P.11 - Multilevel Model for Change

Mihai A. Constantin

29.11.2021

Lab Description

In this assignment you are going to estimate several multilevel models that reproduce the findings discussed in *lecture 11*. Compare your results with the findings reported in the lecture slides. Try to use the lecture slides as a guide through the `R` output.

For this practical you will need the following packages: `lme4`, `ggplot2`, and `psych`. You can install and load these packages using the following code:

```
# Install packages.
install.packages(c("lme4", "ggplot2", "psych"))

# Load the packages.
library(lme4)
library(ggplot2)
library(psych)
```

Questions

Start by loading the `alcohol.csv` data in `R`, then compute basic descriptive statistics. The data is available on Canvas in the module corresponding to the current lab session.

Load the data.

```
# For example.
setwd("/Users/mihai/Downloads")

# Load data.
data <- read.csv("alcohol.csv")

# Inspect the data.
View(data)
```

Check the structure of the data.

```
# List variables.
str(data)
```

```
## 'data.frame':    246 obs. of  9 variables:
##  $ id    : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ age   : int  14 15 16 14 15 16 14 15 16 14 ...
##  $ coa   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ male  : int  0 0 0 1 1 1 1 1 1 1 ...
##  $ age_14: int  0 1 2 0 1 2 0 1 2 0 ...
##  $ alcuse: num  1.73 2 2 0 0 ...
##  $ peer  : num  1.265 1.265 1.265 0.894 0.894 ...
##  $ cpeer : num  0.247 0.247 0.247 -0.124 -0.124 ...
##  $ ccoa  : num  0.549 0.549 0.549 0.549 0.549 0.549 0.549 0.549 0.549 0.549 ...
```

Compute basic descriptive statistics.

```
# Describe the data using `psych`.
describe(data[c("alcuse", "age_14", "coa", "peer")])
```

```
##        vars   n mean   sd median trimmed  mad min  max range  skew kurtosis   se
## alcuse    1 246 0.92 1.06   1.00    0.77 1.48   0 3.61  3.61  0.80    -0.55 0.07
## age_14    2 246 1.00 0.82   1.00    1.00 1.48   0 2.00  2.00  0.00    -1.51 0.05
## coa       3 246 0.45 0.50   0.00    0.44 0.00   0 1.00  1.00  0.19    -1.97 0.03
## peer      4 246 1.02 0.73   0.89    1.00 0.97   0 2.53  2.53 -0.04    -0.99 0.05
```

We can also compute the descriptive statistics per individual (i.e., by `id`). In this case, we use a formula syntax, where ~ denotes *describe what is on the left of ~ by the variable on the right.*

```
# Describe by group, in this case the `id` variable.
describe(alcuse + age_14 + coa + peer ~ id, data = data)
```

Which would result in the following (i.e., trimmed) output:

```
# id: 1
#        vars n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
# alcuse    1 3 1.91 0.15   2.00    1.91 0.00 1.73 2.00  0.27 -0.38    -2.33 0.09
# age_14    2 3 1.00 1.00   1.00    1.00 1.48 0.00 2.00  2.00  0.00    -2.33 0.58
# coa       3 3 1.00 0.00   1.00    1.00 0.00 1.00 1.00  0.00   NaN      NaN 0.00
# peer      4 3 1.26 0.00   1.26    1.26 0.00 1.26 1.26  0.00   NaN      NaN 0.00
# ------------------------------------------------------------------------------
# .
# .
# .
# ------------------------------------------------------------------------------
# id: 82
#        vars n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
# alcuse    1 3 0.80 0.73   1.00    0.80 0.61 0.00 1.41  1.41 -0.25    -2.33 0.42
# age_14    2 3 1.00 1.00   1.00    1.00 1.48 0.00 2.00  2.00  0.00    -2.33 0.58
# coa       3 3 0.00 0.00   0.00    0.00 0.00 0.00 0.00  0.00   NaN      NaN 0.00
# peer      4 3 2.19 0.00   2.19    2.19 0.00 2.19 2.19  0.00   NaN      NaN 0.00
```

Now, we extract only the variables we are interested in and create the factors accordingly.

```
# Vector of variables we are interested in.
variables <- c("id", "age", "coa", "age_14", "peer", "cpeer", "alcuse")

# Subset the data.
```

```r
data <- data[, variables]

# Inspect the first few rows of the data.
head(data)
```

```
##   id age coa age_14      peer      cpeer   alcuse
## 1  1  14   1      0 1.2649111  0.2469111 1.732051
## 2  1  15   1      1 1.2649111  0.2469111 2.000000
## 3  1  16   1      2 1.2649111  0.2469111 2.000000
## 4  2  14   1      0 0.8944272 -0.1235728 0.000000
## 5  2  15   1      1 0.8944272 -0.1235728 0.000000
## 6  2  16   1      2 0.8944272 -0.1235728 1.000000
```

