Supplemental STATA Material for *Longitudinal Analysis: Modeling Within-Person Fluctuation and Change* by Lesa Hoffman (2015, Routledge Taylor & Francis)

General Notes about STATA Syntax

- **Basic rules about STATA syntax and output:**
  - STATA syntax files end in **.do** or **.ado** and are plain text files.
  - By default STATA output goes to the internal results window. To save the output to an external file using syntax, you can use the **log** command to save results as either plain text (**.log**) or STATA-formatted output (**.smcl**). There are other options available, but all I have used in my examples is **.log** (which was then converted to html).
  - For example below, **log** first provides a name the output file to be created. The **replace** option instructs it to write over the file if it already exists, and the **name** option names the output internally. The final line closes the internally-named file:
    - ```
      log using C:\MyFolder\STATA_Output.log, replace name(STATA_Output)
      log close STATA_Output
      ```
  - The **set more off** statement at the beginning of all my STATA example syntax files prevents the output from being stopped until a manual response to click "more" is executed. I also control the line length via the command **set linesize**.
  - I have used the STATA **display as result** command to list titles to appear in your output, in which the text to be printed is shown within double quotes and is colored **red**.
    - ```
      display as result "This is my first title, which will print once when requested"
      display as result "This is my second title, which will print once on a new line"
      ```
  - STATA syntax IS case-sensitive. All STATA commands must be in lower-case, and all variable names must match case exactly (e.g., Var1 is not the same as var1 when referred to in the syntax).
  - In STATA syntax, there is no command terminator required at the end of a line, but STATA is line-sensitive. To extend commands that belong on the same line onto a new syntax line, you need to add **///** at the end of the syntax line before continuing onto a new line.
  - There is no analog in STATA to the "RUN;" or "EXECUTE." statements in SAS and SPSS. All commands are executed immediately by default.
  - STATA primary commands appear in **BLUE** and comments are in **GREEN**. Comments across one or more lines are indicated with * at the beginning of each. You can select just parts of code to run by commenting out the irrelevant lines of syntax beginning with *

- **Getting data into STATA:**
  - In my syntax, I use the **global** command to define an abbreviation for a specific file location from which files should come and go, that way it only needs to be indicated once in a syntax file. For example below, **global** first defines an abbreviation I've named **filesave** that will refer to a specific file location, as defined in quotes:
    - ```
      global filesave "C:\Dropbox"
      ```
  - Existing STATA datasets can be opened in STATA via the syntax below, in which **use** lists the **filesave** abbreviation preceded by a $ and followed by a \ and the name of the STATA .dta data file to be imported, followed by a comma and the command **clear**, which allows STATA to re-import the file. For example:
      ```

```
use "$filesave\myoriginalfile.dta", clear
```

- Data files stored in text or Excel format can be imported directly into STATA using the **import** statement depending on the file format. However, data files from other programs (e.g., SAS, SPSS) must be saved directly as a STATA .dta file in order to be opened within STATA.
- In addition, only one .dta file can be open within a single instance of STATA at a time, although multiple instances of STATA can be running at once.

- **Making new variables, labeling new variables, and using value labels in STATA syntax:**
  - STATA variable names can be up to 32 characters. STATA variable labels (as created by the **LABEL variable** command, as demonstrated below) can provide even more description (up to 80 characters). STATA value labels can be created using the **LABEL values** command to assign descriptive labels to specific values of a variable within a data file (as also demonstrated below).
  - The **gen** statement can be used to make a new variable based on simple mathematical operations (and **egen** can be used to make new variables based on more complex functions, as illustrated later). For example below, **gen** is used to create a new variable called **age80** in which 0 values represent age 80 from the old variable called **age**. **LABEL variable** is used to make a new accompanying variable label as well:

```
gen age = age - 80
label variable age80 "age80: Age in Years (0=80)"
```

  - There are (at least) two ways to transform an existing variable into a new variable based on select values. The **recode** statement is faster but less flexible, whereas the **replace** statement can require more syntax but offer greater flexibility. Each uses an **if** to impose conditional logic. For example, if an existing variable called **demgroup** = 1, then four new variables are to be created as given next (the **gen** statements). After that, values are selectively replaced for the four new variables based on values of **demgroup**. **LABEL variable** is used to make new accompanying variable labels as well:

```
gen demnf=0
gen demnc=0
gen demfn=0
gen demfc=0
replace demnf=0 if demgroup==1
replace demnc=0 if demgroup==1
replace demfn=1 if demgroup==1
replace demfc=0 if demgroup==1
replace demnf=1 if demgroup==2
replace demnc=0 if demgroup==2
replace demfn=0 if demgroup==2
replace demfc=0 if demgroup==2
replace demnf=0 if demgroup==3
replace demnc=1 if demgroup==3
replace demfn=0 if demgroup==3
replace demfc=1 if demgroup==3
label variable demnf "demnf: Dementia Contrast for None=0 vs Future=1"
label variable demnc "demnc: Dementia Contrast for None=0 vs Current=1"
label variable demfn "demfn: Dementia Contrast for Future=0 vs None=1"
label variable demfc "demfc: Dementia Contrast for Future=0 vs Current=1"
```

