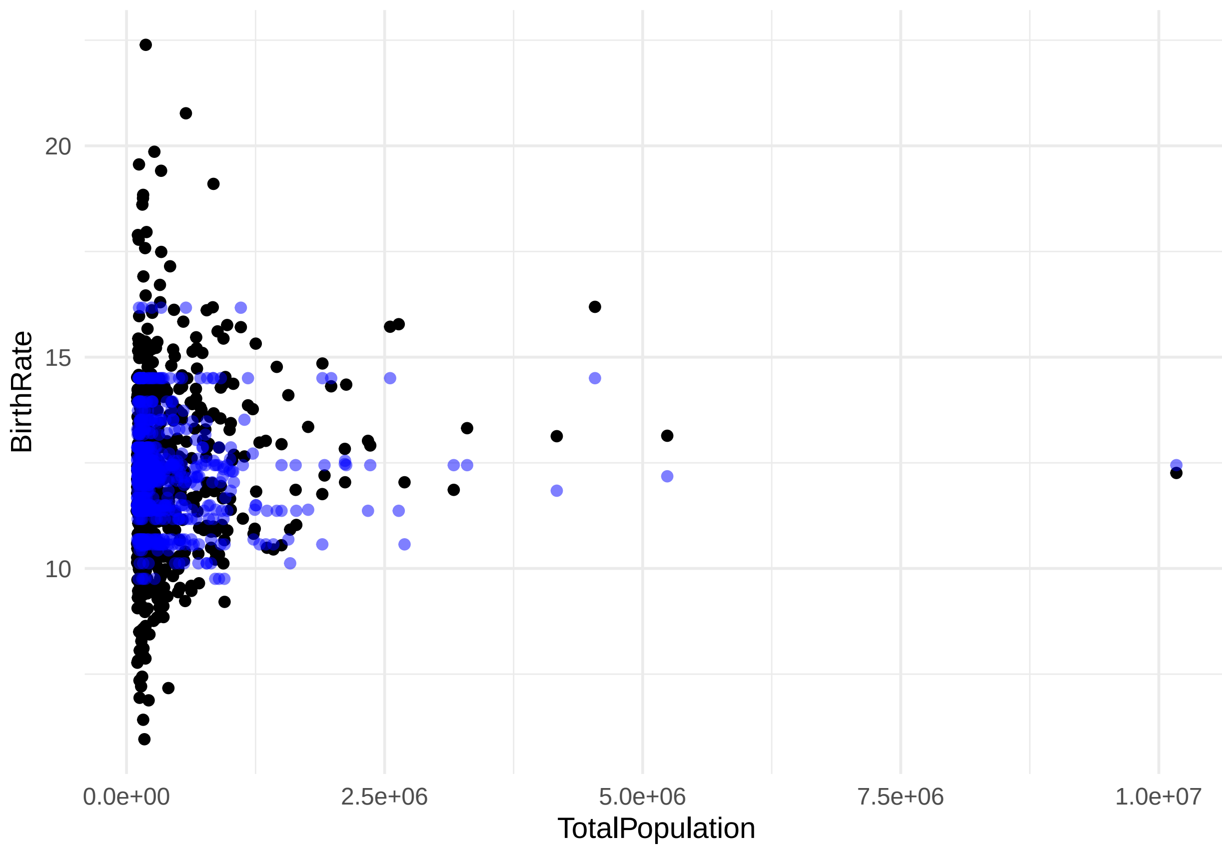



```
countyBirthsData$predictState <- predict(mod1, countyBirthsData)

ggplot(data = countyBirthsData) +
  theme_minimal() +
  geom_point(aes(x = TotalPopulation, y = BirthRate)) +
  geom_point(aes(x = TotalPopulation, y = predictState),
             color = "blue", alpha = 0.5)
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```




```r
# Random-effects intercepts estimated for each state. This allowed us to account for each state having

