Standard Operating Procedure for Assignment_1_Jacob

Purpose:

The purpose of this SOP is to enable a user to easily use the attached template to reformat their data to graph it. It also gives details on the program that reformats the data, enabling the user to potentially edit it to better suit their needs.

Introduction:

This SOP follows a standard, unspecified format to describe the background of the program as well as how to use and edit it. It also includes instructions on how to use the data template. This template only covers the basic function of the program and its use, as well as some methods to edit this program.

Procedure:

1. Save data as a .csv in the same format as the template attached
    a. See category below detailing instructions for formatting
2. Open program in applicable program, such as Google Colaboratory or Jupiter Notebooks.
3. Edit the program to meet any specific qualities or aspects wanted
4. Run “Import Line” and “Definition of the Colour Function”
    a. Prepares rest of the program to run
5. Run “Access of .csv File”
    a. Input the path to the file or the file name when prompted
    b. Input the number of subdivisions per overarching category
6. Run “Creating Graphing Specification File”
    a. Name file by following instructions
    b. Will generate a .txt file
7. Run “Definition of Data Formatting”
8. Run “Formatting Dara for Histogram”
    a. Will generate a .txt file for each column of data in .csv file
9. Download, save, or move the .txt files to be used for graphing in the future

Specifications:

Standard Operating Procedure for Assignment_1_Jacob

The program runs on Python 3.6.9 and was designed using Google Colaboratory. A .csv file in the format of the template is required to enter data into the program. The program returns two or more .txt files based on how many columns of data are entered.

Template:

The first column contains the labeling of each data point. CatA matches the overarching category, while CatB matches the subdivisions. CatA can be made of characters, numbers, spaces, etc. while CatB must be numerical and in a two-digit format where a preceding 0 must be included for divisions 1-9. It must be in the format CatA-CatB. The heading may be whatever the user wishes.

All other columns to the right are for data, with the header being any kind of text that the user wishes while all data must be numerical.

An example, where the overarching category starts at 1 and has 4 subdivisions with two points of data would be:

1-01	0	-0.331
1-04	-213	0.556
2-02	34	-1.993
2-03	4132	0

Should the user wish there to be blank overarching categories preceding or succeeding the range of data, each value for these overarching categories must be included in the template either prior to or after the data. The subdivision should be specified as the first. Zeros, or the base value for those specific data types, should be inputted for the columns to the left. A sample line with two points of data would be:

OverArching-01	0	0

Editing:

- Should the user want to add in more colours:
    - The user must first change the number 14 to the number of total colours being included.

  - They can then add the following lines of code prior to current line 30 in “Definition of the Colour Function” with the proper indentations:

```
elif numb%X == N:
        return “colour”
```

  - *X* stands for the new total number
  - *N* stands for one more than the previous colour
    - Ex: The next colour added would have *N* equal 14
  - *colour* is the name of the colour that the user wants added

- Should the user want to adjust the formatting of the Graphing Specification File
  - They can add, change, or remove values on line 8, but must keep the term *\t* after each value in the line
  - The final value, instead of being succeeded by *\t* must be succeeded by *\n*
- Should there be greater than 99 subdivisions
  - The user must then use three digits when enumerating subdivisions. Two preceding zeros must be included for divisions 1-9 and one preceding zero must be included for divisions 10-99.
  - The following lines in “Definition of Data Formatting” must also be changed
    - Lines 9-10, to be replaced by:

```
if i<9:
        coldata[str(rawaxes[keys])+"-00"+str(i+1)] = 0
elif i<99:
        coldata[str(rawaxes[keys])+"-0"+str(i+1)] = 0
```

    - Lines 21-24, to be replaced by:

```
if i<9:
        histtxt.write(str(keys) +"\t"+ str(i*1000000) +"\t"+
        str(i*1000000+999999) +"\t"+
        str(coldata[str(rawaxes[keys])+"-00"+str(i+1)]) +"\n")
elif i<99:
        histtxt.write(str(keys) +"\t"+ str(i*1000000) +"\t"+
        str(i*1000000+999999) +"\t"+
        str(coldata[str(rawaxes[keys])+"-0"+str(i+1)]) +"\n")
```

History:

The program, template, and SOP were created by Jacob Fuhr on January 12th 2021. Any updates, addendums, or corrections to the program, template, or SOP will be listed below with the date and individual who made the update, addendum, or correction.