  - Value labels are created in two steps. First, **label define** creates a new label format called **fdemgroup** that sets labels for values of 1, 2, or 3. Second, **label values** applies the new label **fdemgroup** to the existing variable **demgroup:** immediately:

```
label define fdemgroup 1 "1None" 2 "2Future" 3 "3Current"
label values demgroup fdemgroup
```

- Certain cases (rows) can be kept or removed based on conditional logic. For example below, the first **drop if** command indicates that cases in which **demgroup** = 1 are be removed from the current data file:

```
drop if demgroup==1
```

- Here is how to select cases to be kept based on whether they have complete data for a series of variables. The first line uses the **egen** statement to create a new variable called **nummiss** that contains the number of missing values across the variables within the **rowmiss** function. Here, we tell STATA that if the number of missing observations is > 0, the case should be removed from the data file:

```
egen nummiss=rowmiss(pred1, dv1, pred2, dv2)
drop if nummiss>0
```

- **Summarizing data into new variables:**
  - Summarizing data into new variables proceeds differently depending on whether one is summarizing across columns or across rows. Here are some examples of each type that are used in the chapter examples.
  - One can summarize across columns into new variables using many built-in functions. More generally in each **egen** statement below, the new variable appears on the left of the equals, the function appears on the right, and what variables the function is to use appear in parentheses. You can either list all the variables or using * to indicate that any variable containing the stem listed in its name should be included:
  - Here are some examples:

```
* Two ways to save the mean of these variables as a new variable
egen meanvar = rowmean(var1 var2 var3 var4 var5)
egen meanvar = rowmean(var*)
* Two ways to save the maximum of these variables as a new variable
egen maxvar = rowmax(var1 var2 var3 var4 var5)
egen maxvar = rowmax(var*)
```

  - Summarizing across rows can also be done via the **egen** statement, such as in my examples using *stacked* data (in which there is one row per occasion per person). For example below, new variables **meanpred** and **maxpred** are created by taking the mean and max of **pred** across rows for the same value of **idvar**:

```
egen meanpred = mean(pred), by(idvar)
egen maxpred  = max(pred),  by(idvar)
```

- **Describing data:**
  - **summarize** can be used to get summary statistics for continuous (quantitative) variables listed after it. By default, this includes N, mean, SD, minimum, and maximum. For example:

```
summarize age grip cognition
```

  - **tabulate** can be used to get marginal and cross-tabulation frequencies for the categorical (grouping) variables listed after it. For example:

```
tabulate sexmw demgroup
```

  - **tabulate** can also be used to get summary statistics for continuous (quantitative) variables per levels of categorical (grouping) variables. For example:

```
tabulate age, summarize(demgroup)
```

- **corr** can be used to get Pearson correlations for continuous (quantitative) variables listed after it. For example:
  - **corr age grip sexmw cognition**
- Summarizing across rows into a new upper-level data file in order to report summary statistics set can be done via the statements below. First, because only one data file can be open in a single instance of STATA at a time, the **preserve** statement tells STATA to retain the original data file. Next, **collapse** lists the variables to be summarized by levels of the variables within the by option. **summarize** requests summary output for these continuous variables, and then **restore** reinstates the original data file. For example:
  -
```
preserve
collapse var1 var2, by(idvar)
summarize var1 var2
restore
```

- **Restructuring data:**
  - STATA has commands with which to restructure data from wide (multivariate) to long (stacked), and code to do so is provided in several examples (e.g., chapters 7b and 9). In the example below, **reshape long** is used to restructure the previous wide (multivariate) file to create each new variable listed next, followed by a comma. The i(idvar) indicates the new variable that indexes which rows belong to the same unit, and the j(occasion) indicates the name of the new variable to distinguish the rows. By default all variables with the stem given in the name will be included in the restructuring, which will be removed from the new stacked file.
    - **reshape long age DV IV, i(idvar) j(occasion)**

- **A note about STATA macro programing:**
  - The SAS syntax files that accompany the chapter examples include custom macro programs I have written to automate otherwise manual tedious calculations (e.g., total R^2, pseudo R^2). Each of these works by requesting the necessary user input explicitly as well as saving the relevant output tables using the SAS Output Delivery System (ODS), manipulating their content and performing the necessary calculations, and then printing the summary results to the output. STATA already has likelihood ratio tests built available, and also has macro programming along with similar output management procedures that could also be used to automate other calculations. But given how rarely I teach in STATA, I have not taken the time to do so. However, for each chapter example I have also provided Excel spreadsheets that include formulas for the same calculations as provided by the SAS macros, but these do require user data entry of the relevant output for each set of calculations.

Understanding the STATA MIXED commands used for Longitudinal and Multilevel Modeling (mostly in order of appearance in my example models)

- As a reminder, STATA syntax is line sensitive. Therefore, when commands that belong on the same line are written on a new syntax line, you need to see **///** at the end of the previous syntax line to indicate the continuance. In addition, most commands end in a comma, regardless of whether they should be on the same line.
- **mixed command:**
  - The first required input for the STATA **mixed** command is the outcome variable to be analyzed, followed by the list of predictors that have fixed effects, followed by a comma. A prefix of **c.** is used to indicate predictors that are to be treated as continuous (i.e., for which a single slope is estimated for each), which is the default. In contrast, a prefix of **i.** is used to indicate predictors that are to be treated as categorical (i.e., for which all possible mean differences across unique levels are estimated instead; see further details below).

