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

General Notes about SPSS Syntax

- **Basic rules about SPSS syntax and output:**
    - SPSS uses three kinds of files: **.sps** is for syntax, **.spv** is for both the log that records any error and the output. The syntax is just a text file (so you can open and edit it in notepad or Word).
    - By default SPSS output goes to a new **.spv** output window. To save the output to an external file using syntax, you can use the `OUTPUT EXPORT` command, whose options include HTML (viewable in a web browser), RTF (rich text file viewable in Word), Excel, PowerPoint, PDF (viewable using Adobe Acrobat), and text format.
    - For example below, the first `OUTPUT NAME` command provides a name for any following output. Then the `OUTPUT EXPORT` statements below save all visible output to a file using the location and name given. Each statement is terminated at the end by a period:
        - **Save as .html:**

```
OUTPUT NAME=SPSS_Output.
OUTPUT EXPORT NAME=SPSS_Output
/CONTENTS EXPORT=VISIBLE LAYERS=VISIBLE MODELVIEWS=VISIBLE
/HTML     DOCUMENTFILE='C:\MyFolder\SPSS_Output.html'
          IMAGEFORMAT=PNG STYLING=YES.
```

        - **Save as .rtf:**

```
OUTPUT NAME=SPSS_Output.
OUTPUT EXPORT NAME=SPSS_Output
/CONTENTS EXPORT=VISIBLE LAYERS=VISIBLE MODELVIEWS=VISIBLE
/DOC      DOCUMENTFILE='C:\MyFolder\SPSS_Output.rtf'.
```

        - **Save as .xls:**

```
OUTPUT NAME=SPSS_Output.
OUTPUT EXPORT NAME=SPSS_Output
/CONTENTS EXPORT=VISIBLE LAYERS=VISIBLE MODELVIEWS=VISIBLE
/XLS      DOCUMENTFILE='C:\MyFolder\SPSS_Output.xls'
          OPERATION=MODIFYSHEET SHEET='SheetName'.
```

    - The `SET` statement at the beginning of all my SPSS example syntax files includes the option `PRINTBACK=NONE` to prevent the syntax from appearing in the output (change to `=LISTING` to have the syntax shown instead) and `HEADER=NO` to shut off the default titles (change to `=ON` to shown titles instead).
    - I have used the SPSS `ECHO` command to list titles to appear in your output, in which the text to be printed is shown within single or double quotes. Multiple separate `ECHO` commands can be used. Although SPSS does have `TITLE` and `SUBTITLE` commands, they are limited to 60 characters, whereas `ECHO` is unlimited in length.
        -

```
ECHO 'This is my first title, which will print once when requested'.
ECHO 'This is my second title, which will print once on the same line'.
```

    - In general, SPSS syntax is not case-sensitive or space-sensitive, and it will continue reading syntax written to any line length. However, in my syntax, I write SPSS keywords in CAPS and data-specific or model-specific syntax in lower-case to make it easier to understand and modify.
    - In SPSS syntax, SPSS primary commands appear in `BLUE`, and each complete command statement ends in a **period**. Only one complete command (terminated by a period) is allowed per line of syntax. Options for a primary command will appear in `RED` with choices in `ORANGE`. Suboptions within a command will appear in `/GREEN` with subcommand keywords in `RED`. Other syntax will appear in `BLACK`. When typing in an SPSS syntax window, the colors will not appear until the final terminating period is included.

- Comments across one or more lines are indicated with `*` at the beginning of each so that the text that follows will turn `GRAY` and are terminated with a period.
- You can select just parts of code to run by commenting out the irrelevant lines of syntax within `*` and `.`
- You can abbreviate lists of variables in contiguous positions using the keyword `TO` (e.g., **`var1x TO var10x`** will refer to **`var1x`**, whatever variable is next in the file, next, next… **`var10x`**). You can also abbreviate all variables in the data file with the keyword **`ALL`**.

- **Getting data into SPSS:**
  - In my syntax, I use the `FILE HANDLE` 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. Multiple instances of `FILE HANDLE` can be created for different file locations. In the example below, `FILE HANDLE` first defines an abbreviation I've named **`filesave`** that will then be used to refer to a specific file location, as defined in quotes on the `/NAME` subcommand, terminated by a period:
    - `FILE HANDLE filesave /NAME = "C:\Dropbox".`
  - Existing SPSS datasets can be opened in SPSS for further modification or use via the syntax below, in which `GET FILE` lists the `FILE HANDLE` location followed by a `/` and the name of the SPSS .sav data file to be imported, terminated with a period. `DATASET NAME` then provides a nickname (such as **`myfile`** below) that you can refer to in the syntax, and `WINDOW=FRONT` makes it the focus of any subsequent syntax commands. For example:

    -
      ```
      GET FILE = "filesave/myoriginalfile.sav".
      DATASET NAME myfile WINDOW=FRONT.
      ```
  - Data files stored in other formats (e.g., SAS, STATA, Excel) must first be imported into SPSS before they can be modified or used, which occurs through some permutation of a `GET DATA` statement depending on the file format, terminated by a period. For example, each of the following statements imports a file called **`myfile`** from the location indicated by **`filesave.`** `DATASET NAME` then gives the new data file a nickname called **`mySPSSfile`** with which to refer to it in subsequent syntax, and brings it into current focus to be used:
    - SAS:
      ```
      GET SAS DATA = "filesave/myfile.sas7bdat".
      DATASET NAME mySPSSfile WINDOW=FRONT.
      ```
    - STATA:
      ```
      GET STAT FILE = "filesave/myfile.dta".
      DATASET NAME mySPSSfile WINDOW=FRONT.
      ```
    - **Excel with .xls**, all the data, from sheet **`mysheet`** in which the first row has the variable names:
      ```
      GET DATA /TYPE = XLS /FILE = "filesave/myfile.xls"
               /SHEET = NAME "mysheet" /CELLRANGE = FULL /READNAMES = ON.
      DATASET NAME mySPSSfile WINDOW=FRONT.
      ```
    - **Excel with .xlsx**, all the data, from sheet **`mysheet`** in which the first row does NOT have variable names:
      ```
      GET DATA /TYPE = XLSX /FILE = "filesave/myfile.xlsx"
               /SHEET = NAME "mysheet" /CELLRANGE = FULL /READNAMES = OFF.
      DATASET NAME mySPSSfile WINDOW=FRONT.
      ```
  - When importing data from other programs, if missing data has been indicated by a recognized missing data value code within that program, it will appear as a dot (period) in SPSS data files for numeric variables or as a blank for string (text) variables.
  - To enter data into a brand-new data file (such as for creating "fake people"), `DATA LIST FREE` indicates that a new file is to be created, followed by a `/` and a list of the variables it should contain (in order), terminated by a period. `BEGIN DATA`, terminated by a period, indicates that raw data values are coming next (with spaces or tabs separating the values into columns matching those variables listed), terminated by `END DATA` and another period. Below, we create a new data file with two rows of data for two fake people (IDvar = −99) and give them values of

