

An overview of complex error structure

(Rough draft; please report any typos and errors.)

Jameson Quinn, Luke Miratrix

2019-11-21

In Unit 6, Lecture 3 we talked about how we can model residuals around an overall population model using different specified structures on the correlation matrices for the students. This handout extends those topics, using the Raudenbush and Bryk Chapter 6 example on National Youth Survey data on deviant attitudes. We're going to do a few things:

1. Reproduce the models in the book, showing you how to get them in R, using the commands `lme` and `gls`.
2. Discuss the relationship between `lme` and `gls`, and what it actually means when you include a `gls`-like “correlation” argument when calling `lme`. To make a long story short: `gls` is cleaner and more principled from a mathematical point of view, but in practice you will probably prefer hybrid calls using `lme`.
3. Give two ways this stuff can actually be useful – heteroskedasticity and AR[1] – and show how to fit realistic models with either and both. We'll interpret and check significance of parameters as appropriate.

National Youth Survey running example

Our running example is the data as described in Raudenbush and Bryk, and we follow the discussion on page 190. These data are the first cohort of the National Youth Survey (NYS). This data comes from a survey in which the same students were asked yearly about their acceptance of 9 “deviant” behaviors (such as smoking marijuana, stealing, etc.). The study began in 1976, and followed two cohorts of children, starting at ages 11 and 14 respectively. We will analyze the first 5 years of data.

At each time point, we have measures of:

- ATIT, the attitude towards deviance, with higher numbers implying higher tolerance for deviant behaviors.
- EXPO, the “exposure”, based on asking the children how many friends they had who had engaged in each of the behaviors. Both of these numbers have been transformed to a logarithmic scale to reduce skew.

For each student, we have:

- Gender (binary)
- Minority status (binary)
- Family income, in units of \$10K.

One reasonable research question would to describe how the cohort evolved. For this question, the parameters of interest would be the average attitudes at each age. Standard deviations and intrasubject correlations are, as is often but not always the case, simply nuisance parameters. Still, the better we can do at realistically modeling these nuisance parameters, the more precision we will have for the measures of interest, and the power we will have to test relevant hypotheses.

Getting the data ready

We'll focus on the first cohort, from ages 11-15. First, let's read the data.

Note that this table is in “wide format”. That is, there is only one row for each student, with all the different observations for that student in different columns of that one row.

```
nyswide = read.csv("nyswide.csv")
head(nyswide)
```

##	ID	ATTIT.11	EXPO.11	ATTIT.12	EXPO.12	ATTIT.13	EXPO.13	ATTIT.14	EXPO.14
## 1	3	0.11	-0.37	0.20	-0.27	0.00	-0.37	0.00	-0.27
## 2	8	0.29	0.42	0.29	0.20	0.11	0.42	0.51	0.20
## 3	9	0.80	0.47	0.58	0.52	0.64	0.20	0.75	0.47
## 4	15	0.44	0.07	0.44	0.32	0.89	0.47	0.75	0.26
## 5	33	0.20	-0.27	0.64	-0.27	0.69	-0.27	NA	NA
## 6	45	0.11	0.26	0.37	-0.17	0.37	0.14	0.37	0.14

##	ATTIT.15	EXPO.15	FEMALE	MINORITY	INCOME
## 1	0.11	-0.17	1	0	3
## 2	0.69	0.20	0	0	4
## 3	0.98	0.47	0	0	3
## 4	0.80	0.47	0	0	4
## 5	0.11	0.07	1	0	4
## 6	0.69	0.32	1	0	4

For our purposes, we want it in “long format”. The `reshape()` command does this for us (reshape allows us to reshape two variables at once, as compared to `gather()` from tidyverse:

```
nys1.na = reshape(nyswide, direction="long", #we want it in long format
                  varying=list(ATTIT=paste("ATTIT",11:15,sep="."),
                                EXPO=paste("EXPO",11:15,sep=".") ),
                  v.names=c("ATTIT","EXPO"), idvar="ID", timevar="AGE", times=11:15)

# Drop missing ATTIT values
nys1 = nys1.na[!is.na(nys1.na$ATTIT),]

head( nys1 )
```

##	ID	FEMALE	MINORITY	INCOME	AGE	ATTIT	EXPO
## 3.11	3	1	0	3	11	0.11	-0.37
## 8.11	8	0	0	4	11	0.29	0.42
## 9.11	9	0	0	3	11	0.80	0.47
## 15.11	15	0	0	4	11	0.44	0.07
## 33.11	33	1	0	4	11	0.20	-0.27
## 45.11	45	1	0	4	11	0.11	0.26

Note, the `paste` command makes the sequence `c("ATTIT.12", "ATTIT.13", ...)` to autogenerate our variable names to reshape:

```
paste("ATTIT",11:15,sep=".")
```