# Let's include a fixed effect: average age of mother.

mod2 <- lmer(formula = BirthRate ~ AverageAgeofMother + (1 | State),
             data = countyBirthsData)

summary(mod2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: BirthRate ~ AverageAgeofMother + (1 | State)
##    Data: countyBirthsData
##
## REML criterion at convergence: 2347.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9602 -0.6086 -0.1042  0.5144  5.2686
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  State    (Intercept) 1.562    1.250
##  Residual             2.920    1.709
## Number of obs: 578, groups:  State, 50
##
```

```
## Fixed effects:
##                     Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)         27.57033    1.81801 575.96198  15.165  < 2e-16 ***
## AverageAgeofMother  -0.53549    0.06349 574.18338  -8.434 2.71e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## AvrgAgfMthr -0.994
```

```r
# Now we want a random-effects slope for each state. A random-effect slope may be estimated for each gr

# Adding total population of each state as a random effect (it's numeric, hence slope not intercept)

countyBirthsData <- countyBirthsData %>%
  mutate(LogTotalPop = log10(TotalPopulation))

mod3 <- lmer(formula = BirthRate ~ AverageAgeofMother + (LogTotalPop | State),
             data = countyBirthsData)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0260043 (tol = 0.002, component 1)
```

```r
# Uncorrelated random effects model (use the || syntax):
mod4 <- lmer(formula = BirthRate ~ AverageAgeofMother + (LogTotalPop || State),
             data = countyBirthsData)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(mod4) # Not good. Need to the more complex model (mod3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: BirthRate ~ AverageAgeofMother + (LogTotalPop || State)
##    Data: countyBirthsData
##
## REML criterion at convergence: 2347.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9602 -0.6086 -0.1042  0.5144  5.2686
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  State    (Intercept) 1.562    1.250
##  State.1  LogTotalPop 0.000    0.000
##  Residual             2.920    1.709
## Number of obs: 578, groups:  State, 50
##
## Fixed effects:
##                     Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)         27.57033    1.81801 575.96198  15.165  < 2e-16 ***
## AverageAgeofMother  -0.53549    0.06349 574.18338  -8.434 2.71e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##             (Intr)
## AvrgAgfMthr -0.994
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
# A predictor can be both a fixed-effect and a random-effect (e.g., mothers age)
mod5 <- lmer(formula = BirthRate ~ AverageAgeofMother + (AverageAgeofMother | State),
             data = countyBirthsData)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0133555 (tol = 0.002, component 1)
```

```r
summary(mod5) # This is helpful for prediction, as you can correct the effect of age of mother by state
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: BirthRate ~ AverageAgeofMother + (AverageAgeofMother | State)
##    Data: countyBirthsData
##
## REML criterion at convergence: 2337.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8402 -0.5965 -0.1132  0.5233  5.1817
##
## Random effects:
##  Groups   Name               Variance Std.Dev. Corr
##  State    (Intercept)        78.33144 8.8505
##           AverageAgeofMother  0.08433 0.2904   -0.99
##  Residual                     2.80345 1.6744
## Number of obs: 578, groups:  State, 50
##
## Fixed effects:
##                    Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)        27.21961    2.41010 39.91438  11.294 5.31e-14 ***
## AverageAgeofMother -0.52344    0.08293 39.42721  -6.312 1.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## AvrgAgfMthr -0.997
## convergence code: 0
## Model failed to converge with max|grad| = 0.0133555 (tol = 0.002, component 1)
```

REML - restricted maximum likelihood method

```r
# Extracting coefficients

# Fixed effects:
fixef(mod5)
```

```
##        (Intercept) AverageAgeofMother
##          27.219605          -0.523442
```