- For example, to analyze the outcome of **cognition** using continuous predictors of **age85** and **grip9** and a categorical predictor of **demgroup**:
  - **mixed cognition i.demgroup c.age85 c.grip9,**

- **Fixed effects specification using the c. and i. options for predictors within the mixed command:**
  - The **i.** option is used to list model predictors to be treated as *categorical*, which here means that STATA will automatically create all necessary pairwise contrasts by which to estimate all possible mean differences across groups (unique values) of these variables. By default, STATA uses the lowest number as the reference group and creates contrasts for each other possible value relative to that reference group. If that is not the reference group you want, then you can change the option on the **i.** prefix. Any other predictors to be treated as continuous must be listed on the **c.** option instead.
  - As discussed in the chapter 2 appendix, predictor variables listed on the **i.** option are marginalized over any interacting variables when estimating the interacting variables' main effects for the Type 3 Tests of Fixed Effects (which are not provided by default in STATA, but which can be requested via **contrast**, as described below). For example, consider an interaction of **demgroup*agegroup**, in which **demgroup** is listed on the **i.** option but **agegroup** is on the **c.** option instead and is thus being treated as a continuous predictor. In the Type 3 Tests of Fixed Effects, the main effect of **agegroup** will refer specifically to its slope *averaged across all levels of* ***demgroup*** (i.e., not just for the reference **demgroup**), but the main effect of **demgroup** will refer to the omnibus group mean difference just for the specific reference **agegroup** = 0. In contrast, if **agegroup** was listed on the **i.** option and was thus also treated as a categorical predictor, then the main effect of **demgroup** will refer to the omnibus group mean difference *averaged across all levels of* ***agegroup*** (i.e., not just for the reference **agegroup** = 0). However, in the Solution for Fixed Effects, all main effects for predictors that are part of an interaction are their simple effects at the reference group/value of their interacting predictors, regardless of whether the predictors are on the **i.** or **c.** options. Thus, the tests of main effects provided in the Solution for Fixed Effects and in the Type 3 Tests of Fixed Effects may not agree when interaction terms are included in the model because they can refer to different main effects (simple or marginal) depending on which predictors are on the **i.** or **c.** options.
  - Only predictor variables that are listed on the **i.** option can have their model-predicted means per group requested via the **margins** subcommand (see below).
  - When used in **lincom** subcommands, predictor variables listed on the **i.** option must have a value listed for each possible level of their variable (i.e., each distinct group; see examples below).
  - Listing predictor variables individually requests their main effects only. Interaction effects can also be requested by listing the variables to be part of an interaction, each joined by a pound sign. For example (assuming the **age85 grip9 sexmw** predictors are modeled as continuous in predicting **cognition**):
  - To include main effects of **age85** and **grip9**:
    ```
    mixed cognition c.age85 c.grip9,
    ```
  - To include main effects of **age85**, **grip9**, and their two-way interaction:
    ```
    mixed cognition c.age85#c.grip9,
    ```
  - To include main effects of **age85**, **grip9**, **sexmw**, as well as their three possible two-way interactions:
    ```
    mixed cognition c.age85#c.grip9 c.age85#c.sexmw c.grip9#c.sexmw
    ```
  - To add their three-way interaction to the previous model:
    ```
    mixed cognition c.age85#c.grip9#c.sexmw
    ```
  - To add a main effect and interaction of **sexmw** with categorical **demgroup** to the previous model:
    ```
    mixed cognition c.age85#c.grip9#c.sexmw i.demgroup#*c.sexmw
    ```

- **Random effects specification within the `mixed` command:**
  - After the first comma terminating the list of predictors with fixed effects, the double bar sign **`||`** indicates the beginning of the random effects. In models with more than two levels, the levels should be identified in descending order. After **`||`** comes the name of the ID variable for that level, followed by a colon, followed by the predictors to be modeled as having random effects, followed by a comma. A random intercept is included for each level mentioned by default; to remove it, use the **`noconstant`** option as described below. If you omit the **`||`** for a level of random effects, then there is no **G** matrix in your model, such that the total variance and covariance across observations would be given in the **R** matrix only. When using the **`||`** to include random effects, then the total variance and covariance across observations will be created in the **V** matrix from **G**, **R**, and **Z**, as $\mathbf{V} = \mathbf{ZGZ}^T + \mathbf{R}$, as described in chapter 5. Thus, without **`||`**, $\mathbf{V} = \mathbf{R}$.
  - The following options have been used in my examples, placed after the comma at the end of each level of random effects:
    - **`noconstant`** removes default random intercept variance from that level; specified per level
    - **`variance`** requests that variance components be given in the output as variances rather than as standard deviations (which is the default instead); only needs specified once
    - **`reml`** use Restricted Maximum Likelihood (the default); only needs specified once
    - **`mle`** use Maximum Likelihood; only needs specified once
    - **`covariance(unstructured)`** Structure of the **G** matrix, which should ALWAYS be unstructured in order to estimate covariances among random effects at the same level; must be specified separately for each level of random effects
  - In STATA **`mixed`**, you must explicitly list any effects for which you want to estimate a random effect variance, excluding the intercept (which is included by default). To facilitate estimation, all random effects should be for continuous predictors only. If you want to estimate a random effect variance for differences across levels of a categorical predictor, those differences should be created via manual contrasts treated as continuous instead.
    - For example, to allow only the intercept to vary randomly across levels of **`PersonID`**:
      ```
      mixed DV c.time, || personid: , variance reml covariance(unstructured)
      ```
    - For example, to allow both the intercept and time slope to vary randomly across levels of **`PersonID`**:
      ```
      mixed DV c.time, || personid: time, variance reml covariance(unstructured)
      ```
  - In my examples the random effects option **`covariance(unstructured)`** is always used for consistency, even though it is not relevant unless there is more than one random effect at the same level.
  - There are several ways to estimate models with crossed random effects within STATA **`mixed`**. One way is to use the **`_all:`** option paired with **`R.`** to indicate a categorical random effect.
    - For example, given crossed subjects and items:
      ```
      mixed DV c.pred, || _all: R.subjectid, covariance(unstructured) ///
                       || _all: R.itemid, covariance(unstructured) variance reml
      ```