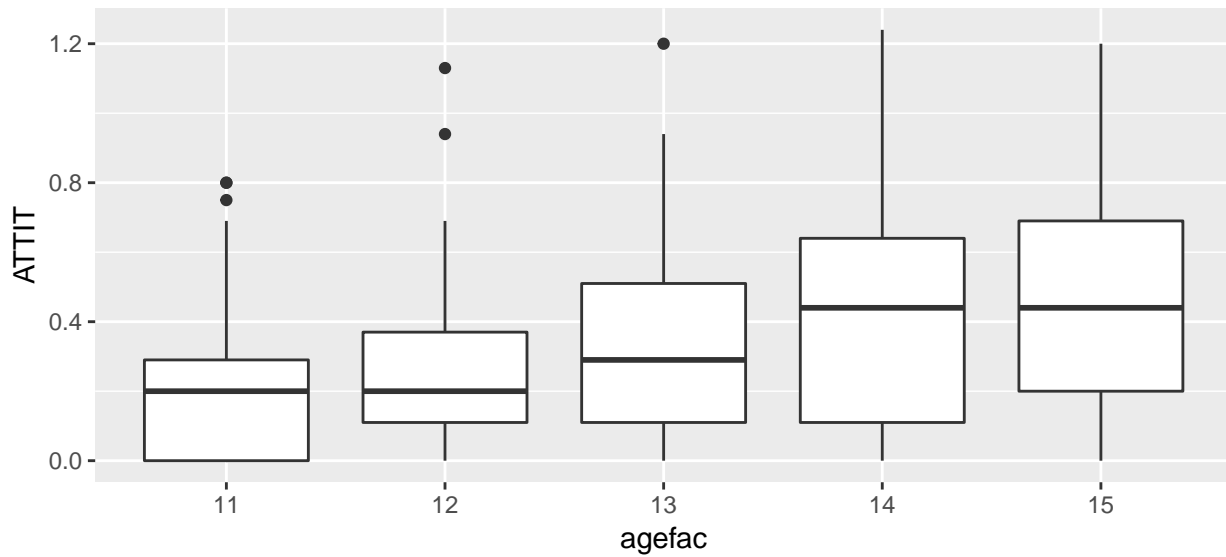
```
## [1] "ATTIT.11" "ATTIT.12" "ATTIT.13" "ATTIT.14" "ATTIT.15"
```

We also need to make our age a factor so it is treated appropriately as an indicator of what wave the data was collected in.

```
nys1$agefac = as.factor(nys1$AGE)
```

Just to get a sense of the data, let’s plot each age as a boxplot

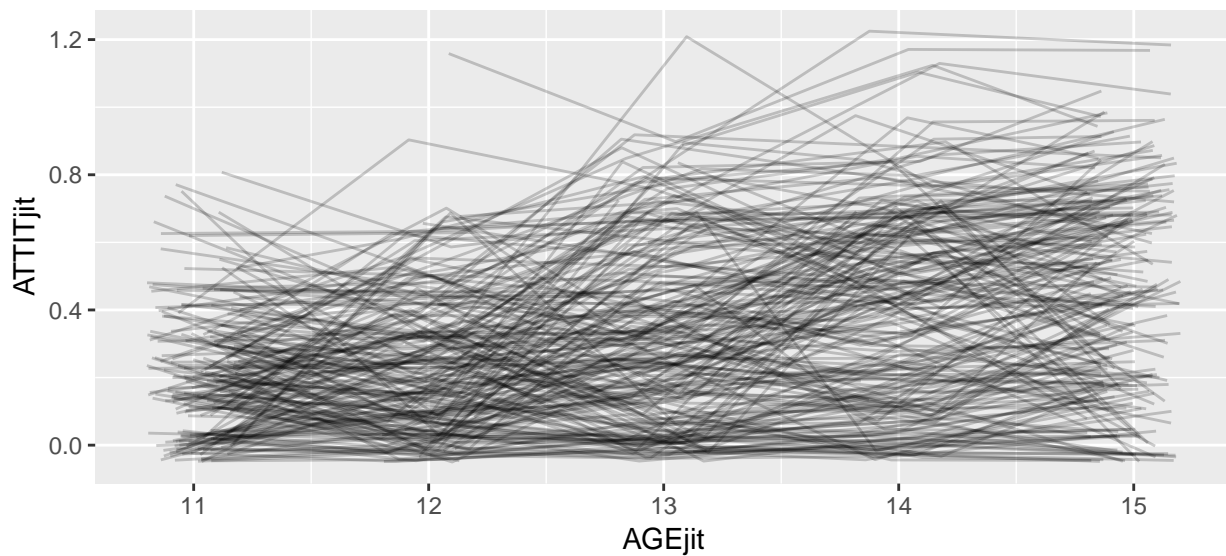
```
ggplot( nys1, aes( agefac, ATTIT ) ) +
  geom_boxplot()
```



Note some features of the data: First, we see that ATTIT goes up over time. Second, we see the variation of points also goes up over time. This is heteroskedasticity.

If we plot individual lines we have

```
nys1$AGEjit = jitter(nys1$AGE)
nys1$ATTITjit = jitter(nys1$ATTIT, amount=0.05)
ggplot( filter( nys1, complete.cases(nys1) ), aes( AGEjit, ATTITjit, group=ID ) ) +
  geom_line( alpha=0.2 )
```



Note how we have correlation of residuals, in that some students are systematically low and some are systematically higher (although there is a lot of bouncing around).

