Alexander Looi
Data Science II
Feb. 22, 2017

This “csvome” takes in a single csv file and computes basic summary of the csv. For all csv’s inputted summary calculated include:

1. Printing of the header
2. Prints the number of rows and columns
3. It also prints the number of cells per row. A lot of my work is with international collaborators that like to use commas as decimal points on occasion. Therefore, it is important to check these csv files for row number consistency.
4. If there are inconsistent row lengths, then it will tell the user and print the row number of where the irregularity is. It should be noted that these are locations where the row length changes, and is not necessarily where the problem is.
5. This code also considers the header as well in its row cell comparisons. It checks to see if the header has the same number of rows as the first row of data.

These general attributes are useful for me because of my international collaborators. I’ve obtained several .xlsx files that use commas as decimal holders. My .xlsx to .csv converter at first did not catch this problem. So, it may be nice to pipe data files from the .xlsx to .csv converter to this script to check for mistakes or inconsistencies.

The second thing this code can do is allow the user to grab specific rows or columns from the data set. The user can grab any number of rows, or none. The entire row is printed as well as the row length. In addition, if the row number referenced is not available then an error is given. The script will notify user if the subscript requested is out of bounds of the list. Grabbing specific rows can be used in figuring out which rows may have more than others, and where potential problems may arise in the data set. For example, if a header cell is split because of a comma, then printing just that row will allow the user to potentially see the split.

Specific columns can be selected as well. The user can specify a column name or a column subscript. If a column does not exist or a number is out of bounds then the user is notified. However, the script will still return those columns that are within the subscript. Here, specific stats about each column are printed, mainly, the number of different data types within each column (excluding the header). Instead of printing the entire data set column this simply prints the counts of each data type condensing what could be thousands or millions of cells into several lines of information in the terminal. This is a quick way of seeing the different kinds of data types in a column. If the user is relatively familiar with their data set, this quick, easy, and intuitive summary will let them know if there are unexpected data types in a column.

Lastly, the PhytoAnnecy_Feb2017_Biomass_ug_l.csv data set is an example where commas are manually entered in by the data provider. As a result the resulting csv has varying numbers of rows and numbers become split up. This csvome can detect some of those instances by tracking when row length changes.

Examples:

```
python csvome.py RawWeatherData_Champlain_ColchesterReef.csv –c 2 6 0 1
'lake' 'lakes'
```

The standard output

```
Rows to be printed:  None
Columns to be printed:  ['2', '6', '0', '1', 'lake', 'lakes']
Opened file: RawWeatherData_Champlain_ColchesterReef.csv
printing header:
['lake', 'met_stationid', 'year', 'wind_height_m', 'date_time',
'air_press_mb', 'air_temp_c', 'rain_mm', 'rel_hum_perc',
'radiation_j_sec_m2', 'max_wind_m_s', 'avg_wind_m_s', 'wind_dir_deg',
'wind_0_m_s', 'wind_45_m_s', 'wind_90_m_s', 'wind_135_m_s',
'wind_180_m_s', 'wind_225_m_s', 'wind_270_m_s', 'wind_315_m_s\n']

number of columns:  21
number of rows:  409094
max number of cells in a row:  21
min number of cells in a row:  21
```

Here “lakes” is not a header but lake is. The script will print:

```
lakes is not a header
```

But will still continue printing the other columns (columns 2, 6, 0, 1, and “lake”) in that order

Here is the summary output of the lakes column.

```
some characteristics of the lake column
There are 0 Integers
There are 0 Floats
There are 0 cells with numerics
There are 0 NA cells
There are 283039 empty cells
There are 126055 Other Data types (strings)
```

Here is sample output summary of the column called air_temp_c

```
some characteristics of the air_temp_c column
There are 934 Integers
There are 127148 Floats
There are 128082 cells with numerics
There are 0 NA cells
There are 281012 empty cells
There are 0 Other Data types (strings)
```

The large number empty cells indicated in the output is a red flag for this data set. Maybe someone deleted rows in excel, but they still exist in csv form.