- **Residual effects specification within the `mixed` command:**
  - The structure of the **R** matrix of variances and covariances (which are *residual* if random effects have been included, but *total* otherwise) is specified as yet another option within the random effects section started above. If you do not explicitly include mention the **R** matrix, by default the model still includes an **`independent R`** matrix, in which all observations are assumed to have the same residual/total variance with no covariances.

- To refer to the **R** matrix, list **residuals** followed in parentheses by the structure, a comma, the letter t, and then in nested parentheses the ID variable for the **R** matrix, followed by a comma before the next section. For example, given an ID variable of **occasion**:
  - **residuals(independent,t(timevar)),** Diagonal **R** matrix
  - **residuals(exchangeable,t(timevar)),** Compound symmetry **R** matrix
  - **residuals(unstructured,t(timevar)),** Unstructured **R** matrix
  - **residuals(ar1,t(timevar)),** First-order autoregressive **R** matrix
  - **residuals(toeplitz6,t(timevar)),** Toeplitz **R** matrix with 6 bands (diagonal + 5 lags)

- **estat commands used in my example code (to print output not provided by default):**
  - **estat vce,** Asymptotic covariance matrix of fixed effects (COVB)
  - **estat icc,** Intraclass correlation—note that any level-3 ICC given is ICC_{L3b}
  - **estat ic, n(#persons),** Information criteria, use #persons for sample size for computing BIC
  - **estat recovariance, relevel(IDvar),** **G** matrix for level of IDvar
  - **estat recovariance, relevel(IDvar) correlation,** **GCORR** matrix for level of IDvar
  - **estat wcorrelation, covariance,** **V** matrix
  - **estat wcorrelation,** **VCORR** matrix

- **To conduct likelihood ratio tests across models using mixed:**
  - Model fit comparisons can be conducted in two steps, as illustrated below, in which the first model given is the baseline for the comparison with a fixed time slope only, and the second model is the augmented model (that also includes a random time slope and covariance with the random intercept). First, the **estimates store** command saves the log-likelihood value from the current model under the name given. Second, in the augmented model, the **lrtest** command requests the likelihood ratio test, in which the model with more parameters is listed first, followed by the model with fewer parameters, followed by a comma. The extra option **force** can be used to require the test even in situations when STATA thinks the models are not nested (as STATA can be wrong).
    - ```
      mixed DV c.time, || personid: , variance reml covariance(unstructured),
      estimates store FixedSlope,
      ```
    - ```
      mixed DV c.time, || personid: time, variance reml covariance(unstructured),
      estimates store RandomSlope,
      lrtest RandomSlope FixedSlope, force
      ```
  - The reference distribution for the likelihood ratio test depends on what models are being compared. The default-provided test of a single-level model (termed "linear regression) against the current model will use a "chi-bar" mixture chi-square distribution if the current model contains only a random intercept, but will use a regular chi-square distribution otherwise. Likewise comparisons of models with only a random intercept versus adding a random slope will use the regular chi-square distribution.

- **test subcommand:**
  - The test subcommand in STATA mixed can be used to request custom hypothesis tests of many different kinds. In my examples, I tend to use the **test** subcommand for multivariate Wald tests of the significance of

multiple fixed effects at once or for custom interaction contrasts. Explanations and examples of each will be shown below.