Representation of error structure

In our data, we have 5 observations y_{it} for each subject i at 5 fixed times $t = 1$ through $t = 5$. Within each person i (where person is our Level-2 group, and time is our Level-1),

$$\begin{pmatrix} y_{i1} \\ y_{i2} \\ y_{i3} \\ y_{i4} \\ y_{i5} \end{pmatrix} \sim N \left[\begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{pmatrix}, \begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} & \tau_{14} & \tau_{15} \\ . & \tau_{22} & \tau_{23} & \tau_{24} & \tau_{25} \\ . & . & \tau_{33} & \tau_{34} & \tau_{35} \\ . & . & . & \tau_{44} & \tau_{45} \\ . & . & . & . & \tau_{55} \end{pmatrix} \right] = N[\mu, \Sigma]$$

Note that the key parts here are the correlations between the residuals at different times. We call our entire covariance matrix Σ . This matrix describes how the residuals within a single individual (with 5 time points of observation) are correlated. The mean vector can easily be separated out:

$$\begin{pmatrix} y_{i1} \\ y_{i2} \\ y_{i3} \\ y_{i4} \\ y_{i5} \end{pmatrix} = \begin{pmatrix} \mu_{1i} \\ \mu_{2i} \\ \mu_{3i} \\ \mu_{4i} \\ \mu_{5i} \end{pmatrix} + \begin{pmatrix} \epsilon_{1i} \\ \epsilon_{2i} \\ \epsilon_{3i} \\ \epsilon_{4i} \\ \epsilon_{5i} \end{pmatrix}$$

$$\begin{pmatrix} \epsilon_{1i} \\ \epsilon_{2i} \\ \epsilon_{3i} \\ \epsilon_{4i} \\ \epsilon_{5i} \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} & \tau_{14} & \tau_{15} \\ . & \tau_{22} & \tau_{23} & \tau_{24} & \tau_{25} \\ . & . & \tau_{33} & \tau_{34} & \tau_{35} \\ . & . & . & \tau_{44} & \tau_{45} \\ . & . & . & . & \tau_{55} \end{pmatrix} \right]$$

Our regression model would give us the mean vector for any given student (e.g., it would be $X'\beta$ for some covariate matrix (design matrix) X and fixed effect parameter vector β . X would have one row per time point and time would be one of the columns, to give our predictions for our 5 time points.

Our error structure model gives us the distribution of the $(\epsilon_{1i}, \dots, \epsilon_{5i})$ for that student. Different ideas about the data generating process lead to different correlation structures here. We saw a couple of those in class.

Reproducing Chapter 6 examples

The above provides a framework for thinking about grouped data: each group is a small world with a linear prediction line and a collection of residuals around that line. Under this view, we specify a specific structure on how the residuals relate to each other. (E.g., for classic OLS we would have i.i.d. normally distributed residuals, represented as our Σ being a diagonal matrix with σ^2 along the diagonal and 0s everywhere else). In R, once we determine what structure we want, we can fit models based on parameterized correlation matrices using the “lme” command from the “nlme” package (You may need to first call `install.packages("nlme")` to get this package). Let’s load the package now:

```
library(nlme)
```

Recall that all of these models include just the single fixed effect (besides intercept) of a linear term on age.

Compound symmetry (random intercept model)

A “compound symmetry” residual covariance structure (all diagonal elements equal, all off-diagonal elements equal) is actually equivalent to a random intercepts model. Thus, there are 2 ways to get this same model:

```
modelRE = lme(ATTIT ~ AGE,
               data=nys1,
               random=~1|ID )
```

and

```
modelCompSymm = gls(ATTIT ~ AGE,
                     data=nys1,
                     correlation=corCompSymm(form=~AGE|ID) )
```

For reference, using the `lme4` package we again have (we use `lme4::` in front of `lmer` to avoid loading the `lme4` package fully):

```
modelRE.lme4 = lme4::lmer(ATTIT ~ AGE + (1|ID), data=nys1 )
```

We can get the correlation matrix for individuals #3:

```
myVarCovs = getVarCov(modelRE,type="marginal", individual=3)
myVarCovs
```

```
## ID 9
## Marginal variance covariance matrix
##      1      2      3      4      5
## 1 0.066450 0.034113 0.034113 0.034113 0.034113
## 2 0.034113 0.066450 0.034113 0.034113 0.034113
## 3 0.034113 0.034113 0.066450 0.034113 0.034113
## 4 0.034113 0.034113 0.034113 0.066450 0.034113
## 5 0.034113 0.034113 0.034113 0.034113 0.066450
## Standard Deviations: 0.25778 0.25778 0.25778 0.25778 0.25778
```

If we look at an individual #5, who only has 4 timepoints we get a 4×4 matrix:

```
getVarCov(modelRE,type="marginal", individual=5)
```

```
## ID 33
## Marginal variance covariance matrix
##      1      2      3      4
## 1 0.066450 0.034113 0.034113 0.034113
## 2 0.034113 0.066450 0.034113 0.034113
## 3 0.034113 0.034113 0.066450 0.034113
## 4 0.034113 0.034113 0.034113 0.066450
## Standard Deviations: 0.25778 0.25778 0.25778 0.25778
```

(Other individuals are the same, if they have the same number of time points, given our model.)