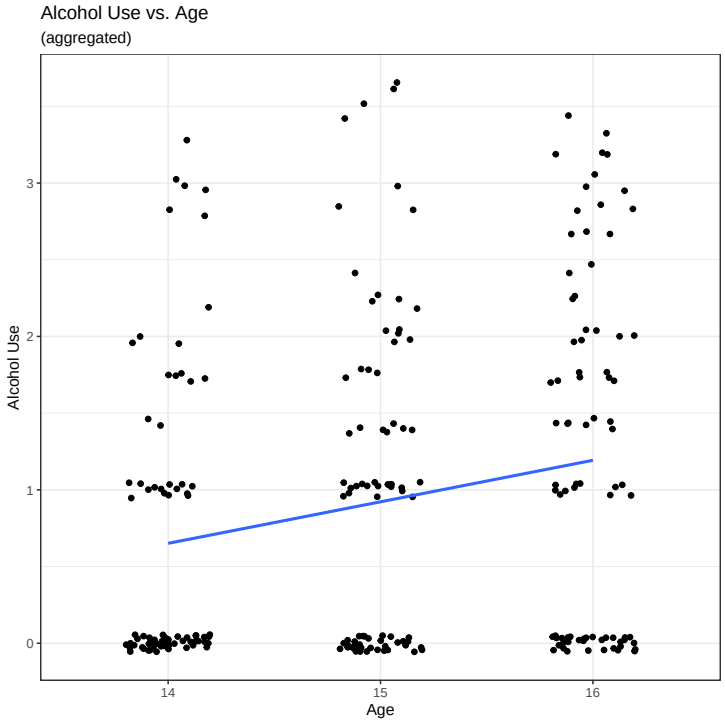
```r
# Create factors for categorical variables.
data$id <- factor(data$id)
data$age <- factor(data$age, levels = c(14, 15, 16), labels = c(14, 15, 16))
data$coa <- factor(data$coa, levels = c(0, 1), labels = c("non-alcoholic parent", "alcoholic parent"))

# Create a mean split variable for the peer alcohol consumption variable.
data$peer_split <- factor(data$peer <= mean(data$peer), levels = c(TRUE, FALSE), labels = c("low", "high"))

# Show the structure of the data.
str(data)
```

```
## 'data.frame':    246 obs. of  8 variables:
##  $ id        : Factor w/ 82 levels "1","2","3","4",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ age       : Factor w/ 3 levels "14","15","16": 1 2 3 1 2 3 1 2 3 1 ...
##  $ coa       : Factor w/ 2 levels "non-alcoholic parent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ age_14    : int  0 1 2 0 1 2 0 1 2 0 ...
##  $ peer      : num  1.265 1.265 1.265 0.894 0.894 ...
##  $ cpeer     : num  0.247 0.247 0.247 -0.124 -0.124 ...
##  $ alcuse    : num  1.73 2 2 0 0 ...
##  $ peer_split: Factor w/ 2 levels "low","high": 2 2 2 1 1 1 1 1 1 2 ...
```

It is always a good idea to visualize the data. We can start by plotting the aggregated data and the corresponding regression line.

```r
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse)) +
    geom_jitter(
        width = 0.2
    ) +
    geom_smooth(
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    theme_bw() +
    theme(
        legend.position = "top"
    ) +
    labs(
```

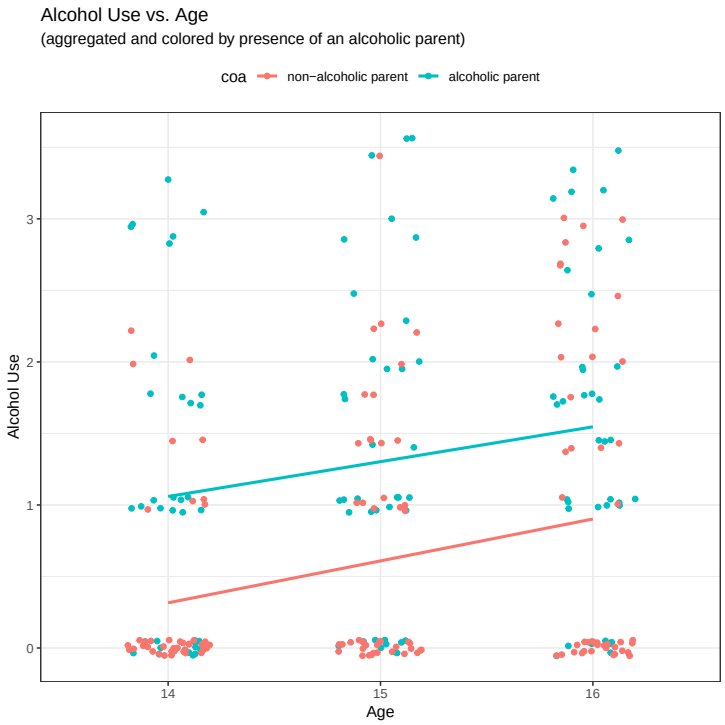
```
        title = "Alcohol Use vs. Age",
        subtitle = "(aggregated)",
        x = "Age",
        y = "Alcohol Use"
    )
```



We can also fit a separate linear regression for each group (e.g., alcoholic parent vs. non-alcoholic parent, or low peer consumption vs. high peer consumption). We start with the `coa` variable.

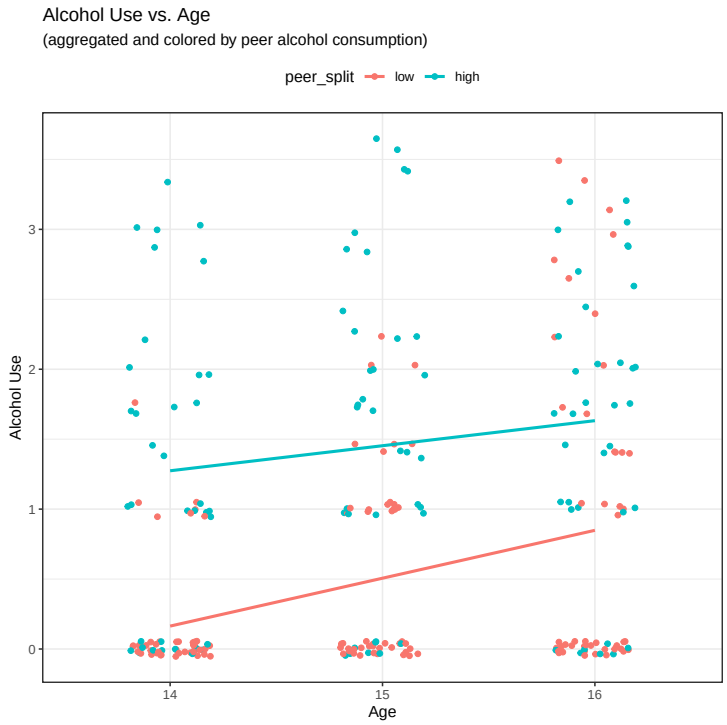
```
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse, color = coa)) +
    geom_jitter(
        width = 0.2
    ) +
    geom_smooth(
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    theme_bw() +
    theme(
        legend.position = "top"
    ) +
    labs(
        title = "Alcohol Use vs. Age",
        subtitle = "(aggregated and colored by presence of an alcoholic parent)",
        x = "Age",
        y = "Alcohol Use"
```

```
    )
```