```r
# Random effects:
ranef(mod5)
```

```
## $State
##      (Intercept) AverageAgeofMother
## AK   4.15003868       -0.109184783
## AL  -4.03378321        0.127790698
## AR   1.21082566       -0.026165323
## AZ  -5.11009217        0.148282653
## CA  11.43254356       -0.373285277
## CO   1.35768165       -0.043219507
## CT  -3.82970951        0.066100814
## DC   1.21115127       -0.012314396
## DE  -2.11510834        0.060412134
## FL -11.31590854        0.334159606
## GA   6.93457372       -0.230274526
## HI  -0.51788167        0.020356335
## IA   3.41174428       -0.085895750
## ID   7.51123552       -0.222596318
## IL  -3.07183968        0.102977648
## IN  -3.02904144        0.097145469
## KS   5.15187643       -0.150220284
## KY   2.30557751       -0.058607045
## LA   5.32029471       -0.154658352
## MA  -5.72651427        0.144383430
## MD  -3.31971034        0.116256669
## ME  -5.76419680        0.145079276
## MI  -5.51723036        0.156052226
## MN  -1.41379992        0.077580405
## MO   0.46288340       -0.013492801
## MS  -0.77739331        0.021041676
## MT   0.07683320       -0.006560404
## NC   4.20869679       -0.159191892
## ND   3.20725583       -0.078130864
## NE   4.97980324       -0.122101047
## NH  -3.02333587        0.055641597
## NJ   6.33991764       -0.220034709
## NM   1.24087579       -0.066463156
## NV   0.35709749       -0.009233999
## NY  -5.34557647        0.162856972
## OH  -7.43833421        0.227092266
## OK   4.73462323       -0.140260716
## OR  -5.64958450        0.170021845
## PA  -8.17679196        0.230593543
## RI  -0.82967532       -0.014536489
## SC  -5.60511798        0.179511398
## SD   4.96670949       -0.128324308
## TN  -1.21831791        0.038711838
## TX  10.90259845       -0.331687671
## UT  14.12700916       -0.369723836
## VA  -8.76482204        0.350357075
## VT  -0.02787975       -0.014042974
## WA   0.30152985       -0.002489047
## WI  -0.35751943       -0.004946489
```

```
## WV  -3.92421158        0.115236390
##
## with conditional variances for "State"
```

```r
# Confidence intervals for fe:
confint(mod5)
```

```
## Computing profile confidence intervals ...

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): unexpected decrease in
## profile: using minstep

## Warning in FUN(X[[i]], ...): non-monotonic profile for .sig02

## Warning in confint.thpr(pp, level = level, zeta = zeta): bad spline fit
## for .sig02: falling back to linear interpolation

##                         2.5 %     97.5 %
## .sig01              4.4104030 13.8610928
## .sig02             -0.9997023 -0.9607253
## .sig03              0.1333576  0.4630192
## .sigma              1.5761207  1.7821071
## (Intercept)        22.4263992 32.1350540
## AverageAgeofMother -0.6923128 -0.3576348
```