Comparing the models

These are two very different ways of specifying the same thing, and the parameter estimates we get out are also not the same. Compare the two summary printouts:

```
summary(modelRE)
```

```
## Linear mixed-effects model fit by REML
## Data: nys1
##      AIC      BIC    logLik
## -204.9696 -185.0418 106.4848
##
## Random effects:
## Formula: ~1 | ID
##      (Intercept) Residual
## StdDev:   0.1846979 0.1798237
##
## Fixed effects: ATTIT ~ AGE
```

```
##               Value Std.Error DF   t-value p-value
## (Intercept) -0.5099954 0.05358498 839 -9.517505      0
## AGE          0.0644387 0.00398784 839 16.158810      0
## Correlation:
##   (Intr)
## AGE -0.969
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.90522949 -0.64353962 -0.01388485  0.60377631  3.26938845
##
## Number of Observations: 1079
## Number of Groups: 239
```

and

```
summary(modelCompSymm)
```

```
## Generalized least squares fit by REML
## Model: ATTIT ~ AGE
## Data: nys1
##      AIC      BIC    logLik
## -204.9696 -185.0418 106.4848
##
## Correlation Structure: Compound symmetry
## Formula: ~AGE | ID
## Parameter estimate(s):
##      Rho
## 0.5133692
##
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept) -0.5099954 0.05358498 -9.517505      0
## AGE          0.0644387 0.00398784 16.158810      0
##
## Correlation:
##   (Intr)
## AGE -0.969
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.77123071 -0.77132300 -0.06434029  0.71151900  3.38387884
##
## Residual standard error: 0.2577787
## Degrees of freedom: 1079 total; 1077 residual
```

These do not look very similar, do they? But wait:

```
logLik(modelCompSymm)
```

```
## 'log Lik.' 106.4848 (df=4)
```

```
logLik(modelRE)
```

```
## 'log Lik.' 106.4848 (df=4)
```

```
logLik(modelRE.lme4)
```

```
## 'log Lik.' 106.4848 (df=4)
```

```
AIC( modelCompSymm )
```

```
## [1] -204.9696
```

```
AIC( modelRE )
```

```
## [1] -204.9696
```

```
AIC( modelRE.lme4 )
```

```
## [1] -204.9696
```

In fact, they have the same AIC, etc., because they are equivalent models.

The lesson is that it's actually quite hard to see the correspondence between a familiar random-effects model and an equivalent model expressed in terms of a covariance matrix. Sure, we could do a bunch of math and see that in the end they are the same; but that math is already daunting here, and this is the simplest possible situation. The fitted parameters of a covariance-based model are just really hard to interpret in familiar terms.

Autoregressive error structure (AR[1])

One typical structure used for longitudinal data is the “autoregressive” structure. The idea is threefold:

1. $Var(u_{it}) = \sigma^2$ - that is, overall marginal variance is staying constant.
2. $Cor(u_{it}, u_{i(t-1)}) = \rho$ - that is, residuals are a little bit “sticky” over time so residuals from nearby time points tend to be similar.
3. $E(u_{it}|u_{i(t-1)}, u_{i(t-2)}) = E(u_{it}|u_{i(t-1)})$ - that is, the only way the two-periods-ago measurement tells you anything about the current one is through the intermediate one, with no longer-term effects or “momentum”.

In this case, the unconditional two-step correlation $Cor(u_{it}, u_{i(t-2)})$ is also easy to calculate. Intuitively, we can say that a portion ρ of the residual “is the same” after each step, so that after two steps the portion that “is the same” is ρ of ρ , or ρ^2 . Clearly, then, after three steps the correlation will be ρ^3 , and so on. In other words, the part that “is the same” is decaying in an exponential pattern. Indeed, one could show that (3.), above, requires the correlated part to decay in a memoryless pattern, leaving the Exponential and Hypergeometric distributions (which both show exponential decay) among the few options.

Thus, the within-subject correlation structure implied by these postulates is:

$$\begin{pmatrix} u_{1i} \\ u_{2i} \\ u_{3i} \\ u_{4i} \\ \dots \\ u_{ni} \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \dots & \rho^{n-1} \\ . & 1 & \rho & \rho^2 & \dots & \rho^{n-2} \\ . & . & 1 & \rho & \dots & \rho^{n-3} \\ . & . & . & 1 & \dots & \rho^{n-4} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ . & . & . & . & \dots & 1 \end{pmatrix} \right]$$

As you can see, this structure takes advantage of the temporal nature of the data sequence to parameterize the covariance matrix with only two underlying parameters: σ and ρ . By contrast, a random intercept model needs the overall σ and variance of intercepts τ —also two parameters! Same complexity, different structure.

Fitting the AR[1] covariance structure

To get a true AR[1] residual covariance structure, we need to leave the world of hierarchical models, and thus use the command `gls`. This is just what we've discussed in class. However, later on in this document, we'll see how to add AR[1] structure on top of a hierarchical model, which is messier from a theoretical point of view, but often more useful and interpretable in practice.

```
modelAR1 = gls(ATTIT ~ AGE,
               data=nys1,
               correlation=corAR1(form=~AGE|ID) )

summary(modelAR1)

## Generalized least squares fit by REML
##   Model: ATTIT ~ AGE
##   Data: nys1
##           AIC      BIC    logLik
##   -250.4103 -230.4826 129.2051
##
## Correlation Structure: ARMA(1,0)
## Formula: ~AGE | ID
## Parameter estimate(s):
##   Phil
## 0.6159857
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept) -0.4534647 0.07515703 -6.033564      0
## AGE          0.0601205 0.00569797 10.551218      0
##
## Correlation:
##   (Intr)
## AGE -0.987
##
## Standardized residuals:
##           Min           Q1           Med           Q3           Max
## -1.75013168 -0.81139621 -0.03256558  0.74814629  3.40350724
##
## Residual standard error: 0.2561765
## Degrees of freedom: 1079 total; 1077 residual

getVarCov(modelAR1,type="marginal")

## Marginal variance covariance matrix
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.0656260 0.040425 0.024901 0.015339 0.0094485
## [2,] 0.0404250 0.065626 0.040425 0.024901 0.0153390
## [3,] 0.0249010 0.040425 0.065626 0.040425 0.0249010
## [4,] 0.0153390 0.024901 0.040425 0.065626 0.0404250
## [5,] 0.0094485 0.015339 0.024901 0.040425 0.0656260
##   Standard Deviations: 0.25618 0.25618 0.25618 0.25618 0.25618
```