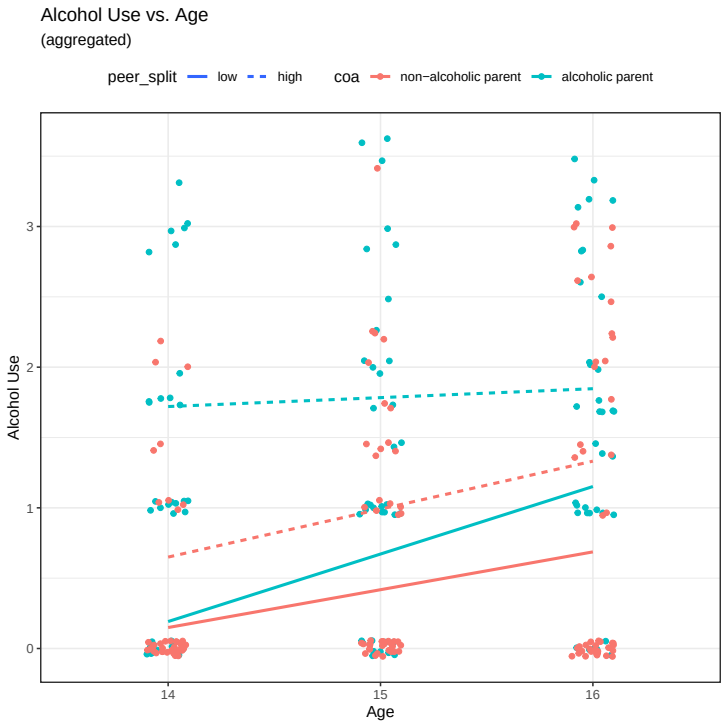
And now we do the same for the peer consumption, using the `peer_split` variable we created that splits the peer alcohol consumption measure into *low* (i.e., below the mean) and *hight* (i.e., above the mean).

```
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse, color = peer_split)) +
    geom_jitter(
        width = 0.2
    ) +
    geom_smooth(
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    theme_bw() +
    theme(
        legend.position = "top"
    ) +
    labs(
        title = "Alcohol Use vs. Age",
        subtitle = "(aggregated and colored by peer alcohol consumption)",
        x = "Age",
        y = "Alcohol Use"
    )
```



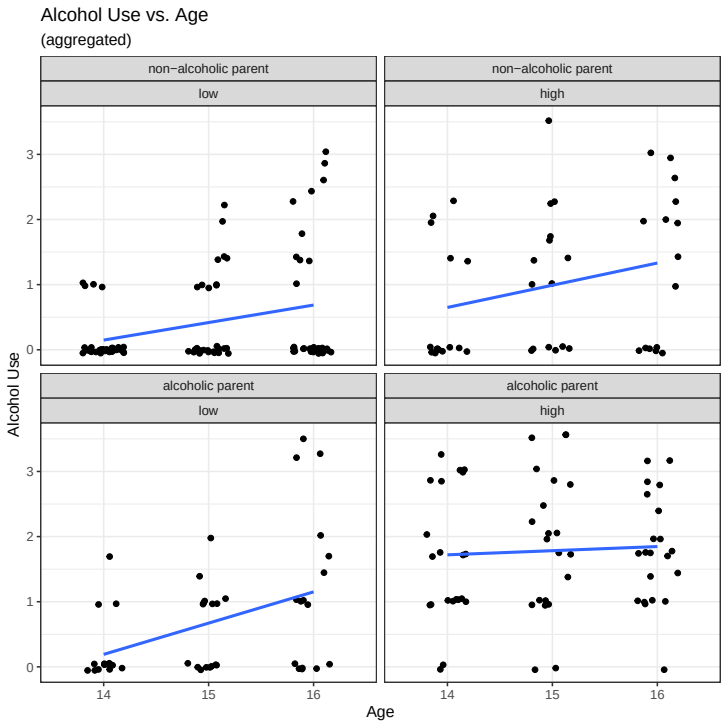
Or, we can view the regression lines for the aggregated data, both by `coa` and `peer_split`, where the line color indicates the presence or absence of a non-alcoholic parent, and the line type indicates the peer alcohol consumption.

```
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse, color = coa, linetype = peer_split)) +
    geom_jitter(
        width = 0.1
    ) +
    geom_smooth(
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    theme_bw() +
    theme(
        legend.position = "top"
    ) +
    labs(
        title = "Alcohol Use vs. Age",
        subtitle = "(aggregated)",
        x = "Age",
        y = "Alcohol Use"
    )
```