```r
tidy(mod5, conf.int = TRUE)
```

```
## # A tibble: 6 x 10
##   effect group term  estimate std.error statistic    df   p.value conf.low
##   <chr>  <chr> <chr>    <dbl>     <dbl>     <dbl> <dbl>     <dbl>    <dbl>
## 1 fixed  <NA>  (Int~   27.2      2.41       11.3   39.9  5.31e-14   22.5
## 2 fixed  <NA>  Aver~   -0.523    0.0829     -6.31  39.4  1.83e- 7   -0.686
## 3 ran_p~ State sd__~    8.85    NA          NA     NA   NA          NA
## 4 ran_p~ State sd__~    0.290   NA          NA     NA   NA          NA
## 5 ran_p~ State cor_~   -0.992   NA          NA     NA   NA          NA
## 6 ran_p~ Resi~ sd__~    1.67    NA          NA     NA   NA          NA
## # ... with 1 more variable: conf.high <dbl>
```

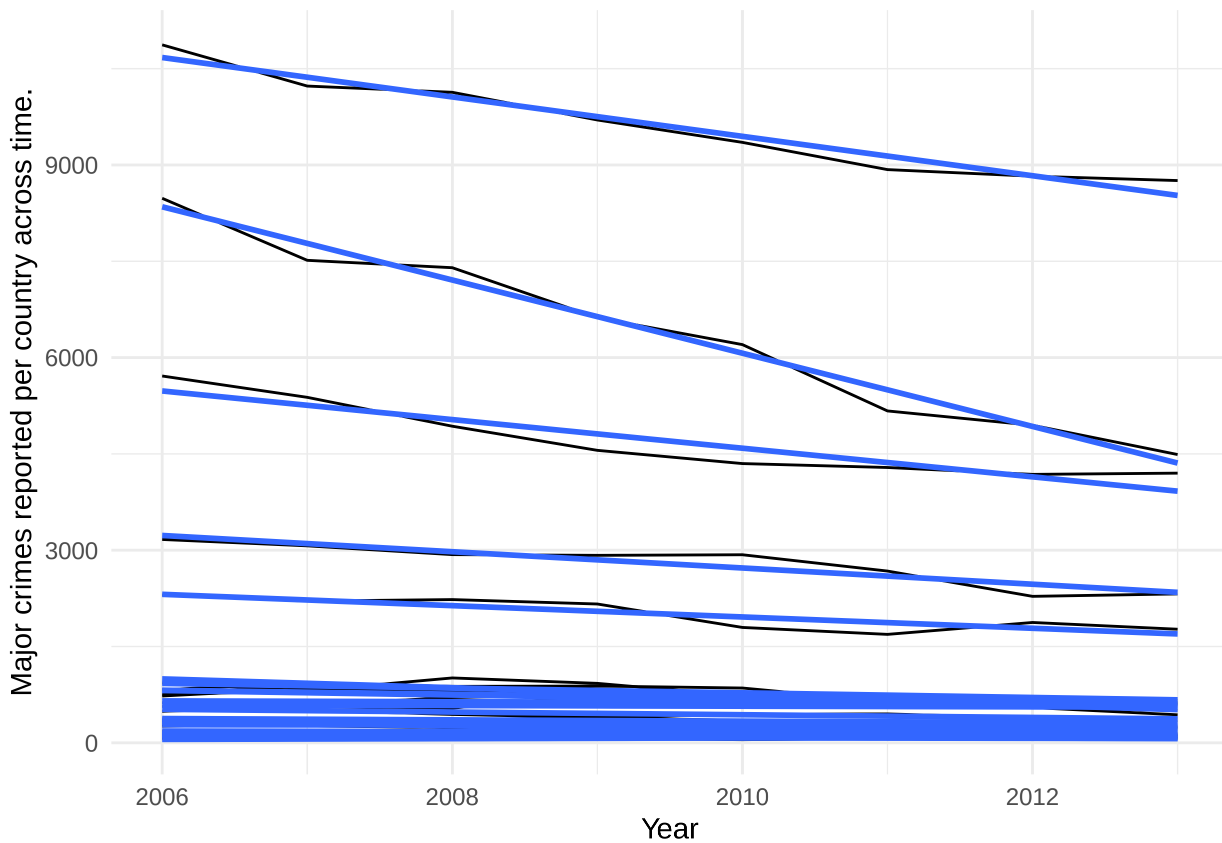
Using Maryland Crime data

```r
MDcrime <- read_csv("https://assets.datacamp.com/production/repositories/1803/datasets/e5e076efd3c3b766
```

```
## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   County = col_character(),
##   Year = col_double(),
##   Crime = col_double(),
##   Year2 = col_double()
## )
```

```r
ggplot(data = MDcrime, aes(x = Year, y = Crime, group = County)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  ylab("Major crimes reported per country across time.")
```




```
# Looks like we will require a random-effect intercept, and likely a random-effect slope.

# Try a linear model:
mod1 <- lm(formula = Crime ~ Year, data = MDcrime)
summary(mod1) # Year(2) not significant. Note: need to use Year2 variable.
```

```
##
## Call:
## lm(formula = Crime ~ Year, data = MDcrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1514.3 -1156.6  -930.9  -511.5  9293.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 136642.97  147510.24   0.926    0.355
## Year           -67.33      73.41  -0.917    0.360
##
## Residual standard error: 2331 on 190 degrees of freedom
## Multiple R-squared:  0.004408,   Adjusted R-squared:  -0.0008315
## F-statistic: 0.8413 on 1 and 190 DF,  p-value: 0.3602
```