- An example of a multivariate Wald test is the traditional test of the significance for the variance accounted for by a regression model (i.e., the *F*-test of the model R^2). For *continuous* predictors (those with a prefix of **c.** for which slopes are estimated), each fixed effect to be included is listed as = 0 within parentheses.
- For example, given the following model with main effects of two *continuous* predictors, the **test** subcommand can provide a test of their joint significance (i.e., test of the significance of the model R^2) with two degrees of freedom as follows:
  - ```
    mixed cognition c.age85 c.grip9,
          test (c.age85=0) (c.grip9=0),
    ```
- If the main effect of a *categorical* predictor (those with a prefix of **i.** for which mean differences across unique levels or groups are estimated) with two groups is added to the model (such as **sexmw** below), then the group difference must be indicated by setting the contrast for the higher-coded group=0 instead. For instance, given **sexmw** values of **0** or **1**, **0** is first and **1** is second.
- For example, the **test** subcommand below now provides a joint test of significance of the model predictors with three degrees of freedom as follows:
  - ```
    mixed cognition c.age85 c.grip9 i.sexmw,
          test (c. age85=0) (c.grip9=0) (1.sexmw=0),
    ```
- If the main effect of a *categorical* predictor with three groups is added to the model (such as **demgroup** below), then the differences among those three groups must be indicated by setting the contrast for each non-reference, higher-coded group =0. More generally, the number of separate contrasts needed is equal to one fewer than the number of groups; this set of group contrasts is otherwise known as an "omnibus" test of mean differences in an ANOVA context. The **test** subcommand below provides a joint test of significance of the model predictors (including demgroup with **demgroup** values of 1, 2, 3) with five degrees of freedom as follows:
  - ```
    mixed cognition c.age85 c.grip9 i.sexmw i.demgroup,
          test (c.age85=0) (c.grip9=0) (1.sexmw=0) (2.demgroup=0) (3.demgroup=0),
    ```

- **contrast subcommand:**
  - The **contrast** subcommand in STATA **mixed** can be used to request custom hypothesis tests involving categorical predictors. For example, the omnibus main effects of two *categorical* predictors (those with a prefix of **i.** for which mean differences across unique levels or groups are estimated) with degrees of freedom equal to the number of groups – 1 can be requested as follows:
    - ```
      mixed cognition i.sexmw i.demgroup,
            contrast i.sexmw i.demgroup,
      ```
  - The **contrast** subcommand can also be used to test custom contrasts within an interaction involving one or more *categorical* predictors. In the example below, the interaction of the two-group **sexmw** predictor and the three-group **demgroup** predictor is represented by the six values after their interaction term, in which the order of the values is determined by the order of these predictors when listed as fixed effects. The first three contrasts request the simple main effect of **sexmw** within each level of **demgroup**. The next three contrasts then request how those simple main effects of **sexmw** differ across levels of **demgroup** (i.e., single degree-of-freedom interactions within the omnibus two degree-of-freedom **sexmw*demgroup** interaction), which can be found by literally subtracting the contrast values between the relevant lines:
    - ```
      * Value order for sexmw (s) and demgroup (d): s0d1 s0d2 s0d3 s1d1 s1d2 s1d3
      * First 3 contrasts provide sexmw simple effects within each demgroup
      ```
      ```

```
* Second 3 contrasts provide sexmw simple effect differences across demgroups
* Comments are indicated by // within a line
mixed cognition i.sexmw i.demgroup,
     contrast {i.sexmw#i.demgroup} -1  0  0  1  0  0,  // Sex diff in demgroup 1
     contrast {i.sexmw#i.demgroup}  0 -1  0  0  1  0,  // Sex diff in demgroup 2
     contrast {i.sexmw#i.demgroup}  0  0 -1  0  0  1,  // Sex diff in demgroup 3
     contrast {i.sexmw#i.demgroup} -1  1  0  1 -1  0,  // Sex diff: demgroup 1v2
     contrast {i.sexmw#i.demgroup} -1  0  1  1  0 -1,  // Sex diff: demgroup 1v3
     contrast {i.sexmw#i.demgroup}  0 -1  1  0  1 -1,  // Sex diff: demgroup 2v3
```

- **`lincom` subcommand:**
  - The **`lincom`** subcommand in STATA **`mixed`** can be used to obtain estimates, standard errors, and *p*-values for significance tests against a null hypothesis of 0 for model-implied effects of a single degree of freedom, such as for model-predicted outcomes (intercepts), comparisons of coefficients, or simple effects within higher-order interaction terms (the logic of which is presented in more detail in chapter 2).
  - Below is an example of how to request model-predicted outcomes (i.e., intercepts). The term _cons is used to represent the intercept, and otherwise the values given should indicate the corresponding values for the predictor variables in the model equation:

```
mixed cognition c.age85 c.grip9 i.sexmw i.demgroup,
      // Cognition for age85=-5, grip9=3, sexmw=m, demgroup=2
      lincom _cons*1 + c.age85*-5 + c.grip9*3 + 0.sexMW + 2.demgroup,
      // Cognition for age85=-5, grip9=3, sexmw=w, demgroup=3
      lincom _cons*1 + c.age85*-5 + c.grip9*3 + 1.sexMW + 3.demgroup,
```

  - Note that in requesting model-predicted outcomes, the intercept (via **`_cons`**) should always be included with a value of **`1`** (because the fixed intercept always contributes once to any predicted outcome), and values should be specified for every other predictor in the model. Otherwise, any predictor without a value given is assumed to be held to **`0`** (i.e., held to its reference value or reference group).
  - Below is an example of how to compare coefficients using the **`lincom`** subcommand. Note that in order for such comparisons to make sense, the coefficients should be on the same scale (i.e., a one-unit change means the same thing in both predictors). The **`lincom`** subcommand below provides the test of the difference in the slope of one additional year since birth (via **`YearsSinceBirth`**) and the slope of one additional year further away from death (via **`YearsUntilDeath`**), in which the common unit of prediction is one year:

```
mixed cognition c.YearsSinceBirth c.YearsUntilDeath,
     // Difference in year coefficients
     YearsSinceBirth -1 YearsUntilDeath 1,
```

  - Here is another example of comparing coefficients but in a different context. In this example, the three-group i.**`demgroup`** categorical predictor has instead been represented by two manually-created dummy-coded continuous predictors of **`c.demnf`** and **`c.demnc`** (for none vs. future and none vs. current, as described in chapter 2). The **`lincom`** subcommand below provides the missing comparison of future vs. current, which is the difference of **`c.demNF`** and **`c.demNC`** coefficients created by the subtraction of their values:

```
mixed cognition c.age85 c.grip9 c.demnf c.demnc,
     // Future vs. Current when demgroup is not categorical
     lincom demnf -1 demnc 1,
```

  - Below is an example of how to request model-implied simple effects of a two-way interaction among *continuous* predictors (i.e., those listed on the **`c.`** option of the **`mixed`** command). Each **`lincom`** subcommand below contains the simple effect to be estimated and how it is modified by any interacting predictors as specified in the fixed effects. The interaction terms in the **`lincom`** subcommand should be read in pieces—whichever is the

simple effect of interest is given a value of **1**, and the value given for the interacting predictor creates the new conditional simple effect, in which positive values create addition and negative values create subtraction. In the example below, the first two **lincom** subcommands build an **c.age85** slope by adding its simple effect given directly by the model (when **c.grip9**=0) to the interaction value for how the **c.age85** slope changes per unit of **c.grip9**, whereas the second two **lincom** subcommands build a **c.grip9** slope by adding its simple effect given directly by the model (when **c.ge85**=0) to the interaction value for how the **c.grip9** slope changes per unit of **c.age85**. Note that because **i.sexmw** and **i.demgroup** do not interact with **c.age85** or c.**grip9** in the model, they should not be included in their simple effect **lincom** subcommands:

- ```
  * First 2 lincom provide simple effects of age85 conditional on grip9
  * Second 2 lincom provide simple effects of grip9 conditional on age85
  mixed cognition i.sexmw i.demgroup c.age85 c.grip9 c.age85#c.grip9,
       lincom c.age85*1 + c.age85#c.grip9*-3,  // Age85 Slope if Grip9=-3
       lincom c.age85*1 + c.age85#c.grip9*3,   // Age85 Slope if Grip9= 3
       lincom c.grip9*1 + c.age85#c.grip9*-5,  // Grip9 Slope if Age85=-5
       lincom c.grip9*1 + c.age85#c.grip9*5,   // Grip9 Slope if Age85= 5
  ```

- Below is an example of how to request model-implied simple effects of a two-way interaction of a *continuous* predictor (listed on the **c.** option of the **mixed** command) with a *categorical* predictor (listed on the **i.** option of the **mixed** command). The logic is the same, but the categorical predictor must have a value for the intended unique level. The first set of **lincom** subcommands provide the simple effect of **c.age85** for each level of **i.demgroup** (via the value within **i.demgroup**), whereas the second set provides the differences in the simple effects of **c.age85** between levels of **i.demgroup** (via the values within **i.demgroup**). The third set of statements require the **margins** subcommand instead to provide the simple differences across levels of **i.demgroup** for specific values of **c.age85**, as explained in greater detail in the next section.

  - ```
    * First 3 lincom provide simple effects of age85 conditional on demgroup
    * Second 3 lincom provide age85 simple effect differences between demgroups
    * Last 2 margins provides simple diffs for all demgroups conditional on age85
    mixed cognition i.sexmw i.demgroup c.age85 c.grip9 c.age85#i.demgroup,
     lincom c.age85*1 + c.age85#i1.demgroup,  // Age85 Slope if DemGroup=1
     lincom c.age85*1 + c.age85#i2.demgroup,  // Age85 Slope if DemGroup=2
     lincom c.age85*1 + c.age85#i3.demgroup,  // Age85 Slope if DemGroup=3
     lincom c.age85#i2.demgroup - c.age85#i1.demgroup, // Age85 Slope: DemGroup=1vs2
     lincom c.age85#i3.demgroup - c.age85#i1.demgroup, // Age85 Slope: DemGroup=1vs3
     margins i.demgroup, at (c.age85=5)  pwcompare(pveffects),
     margins i.demgroup, at (c.age85=-5) pwcompare(pveffects),
    ```

- Given the number of distinct values to be specified in order to represent all possible unique group combinations, simple effects within interactions among *categorical* predictors (i.e., those listed on the **i.** option of the **mixed** command) are much easier to specify using **margins**, as shown next.

- **margins subcommand:**

  - The **margins** subcommand can be used to request model-predicted conditional means for any combination of continuous or categorical predictors, in which the **at** subcommand denotes values of continuous predictors. There are many variations of margins possible; the example below are those used in the models reported.

  - The **margins** below requests 27 predicted outcomes for **i.demgroup** = 1, 2, or 3, with values of –5, 0, and 5 for **c.age85** (from −5 to 5 in increments of 5) and values of –3, 0, and 3 for **c.grip9** (from −3 to 3 in increments of 3), holding **c.sexmw** constant at 0:

    - ```
      mixed cognition  c.sexmw i.demgroup c.age85 c.grip9 c.age85#c.grip9,
        margins i.demgroup, at (c.age85=(-5(5)5) c.grip9=(-3(3)3) c.sexmw=0),
      ```
      ```