Another way is to put each variable int its own quadrant using the function `facet_wrap`.

```
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse)) +
    geom_jitter(
        width = 0.2
    ) +
    geom_smooth(
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    facet_wrap(
        facets = coa ~ peer_split
    ) +
    theme_bw() +
    theme(
        legend.position = "top"
    ) +
    labs(
        title = "Alcohol Use vs. Age",
        subtitle = "(aggregated)",
        x = "Age",
        y = "Alcohol Use"
    )
```



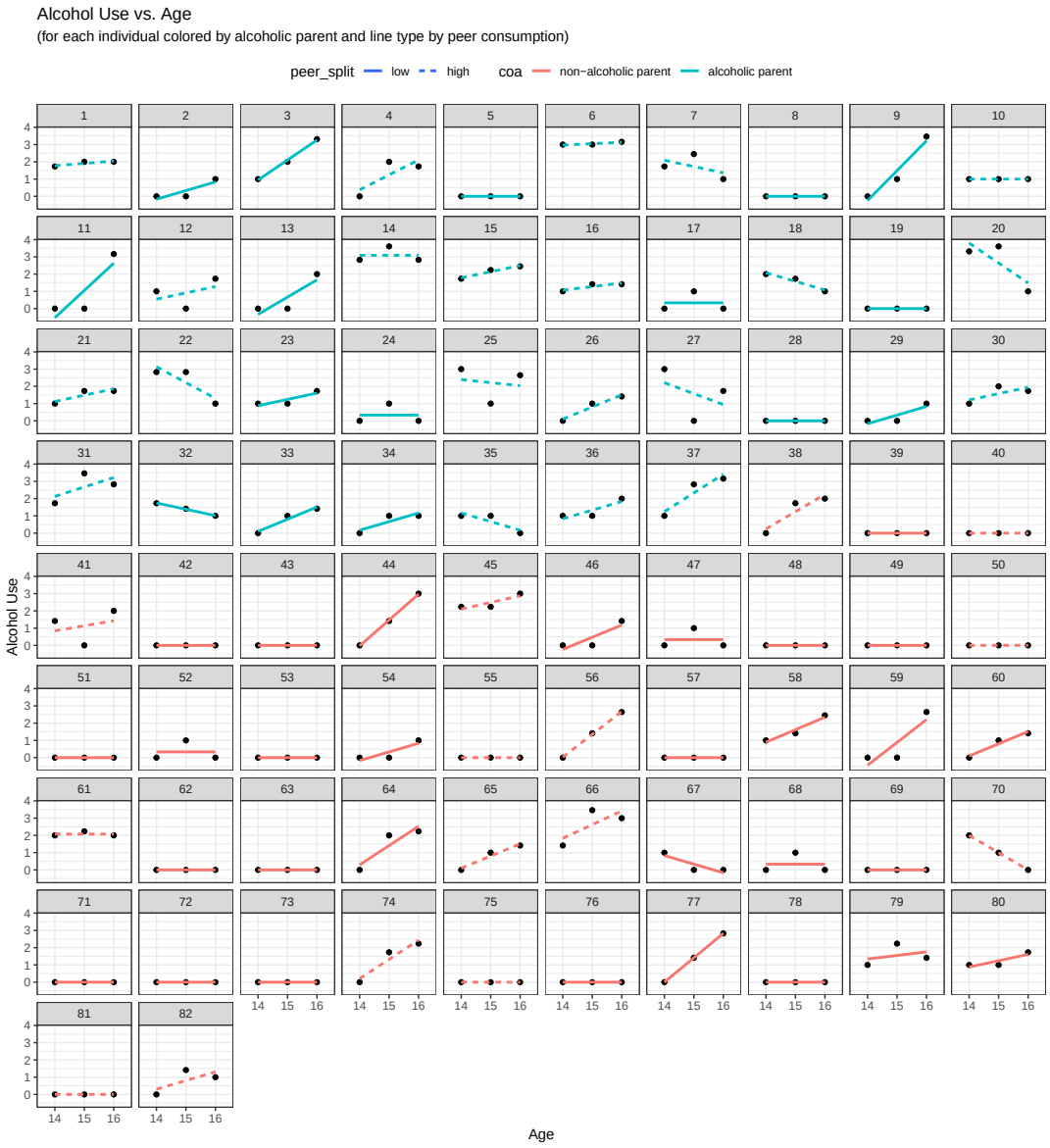
However, since we have nested data with multiple measurements per participant (i.e., three measurements, at ages 14, 15 and 16), aggregating these measurements it is not a good idea because we violate the assumption of independent observations. If we take a look at the first six rows in our data

```
##   id age              coa age_14      peer      cpeer   alcuse peer_split
## 1  1  14 alcoholic parent      0 1.2649111  0.2469111 1.732051       high
## 2  1  15 alcoholic parent      1 1.2649111  0.2469111 2.000000       high
## 3  1  16 alcoholic parent      2 1.2649111  0.2469111 2.000000       high
## 4  2  14 alcoholic parent      0 0.8944272 -0.1235728 0.000000        low
## 5  2  15 alcoholic parent      1 0.8944272 -0.1235728 0.000000        low
## 6  2  16 alcoholic parent      2 0.8944272 -0.1235728 1.000000        low
```

we see that the first three observations belong to the first participant (i.e., $id = 1$), and the other three belong to the second participant (i.e., $id = 2$). This is what we call nested data (e.g., some measurements belong to a participant, while others to another participant and so on). By fitting a linear model like we did above, we do not respect the nested structure of the data and, in fact, we aggregated over the entire rows. This means that we end up treating each row as an independent observation (e.g., row number one is independent of row number two), when this is not the case (e.g., row number one is *not independent* of row number two because both of these observations were produced by the same individual, i.e., the person with $id = 1$). Therefore, we are better of taking into account such dependencies in the data and allow each individual to have his or her own regression line. Let us see how we can illustrate this idea graphically.

```
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse, color = coa, linetype = peer_split)) +
    geom_point(
        color = "#000000"
    ) +
    geom_smooth(
```

