Survey Pivoter

Functional Specifications and Requirements

|  |  |
| --- | --- |
| Overview | Bowdoin conducts surveys using Qualtrics, an online survey tool, and performs analysis on the results. Typical survey data files are provided by Qualtrics in a one row per survey respondent format. This format can make it difficult to perform certain types of analysis, particularly when using analysis and data visualization tools such as Tableau. There is a need to be able to easily “pivot” this data into a one row per question per response format that would be easier to analyze. |

# Input Files

Survey files will either come from Qualtrics directly or from COFHE. In either case, they will adhere to a similar format. There will be four sources of data to work with:

|  |  |
| --- | --- |
| Responses as text | A spreadsheet containing one row per student and one column per question. Responses to the questions are recorded as text (e.g. “Satisfied”). |
| Responses as values | A spreadsheet containing one row per student and one column per question. Responses to the questions are recorded as values (e.g. “5”). |
| Question names & labels | A spreadsheet containing one row per question, with a column titled “Name” containing the question number (e.g. Q1, Q2, etc.) and a column titled “Label” containing the question text (e.g. “How satisfied were you with ...”). This spreadsheet may have other columns.  Note: This spreadsheet is likely found in the same file as the “Response sets for each question” spreadsheet but may not always be. |
| Response sets for each question | A spreadsheet containing the set of possible responses to each question.  Note: This spreadsheet is likely found in the same file as the “Question names & labels” spreadsheet but may not always be. |

# Output

|  |  |
| --- | --- |
| 1 | The script should output a single new file that’s a .xlsx file. |
| 2 | The output file should be in the format of one row per respondent per response. |
| 3 | There should be a single column containing the **question text** for each row. |
| 4 | There should be a single column containing the **question number** for each row. |
| 5 | There should be a single column containing the **numerical response** for each row. Rows for which a numerical response is not relevant should have a NULL value or empty string in this column. |
| 6 | There should be a single column containing the **textual response** for each row. Rows for which a textual response is not relevant should have the value in the numerical response column repeated in the textual response column. |
| 7 | There should be a single column containing the question **group number** for each row. Question grouping is a way to group similar questions together. See below for more details. |
| 8 | There should be a single column containing the question **group text** for each row. Question grouping is a way to group similar questions together. See below for more details. |
| 9 | Question grouping can be determined using question numbers. Questions which belong in a group are numbered using the following patterns:   * **~~Q#n~~** ~~– Where “#” is a number and “n” is a letter. For example, Q1 and Q1a should belong in the same group~~. * **Q#\_#** - Where “#” is a number. For example, Q2\_1 and Q2\_2 should belong in the same group.   **Determining Grouped Question Numbers**  Using the above patterns, the question group number assigned to a group should be the common elements between questions, removing underscores. For example, Q1 and Q1a should be grouped together with question number Q1 and similarly Q2\_1 and Q2\_2 should be grouped together as Q2.  For questions that aren’t part of a group, the question number should be repeated as the question group number. For example, Q3 should have a question group number of Q3.  **Determining Grouped Question Text**  Using the above patterns, the question group text assigned to a group should be the common elements between the question text. If a common string does not exist for questions in a group, the group number should be repeated as the grouped question text. |
| 10 | A number of columns from the input files should be treated as attributes for the respondent and repeated on every row instead of being treated as questions. In general, these columns will have a “name” starting with something other than “Q” (e.g. V5). These attributes should be added as separate columns with a column name of the “Label” (e.g. if the label for V5 is “Email” then “Email” should be the column name). |
| 11 | Certain questions should be able to be treated as both questions (i.e. one row per response with the response in the “Response” columns) and as attributes (i.e. added as a column with a column name of the question label and values of the response text).  The specification of these questions needs to be determined. It may be an additional input file, or a configuration setting, or some other means. TBD. |
| 12 | An additional column called “Count Negative” should be created. The values for this column should be based on the range of responses for each question with the top half of the responses receiving positive values and the bottom half negative values. For questions with an odd number of response types, the middle response should receive a value of zero. What about even number of response types? |
| 13 | The order of columns in the output file should be sorted as follows:   1. Attribute columns 2. Question group columns 3. Question columns 4. Response columns |

**Notes**:

1. Regarding count\_negative:
   * We need to modify the current ***count\_negative*** function, but this should be done easily.
   * What to do when there is a “Not-a-response” value (9 currently) that is not part of the response?
   * What to do with Nan values?
2. Right now, the STUID is being included twice. Do we want to disable the automatic inclusion of ID?
3. Should there be an option to completely disregard certain columns? (i.e do not include that column in the output file at all – this will help speed up the pivoting process)
4. Should we replace all missing values in the data with empty string? As of right now, missing (NULL) data are outputted as “nan”.