A comprehensive list of the changes made in the project files:

1. Download dependencies
2. Download and install openpyxl (library)

# Extraction of data

1. Reading data into Pandas DF
2. Brief description of dataframe using .info()
3. Getting the columns within the dataframe

# Creation of Category and Sub-Category DF

1. Segmentation of column values within dataframe, like module 1 challenge:

* Reference (column segmentation)

<https://pandas.pydata.org/docs/reference/api/pandas.Series.str.split.html#pandas.Series.str.split>

1. getting the unique values within columns and creation of lists

* Reference (unique values)

<https://pandas.pydata.org/docs/reference/api/pandas.api.extensions.ExtensionArray.unique.html#pandas.api.extensions.ExtensionArray.unique>

* Reference (lists)

<https://pandas.pydata.org/docs/reference/api/pandas.api.extensions.ExtensionArray.tolist.html#pandas.api.extensions.ExtensionArray.tolist>

1. Creation of f string to depict category and sub-category (will use a separate cell to ensure depicted values are correct)
   * <https://www.geeksforgeeks.org/formatted-string-literals-f-strings-python/?ref=header_outind>
2. Using LEN function to determine number of distinct variables within category
   * <https://pandas.pydata.org/docs/reference/api/pandas.Series.str.len.html#pandas.Series.str.len>
3. Created a new list to append the word((cat) short for category) to all ids within previous array. Usage of list comprehension.
   * <https://pandas.pydata.org/docs/user_guide/indexing.html>
4. DataFrame creation for category and sub-category
   * <https://www.geeksforgeeks.org/different-ways-to-create-pandas-dataframe/?ref=header_outind>
5. Writing the dataframes into their own csv files respectively
   * <https://www.geeksforgeeks.org/export-pandas-dataframe-to-a-csv-file/?ref=header_outind>

# Campaign DataFrame

1. Creation of dataframe copy using .copy() function
2. Renaming the columns within the new(copied) dataframe
   * The changes should be made accordingly with the projected answer (found in file before any changes made
   * <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rename.html#pandas.DataFrame.rename>
3. Conversion of data types from columns goal and pledged
   * <https://www.geeksforgeeks.org/change-data-type-for-one-or-more-columns-in-pandas-dataframe/?ref=header_outind>
4. Checking the data types
   * <https://www.geeksforgeeks.org/get-the-datatypes-of-columns-of-a-pandas-dataframe/?ref=header_outind>
5. Formatting / changing the columns format.
   * The units are set in seconds
   * The units are of Unix timestamp format and should be essentially converted
   * <https://www.geeksforgeeks.org/convert-datetime-string-to-yyyy-mm-dd-hhmmss-format-in-python/?ref=header_outind>
   * Several issues faced when determining timestanmp values for these columns and adjust accordingly. Use forward slash to indicate continuation on next line
6. Merging the two data frames created via category and sub-category
   * <https://www.geeksforgeeks.org/joining-two-pandas-dataframes-using-merge/?ref=header_outind>
7. Dropping the unwanted columns within the new dataframe (can be further discussed but I went through each of the columns and have determined that the following can be dropped: staff\_pick, spotlight, category, sub-category. This is due to the fact that both the category and sub have their own dataframes respectively. This was not the case as the initial solution projects differently. After comparison I found that the columns to drop were staff\_pick, spotlight, category & sub-category, category, sub-category. I dropped these
8. Saved the new (cleaned dataframe) to a csv file and pushed work to git (6th push)