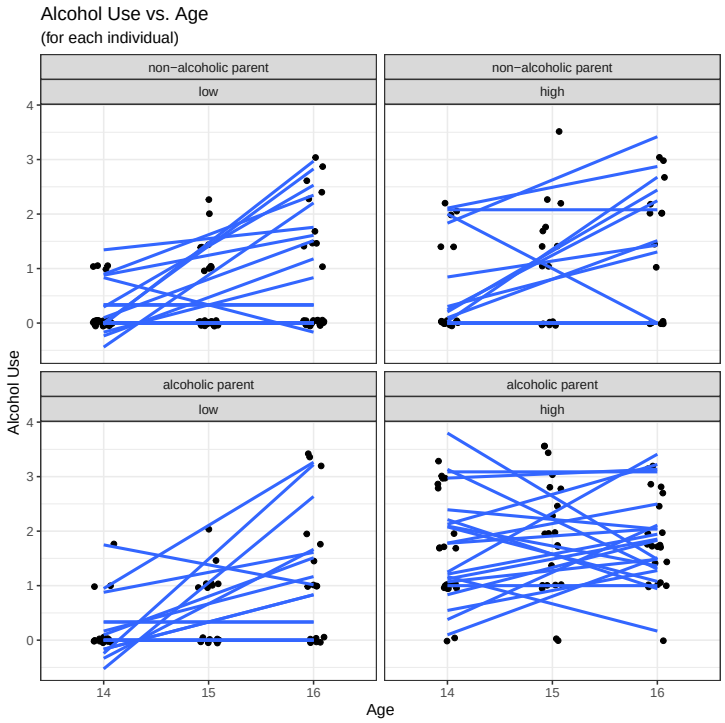
```
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    facet_wrap(
        facets = ~ id
    ) +
    theme_bw() +
    theme(
        legend.position = "top"
    ) +
    labs(
        title = "Alcohol Use vs. Age",
        subtitle = "(for each individual colored by alcoholic parent and line type by peer consumption)",
        x = "Age",
        y = "Alcohol Use"
    )
```



Or, more easier to see.

```
# Create ggplot.
ggplot(data, aes(x = age, y = alcuse, group = id)) +
    geom_jitter(
        width = 0.1
    ) +
    geom_smooth(
        mapping = aes(x = as.numeric(age)),
        method = lm,
        formula = y ~ x,
        se = FALSE
    ) +
    facet_wrap(
        facets = coa ~ peer_split
    ) +
```

```
    theme_bw() +
    theme(
        legend.position = "none"
    ) +
    labs(
        title = "Alcohol Use vs. Age",
        subtitle = "(for each individual)",
        x = "Age",
        y = "Alcohol Use"
    )
```



Unsurprisingly, we see that there is quite some variability both in the intercepts (i.e., the initial status of alcohol consumption) and the slopes (i.e., the yearly rate of alcohol consumption). So, with this in mind, we can now go on a fit some multi-level models.

You can check the documentation for the `lme4` package to find out how random intercepts and slopes are indicated in the model syntax. Or, if you want to find out more, can also check the incredibly popular paper for the `lme4` package (i.e., 42713 citations to date). In *Figure 1* you can see an overview of the syntax of `lme4`. In a nutshell, we specify the models using the `R` formula interface, where what is left of the `~` symbol represents the dependent variable, and what is on the right represents independent variables, iteraction terms and so on. For example, if we want to fit a simple linear regression with only an intercept, we can use `lm(y ~ 1)`, where `y ~ 1` can be read as `y` is predicted by its intercept. If we want to add a predictor `x`, then the formula becomes `y ~ 1 + x`, where now `y` is predicted by its intercept and `x`. We can simplify the above formula to `y ~ x`, and, the `lm`, or the corresponding function of `lme4` we use, will still estimate the intercept for us. When using `lme4` to specify random effects, we specify such effects in between brackets (i.e., `(` and `)`) together with a pipe operator (i.e., `|`). The pipe operator `|` follows a similar logic to the tilde `~` operator.

What is on the left of the `|` operator it is allowed to vary for the levels of (i.e., by) what is on the right. For example, the following complete formula `y ~ x + (1 | id)` means that `y` is predicted by `x` and that the intercept is allowed to vary by `id` (i.e., each individual gets his or her own initial status).

Formula	Alternative	Meaning
`(1 | g)`	`1 + (1 | g)`	Random intercept with fixed mean.
`0 + offset(o) + (1 | g)`	`-1 + offset(o) + (1 | g)`	Random intercept with *a priori* means.
`(1 | g1/g2)`	`(1 | g1) + (1 | g1:g2)`	Intercept varying among `g1` and `g2` within `g1`.
`(1 | g1) + (1 | g2)`	`1 + (1 | g1) + (1 | g2)`	Intercept varying among `g1` and `g2`.
`x + (x | g)`	`1 + x + (1 + x | g)`	Correlated random intercept and slope.
`x + (x || g)`	`1 + x + (1 | g) + (0 + x | g)`	Uncorrelated random intercept and slope.

Table 2: Examples of the right-hand-sides of mixed-effects model formulas. The names of grouping factors are denoted `g`, `g1`, and `g2`, and covariates and *a priori* known offsets as `x` and `o`.

Figure 1: Common syntax for the `R` packge `lme4` reproduced from Bates et al. (2015, p. 7).