```
# Fit the model with Year(2) as both a fixed and random-effect:
mod2 <- lmer(formula = Crime ~ Year2 + ( 1 + Year2 | County), data = MDcrime)
summary(mod2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Crime ~ Year2 + (1 + Year2 | County)
##    Data: MDcrime
##
## REML criterion at convergence: 2535.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8080 -0.2235 -0.0390  0.2837  3.0767
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  County   (Intercept) 7584514  2754.00
##           Year2         16940   130.15  -0.91
##  Residual                8425    91.79
## Number of obs: 192, groups:  County, 24
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  1577.28     562.29   23.02   2.805   0.0100 *
## Year2         -67.33      26.72   23.01  -2.519   0.0191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## Year2 -0.906
```

```r
# ANOVA - Analysis of Variance
# Build a null model with only County as a random-effect
null_model <- lmer(Crime ~ (1 | County), data = MDcrime)
# Build alternative model, with Year2 as a fixed and random slope and County as a random effect
alt_model <- lmer(Crime ~ Year2 + (1 + Year2 | County), data = MDcrime)

# Compare models - look at Chi square test.
anova(null_model, alt_model)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: MDcrime
## Models:
## null_model: Crime ~ (1 | County)
## alt_model: Crime ~ Year2 + (1 + Year2 | County)
##            Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## null_model  3 2954.4 2964.2 -1474.2   2948.4
## alt_model   6 2568.9 2588.4 -1278.4   2556.9 391.52      3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 3: Generalized Linear Mixed-Effects Models

Topics covered in this chapter (for future reference):

- Logisitic regression with glm() - family = "binomial"
- Poisson regression with glm() - family = "poisson" -> for count data.
- Plotting logistic regression results with ggplot - stat_smooth(method = "glm", method.args = list(family = "binomial"))
- Using glmer() to estimate glms with mixed effects.
- Handling different data inputs (e.g. matrix with cbind)
- Calculating odds-ratios with exp(fixef()) and exp(confint()). Can also be done via tidy()
- Using ggplot to visualize random-effects poisson models

Chapter 4: Repeated Measures

Paired t-test -> special case of a t-test

t.test(paired = TRUE) -> does not assume equal variance of both groups.

Repeated measures ANOVA -> tests if means are constant across time. ^ simply a special type of mixed effects model.

```
set.seed(1234)

n_ind <- 10
before <- rnorm(n = n_ind, mean = 0, sd = 0.5)
after <- before + rnorm(n = n_ind, mean = 4.5, sd = 5)

t.test(before, after, paired = F)
```

```
##
##  Welch Two Sample t-test
##
## data:  before and after
## t = -2.3343, df = 9.1605, p-value = 0.04396
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.6873743 -0.1309186
## sample estimates:
##  mean of x  mean of y
## -0.1915787  3.7175678
```

```
t.test(before, after, paired = T) # Paired is more powerful.
```

```
##
##  Paired t-test
##
## data:  before and after
## t = -2.3164, df = 9, p-value = 0.04576
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.72678305 -0.09150987
## sample estimates:
## mean of the differences
##               -3.909146
```

```
dat <- data.frame(y = c(before, after),
                  trial = rep(c("before", "after"), each = n_ind),
                  ind = rep(letters[1:n_ind], times = 2))
```

```r
# Now do it with lmer:
mod <- lmer(y ~ trial + (1|ind), data = dat)
summary(mod) # Woot. Pretty much the same (same t value, p value)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: y ~ trial + (1 | ind)
##    Data: dat
##
## REML criterion at convergence: 103.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0871 -0.5599 -0.0249  0.1110  3.3158
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  ind      (Intercept) 9.047e-07 0.0009512
##  Residual             1.402e+01 3.7446323
## Number of obs: 20, groups:  ind, 10
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)    3.718      1.184 18.000   3.139  0.00567 **
## trialbefore   -3.909      1.675 18.000  -2.334  0.03137 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## trialbefore -0.707
```