- The first **margins** below requests and compares the cell means for each unique combination of **i.sexmw** by **i.demgroup**, holding the covariates listed in the **at** subcommand to 0. The second **margins** requests the simple omnibus main effects for **i.demgroup** at each level of **i.sexmw**, and the third **margins** does the opposite:

```
mixed cognition c.age85 c.grip9 i.sexmw i.demgroup i.sexmw#i.demgroup,
  margins i.sexmw#i.demgroup, at (c.age85=0 c.grip9=0) pwcompare(pveffects),
  margins i.demgroup@i.sexmw, at (c.age85=0 c.grip9=0),
  margins i.sexmw@i.demgroup, at (c.age85=0 c.grip9=0),
```

- **predict subcommand:**
  - The **predict** subcommand in STATA **mixed** can be used to save model-predicted outcomes to a dataset using the name given as the new column. The **xb** suboption specifies predictions from fixed effects only:
    -

```
mixed cognition c.age85 c.grip9 i.sexmw i.demgroup,
      predict PredValues, xb,
```

Model-Specific Notes by Chapter Example

- **A note about model fit, degrees of freedom, and information criteria across programs:**
  - As presented in chapters 3 and 5, **AIC** is computed as AIC = −2LL + 2*df, in which *df* = degrees of freedom. But within most programs (including STATA and SAS but excluding STATA), the degrees of freedom used in computing AIC are different when using ML versus REML estimation. In ML, degrees of freedom include ALL model parameters (fixed effects in the model for the means plus all variances, covariances, and other variance model parameters), but will exclude the fixed effects when using REML. Although this may seem arbitrary, it makes sense if you remember that models that differ in fixed effects cannot have their fit compared using −2LL, AIC, or BIC when using REML (and thus the number of fixed effects is irrelevant for those statistics). In STATA, however, the df used in computing AIC and BIC within REML do include fixed effects in the model for the means, so AIC and BIC computed using STATA REML will not agree with those from SPSS or SAS.
  - As also presented in chapters 3 and 5, **BIC** is computed as BIC = −2LL + Log(N)*df, in which *df* = degrees of freedom (which differs between ML and REML for the same programs as noted for AIC), and *N* refers to some kind of sample size. But what value of *N* is used differs by program. In STATA and MPLUS, *N* = the total number of observations, whereas *N* = the sample size at the highest level of the model in SAS. More specifically, in SAS *N* is the number of unique repetitions of the **V** matrix, which means that *N* = 1 in models with crossed random effects. In STATA, the user can tell the program what *N* should be (and in my examples, I follow the SAS convention). Consequently, BIC will differ across these programs for any multilevel model.

- **Chapter 2**
  - Because chapter 2 includes single-level models only, all references to the **personid** variable on the random effects section were followed by **noconstant**, such no random intercept variance was estimated. The default **R** matrix type is **independent**, in which the residual variance is assumed constant over all observations with no covariance across observations.
  - The method of fake people was replaced by the more efficient use of **margins** to calculate predicted values.

- **Chapter 3a**
  - The data were initially in multivariate (wide) format, in which one row contains all observations for a person. The **`reshape`** restructured the data into stacked (long) format of one row per occasion per person, as is needed for **`mixed`**.

- **Chapter 3b**
  - The data to be read in were already in stacked format, so no further data transformation was necessary.
  - Note that the time ID variable, **`i.session`** as listed in the fixed effects section, is being treated as a categorical predictor, thus estimating all possible mean differences across sessions. The **`session`** ID variable is also used to structure the **R** matrix in the **`residuals`** subcommand in each model.

- **Chapter 4**
  - The data to be read in were already in stacked format, so no further data transformation was necessary.
  - Note that only the **`residuals`** subcommand was used for the **R**-only models, whereas both the **`||`** for random effects and **`residuals`** subcommands were used for the **G** and **R** combined models (to control **G** and **R**, respectively). However, the **G** and **R** combined models with a reduced-order **R** matrix with heterogeneous variances were not able to be estimated in STATA.

- **Chapter 5**
  - The data to be read in were already in stacked format, so no further data transformation was necessary.
  - Given the use of both the **G** and **R** matrices, both the **`||`** for random effects and **`residuals`** subcommands were used. However, the **`residuals`** subcommand with an independent structure was technically unnecessary to include, as that is already the default.

- **Chapter 6**
  - The data to be read in were already in stacked format, so no further data transformation was necessary, except for the creation of the specific time predictors needed (e.g., linear and quadratic time, piecewise time slopes).
  - The negative exponential models could not be estimated using STATA.

- **Chapter 7a**
  - The data to be read in were already in stacked format, so no further data transformation was necessary, except for the creation of the specific centered predictors needed.
  - Because time was unbalanced, the **`residuals`** subcommand was not used given that it was never anything other than the default independent structure (the other types are for balanced data).
  - The heterogeneous variance models could not be estimated using STATA.

- **Chapter 7b**
  - The data were initially in multivariate (wide) format, in which one row contains all observations for a person. The **`reshape`** restructured the data into stacked (long) format of one row per occasion per person, as is needed for **`mixed`**.

- The method of fake people was replaced by the more efficient use of **margins** to calculate predicted values.
- Because time was unbalanced, the **residuals** subcommand was not used given that it was never anything other than the default independent structure (the other types are for balanced data).

- **Chapter 8**
  - The data to be read in were already in stacked format, so no further data transformation was necessary, except for the creation of the specific centered predictors needed.
  - All level-1 predictors were treated as observed variables that were not included in the likelihood.
  - **margins** subcommands were used to create predicted outcomes instead of separate syntax statements.
  - The heterogeneous variance models could not be estimated using STATA.