For this exercise, and the rest of the exercises, we are interested in the `lmer` function of `lme4`. However, by default, the `lme4` package does not provide *p*-values. However, we can rely on the `lmerTest` package that will "mask" the `lmer` function of `lme4` and also include *p*-values for the model parameters in the output. In other words, the `lmerTest` package provides functions that are identical to those of `lme4` (e.g., `lmer`) in terms of input (i.e., arguments), but also include *p*-values in the output. As far as we are concerned, we work with the `lmerTest::lmer` as if we are working with the `lme4::lmer`. Therefore, for the rest of the lab, we are going to proceed with package `lmerTest` loaded.

```
# Install packages.
install.packages("lmerTest")


# Load package.
library(lmerTest)
```

1. Estimate the the *unconditional means model* (i.e., as `model_a`). In this model, the variable `alcuse` (i.e., alcohol use) is the dependent variable, which is only predicted by the intercept.
    - *Tip.* Recall how intercepts are modeled in simple linear regression, and how to allow for the intercepts to vary across individuals.

The model we are about to fit is what we call the **intercept-only** model. In equations, the model takes the

follow form:

Level 1:

$$\text{alcuse}_{ij} = \pi_{0i} + \varepsilon_{ij}$$
$$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2_\varepsilon)$$

Level 2:

$$\pi_{0i} = \gamma_{00} + \zeta_{0i}$$
$$\zeta_{0i} \sim \mathcal{N}(0, \sigma^2_0)$$

We can run the model using the code below. Note that we use the `REML = FALSE` argument to tell `lme4` to use the maximum-likelihood (ML) approach (i.e., the same estimation method used to obtain the results in the lecture slides).

```
# Model syntax.
model_a <- lmer(alcuse ~ 1 + (1 | id), data = data, REML = FALSE)

# Or simpler.
model_a <- lmer(alcuse ~ (1 | id), data = data, REML = FALSE)

# Model summary.
summary(model_a)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: alcuse ~ (1 | id)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    676.2    686.7   -335.1    670.2      243
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8865 -0.3076 -0.3067  0.6137  2.8567
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 0.5639   0.7509
##  Residual             0.5617   0.7495
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  0.92195    0.09571 81.99997   9.633 3.97e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Overall, there two main sections in the output that we want to look at. Starting with the `Fixed effects` section, we observe that the grand mean across individuals and measurement occasions in 0.92195. This parameter corresponds to the γ_{00} parameter in the lecture slides (i.e., or the initial status). Next, we take a look at the `Random effects` section where we see two parameters:

- the `intercept` parameter (i.e., σ_0^2) represents our between-person variance component. In other words, this parameter describes the variability in the individual intercepts, i.e., $\sigma_0^2 = 0.5639$.
- the `residual` parameter (i.e., σ_ε^2) represents our within-person variance component. In other words, this parameter describes how much variance is unexplained, after we allowed for the intercepts to vary, i.e., $\sigma_\varepsilon^2 = 0.5617$.

We can extract the coefficients (i.e., in this case only the individual intercepts, or the person-specific means) estimated for each individual via the `coef` function in `R`.

```
# Extract the coefficients for each person.
coef(model_a)
```

We can also compute the R^2 value using the `mitml` package, but make sure you first read the relevant papers to understand how exactly it is computed. Thanks to Stella for mentioning the package!

```
# Install packages.
install.packages("mitml")

# Load package.
library(mitml)

# R^2 values.
multilevelR2(model_a, print = c("RB2"))
```

If you want to see an APA-style table for your model parameters, you can use the package`sjPlot`:

```
# Install package.
install.package("sjPlot")

# Load package.
library(sjPlot)

# Create APA-style table.
tab_model(model_a)
```

2. Calculate the *interclass correlation coefficient* (ICC) from `model_a`.

```
# Print the random effects (i.e., standard deviations).
VarCorr(model_a)
```

```
##  Groups   Name        Std.Dev.
##  id       (Intercept) 0.75091
##  Residual             0.74950
```

```
# Print the random effects (i.e., variances).
print(VarCorr(model_a), comp = "Variance")
```

```
##  Groups   Name        Variance
##  id       (Intercept) 0.56386
##  Residual             0.56175
```

```
# The ICC is the proportion of between-persons variance.
0.56386 / (0.56386 + 0.56175)
```

```
## [1] 0.5009373
```

We obtain the same if we use the function `icc` in the `R` package `performance`, after we install and load it.

```
# Install package.
install.package("performance")

# Load package.
library(performance)
```

Compute the ICC via `performance::icc`.

```
# Compute ICC.
icc(model_a)
```

```
## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.501
##   Conditional ICC: 0.501
```

You probably noticed that I am referencing quite a number of package. Since `lme4` is incredibly popular, many authors have written handy packages and wrappers around `lme4` to facilitate multi-level analysis.

3. Estimate the *unconditional growth model* (i.e., as `model_b`). In this model, allow for random variation in the `age_14` variable, which captures the effect of time.
    - *Note.* The variable `age_14` by subtracting 14 from the variable `age`. Therefore, variable `age_14` holds 0 for age 14, 1 for age 15, and 2 for age 16.

In addition to the previous model, we now include another predictor (i.e., `age_14`) that accounts for the change over time. Furthermore, now we allow both the intercept (i.e., π_{0i}) and the slope for the time predictor (i.e., π_{1i}) to vary by individual. Substantively, this means that each individual gets his or her own initial status and alcohol consumption rate of change over time. In equations, the model takes the follow form:

Level 1:

$$\text{alcuse}_{ij} = \pi_{0i} + \pi_{1i} \times \text{age\_14}_{ij} + \varepsilon_{ij}$$

$$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2_\varepsilon)$$

Level 2:

$$\pi_{0i} = \gamma_{00} + \zeta_{0i}$$

$$\pi_{1i} = \gamma_{10} + \zeta_{1i}$$

$$\begin{bmatrix} \zeta_{0i} \\ \zeta_{1i} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_0 & \sigma_{01} \\ \sigma_{10} & \sigma^2_1 \end{bmatrix} \right)$$

The combined model (i.e., levels one and two put together):

$$\text{alcuse}_{ij} = \gamma_{00} + \gamma_{10} \times \text{age\_14}_{ij} + \left(\zeta_{0i} + \zeta_{1i} \times \text{age\_14}_{ij} + \varepsilon_{ij}\right)$$

We can drop the intercept from the model syntax (i.e., the 1) and `lme4` will still estimate it for us by default. It is only really necessary to mention it, when you explicitly want an intercept-only model.