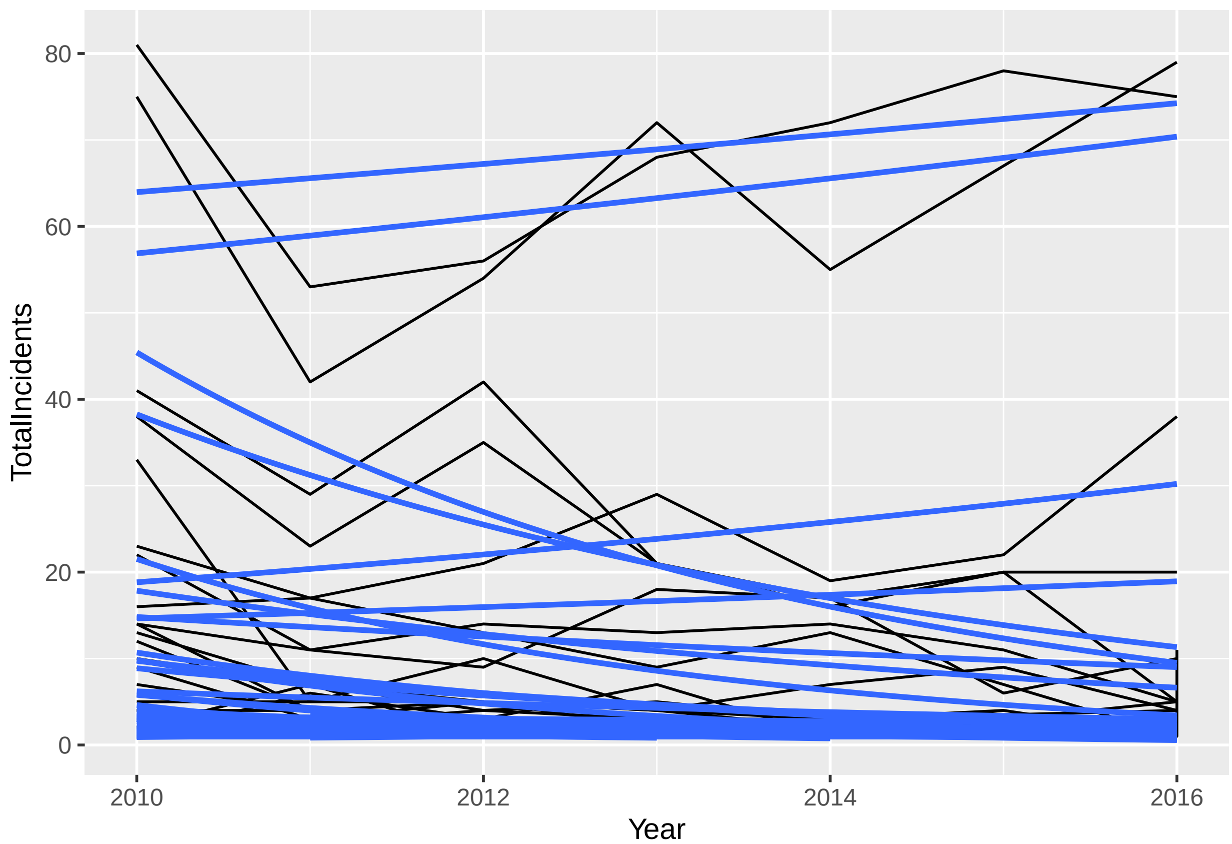
```r
htcrime <- read_csv("https://assets.datacamp.com/production/repositories/1803/datasets/45e88fe1bc8d1d76
```

```
## Parsed with column specification:
## cols(
##   Year = col_double(),
##   County = col_character(),
##   TotalIncidents = col_double(),
##   Year2 = col_double()
## )
```

```r
# Is the number of hate crimes changing over time in NY counties?
# 1) Is the state-wide number of hate crimes changing?
# 2) Are the number of hate crimes changing differently in each county?

# Step 1. Visualize

ggplot(data = htcrime, mapping = aes(x = Year, y = TotalIncidents, group = County)) +
  geom_line() +
  geom_smooth(method = "glm", method.args = c("poisson"), se = FALSE)
```




```
# We can see that different counties have different trends. Therefore it would be appropriate to use di
```