0 or 1 for the **sexMW** variable and values of 80 or 90 for the **age** variable. The command **DATASET NAME** then gives the new data file a nickname of **fakepeople** to be referred to in the syntax.

-
  ```
  DATA LIST FREE / IDvar sexMW age.
  BEGIN DATA.
  -99  0  80
  -99  1  90
  END DATA.
  DATASET NAME fakepeople.
  ```

- There are two different syntax commands used to merge SPSS data files. If you need to combine data files with different cases but the same variables, use **ADD FILES**. For example below, the **ADD FILES** statement combines the existing **myfile** and **fakepeople** data files as listed on the **FILE** and **/FILE** suboptions, and the statement is terminated by a period. **DATASET NAME** creates a nickname for the new data file called **myfilemerged**:

  -
    ```
    ADD FILES FILE=myfile /FILE=fakepeople.
    DATASET NAME myfilemerged.
    ```

- Alternatively, if you need to combine data files with the same cases but different variables, use **MATCH FILES**. Given that the data files will need to be matched on one or more ID variables (such as **IDvar** here), each data file should be sorted by those ID variables first. In the example below, **DATASET ACTIVATE** is used to direct the syntax to operate on each of the data files to be merged, which are each then sorted by **IDvar** using **SORT CASES**. Each statement is terminated by a period. The **MATCH FILES** statement combines the existing **myfile** and **fakepeople** data files as listed on the **FILE** and **/FILE** suboptions, the **/BY** suboption matches the data files based on **IDvar**, and the statement is terminated by a period. Finally, **DATASET NAME** creates a nickname for the new data file called **myfilemerged**:

  -
    ```
    DATASET ACTIVATE fakepeople.
    SORT CASES BY IDvar.
    DATASET ACTIVATE myfile.
    SORT CASES BY IDvar.
    MATCH FILES FILE=myfile /FILE=fakepeople /BY IDvar.
    DATASET NAME myfilemerged.
    ```

- A special case of **MATCH FILES** is needed when the data files to be merged have different structures, such as when merging multivariate data into stacked data (i.e., merging data with one row per person into data with more than one row per person). Given that the data files will still need to be matched on one or more ID variables (such as **IDvar** here), each data file should be sorted by those ID variables first. In the example below, **DATASET ACTIVATE** is used to direct the syntax to operate on each of the data files to be merged, which are each then sorted by **IDvar** using **SORT CASES**. Each statement is terminated by a period. The **MATCH FILES** statement then creates a new file by combining the stacked file **mystackedfile** and the multivariate file **mymultivfile** data files as listed on the **/FILE** and **/TABLE** suboptions (which are not interchangeable), the **/BY** suboption matches the data files based on **IDvar**, and the statement is terminated by a period. Finally, **DATASET NAME** creates a nickname for the new data file called **myfilemerged**:

  -
    ```
    DATASET ACTIVATE mymultivfile.
    SORT CASES BY IDvar.
    DATASET ACTIVATE mystackedfile.
    SORT CASES BY IDvar.
    MATCH FILES /FILE=mystackedfile /TABLE=mymultivfile /BY IDvar.
    DATASET NAME myfilemerged.
    ```

- **Making new variables, labeling new variables, and using value labels in SPSS syntax:**
  - SPSS variable names can be up to 64 characters. SPSS variable labels (as created by the `VARIABLE LABELS` command, as demonstrated below) can provide even more description (up to 255 characters). SPSS value labels can be created using the `VALUE LABELS` command to assign descriptive labels to specific values of a variable within a data file (as also demonstrated below).
  - The `COMPUTE` statement can be used to make a new variable based on mathematical operations, or the `RECODE` statement can be used to transform an existing variable into a new variable based on select values. Notably, although such data modifications can happen anywhere in the syntax, they will be applied to whichever data file is currently "active" (meaning it was clicked on last or operated on last). Thus, if you have multiple data files open, it is a good idea to use `DATASET ACTIVATE` to direct the syntax to a specific data file prior to any data transformations to ensure they occur in the intended data file.
  - For example below, the `DATASET ACTIVATE` command first directs the syntax to the data file `myfile`, and then `COMPUTE` is used to create a new variable called `age80` in which 0 values represent age 80 from the old variable called `age`. `VARIABLE LABELS` is used to make a new accompanying variable label as well. Each statement is terminated by a period. The `EXECUTE` command makes the transformation happen immediately:

    -
      ```
      DATASET ACTIVATE myfile.
      COMPUTE age = age – 80.
      VARIABLE LABELS age80 "age80: Age in Years (0=80)".
      EXECUTE.
      ```
  - Another way of creating new variables is through conditional logic: a combination of `DO IF`, `ELSE IF`, and `END IF`. `DO IF` lists the first condition (terminated by a period); if that condition is not met, `ELSE IF` lists the others (also terminated by a period). The last condition must be closed with an `END IF` statement (terminated by a period). For example, if an existing variable called `demgroup` = 1, then four new variables are to be created as given next (the `COMPUTE` statements after `DO IF`). Otherwise (`ELSE IF`) `demgroup` = 2, the new variables take on different values, otherwise (`ELSE IF`)… and so on until `END IF`. `VALUE LABELS` provides labels for the values for the original demgroup variable, and `VARIABLE LABELS` creates labels for the new variables. Each statement is terminated once by a period at the end. `EXECUTE` makes the transformation happen immediately:

    -
      ```
      DATASET ACTIVATE myfile.
      DO IF (demgroup=1).
      COMPUTE demNF=0.
      COMPUTE demNC=0.
      COMPUTE demFN=1.
      COMPUTE demFC=0.
      ELSE IF (demgroup=2).
      COMPUTE demNF=1.
      COMPUTE demNC=0.
      COMPUTE demFN=0.
      COMPUTE demFC=0.
      ELSE IF (demgroup=3).
      COMPUTE demNF=0.
      COMPUTE demNC=1.
      COMPUTE demFN=0.
      COMPUTE demFC=1.
      END IF.
      VALUE LABELS demgroup 1 "1None" 2 "2Future" 3 "3Current".
      VARIABLE LABELS demNF "demNF: Dementia Contrast for None=0 vs Future=1"
                      demNC "demNC: Dementia Contrast for None=0 vs Current=1"
                      demFN "demFN: Dementia Contrast for Future=0 vs None=1"
                      demFC "demFC: Dementia Contrast for Future=0 vs Current=1".
      EXECUTE.
      ```

- Some of my examples use a series of `DO IF` and `END IF` commands instead of the `DO IF`, `ELSE IF`, and `END IF` sequence illustrated above. The difference is the idea of *otherwise*: above, each `ELSE IF` will not be read unless the preceding conditions have not been met, whereas any `DO IF` statement will be read regardless. If your conditions are truly mutual exclusive, either will have the same result, but the above is admittedly better practice.
- Certain cases (rows) can be kept or removed based on conditional logic. For example below, the `SELECT IF` command indicates that only cases in which `demgroup` = 1 are to remain in the current data file:

  -
    ```
    DATASET ACTIVATE myfile.
    SELECT IF (demgroup=1).
    EXECUTE.
    ```
- Here is how to select cases to be kept based on whether they have complete data for a series of variables. The `NVALID` function counts the number of non-missing values across the numeric variables listed in its parentheses. Here, we tell SPSS that if the number of valid observations not 4, the case should be removed from the data file:

  -
    ```
    DATASET ACTIVATE myfile.
    SELECT IF NVALID(pred1, dv1, pred2, dv2)=4.
    EXECUTE.
    ```

- **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 `COMPUTE` 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, abbreviated by `TO` and/or separated by commas:
  - Here are some examples:

```
DATASET ACTIVATE myfile.
* Save the mean of these variables as a new variable.
  COMPUTE meanvar = MEAN(var1 TO var10).
* Save the maximum of these variables as a new variable.
  COMPUTE maxvar = MAX(var1 TO var10).
EXECUTE.
```

  - Summarizing across rows can be done using the `AGGREGATE` command, such as in my examples using *stacked* data (in which there is one row per occasion per person). The general procedure is to save the mean across occasions for the desired variables for each person. Those person means can be saved either directly to the current data file or to a new summary data file (in which each person has only one row). In my examples, I usually save the summarized data to a new file for use in generating person-level descriptive statistics for the sample, and then merge the summary data file into the original stacked file.
  - In the example below, `SORT CASES` first sorts the `stacked` data file by the variable to be the unit of summary across rows (`IDvar`). `DATASET DECLARE` then names to the to-be-created summary data file. The `AGGREGATE` command summarizes variables from the pre-sorted `stacked` data file (as indicated by the `/PRESORTED` subcommand) per unique value of `IDvar` (as indicated on the `/BREAK=` subcommand) and saves the summaries to a new data file called `IDvarMeans` (as indicated on the `/OUTFILE=` subcommand). The second line of the `AGGREGATE` command indicates how the specific variables are to be summarized—here, that new variables `meanpred1` and `meanpred2` are to be created in the new `IDvarMeans` data file by taking the mean across rows per unique `IDvar` of variables `pred1` and `pred2` from the existing `stacked` data file. Then, in preparation of merging, the original `stacked` data file is brought back into focus via `DATAEST ACTIVATE`. `MATCH FILES` requests that the new summary data file of means per unique IDvar (via `/TABLE=IDvarMeans`) is then merged back into the currently in focus `stacked` data file (via `/FILE=*`), matching on `IDvar` (via `/BY`):

-
  ```
  DATASET ACTIVATE stacked.
  SORT CASES BY IDvar.
  DATASET DECLARE IDvarMeans.
  AGGREGATE /OUTFILE="IDvarMeans" /PRESORTED /BREAK=IDvar
            /meanpred1=MEAN(pred1) /meanpred2=MEAN(pred2).
  DATASET ACTIVATE stacked.
  MATCH FILES /FILE=* /TABLE="IDvarMeans" /BY IDvar.
  ```

- **Describing data:**
  - If multiple SPSS data files are open simultaneously, then when requesting output of any kind, it is a good idea to first include a `DATA ACTIVATE` command to first tell SPSS which data file to use (such as `myfile` below):

    -
      ```
      DATA ACTIVATE myfile.
      ```
  - `DESCRIPTIVES` can be used to get summary statistics for continuous (quantitative) variables. By default, this includes N, mean, SD, minimum, and maximum (other statistics can be added or removed by listing them on a `/STATISTICS=` subcommand). In the example below, `VARIABLES=` lists the variables to be summarized, and the `DESCRIPTIVES` command is terminated with a period:

    -
      ```
      DESCRIPTIVES VARIABLES=age grip cognition.
      ```
  - Another procedure for summarizing continuous variables is `SUMMARIZE`, which can provide the same statistics as `DESCRIPTIVES` for the entire sample or for subsets of the sample. As an example of the latter below, the subcommand `/TABLES` first lists the variables to be summarized, followed by `BY` and a list of the variables for which summaries should be provided for each unique value. The suboption `/FORMAT=NOLIST TOTAL` indicates that a per-case list should not be printed, but a total summary should be printed. The suboption `/CELLS=` then lists the statistics to be printed, and the `SUMMARIZE` command is terminated with a period:

    -
      ```
      SUMMARIZE /TABLES=age grip cognition BY demgroup
                /FORMAT=NOLIST TOTAL /CELLS=COUNT MEAN STDDEV MIN MAX.
      ```
  - `FREQUENCIES` can be used to get marginal frequencies for the categorical (grouping) variables listed on the `VARIABLES=` option. `CROSSTABS` can be used to get cross-tabulations across categorical variables listed on the `TABLES=` option, each separated by `BY`. `CROSSTABS` by default includes frequencies and percentages per cell, per row, and per column (others can be added or removed by listing them after a `/CELLS=` subcommand). Additional statistics can be requested in either procedure by listing them after a `/STATISTICS=` subcommand. For example below, `FREQUENCIES` provides marginal frequencies for the `sexMW` variable, whereas `CROSSTABS` provides the cross-tabulation of `sexMW` and `demgroup`. Each command is terminated by a period:

    -
      ```
      FREQUENCIES VARIABLES=sexMW.
      CROSSTABS TABLES=sexMW BY demgroup.
      ```
  - `CORRELATIONS` can be used to get Pearson correlations for continuous (quantitative) variables as listed on the implicit `VARIABLES=` option, and the command is terminated by a period:

    -
      ```
      CORRELATIONS VARIABLES=age grip cognition.
      ```

- **Restructuring data:**
  - SPSS 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, `DATASET ACTIVATE` indicates the multivariate data file to be stacked, and then `VARSTOCASES` is used to restructure the previous wide (multivariate) file. Each `/MAKE` statement creates a new stacked variable `FROM` existing multivariate variables, and the `/INDEX` line indicates the name of the new variable to distinguish the rows. The `/KEEP` line instructs SPSS to keep all other variables not mentioned, although the previous multivariate variables will be removed. Finally, `DATASET NAME` indicates the name of the new stacked file.