```r
# Model syntax.
model_b <- lmer(alcuse ~ age_14 + (age_14 | id), data = data, REML = FALSE)

# Model summary.
summary(model_b)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: alcuse ~ age_14 + (age_14 | id)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    648.6    669.6   -318.3    636.6      240
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.47999 -0.38401 -0.07553  0.39001  2.50685
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept) 0.6244   0.7902
##           age_14      0.1512   0.3888   -0.22
##  Residual             0.3373   0.5808
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  0.65130    0.10508 82.00018   6.198 2.19e-08 ***
## age_14       0.27065    0.06245 81.99974   4.334 4.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
## age_14 -0.441
```

Now, in the output for our model we have values for the following parameters:

- **Fixed effects**:
  - `intercept` (i.e., γ_{00}) equal to 0.65130, which represents the average initial status across all participants
  - `age_14`(i.e., γ_{10}) equal to 0.27065, which represents the average true rate of change across all participants
- **Random effects**:
  - `intercept` (i.e., ζ_{0i}) equal to 0.6244, which represents the variance in the individual intercepts (i.e., in the initial alcohol consumption)
  - `age_14` (i.e., the ζ_{1i}) equal to 0.1512, which represents the variance in the individual slopes (i.e., in the rate of change in alcohol consumption)
  - `residual` (i.e., σ^2_ε) equal to 0.3373, which represents the unexplained within-person variance

4. Estimate another model (i.e., `model_c`), where the variable `coa` predicts both the initial status and the rate of change in variable `alcuse`.

    - *Note.* The variable `coa` refers to whether the children belongs to a family with an alcoholic parent, i.e., coded as 1, and 0 otherwise.

We follow the same logic as above, but this time, we an interaction term between `coa` and `age_14`.

```
# Model syntax.
model_c <- lmer(alcuse ~ coa * age_14 + (age_14 | id), data = data, REML = FALSE)

# Model summary.
summary(model_c)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: alcuse ~ coa * age_14 + (age_14 | id)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    637.2    665.2   -310.6    621.2      238
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5480 -0.3880 -0.1058  0.3602  2.3961
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept) 0.4876   0.6983
##           age_14      0.1506   0.3881   -0.22
##  Residual             0.3373   0.5808
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##                             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)                  0.31595    0.13070 81.99989   2.417 0.017846 *
## coaalcoholic parent          0.74321    0.19457 81.99989   3.820 0.000259 ***
## age_14                       0.29296    0.08423 82.00054   3.478 0.000811 ***
## coaalcoholic parent:age_14 -0.04943    0.12539 82.00054  -0.394 0.694448
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) clchlp age_14
## colchlcprnt -0.672
## age_14      -0.460  0.309
## clprnt:g_14  0.309 -0.460 -0.672
```

5. Calculate the proportional reduction in variance in the initial status and the rate of change due to including the `coa` predictor in the model.

```
# Variances for both models.
print(VarCorr(model_b), comp = "Variance")
```

```
##  Groups   Name        Variance Corr
##  id       (Intercept) 0.62436
##           age_14      0.15120  -0.223
##  Residual             0.33729
```

```
print(VarCorr(model_c), comp = "Variance")
```

```
##  Groups   Name        Variance Corr
##  id       (Intercept) 0.48758
##           age_14      0.15060  -0.219
##  Residual             0.33729
```

```
# Initial status.
(0.62436 - 0.48758) / 0.62436
```

```
## [1] 0.2190723
```

```
# Rate of change.
(0.15120 - 0.15060) / 0.15120
```

```
## [1] 0.003968254
```

We see that the reduction in variance is about 22 for the intercepts, but close to 0 for the slopes.

6. Estimate another (i.e., `model_d`) in which the variable `peer` is added to `model_c` to explain the initial status and the rate of change in `alcuse`.

    - *Note.* The variable `peer` is a measure of peer alcohol use.

```
# Model syntax.
model_d <- lmer(alcuse ~ coa * age_14 + peer * age_14 + (age_14 | id), data = data, REML = FALSE)

# Model summary.
summary(model_d)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: alcuse ~ coa * age_14 + peer * age_14 + (age_14 | id)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    608.7    643.7   -294.3    588.7      236
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.59554 -0.40005 -0.07769  0.46003  2.29373
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept) 0.2409   0.4908
##           age_14      0.1391   0.3730   -0.03
##  Residual             0.3373   0.5808
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##                             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)                -0.31651    0.14806 82.00122  -2.138 0.035517 *
```

```
## coaalcoholic parent         0.57917    0.16249 82.00122   3.564 0.000611 ***
## age_14                      0.42943    0.11369 81.99960   3.777 0.000299 ***
## peer                        0.69430    0.11153 82.00121   6.225 1.95e-08 ***
## coaalcoholic parent:age_14 -0.01403    0.12477 81.99959  -0.112 0.910729
## age_14:peer                -0.14982    0.08564 81.99959  -1.749 0.083975 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) clchlp age_14 peer   cp:_14
## colchlcprnt -0.371
## age_14      -0.436  0.162
## peer        -0.686 -0.162  0.299
## clprnt:g_14  0.162 -0.436 -0.371  0.071
## age_14:peer  0.299  0.071 -0.686 -0.436 -0.162
```

7. Calculate the proportional reduction in variance in the initial status and the rate of change due to including the `peer` predictor in the model.