You have to dig around in the large amount of output to find the parameter estimates, but they are there. Phil is the auto-correlation parameter. And the covariance of residuals:


```
summary(modelAR1)$AIC
```

```
## [1] -250.4103
```

Note that the AIC of our AR[1] model is lower by about 45 than the random intercept model; clearly far superior because it is getting nearby residuals being more correlated, while the random intercept model does not do this. Also see the banding structure of the residual correlation matrix.

Random slopes

In theory, a random slopes model could be done with `gls` as well as with `lme`; in practice, it's much more practical just to do it as a hierarchical model with `lme`:

```
modelRS = lme(ATTIT ~ 1 + AGE,
              data=nys1,
              random=~AGE|ID )
```

We have separated our fixed and random components with `lme()`. We first include a formula with only fixed effects, and then give a right-side-only formula with terms similar to what you'd put in parentheses with `lmer()` for the random effects.

Our results:

```
summary(modelRS)
```

```
## Linear mixed-effects model fit by REML
## Data: nys1
##      AIC      BIC    logLik
##   -310.125 -280.2334 161.0625
##
## Random effects:
## Formula: ~AGE | ID
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 0.51024133 (Intr)
## AGE          0.05038614 -0.98
## Residual    0.16265428
##
## Fixed effects: ATTIT ~ 1 + AGE
##           Value Std.Error DF   t-value p-value
## (Intercept) -0.5133250 0.05834087 839 -8.798719      0
## AGE          0.0646849 0.00492904 839 13.123228      0
## Correlation:
##      (Intr)
## AGE -0.981
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
##  -2.87852426 -0.55971198 -0.07521191  0.57495076  3.45648137
##
## Number of Observations: 1079
## Number of Groups: 239
```

```
getVarCov(modelRS,type="marginal", individual=3)
```

```
## ID 9
```

```
## Marginal variance covariance matrix
##      1      2      3      4      5
## 1 0.039649 0.015922 0.018650 0.021379 0.024108
## 2 0.015922 0.047646 0.026457 0.031725 0.036992
## 3 0.018650 0.026457 0.060720 0.042070 0.049876
## 4 0.021379 0.031725 0.042070 0.078872 0.062760
## 5 0.024108 0.036992 0.049876 0.062760 0.102100
## Standard Deviations: 0.19912 0.21828 0.24641 0.28084 0.31953
```

```
summary(modelRS)$AIC
```

```
## [1] -310.125
```

The first thing to note is the residual covariance matrix comes from the structure of the random intercept and random slope. If you squint hard enough at it, you can begin to see the linear structures in its diagonal and off-diagonal elements. If you graphed it, those structures would jump out more clearly. But in practice, it's much easier to think of things in terms of the hierarchical model, not in terms of linear structures in a covariance matrix.

Note also that the AIC has dropped by another 60 points or so; we're continuing to improve the model.

Random slopes with heteroskedasticity

Relaxing the homoskedasticity assumption in the random slopes model leaves us a bit in between worlds. We're not fully into the world of GLS, because there are still random effects; but we're not fully in the world of hierarchical models because there is structure in the residuals within groups. We'll talk more about this compromise below; for now, let's just do it.

```
modelRSH = lme(ATTIT ~ AGE,
               data=nys1,
               random=~AGE|ID,
               weights=varIdent(form=~1|agefac) )
```

The key line is the `varIdent` line: we are saying each age factor level gets its own weight (rescaling) of the residuals—this is heteroskedasticity. In particular, the above says our residual variance will be weighted by a weight for each age factor, so each age level effectively gets its own variance. This is where these models start to get a bit exciting—we have random slopes, and then heteroskedastic residuals (homoskedastic for any given age level), all together. Our fit model:

```
summary(modelRSH)
```

```
## Linear mixed-effects model fit by REML
## Data: nys1
##      AIC      BIC    logLik
## -312.5801 -262.7608 166.2901
##
## Random effects:
## Formula: ~AGE | ID
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 0.57693635 (Intr)
## AGE         0.05431371 -0.979
## Residual    0.14054173
##
## Variance function:
## Structure: Different standard deviations per stratum
```

```
## Formula: ~1 | agefac
## Parameter estimates:
##      11      12      13      14      15
## 1.0000000 1.1956045 1.3095883 1.1255244 0.9802308
## Fixed effects: ATTIT ~ AGE
##           Value Std.Error DF   t-value p-value
## (Intercept) -0.4929012 0.05715888 839 -8.623352      0
## AGE          0.0631404 0.00483385 839 13.062119      0
## Correlation:
## (Intr)
## AGE -0.981
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.91635759 -0.54982016 -0.07583344  0.54829414  3.23703121
##
## Number of Observations: 1079
## Number of Groups: 239
```

Note how we have 5 parameter estimates for the residuals, listed under `agefac`. It appears as if we have more variation in age 13 than other ages. Age 11, the baseline, is 1.0; it is our reference scaling. These numbers are all scaling the overall residual variance parameter σ^2 of 0.1405².

