To start off with Data Wrangling I imported the CSV's into a list of Dataframes and began inspecting. I found that the earlier questionnaires had much fewer questions and combined to a total of 2692 respondents, so I thought it would be a good idea to normalize the data based on the fewest amount of questions and drop the bulk that would have a majority of missing entries.

To start with normalizing the data I inspected the intersection of columns for each data set with the standard I chose to use, and renamed applicable cols to create consistency. Once all of the columns were renamed I redefined the data sets to be subsets only including the questions I wanted.

Once I had all of my data sets with normalized column headings I began to inspect values and create dictionaries to transform text to numeric values.

ex . of some dicts I created:

```
family_history_map = {'No':0, 'Yes':1, "I don't know":2}
treatment_map = {'No': 0, 'Yes': 1}
work_interfere_1_map = {'No':0, 'Yes':1, "Not applicable to me":2}
```

Once I had my dictionaries set I used Pandas.Series.map() function to map data to numeric values. Gender had more variation so I ran a for loop to cover the cases. Finally, country and state were the final columns left that I need to transform to numeric values.

To get the list of countries I ran a for loop and created a dictionary to store both the unique countries and the count of their occurrence. I decided that for my purposes I would only use countries with at least 50 entries, and the rest would fall under ‘other’. This left me with 7 countries and ‘other’.

The state mapping would have been nicer, but just one of the data sets decided to use abbreviations instead of the full spelling, so it became a two-part process. First I mapped the abbreviated data to use the full spelling of the state, then I ran a for loop to map all of the data sets’ states to numeric values.

During the mappings wherever it made sense in the context of the question I chose to replace null values with a number representing ‘I don’t know’. If the context did not allow I chose to leave null entries.

With all of my headings normalized and values, numeric I merged all of the data sets into one data frame and wrote it to a csv file titled ‘data_cleaned.csv’

---------- Update ----------

After discussing with my mentor we decided to use Panda's GetDummies, as this solution is more versatile with larger datasets. I went back to the dictionaries I had created, and rather than setting to numerical values I just made sure to map all of the text answers to be consistent. For example, if some years answered 'I don't know' and other years answered 'Don't know' for a column, I mapped all of the values to 'Don't know'.

Once text values were consistent I merged the DataFrames into 1 df. From there I looped through the Age values, and set any NaN values to the average of the Age column.

Finally, I used pd.getDummies to get dummy variables and wrote the dataframe with dummy vars to a csv file.

Check out code here:
https://github.com/saturnianangel/Springboard/blob/master/Capstone_I/CapstoneProject1_DataWrangling.ipynb