<u>Detailed steps for basic cleaning and merging of research site study with 5 data files</u>

1. Explicit file – tall format organized by *session_id*
    - a) Import explicit file. Set first data row to 2 (header is first row)
        - i) Increase length of long variables: If have long question and questionnaire names, increase length of *question_name* and *questionnaire_name* variables from the program default. If have text box responses, increase length of *question_response* variable.
            - (1) Warning: some programs have a limit on how many characters can be input from a single line and may cut off long responses
            - (2) If have long text box responses, errors may occur importing data. Suggestion: edit raw text file and move open response text to another file before importing
        - ii) *study_name* should be your study – if not, be alarmed
        - iii) Suggestion: remove 'feedback', 'text', and 'd' from *question_name*
    - b) Sort by *session_id*, *questionnaire_name*, *question_name*
    - c) Identify and delete repeat rows in the data
        - i) If the same values exist for *session_id*, *questionnaire_name*, and *question_name* in two consecutive rows, then some kind of server or user error caused a row of data to repeat.
        - ii) Delete the repeated row.
        - iii) If there are many repeats, suggests a problem with data collection. Should be less than .5%
    - d) Transpose data from tall to wide format
        - i) Designate *session_id* as the rows
        - ii) Designate *question_name* as columns
        - iii) Designate *question_response* as data values (cells in the dataset)
        - iv) Errors on this step result from questions being named the same thing. The program can't place 2 data points into one cell. If two questions are named the same thing in two different questionnaires, the data can be salvaged by renaming one of the questions prior to transposing the data. If two questions are named the same thing within a questionnaire, the data has likely been overwritten.
            - (1) Hint: NEVER name two things the same
        - v) Sometimes a server error will cause this to happen with one or two sessions and doesn't mean there was a coding error.
            - (1) Identify sessions with repeat data and investigate in the raw data file. These will often require deletion. If so, remove the entire session from the raw explicit file and re-import the explicit file.
            - (2) Keep track of these sessions and report as 'computer error' in method section.

vi) Much Project Implicit data is saved as character data. Recode numeric variables that are saved as character variables into numeric variables

e) Examine descriptives (mean, sd, min, max) of all numeric variables for any coding errors.
   - i) If find coding errors, reexamine study files for original coding and recode the values to reflect how the variables should have been coded originally
     - (1) .jsp files will sometimes have '-900' values from two possible sources:
       - (a) If the study used dependency questions in .jsp files, will get '-900' values for questions that were not assigned to the participant
       - (b) On Demo site data, '-900' values are assigned to dependent demographic variables that were not assigned to the participant
       - (c) In most cases, set these values to missing
     - (2) .jsp files will also have extra variables that will be the *question_name* followed by rt and trt. RT variables are the time it takes to continue to the next page and TRT is the total time spent on that jsp (through all the items).
       - (a) These variables will be 0 if the item was not assigned because of a dependency
     - (3) '-900,' 'rt' and 'trt' variables are not typically found in other files formats than .jsp (xml, html)

2. Sessions file – wide format organized by *session_id* and *user_id*
   - a) Import session file. Set first row to 2 (header is first row)
     - i) Set the following variables to date/time format: *session_date*, *last_update_date*, *creation_date*
   - b) *study_name* should be your study – if not, be alarmed
   - c) Sort by *user_id*

3. Demographics file– tall format organized by *user_id*
   - a) Import demographics file. Set first row to 2 (header is first row)
     - i) Increase length of characteristic and *value*
   - b) Sort data by *user_id*
   - c) Transpose data from tall to wide format
     - i) Designate *user_id* as the rows
     - ii) Designate *value* as columns
     - iii) Designate *characteristic* as data values (cells in the dataset)
   - d) See 'PI Demographics Data Dictionary' document for demographics variable information

4. SessionTasks file – wide format organized by *session_id*
   - a) Import sessionTasks file. Set first row to 2 (header is first row)

        i) Many variables are repeats from sessions file and can be disregarded. Variables of interest are *task_id* and *task_number*
        ii) Increase length of *task_id* variable
    b) Sort by *session_id* and *task_id*
    c) Identify and delete repeat rows in the data
        i) If the same values exist for *session_id and task_id* in two consecutive rows, then some kind of server or user error caused a row of data to repeat.
        ii) Identify sessions with repeat data and investigate in the raw data file. These may require deletion. If so, remove the entire session from the raw explicit file and re-import the sessionTasks file.
            (1) Repeat rows are rare in the sessionTasks file
    d) Transpose data from tall to wide format
        i) Designate *session_id* as the rows
        ii) Designate *task_id* as columns
        iii) Designate *task_number* as data values (cells in the dataset)
    e) Identify experimental conditions and counterbalances
        i) Run descriptives on each variable – values are the *task_id*
        ii) Create new variables that code for conditions and counterbalances
    f) Create new variables that code for study session information
        i) Whether people consented (if consent=1)
        ii) Whether people were debriefed (if debrief=1)
    g) Retain *session_id*, as well as new condition, counterbalance, consented and debriefed variables
        i) Drop all other variables

5. Iat file– tall format organized by *session_id*
    a) Import iat file. Set first row to 2 (header is first row)
        i) Drop unnecessary variables: *block_number*, *block_trial_count*, *study_name*, *task_number*, *trial_name*, *trial_response*
    b) Sort by *session_id*, *task_name*, *block_name*, *trial_number*
    c) Identify and delete repeat rows in the data
        i) If the same values exist for *session_id, task_name, block_name,* and *trial_number* in two consecutive rows, then some kind of server or user error caused a row of data to repeat.
        ii) Identify rows with repeat data and delete the repeat rows
            (1) Repeat rows are relatively common in the iat file and are usually because of server errors – simply delete the repeated row
    d) Run IAT algorithm

6. Sort and merge data files
   a) Sort and merge sessions and demographics files
      i) Sort sessions file by *user_id*
      ii) Sort demographics file by *user_id*
      iii) Merge sessions and demographics files by *user_id*
   b) Sort and merge session_demographics merged file with other files
      i) Sort session_demographics file by *session_id*
      ii) Sort explicit file by *session_id*
      iii) Sort iat file by *session_id*
      iv) Sort sessionTasks file by *session_id*
      v) Merge session_demographics file, explicit file, iat file, and sessionTasks file by *session_id*
   c) Check log when merging files and ensure that the number of rows per file is similar for sessions and sessionTasks, then iat and explicit will be similar but have fewer, and demographics will have many more.
   d) You should now have a dataset that is organized with one participant per row

7. Basic recommended cleaning and coding
   a) Create age by subtracting birth information (*byear* and *bmonth*) from the date they took the study (*session_date*)
   b) Delete people who did not consent – these do not count as sessions since they were not started
   c) Identify IAT data with high error rate (>30% overall or >40% in any one block) and set IAT score to missing
   d) Set IAT scores to missing if subexcl is not 0

8. Areas to identify bad data and participant misbehavior
   a) Participants who mindlessly click through .jsp files – can use RT on individual .jsp items
   b) Fast latencies on latency tasks
      i) sessionTasks file has information on time spent on every task