For looking at the covariance structure of the residuals, at this point we have to warn you: there appears to be a bug in `getVarCov` which rears its head when you use the `weights` argument to either `lme` or `gls`. It has to do with the order of the rows of the data set, something which obviously should not matter; and it means that you get simply wrong numbers for marginal variances, though correlations should still be correct. Jameson Quinn wrote a fix for this function, which is in the source file `getVarCov2.R`. This fix is used below, using the `source` command to load the fixed version of the function. But beware: while we've tested that this fix gives the right answers for this data set, and that it gives the right answers when the `weights` argument is not used (that is, when the old version was already right), we have not done the extensive checking it would take to say we trust it in all cases. This bug was reported back in 2016, and hopefully it will be fixed in later versions of R; but for now, tread with care, and double-check that the numbers you're getting make sense.

```
source('getVarCov2.R')
myVarCov = getVarCovFixedLme(modelRSH,type="marginal", individual=3)
myVarCov
```

```
## ID 9
## Marginal variance covariance matrix
##      1      2      3      4      5
## 1 0.034915 0.016947 0.018731 0.020515 0.022300
## 2 0.016947 0.049916 0.026415 0.031150 0.035884
## 3 0.018731 0.026415 0.067975 0.041784 0.049468
## 4 0.020515 0.031150 0.041784 0.077440 0.063052
## 5 0.022300 0.035884 0.049468 0.063052 0.095615
## Standard Deviations: 0.18685 0.22342 0.26072 0.27828 0.30922
```

We get lists of matrices back from our call. We can convert any one to a correlation matrix:

```
cov2cor(myVarCov[[1]])

##      1      2      3      4      5
## 1 1.0000000 0.4059443 0.3844926 0.3945439 0.3859519
## 2 0.4059443 1.0000000 0.4534860 0.5010154 0.5194178
## 3 0.3844926 0.4534860 1.0000000 0.5759074 0.6136043
## 4 0.3945439 0.5010154 0.5759074 1.0000000 0.7327485
```

```
## 5 0.3859519 0.5194178 0.6136043 0.7327485 1.0000000
```

And our AIC:

```
summary(modelRSH)$AIC
```

```
## [1] -312.5801
```

No amount of squinting will show the structure in that covariance matrix. But when you convert to a correlation matrix, you can again squint and begin to see the linear structures in its diagonal and off-diagonal elements. The same comment as above still applies: in practice, it's much easier to think of things in terms of the hierarchical model, and only read the diagonals of the covariance matrix.

Note that the AIC has dropped by only another 2.5 points or so; that corresponds to the idea that if one of these two models were exactly true, the odds are about $e^{2.5/2} \cong 3.5$ in favor of the more complex model. Aside from the fact that that premise is silly – we are pretty sure that neither of these models is the exact truth; and in that case, something like BIC would probably be better than AIC – those odds are also pretty weak; the simpler model is probably better here.

Here's the reported BICs, by the way: -280.2334145 for the homoskedastic one, and -262.7607678 for the heteroskedastic. As we expected, the simpler model wins that fight. (Though what N to use for BIC is sometimes not obvious with hierarchical models, so you can't trust those numbers too much. We will discuss AIC and BIC more later in the course.)

Fully unrestricted model

OK, let's go whole hog, and fit the unrestricted model. Again, this means leaving the world of hierarchical models and using gls.

```
modelUnrestricted = gls(ATTIT ~ AGE,
  data=nys1,
  correlation=corSymm(form=~1|ID),
  weights=varIdent(form=~1|agefac) )
```

```
summary(modelUnrestricted)
```

```
## Generalized least squares fit by REML
## Model: ATTIT ~ AGE
## Data: nys1
##      AIC      BIC  logLik
## -319.262 -234.5691 176.631
##
## Correlation Structure: General
## Formula: ~1 | ID
## Parameter estimate(s):
## Correlation:
##  1      2      3      4
## 2 0.458
## 3 0.372 0.511
## 4 0.441 0.437 0.663
## 5 0.468 0.443 0.597 0.764
## Variance function:
## Structure: Different standard deviations per stratum
## Formula: ~1 | agefac
## Parameter estimates:
##      11      12      13      14      15
```

```
## 1.000000 1.118479 1.414270 1.522511 1.560072
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept) -0.4557089 0.05465560 -8.337826      0
## AGE          0.0597274 0.00458343 13.031153      0
##
## Correlation:
##      (Intr)
## AGE -0.979
##
## Standardized residuals:
##      Min          Q1          Med          Q3          Max
## -1.482607475 -0.809004625 -0.006791829  0.840804662  4.082257854
##
## Residual standard error: 0.1903187
## Degrees of freedom: 1079 total; 1077 residual
```

And our residual structure:

```
source('getVarCov2.R')
myvc = getVarCovFixedGls(modelUnrestricted,type="marginal", individual=3)
myvc
```

```
## Marginal variance covariance matrix
##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.036221 0.018541 0.019071 0.024335 0.026466
## [2,] 0.018541 0.045313 0.029251 0.026924 0.027989
## [3,] 0.019071 0.029251 0.072448 0.051704 0.047736
## [4,] 0.024335 0.026924 0.051704 0.083962 0.065764
## [5,] 0.026466 0.027989 0.047736 0.065764 0.088156
## Standard Deviations: 0.19032 0.21287 0.26916 0.28976 0.29691
```

And AIC:

```
AIC( modelUnrestricted )
```

```
## [1] -319.262
```

This unrestricted covariance and correlation matrices have the same structures discussed in the book and in class. The AIC has improved by another 6 or 7 points; that's marginally “significant”, but in practice probably not substantial enough to make up for the loss of interpretability. The lesson we should take from that is that there's not a whole lot of room for improvement just by tinkering with the residual covariance structure; if we want a much better model, we would have to add new fixed or random effects; perhaps other covariates or perhaps a quadratic term in time.