  -
    ```
    DATASET ACTIVATE multivariatedata.
    VARSTOCASES
    /MAKE age FROM age12 age13 age14
    /MAKE DV  FROM DV12 DV13 DV14
    /MAKE IV  FROM IV12 IV13 DV14
    /INDEX = occasion (3)
    /KEEP = ALL.
    DATASET NAME stackeddata.
    ```

- **A note about SPSS 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, likelihood ratio tests). 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 window. SPSS has macro programming along with a similar procedure called the Output Management System (OMS) that could also be used to create such programs. But given how rarely I teach in SPSS, 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 SPSS MIXED commands used for Longitudinal and Multilevel Modeling (mostly in order of appearance in my example models)

- Note that there is only a single period at the very end of the SPSS `MIXED` command (which I have placed on a new line in my examples for emphasis). This means that comments cannot be placed within the subcommands, and must be place either before or after the entire `MIXED` command.
- **`MIXED` command:**
  - The first required input for the SPSS `MIXED` command is the outcome variable to be analyzed. The option `WITH` is required to indicate model predictors that are to be treated as continuous (i.e., for which a single slope is estimated for each). The option `BY` is used to indicate model 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). If there are no continuous predictors, `WITH` can be omitted; if there are no categorical predictors and no nesting variables (i.e., as used in `/RANDOM=` or `/REPEATED=` subcommands, see below), `BY` can be omitted.
  - For example, to analyze the outcome of `cognition` using continuous predictors of `age85` and `grip9` and a categorical predictor of `demgroup`:
    - `MIXED cognition BY demgroup WITH age85 grip9`

- **Model specification based on the `WITH` and `BY` options of the `MIXED` command:**
  - The `BY` option is used to list model predictors to be treated as *categorical*, which here means that SPSS will automatically create all necessary pairwise contrasts by which to estimate all possible mean differences across groups (unique values) of these variables. By default, SPSS uses the highest number or last string alphabetically 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 will need to re-code your predictor variable so that your desired reference group is indeed the highest numeric or last alphabetic value. Any other predictors to be treated as continuous must be listed on the `WITH` option instead.

- As discussed in the chapter 2 appendix, predictor variables listed on the **BY** option are marginalized over any interacting variables when estimating the interacting variables' main effects for the Type 3 Tests of Fixed Effects. For example, consider an interaction of **demgroup*agegroup**, in which **demgroup** is listed on the **BY** option but **agegroup** is on the **WITH** 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 **BY** 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 **BY** or **WITH** 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 **BY** or **WITH** options.
- Only predictor variables that are listed on the **BY** option can have their model-predicted means per group requested via **/EMMEANS** subcommands (see below).
- When used in **/TEST** subcommands, predictor variables listed on the **BY** option must have a value listed for each possible level of their variable (i.e., each distinct group; see examples below).

- **/METHOD= subcommand**: Select either REML or ML estimation (default is REML): **/METHOD = REML**
- **/PRINT= subcommand options used in my example code (each not printed to the output by default):**
  - **SOLUTION** Print fixed effects solution
  - **TESTCOV** Print SEs and *p*-values for significance test of variance estimates
  - **R** Print the **R** matrix of variances and covariances
  - **G** Print the **G** matrix of random effects variances and covariances
  - **COVB** Print asymptotic covariance matrix of the fixed effects (used for regions of significance)
  - Note that as of SPSS v. 22, the **RCORR**, **GCORR**, **V**, and **VCORR** matrices are not available. Also not available are the predicted random effects (empirical Bayes estimates per higher-level unit).
  - A complete example of the **MIXED** command with the **/METHOD=** and **/PRINT=** subcommands:

```
MIXED cognition BY demgroup WITH age85 grip9
      /METHOD= REML  /PRINT= SOLUTION TESTCOV R G COVB
```

- **/FIXED= subcommand:**
  - The **/FIXED=** subcommand is used to list all fixed effects of model predictors. The fixed intercept is included in the model by default to create at least an empty means model, and does not need to be listed with the other predictors. You can include the fixed intercept explicitly listing INTERCEPT first on the **/FIXED=** subcommand. A empty **/FIXED=** subcommand (i.e., for an empty means model) will generate an error message you can ignore.
  - To remove the default fixed intercept (i.e., as in multivariate models), include the option **| NOINT** to the end of the **/FIXED=** subcommand.
  - As noted above, any predictors listed on the **BY** option of the **MIXED** command will be treated as *categorical*, such that all possible contrasts between a reference group and each other group will be estimated, whereas any predictors listed on the **WITH** option of the **MIXED** command will be treated as *continuous*, such that their slopes

will be estimated instead. This distinction (*categorical* means on the `BY` option of the `MIXED` command, *continuous* means on the `WITH` option of the `MIXED` command) is also maintained when predictors are listed on the `/EMMEANS=` and `/TEST=` subcommands described 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 an asterisk. For example:
- To include main effects of `age85` and `grip9`:

```
     /FIXED= age85 grip9
```

