Software Spend Reporter

Author: Mandeep Singh
Contact: msingh98@uw.edu
Developed for: ServiceNow

This project attempts to make sense of software spend data of various companies. The required information to produce this report comes from a CSV file, that must contain the columns of 'Vendor', 'Product' and 'Amount' in no specific order.

The sample output is a two-level tree that shows data in a two-level tree format (as in *fig1*) for the CSV file in *fig2*:

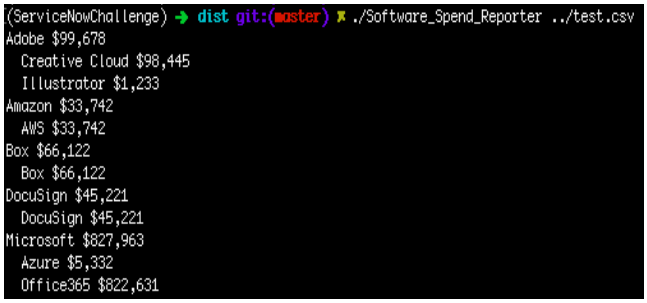

Fig 1.

Transaction Date	Vendor	Product	Amount
1/28/19	Microsoft	Office365	432854
1/27/19	Adobe	Creative Cloud	98445
1/24/19	Amazon	AWS	12443
1/19/19	Microsoft	Azure	5332
1/11/19	Adobe	Illustrator	1233
12/24/18	Amazon	AWS	11977
12/5/18	Box	Box	66122
11/24/18	Amazon	AWS	9322
11/3/18	DocuSign	DocuSign	45221
1/28/18	Microsoft	Office365	389777

Fig 2.

Quick Installation:

The executable was made on a Mac OSx and will run on similar Operating Systems.

Once the repository has been cloned, the "dict" folder contains a standalone executable (*Software_Spend_Reporter*) file that can be used to get the required information. The executable will require a valid path to a CSV file to be passed as a parameter.

```
./Software_Spend_Reporter  /path/to/file
```

Note: The executable is a standalone and can be placed anywhere in your file system.

If the above command results in an error, try to add executable permissions to the file by typing the following:

```
Chmod +x Software_Spend_Reporter
```

Development:

The code for the executable was written in Python 3.6. The additional libraries used were: *pandas, os* and *sys.* I then created a standalone executable using *pyInstaller* which created the various directories in the root folder and the *Software_Spend_Reporter.spec* file.

The details of the python environment (made using anaconda) for the development of the code can be found in the *environment.yaml* file in the root folder of the project. If you would like to setup the environment on your local machine, you can build the environment using:

```
conda env create -f environment.yml
```

To activate the environment type:

```
source activate ServiceNowChallenge
```

You should now be able to use the python script.

Further Possible Improvements:

I chose to use pandas as a framework to read CSV files because further aggregations on the CSV files are easier with pandas. For example: we can pass additional parameters that can define a time period for which the total spend report is to be calculated.

Further aggregations on the file can be made possible if we choose to create a properties file to define various parameters and use the path of the properties file as a parameter instead of the path to a CSV file.