Mixing AR[1] and Random Slopes

Let's look at an AR1 residual structure along with some covariates in our main model. The following has AR[1] and *also* a random intercept and slope:

```
model11 = lme(fixed=ATTIT ~ EXPO + FEMALE + MINORITY + log(INCOME + 1),
              data=nys1,
              random=~AGE|ID,
              correlation=corAR1() )
```

Compare to same model without AR1 correlation

```
modell1simple = lme(fixed=ATTIT ~ EXPO + FEMALE + MINORITY + log(INCOME + 1),
  data=nys1,
  random=~AGE|ID )
screenreg( list( modell1, modell1simple ) )
```

```
##
## =====
##               Model 1      Model 2
## -----
## (Intercept)      0.33 ***      0.32 ***
##                (0.04)      (0.04)
## EXPO              0.36 ***      0.37 ***
##                (0.03)      (0.03)
## FEMALE            -0.01          -0.01
##                (0.02)      (0.02)
## MINORITY          -0.06 *        -0.06 *
##                (0.03)      (0.03)
## log(INCOME + 1)  -0.01          -0.01
##                (0.02)      (0.02)
## -----
## AIC              -359.46      -350.54
## BIC              -309.67      -305.73
## Log Likelihood    189.73      184.27
## Num. obs.         1079        1079
## Num. groups       239         239
## =====
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

The AR1 model has notably lower AIC and thus is better. (A difference of ~9 in AIC, which can be interpreted as a factor of $e^{4.5}$ in odds.) Here are the log likelihoods with degrees of freedom:

```
logLik( modell1 )
```

```
## 'log Lik.' 189.7308 (df=10)
```

```
logLik( modell1simple )
```

```
## 'log Lik.' 184.269 (df=9)
```

Our model is actually kind of mixed up, conceptually. We allowed a random slope on age, and also an autoregressive component by age. Thus, we effectively allowed the covariance matrix to vary in two different ways, at two different levels of our modeling.

In fact, as we've seen in class, any random effects, whether they be on slope or intercept, are actually equivalent to certain ways of varying the [variance-]covariance matrix of the residuals within each group. For instance, random intercepts are equivalent to compound symmetry. Thus, by including both random intercepts and AR1 correlation in the above model, we've effectively fit a model that allows any covariance matrix that can be expressed as a sum of a random slope covariance matrix (with 2 parameters plus a scaling factor) and an AR1 covariance matrix (with 1 parameter plus a scaling factor). That makes 5 degrees of freedom total for our covariance matrix.

But conceptually this model is nice: people have linear growth trends, but vary around those growth trends in an autoregressive way.

The Kitchen sink: building complex models

Which brings us to the next point: how do you actually use this stuff in practice. Ideally, you'd like both the interpretability (and robustness against MAR missingness) of hierarchical models, along with the ability to add additional residual structure such as AR[1] and/or heteroskedastic residuals. The good news is, you can get both. The bad news is, there's a bit of a potential for bias due to overfitting.

For instance, imagine you use both random effects and AR[1]. Say that for a given subject you have 5 time points, and all of them are above the values you would have predicted based on fixed effects alone. That might be explained by an above-average random effect, or by a set of correlated residuals that all came in high. Whichever one of these is the “true” explanation, the MLE will tend to parcel it out between the two. This can lead to downward bias in variance and/or correlation parameter estimates, especially with small numbers of observations per subject.

Still, as long as your focus is on location parameters such as true means or slopes, this can be a good way to proceed. Let's explore this by first fitting a “kitchen sink” model for this data, in which we use all available covariates; and seeing how adding heteroskedasticity, AR[1] structure, or both changes it (or doesn't).

What do we want in this “kitchen sink” model? Let's first fit a very simple random intercept model with fixed effects for gender, minority status, “exposure”, and log(income), to see which of these covariates to focus on. We use the `lmerTest` package to get some early *p*-values for these fixed effects.

```
modelKS0 = lmerTest::lmer(ATTIT ~ FEMALE + MINORITY + log(INCOME + 1) + EXPO + (1|ID), data=nys1)
summary(modelKS0, correlation=FALSE)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: ATTIT ~ FEMALE + MINORITY + log(INCOME + 1) + EXPO + (1 | ID)
## Data: nys1
##
## REML criterion at convergence: -265.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.1840 -0.5922 -0.0797  0.6043  2.6319
##
## Random effects:
## Groups Name Variance Std.Dev.
## ID      (Intercept) 0.01756 0.1325
## Residual          0.03444 0.1856
## Number of obs: 1079, groups: ID, 239
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   3.480e-01  4.195e-02 2.301e+02  8.297    9e-15 ***
## FEMALE        -1.835e-02  2.094e-02 2.327e+02 -0.876    0.3819
## MINORITY      -5.698e-02  2.789e-02 2.279e+02 -2.043    0.0422 *
## log(INCOME + 1) 2.102e-03  2.449e-02 2.272e+02  0.086    0.9317
## EXPO          4.516e-01  2.492e-02 1.041e+03 18.122   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(The `correlation=FALSE` shortens the printout.)