- To include main effects of `age85`, `grip9`, and their two-way interaction:

```
     /FIXED= age85 grip9 age85*grip9
```

- To include main effects of `age85`, `grip9`, `sexMW`, as well as their three possible two-way interactions:

```
     /FIXED= age85 grip9 sexMW age85*grip9 age85*sexMW grip9*sexMW
```

- To add their three-way interaction to the previous model:

```
     /FIXED= age85 grip9 sexMW age85*grip9 age85*sexMW grip9*sexMW age85*grip9*sexMW
```

- To add a main effect and interaction of `sexMW` with `demgroup` to the previous model:

```
     /FIXED= age85 grip9 sexMW age85*grip9 age85*sexMW grip9*sexMW age85*grip9*sexMW
             demgroup demgroup*sexMW
```

- **`/TEST=` subcommand:**
  - The `/TEST=` subcommand in SPSS `MIXED` is used for both the `CONTRAST` and `ESTIMATE` statement functions in SAS `PROC MIXED`, and so each of these functions will be discussed in turn.
  - **First, the `/TEST=` subcommand is 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.
  - Each `/TEST=` subcommand statement allows an optional user-defined label of up to 255 characters included in single or double quotes (which is helpful for documentation more generally). Unfortunately, only one label can be specified for each `/TEST=` subcommand, such that multiple `/TEST=` subcommands must be used in order for each to be labeled, resulting in a separate box of output for each (as opposed to one box for all of them).
  - 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 on the `WITH` option of the `MIXED` command, and for which slopes are estimated), each fixed effect to be included is listed followed by a `1` (i.e., to "turn it on" because you want to test 1 fixed effect for it). Semi-colons are used to separate each effect in order to indicator the degrees of freedom for the test.
  - 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 WITH age85 grip9`

```
      /FIXED= age85 grip9
      /TEST= 'Model R2 Test' age85 1; grip9 1
```

  - If the main effect of a *categorical* predictor (i.e., as listed on the `BY` option of the `MIXED` command, such that 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 through the use of the contrast `-1` and `1` instead. One value for each possible group must be listed after the predictor. The original values of the categorical predictor do not matter, as SPSS simply refers to their numeric or alphabetic order (i.e., given `sexMW` values of `0` or `1`, `0` is first and `1` is second; given string variable values of `M` and `W`, M is first and W 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 BY sexMW WITH age85 grip9
         /FIXED= age85 grip9 sexMW
         /TEST= 'Model R2 Test' age85 1; grip9 1; sexMW -1 1
    ```

- 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 through the use of two separate contrasts involving **-1** and **1** instead. One value for each group must be listed after the predictor in both contrasts, but which specific groups get the **-1** and **1** doesn't matter, so long as the two contrasts are indeed different. 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. Given the interchangeability of the contrast values, either **/TEST=** subcommand below provides an equivalent joint test of significance of the model predictors with five degrees of freedom as follows:

  -
    ```
    MIXED cognition BY sexMW demgroup WITH age85 grip9
         /FIXED= age85 grip9 sexMW demgroup
         /TEST= 'Model R2 Test version 1' age85 1; grip9 1; sexMW -1 1;
                                          demgroup -1 1 0; demgroup -1 0 1
         /TEST= 'Model R2 Test version 2' age85 1; grip9 1; sexMW -1 1;
                                          demgroup 0 -1 1; demgroup 1 0 -1
    ```

- The **/TEST=** 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 on the **BY** option. 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): s1d1 s1d2 s1d3 s2d1 s2d2 s2d3.
    * First 3 /TESTs provide sexMW simple effects within each demgroup.
    * Second 3 /TESTs provide sexMW simple effect differences across demgroups.
    MIXED cognition BY sexMW demgroup WITH age85 grip9
         /FIXED= age85 grip9 sexMW demgroup
         /TEST= 'Sex diff in demgroup 1' sexMW*demgroup -1  0  0  1  0  0
         /TEST= 'Sex diff in demgroup 2' sexMW*demgroup  0 -1  0  0  1  0
         /TEST= 'Sex diff in demgroup 3' sexMW*demgroup  0  0 -1  0  0  1
         /TEST= 'Sex diff: demgroup 1v2' sexMW*demgroup -1  1  0  1 -1  0
         /TEST= 'Sex diff: demgroup 1v3' sexMW*demgroup -1  0  1  1  0 -1
         /TEST= 'Sex diff: demgroup 2v3' sexMW*demgroup  0 -1  1  0  1 -1
    ```

- **A second use of the /TEST= subcommand** is 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), in which the values given should indicate the corresponding values for the predictor variables in the model equation:

  -
    ```
    MIXED cognition BY sexMW demgroup WITH age85 grip9
         /FIXED= age85 grip9 sexMW demgroup
         /TEST= 'Cognition for age85=-5, grip9=3, sexMW=M, demgroup=2'
                 intercept 1 age85 -5 grip9 3 sexMW 1 0 demgroup 0 1 0
         /TEST= 'Cognition for age85=-5, grip9=3, sexMW=W, demgroup=3'
                 intercept 1 age85 -5 grip9 3 sexMW 0 1 demgroup 0 0 1
    ```

- Note that in requesting model-predicted outcomes, the intercept 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 *continuous* predictor (those listed on the **WITH** option of the **MIXED** command) without a value given is assumed to be held to **0** (i.e., held to its reference value), but the same is not true for any *categorical* predictors (i.e., those listed on the **BY** option of the **MIXED** command), for which a value must be given explicitly. For instance, the following would result in an error message because of the missing values for **demgroup**:

  ```
  MIXED cognition BY sexMW demgroup WITH age85 grip9
       /FIXED= age85 grip9 sexMW demgroup
       /TEST= 'Cognition for age85=-5, grip9=3, sexMW=W, demgroup=???'
               intercept 1 age85 -5 grip9 3 sexMW 0 1
  ```