- **Chapter 9**
  - The data to be read in were initially in multivariate (wide) format, in which one row contains all observations for a person. The initial data step restructured the data into stacked (long) format of one row per occasion per person, as is needed for **mixed**.
  - **mixed** would not estimate the multivariate longitudinal models in which both time-varying variables (the risky behavior outcome and the monitoring predictor) were included in the likelihood, such that their random intercept, random linear age slope, and residual variances were partitioned by the model.
  - The random effects version of the slopes-as-outcomes model was estimated using **mixed** given that empirical Bayes estimates for the predicted random effects are directly available. Although the fixed effects version is also possible, it would require additional data transformation steps to save the individual-specific fixed effects (i.e., either by using additional syntax or manually copying them out of the output). Given that the slopes-as-outcomes models cannot be recommended in the first place, I didn't bother doing this.

- **Chapter 10a**
  - The data to be read in were already in stacked format, so no further data transformation was necessary, except for the creation of the specific centered predictors needed.
  - The method of fake people was replaced by the more efficient use of **margins** to calculate predicted values. Note that missing data were used instead for cases with impossible values (i.e., that would create dead people).
  - Because time was unbalanced, the **residuals** subcommand was not used given that it was never anything other than the default independent structure (the other types are for balanced data).

- **Chapter 10b**
  - The data to be read in were already in stacked format, so no further data transformation was necessary, except for the creation of the specific centered predictors needed.
  - The method of fake people was replaced by the more efficient use of **margins** to calculate predicted values.
  - The three-level models required two **||** subcommands (to structure the **G** matrix for levels 2 and 3), which can be distinguished by the ID variables given. In this example, **|| personid** indicated level 3 (for unique persons), whereas **|| burst** indicated level 2 (for the unique combination of person and burst). The random level listed

first is implicitly the highest, such that **`burst`** is assumed to be nested within **`personid`** even if the values are repeated (which they are here for bursts 1–5).

- Although it could have been used given balanced sessions at level 1, the **`residuals`** subcommand was not used given that it was never anything other than the default independent structure in this example.
- The multivariate model could not be estimated within **`mixed`**.

- **Chapter 11a**
  - The data to be read in were in multivariate format with respect to waves within students, but were stacked with respect to students within classes. Although the summary variables needed with which to build the model predictors could be calculated within one file, separate files were needed to summarize data at the different levels of analysis and for some specific models as well (as indicated by the use of **`preserve`**, **`collapse`**, **`summarize`**, and **`restore`**).
  - The three-level models again required two **`||`** subcommands (to structure the **G** matrix for levels 2 and 3), which can be distinguished by the ID variables given. In this example, **`|| classid`** indicated level 3 (for unique classes), whereas **`|| studentid`** indicated level 2 (for the unique combination of class and student). Students were assumed to be nested within classes given the order in which the **`||`** subcommands were listed.
  - The saturated means models included **G** and **R** matrices for classes and students that were unstructured across occasions, such that the **`||`** random effects subcommand did not include an intercept, and instead included the three dummy codes to distinguish each level of **`i.wave`** (which was used to structure the **R** matrix).
  - All level-1 and level-2 predictors were treated as observed variables that were not included in the likelihood.
  - Note that some of the models would not converge, but their syntax was included for reference purposes.

- **Chapter 11b**
  - The data to be read in were in multivariate format with respect to waves within students, but were stacked with respect to students within classes. Although the summary variables needed with which to build the model predictors could be calculated within one file, separate files were needed to summarize data at the different levels of analysis and for some specific models as well (as indicated by the use of **`preserve`**, **`collapse`**, **`tabulate`**, and **`restore`**).
  - The specification of fixed effects of year-specific classes relied on a similar logic as in the multivariate models. Because the class effects are nested within years, the differences between classes in a given year were created through what look like interaction terms between year-specific indicators and class ID categorical predictors (e.g., **`i.classID_year0#c.aclass0`**), but in which the indicator variables function as a "switch" to only allow the class differences for the relevant cases. The year-specific effects of class grade were specified this way as well.
  - The models for crossed random effects of year-specific classes included one **`||`** subcommand per year (whose ID variable is the class ID for that year), but none of them included a general random intercept. Instead, each included a random effect for its year-specific indicator variable, which created a random intercept that was switched on or off for each case as needed (see chapter 11 for more explanation). Because these custom random intercepts were included in separate **`||`** subcommands, no covariances were estimated among them. Further, because the keyword **`_all:`** was used for each, none were viewed as nested within each other.

- **Chapter 12**
  - The data to be read in were already in stacked format, so no further data transformation was necessary, except for the creation of the specific centered predictors needed.
  - The method of fake people was again used as a means to create predicted outcomes for use in plotting.
  - The models for crossed random effects for subjects and items included one `||` subcommand for each level-2 part of the **G** matrix, which can be distinguished via the ID variables. A general random intercept was included for both subjects and items, in which the item random intercept was referred to as non-nested with respect to subjects via the keyword `_all:` The use of two separate `||` subcommands means that any random effects for subjects will not have covariances with any random effects for items. None of the models with random slopes would convergence, but their syntax is provided for reference.