Apparently, MINORITY and EXPO are the covariates with significant effects; minority status is correlated with a lower tolerance for deviance, while “deviant” friends are of course correlated positively with tolerance of deviance. Let's build a few hierarchical models including these in various specifications (can you identify

what models are what? Some of these models are not necessarily good choices). We first center our age so we have meaningful intercepts.

```
nys1$age13 = nys1$AGE - 13

modelKS1 = lme(ATTIT ~ MINORITY,
               data=nys1,
               random=~age13 + EXPO|ID )

modelKS2 = lme(ATTIT ~ MINORITY + age13,
               data=nys1,
               random=~age13 + EXPO|ID )

modelKS3 = lme(ATTIT ~ MINORITY + age13,
               data=nys1,
               random=~EXPO|ID )

modelKS4 = lme(ATTIT ~ MINORITY + age13 + EXPO,
               data=nys1,
               random=~1|ID )
```

And now we examine them:

```
library( texreg )
screenreg( list( modelKS1, modelKS2, modelKS3, modelKS4 ) )
```

```
##
## =====
##               Model 1      Model 2      Model 3      Model 4
## -----
## (Intercept)      0.24 ***      0.31 ***      0.29 ***      0.34 ***
##                (0.01)      (0.01)      (0.01)      (0.01)
## MINORITY         -0.05         -0.05 *      -0.04         -0.06 *
##                (0.02)      (0.02)      (0.03)      (0.03)
## age13                        0.06 ***      0.05 ***      0.05 ***
##                (0.00)      (0.00)      (0.00)      (0.00)
## EXPO                                0.37 ***
##                                (0.02)
## -----
## AIC              -305.22      -374.18      -312.13      -394.06
## BIC              -260.38      -324.37      -277.27      -364.18
## Log Likelihood   161.61      197.09      163.07      203.03
## Num. obs.        1079        1079        1079        1079
## Num. groups      239         239         239         239
## =====
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

OK, Number 4 seems like a pretty good model. Let's see how much it improves when we add AR[1]:

```
modelKS5 = lme(ATTIT ~ MINORITY + age13 + EXPO,
               data=nys1,
               random=~1|ID,
               correlation=corAR1(form=~AGE|ID) )
AIC( modelKS5 )
```

```
## [1] -433.5943
```



```
fixef( modelKS4 )
```

```
## (Intercept)    MINORITY      age13      EXPO
##  0.34054593 -0.05606947  0.04830698  0.36778951
```

```
fixef( modelKS5 )
```

```
## (Intercept)    MINORITY      age13      EXPO
##  0.34083507 -0.05704474  0.04733879  0.35207989
```

Note that the estimates for all the effects are essentially unchanged. However, the AIC is almost 40 points better. Also, because the model has done a better job explaining residual variance, the p-value for the coefficient on MINORITY has dropped from 0.032 to 0.029, as we can see on the summary display below. This is not a large drop, but a noticeable one:

```
summary( modelKS5 )
```

```
## Linear mixed-effects model fit by REML
## Data: nys1
##      AIC      BIC   logLik
## -433.5943 -398.7338 223.7972
##
## Random effects:
## Formula: ~1 | ID
##      (Intercept) Residual
## StdDev:   0.1186009 0.1864592
##
## Correlation Structure: ARMA(1,0)
## Formula: ~AGE | ID
## Parameter estimate(s):
##      Phi1
## 0.3212696
## Fixed effects: ATTIT ~ MINORITY + age13 + EXPO
##              Value Std.Error DF   t-value p-value
## (Intercept)  0.3408351 0.011858053 838 28.742920  0.000
## MINORITY    -0.0570447 0.025968587 237 -2.196683  0.029
## age13        0.0473388 0.004523745 838 10.464512  0.000
## EXPO         0.3520799 0.024451130 838 14.399330  0.000
## Correlation:
##      (Intr) MINORI age13
## MINORITY -0.457
## age13    -0.013  0.010
## EXPO      0.006 -0.001 -0.224
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.82097989 -0.65136103 -0.08846076  0.61819746  2.77303796
##
## Number of Observations: 1079
## Number of Groups: 239
```

Is any of this drop in the p -value due to overfitting? Given the size of the change in AIC, it seems doubtful that that's a significant factor.

Let's try including heteroskedasticity, without AR[1]:

```

modelKS6 = lme(ATTIT ~ MINORITY + age13 + EXP0,
               data=nys1,
               random=~1|ID,
               weights=varIdent(form=~1|agefac) )
AIC( modelKS6 )

```

```
## [1] -389.5696
```

This did not improve AIC in this case, so we can avoid looking at this model further.

For completeness, let's look at a model with both AR(1) and heteroskedasticity:

```

modelKS7 = lme(ATTIT ~ MINORITY + age13 + EXP0,
               data=nys1,
               random=~1|ID,
               correlation=corAR1(form=~AGE|ID),
               weights=varIdent(form=~1|agefac) )
AIC( modelKS7 )

```

```
## [1] -431.1943
```

Again, no improvement. So we settle with our AR[1] model with a random intercept to get overall level of a student.