- If you want to average across levels of a *categorical* predictor, then each of its possible levels needs to be included explicitly in forming that average. In the example below, the **DIVISOR** option is used in order to specify fractional values by which to create an average across groups. Whole numbers are given as values for the numerator, and **DIVISOR** then provides the denominator value. Here, we assign values of **1/3** to each possible level of **demgroup** (and thus all other values given must be multiplied by **3** to counter-act the divisor):

  ```
  MIXED cognition BY sexMW demgroup WITH age85 grip9
       /FIXED= age85 grip9 sexMW demgroup
       /TEST= 'Cognition for age85=-5, grip9=3, sexMW=W, demgroup=Mean'
               intercept 3 age85 -15 grip9 9 sexMW 0 3 demgroup 1 1 1 DIVISOR=3
  ```

- Below is an example of how to compare coefficients using the **/TEST=** 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 **/TEST=** 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 WITH YearsSinceBirth YearsUntilDeath
       /FIXED= YearsSinceBirth YearsUntilDeath
       /TEST= '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 **demgroup** categorical predictor that was on the **BY** option of the **MIXED** command has instead been represented by two manually-created dummy-coded continuous predictors of **demNF** and **demNC** (for none vs. future and none vs. current (included on the **WITH** option of the **MIXED** command instead, as described in chapter 2). The **/TEST=** subcommand below provides the missing comparison of future vs. current, which is the difference of **demNF** and **demNC** coefficients created by the subtraction of their values:

  ```
  MIXED cognition BY sexMW WITH age85 grip9 demNF demNC
       /FIXED= age85 grip9 sexMW demNF demNC
       /TEST= 'Future vs. Current when demgroup is not categorical'
               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 **WITH** option of the **MIXED** command). Each **/TEST=** subcommand below contains the simple effect to be estimated and how it is modified by any interacting predictors as specified in the **/FIXED=** subcommand. The interaction terms in the **/TEST=** 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 **/TEST=** subcommands build an **age85** slope by adding its simple effect given directly by the model (when **grip9**=0) to the interaction value for how the **age85** slope changes per unit of

grip9, whereas the second two /TEST= subcommands build a **grip9** slope by adding its simple effect given directly by the model (when **age85**=0) to the interaction value for how the **grip9** slope changes per unit of **age85**. Note that because **sexMW** and **demgroup** do not interact with **age85** or **grip9** in the /FIXED= subcommand, they should not be included in their simple effect /TEST= subcommands:

-
  ```
  * First 2 /TEST= provide simple effects of age85 conditional on grip9.
  * Second 2 /TEST= provide simple effects of grip9 conditional on age85.
  MIXED cognition BY sexMW demgroup WITH age85 grip9
       /FIXED= age85 grip9 sexMW demgroup age85*grip9
       /TEST= 'Age85 Slope if Grip9=-3' age85 1 age85*grip9 -3
       /TEST= 'Age85 Slope if Grip9= 3' age85 1 age85*grip9  3
       /TEST= 'Grip9 Slope if Age85=-5' grip9 1 age85*grip9 -5
       /TEST= 'Grip9 Slope if Age85= 5' grip9 1 age85*grip9  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 **WITH** option of the **MIXED** command) with a *categorical* predictor (listed on the **BY** option of the **MIXED** command). The logic is the same, but the categorical predictor must have values given for each of its unique levels. The first set of /TEST= subcommands provide the simple effect of **age85** for each level of **demgroup** (via the placement of the **1** for **age85*demgroup**), whereas the second set provides the differences in the simple effects of **age85** between levels of **demgroup** (via the placement of the **-1** and **1** for **age85*demgroup**). The third set provides the simple effects of **demgroup** for specific levels of **age85**, in which the contrast values of **-1** and **1** are multiplied by the values for **age85** (note that when **age85**= **-5**, the signs reverse accordingly):

  -
    ```
    * First 3 /TEST= provide simple effects of age85 conditional on demgroup.
    * Second 3 /TEST= provide age85 simple effect differences between demgroups.
    * Last 6 /TEST= provide simple differences for demgroup conditional on age85.
    MIXED cognition BY sexMW demgroup WITH age85 grip9
         /FIXED= age85 grip9 sexMW demgroup age85*demgroup
         /TEST= 'Age85 Slope if DemGroup=1' age85 1 age85*demgroup 1 0 0
         /TEST= 'Age85 Slope if DemGroup=2' age85 1 age85*demgroup 0 1 0
         /TEST= 'Age85 Slope if DemGroup=3' age85 1 age85*demgroup 0 0 1
         /TEST= 'Age85 Slope: DemGroup=1vs2' age85*demgroup -1  1  0
         /TEST= 'Age85 Slope: DemGroup=1vs3' age85*demgroup -1  0  1
         /TEST= 'Age85 Slope: DemGroup=2vs3' age85*demgroup  0 -1  1
         /TEST= 'Demgroup 1vs2 if Age85=5'  demgroup -1  1 0 age85*demgroup -5  5  0
         /TEST= 'Demgroup 1vs3 if Age85=5'  demgroup -1  0 1 age85*demgroup -5  0  5
         /TEST= 'Demgroup 2vs3 if Age85=5'  demgroup  0 -1 1 age85*demgroup  0 -5  5
         /TEST= 'Demgroup 1vs2 if Age85=-5' demgroup -1  1 0 age85*demgroup  5 -5  0
         /TEST= 'Demgroup 1vs3 if Age85=-5' demgroup -1  0 1 age85*demgroup  5  0 -5
         /TEST= 'Demgroup 2vs3 if Age85=-5' demgroup  0 -1 1 age85*demgroup  0  5 -5
    ```

- Given the number of distinct values to be specified in a /TEST= subcommand to represent all possible unique group combinations, simple effects within interactions among *categorical* predictors (i.e., those included on the **BY** option) are much easier to specify using /EMMEANS, as shown next.

- /EMMEANS **subcommand:**

  - The /EMMEANS= subcommand is used to request model-predicted means for *categorical* predictors only (i.e., those listed on the **BY** option of the **MIXED** command). Similar to their listing on the /FIXED= subcommand, listing a categorical predictor by itself within the **TABLES( )** option of the /EMMEANS= subcommand requests its marginal means, whereas listing its interaction with another categorical predictor requests the mean for each unique combination of groups implied by the interaction (i.e., cell means or simple means). However, a separate /EMMEANS= subcommand is needed for each set of means to be requested. For example:

- To request only the marginal means for `sexMW` and for `demgroup`:

```
     MIXED cognition BY sexMW demgroup
          /FIXED= sexMW demgroup
          /EMMEANS= TABLES(sexMW)
          /EMMEANS= TABLES(demgroup)
```

- To request only the cell means for each unique combination of `sexMW` by `demgroup`:

```
     MIXED cognition BY sexMW demgroup
          /FIXED= sexMW demgroup sexMW*demgroup
          /EMMEANS= TABLES(sexMW*demgroup)
```

- To request both marginal means and cell means for each unique combination of `sexMW` by `demgroup`:

```
     MIXED cognition BY sexMW demgroup
          /FIXED= sexMW demgroup sexMW*demgroup
          /EMMEANS= TABLES(sexMW)
          /EMMEANS= TABLES(demgroup)
          /EMMEANS= TABLES(sexMW*demgroup)
```

- The `COMPARE( )` option of the `/EMMEANS=` subcommand can be used to obtain simple main effects and pairwise comparisons for the predictor listed within `( )` for decomposing interactions. Unadjusted *p*-values are printed by default, but more conservative *p*-values can be obtained by adding the `ADJUST( )` option.

- For example, the first `/EMMEANS=` subcommand `COMPARE( )` option requests the omnibus main effect and pairwise comparisons across `demgroup` for each level of `sexMW`, and the second instance does the opposite:

  -
```
MIXED cognition BY sexMW demgroup
     /FIXED= sexMW demgroup sexMW*demgroup
     /EMMEANS= TABLES(sexMW*demgroup) COMPARE(demgroup)
     /EMMEANS= TABLES(sexMW*demgroup) COMPARE(sexMW)
```

- The `WITH( )` option of the `/EMMEANS=` subcommand is used to list continuous predictors (those on the `WITH` option of the `MIXED` command) and their values at which to hold constant in estimating means for levels of the categorical predictors.

- For example, to hold `age85` at `5` and `grip9` at `3` in estimating conditional cell means and differences for each combination of `demgroup*sexMW`:

  -
```
MIXED cognition BY sexMW demgroup WITH age85 grip9
     /FIXED= sexMW demgroup sexMW*demgroup age85 grip9
     /EMMEANS= TABLES(sexMW*demgroup) COMPARE(demgroup)
               WITH (age85=5 grip9=3)
     /EMMEANS= TABLES(sexMW*demgroup) COMPARE(sexMW)
               WITH (age85=5 grip9=3)
```

- For example, to hold `age85` and `grip9` at their means instead (which is the default otherwise):

  -
```
MIXED cognition BY sexMW demgroup WITH age85 grip9
     /FIXED= sexMW demgroup sexMW*demgroup WITH age85 grip9
     /EMMEANS= TABLES(sexMW*demgroup) COMPARE(demgroup)
               WITH (age85=MEAN grip9=MEAN)
     /EMMEANS= TABLES(sexMW*demgroup) COMPARE(sexMW)
               WITH (age85=MEAN grip9=MEAN)
```

- `/RANDOM=` **subcommand:**
  - The `/RANDOM=` subcommand is used to indicate the structure of the **G** matrix of random effects variances and covariances. If you omit the `/RANDOM` subcommand, 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 a `/RANDOM=`

subcommand, then the total variance and covariance across observations will be created in the **V** matrix from **G**, **R**, and **Z**, as **V** = **ZGZ**^{T} + **R**, as described in chapter 5. Thus, without the `/RANDOM=` subcommand, **V** = **R**.

- The following options have been used in my examples, placed after the `|` on the `/RANDOM=` subcommand:
  - `SUBJECT( )` Identification variable for how rows of data should be joined together (random per what?) (ID variable is treated as categorical, so it should be listed on the `BY` option of the `MIXED` command)
    Note that multiple ID variables can be specified as joined by an asterisk to create a unique value for each case, e.g., `SUBJECT(GroupID*PersonID)`
  - `COVTYPE( )` Structure of the **G** matrix, which should ALWAYS be `COVTYPE(UN)` for unstructured in order to estimate covariances among random effects at the same level (see chapter 5), even though the default is `COVTYPE(VC)` for uncorrelated random effects instead
    Note that when only a random intercept variance is estimated, and thus **G** is a scalar, `COVTYPE(UN)` will generate an error message you can ignore
- In multilevel models, you must explicitly list any effects for which you want to estimate a random effect variance, including the intercept (which is not included by default). Notably, to prevent estimation problems, all random effects should be for continuous predictors only (those listed on the `WITH` option of the `MIXED` command). If you want to estimate a random effect variance for differences across levels of a categorical predictor, those differences should be represented via manual contrasts treated as continuous predictors instead.
  - For example, to allow only the intercept to vary randomly across levels of `PersonID`:

```
MIXED cognition BY PersonID WITH time
     /FIXED= time
     /RANDOM= INTERCEPT | COVTYPE(UN) SUBJECT(PersonID)
```

  - For example, to allow both the intercept and time slope to vary randomly across levels of `PersonID`:

```
MIXED cognition BY PersonID WITH time
     /FIXED= time
     /RANDOM= INTERCEPT time | COVTYPE(UN) SUBJECT(PersonID)
```

- **`/REPEATED` subcommand:**
  - The `/REPEATED=` subcommand is used to indicate the structure of the **R** matrix of variances and covariances (which are *residual* if a `/RANDOM=` subcommand has been included, but *total* otherwise). If you do not explicitly include a `/REPEATED=` subcommand, by default the model still includes a `COVTYPE(ID)` diagonal **R** matrix, in which all observations are assumed to have the same residual/total variance with no covariances.
  - The following options have been used in my examples, placed after the `|` on the `/REPEATED=` subcommand:
    - `SUBJECT( )` Identification variable for how rows of data should be joined together
      (ID variable is treated as categorical, so it should be listed on the `BY` option of the `MIXED` command)
      Note that multiple ID variables can be specified as joined by an asterisk to create a unique value for each case, e.g., `SUBJECT(GroupID*PersonID)`
    - `COVTYPE( )` Structure of the **R** matrix (see chapter 4 for details), such as: `ID` for Independent (same as `VC` for Variance Components), `CS` for Compound Symmetry, `CSH` for Compound Symmetry Heterogeneous, `AR1` for First-Order Auto-Regressive, `ARH1` for First-Order Auto-Regressive Heterogeneous, `TP` for Toeplitz, `TPH` for Toeplitz Heterogeneous, and `UN` for Unstructured
      Note that the fullest possible order is the only option for the `TP`, `TPH`, and `UN` **R** matrices
  - In longitudinal data, you may list a time-level ID variable by which to organize the rows and columns of the **R** matrix (this ID variable must also be listed on the `BY` option of the `MIXED` command). This distinction is only

relevant in the presence of missing data (in which case you want the correct value to go to the correct row and column). For example, to structure the **R** matrix by the time-level variable `studyday` per `PersonID`:

  -
    ```
    MIXED cognition BY studyday PersonID
          /REPEATED= studyday | COVTYPE(UN) SUBJECT(PersonID)
    ```

- `/SAVE=` **subcommand:**
  - The `/SAVE=` subcommand in SPSS `MIXED` can be used to save model-predicted outcomes to a dataset using the name given as the new column. The `FIXPRED` suboption specifies predictions from fixed effects only:
    - `/SAVE= FIXPRED(PredValues)`

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 SPSS 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).
  - 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 SPSS 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, no `/REPEATED=` or `/RANDOM=` subcommands were specified. The default `/REPEATED=` subcommand if one is not provided is a diagonal **R** matrix of `TYPE(ID)` in which the residual variance is assumed constant over all observations with no covariance across observations.
  - The method of fake people was used as a means to create predicted outcomes for use in plotting interactions. In this method (as introduced in chapter 2), a data file of fake cases with desired values of the model predictors is created and merged into the real data, and predicted outcomes are then requested for each case. The fake cases will not contribute to the model given that they do not have values for the outcome variable, but they will receive model-predicted values you can then plot.

- **Chapter 3a**
  - 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`.

- **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, **session**, in the **/FIXED=** subcommand also appears in the **BY** option of the **MIXED** command, such that it 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 **/REPEATED=** 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 **/REPEATED=** subcommand was used for the **R**-only models, whereas both the **/RANDOM=** and **/REPEATED=** 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 were not able to be estimated in SPSS.

- **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 **/RANDOM=** and **/REPEATED=** subcommands were used. However, the **/REPEATED=** subcommands in which **TYPE(ID)** were 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 SPSS.

- **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 **/REPEATED=** subcommand was not used given that it was never anything other than the default **TYPE(ID)** (the other types are for balanced data).
  - The heterogeneous variance models could not be estimated using SPSS.

- **Chapter 7b**
  - 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**.
  - The method of fake people was again used as a means to create predicted outcomes for use in plotting interactions.
  - Because time was unbalanced, the **/REPEATED=** subcommand was not used given that it was never anything other than the default **TYPE(ID)** (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.
  - **/TEST=** subcommands were used to create predicted outcomes instead of the method of fake people.
  - The heterogeneous variance models could not be estimated using SPSS.

- **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** was used to estimate the multivariate longitudinal models 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. However, this was only possible for the undirected effects version (in which the relationships between monitoring and risky behavior were specified using covariances rather than directed effects).
  - The slopes-as-outcomes models were also not estimated using **MIXED**. The random effects version would not be possible given that empirical Bayes estimates for the predicted random effects are not directly available. Although the fixed effects version is possible, it would require additional data transformation steps to save the individual-specific fixed effects (i.e., either by using the OMS system 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 again used as a means to create predicted outcomes for use in plotting interactions. Note that missing data were deliberately left for cases with impossible values (i.e., that would create dead people).
  - Because time was unbalanced, the **/REPEATED=** subcommand was not used given that it was never anything other than the default **TYPE(ID)** (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 again used as a means to create predicted outcomes for use in plotting.
  - The three-level models required two **/RANDOM=** subcommands (to structure the **G** matrix for levels 2 and 3), which can be distinguished by the ID variables given in their **SUBJECT( )** options. In this example, **SUBJECT(PersonID)** indicated level 3 (for unique persons), whereas **SUBJECT(PersonID*burst)** indicated level 2 (for the unique combination of person and burst).
  - Although it could have been used given balanced sessions at level 1, the **/REPEATED=** subcommand was not used for the univariate models given that it was never anything other than the default **TYPE(ID)** in this example.
  - For the multivariate model, the **/REPEATED=** subcommand was used to structure the **R** matrix for repeated **DV** outcomes within a session (within a burst within a person) via **SUBJECT(PersonID*burst*session)**.

- **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. A significant amount of data manipulation was required to create all the summary variables needed with which to build the model predictors, resulting in several intermediate data files. These were then used to summarize data at the different levels of analysis and for some specific models as well.
- The three-level models again required two `/RANDOM=` subcommands (to structure the **G** matrix for levels 2 and 3), which can be distinguished by the ID variables given in their `SUBJECT( )` options. In this example, `SUBJECT(ClassID)` indicated level 3 (for unique classes), whereas `SUBJECT(ClassID*StudentID)` indicated level 2 (for the unique combination of class and student).
- The saturated means models included **G** and **R** matrices for classes and students that were unstructured across occasions, such that the `/RANDOM=` subcommand did not include an intercept, and instead included the categorical predictor of `wave` (as did the `/REPEATED=` subcommand).
- All level-1 and level-2 predictors were treated as observed variables that were not included in the likelihood.
- Note that most 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. A significant amount of data manipulation was required to create all the summary variables needed with which to build the model predictors, resulting in several intermediate data files. These were then used to summarize data at the different levels of analysis.
  - 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., `ClassID_year0*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 `/RANDOM=` subcommand per year (whose `SUBJECT( )` 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 `/RANDOM=` subcommands, no covariances were estimated among them.

- **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 `/RANDOM=` subcommand for each level-2 part of the **G** matrix, which can be distinguished via the ID variables in their `SUBJECT( )` option. A general random intercept was included for both subjects and items, as well as random effects for item predictors across subjects. However, the use of two separate `/RANDOM=` subcommands means that any random effects for subjects will not have covariances with any random effects for items, but that random effects from the same dimension will have covariances given the `TYPE(UN)` option on each `/RANDOM=` subcommand.