```
# Variances for both models.
print(VarCorr(model_c), comp = "Variance")
```

```
##  Groups   Name        Variance Corr
##  id       (Intercept) 0.48758
##           age_14      0.15060  -0.219
##  Residual             0.33729
```

```
print(VarCorr(model_d), comp = "Variance")
```

```
##  Groups   Name        Variance Corr
##  id       (Intercept) 0.24090
##           age_14      0.13912  -0.033
##  Residual             0.33729
```

```
# Initial status.
(0.48758 - 0.24090) / 0.48758
```

```
## [1] 0.5059272
```

```
# Rate of change.
(0.15060 - 0.13912) / 0.15060
```

```
## [1] 0.07622842
```

Including the variable `peer`, we see that the reduction in variance is about 50 for the intercepts, and about 8 for the slopes.

8. Estimate another model (i.e., `model_e`), in which the non-significant effect of variable `coa` on the rate of change is removed.

Now, we are removing the interaction between `coa` and the rate of change (i.e., the slope).

```
# Model syntax.
model_e <- lmer(alcuse ~ coa + peer * age_14 + (age_14 | id), data = data, REML = FALSE)


# Model summary.
```

```
summary(model_e)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: alcuse ~ coa + peer * age_14 + (age_14 | id)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    606.7    638.3   -294.4    588.7      237
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.59554 -0.40414 -0.08352  0.45550  2.29975
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept) 0.2409   0.4908
##           age_14      0.1392   0.3730   -0.03
##  Residual             0.3373   0.5808
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##                     Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)         -0.31382    0.14611 85.28183  -2.148 0.034569 *
## coaalcoholic parent  0.57120    0.14623 81.99992   3.906 0.000192 ***
## peer                 0.69518    0.11126 82.65467   6.249 1.72e-08 ***
## age_14               0.42469    0.10559 82.00024   4.022 0.000128 ***
## peer:age_14         -0.15138    0.08451 82.00024  -1.791 0.076957 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) clchlp peer   age_14
## colchlcprnt -0.338
## peer        -0.709 -0.146
## age_14      -0.410  0.000  0.351
## peer:age_14  0.334  0.000 -0.431 -0.814
```

9. Estimate another model (i.e., `model_f`) based on `model_e`, but with intercepts that describe a child of non-alcoholic parents with an average value of `peer` (i.e., use the centered variable `cpeer`).

To do this, replace the `peer` variable with a centered version of the same variable (i.e., `cpeer` in our data set).

```
# Model syntax.
model_f <- lmer(alcuse ~ coa + cpeer * age_14 + (age_14 | id), data = data, REML = FALSE)

# Model summary.
summary(model_f)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: alcuse ~ coa + cpeer * age_14 + (age_14 | id)
##    Data: data
##
```

```
##      AIC      BIC   logLik deviance df.resid
##    606.7    638.3   -294.4    588.7      237
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.59554 -0.40414 -0.08352  0.45550  2.29975
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  id       (Intercept) 0.2409   0.4908
##           age_14      0.1392   0.3730   -0.03
##  Residual             0.3373   0.5808
## Number of obs: 246, groups:  id, 82
##
## Fixed effects:
##                     Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)          0.39387    0.10354 90.27911   3.804 0.000259 ***
## coaalcoholic parent  0.57120    0.14623 81.99992   3.906 0.000192 ***
## cpeer                0.69518    0.11126 82.65467   6.249 1.72e-08 ***
## age_14               0.27058    0.06127 82.00022   4.416 3.04e-05 ***
## cpeer:age_14        -0.15138    0.08451 82.00022  -1.791 0.076957 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) clchlp cpeer  age_14
## colchlcprnt -0.637
## cpeer        0.094 -0.146
## age_14      -0.336  0.000  0.000
## cpeer:ag_14  0.000  0.000 -0.431  0.001
```

10. Perform a *Likelihood-Ratio Test* (LRT) in which you simultaneously compare `model_c`, `model_d`, and `model_e`. What do you conclude?

```
# Perform the LRT.
anova(model_c, model_d, model_e)
```

```
## Data: data
## Models:
## model_c: alcuse ~ coa * age_14 + (age_14 | id)
## model_e: alcuse ~ coa + peer * age_14 + (age_14 | id)
## model_d: alcuse ~ coa * age_14 + peer * age_14 + (age_14 | id)
##         npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## model_c    8 637.20 665.25 -310.60   621.20
## model_e    9 606.70 638.25 -294.35   588.70 32.4993  1  1.192e-08 ***
## model_d   10 608.69 643.74 -294.35   588.69  0.0126  1     0.9105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that `model_c` with only `coa` (i.e., so the constrained model in which the effect of `peer` in initial status and rate of change is set to 0) does not fit equally well as the unconstrained model with both `coa` and `peer` (i.e. *LRT* test is significant). So, we prefer the more elaborate `model_d`. However, `model_e`, with

nonsignificant effect of `coa` on the rate of change removed (i.e., constrained, simpler model), fits equally well as the more elaborate `model_d`. So, we prefer the constrained model `model_e` because that model does not fit worse than the model with the nonsignificant effect of `coa` included.

References

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using Lme4. *Journal of Statistical Software*, *67*(1). https://doi.org/10.18637/jss.v067.i01