UWA Data Analysis

Provide a Python file to perform basic cleansing, statistical analysis and error detection on data provided via a csv file.

This document will give step-by-step instructions on how to set up this program to run on your local machines and teaches you how to use it. It also documents what data types and type of analysis and error detection it can do.

Getting Started

1. Install the Anaconda package for your operating system for Python 3 and follow the installation instructions.
2. If you haven't navigated using the terminal/command line before we recommend reading this tutorial

Mac/Ubuntu

Open the terminal, navigate to the directory containing the application.py file and run using

```
python3 application.py csv_filename_here
```

Where *csv_filename_here* is replaced with the path to the file you want to run. For example: to run the SampleFile.csv file found in csv_files in the program folder enter the following:

```
python3 application.py csv_files/SampleFile.csv
```

If you don't want to list out the path to the file through typing you can drag and drop the file onto the terminal to paste its path in.

Windows

Open windows powershell, navigate to the directory containing the application.py file and run using

```
python application.py csv_filename_here
```

Where *csv_filename_here* is replaced with the path to the file you want to run. For example: to run the SampleFile.csv file found in csv_files in the program folder enter the following:

```
python application.py csv_files/SampleFile.csv
```

If you don't want to list out the path to the file through typing you can drag and drop the file onto the terminal to paste its path in.

Usage

You can specify multiple files using:

> python application.py *csv_fileame1 csv_filename2*

The reports generated are html files which can be opening with any internet browser and are saved in the same location as the original csv file.

You can use templates using the -t flag see the documentation for more information on templates:

> python application.py *csv_filaname* -t *template_name*

You can specify entire directories (sub directories are not recursed, only csv files) by specifying a directory in the program directory running:

> python application.py csv_files\

Files can either be csv files or excel files. We recommend saving excel spreadsheets as csv files using the save as function in excel as there are errors in used modules that prevent some excel files being run. If using excel files the program will create a csv file for each sheet in your excel file. Each sheet is analysed independently and a new report is generated for each. All these are saved in a new directory located in the same locations as the original excel file.

If multiple files are given with only one template all files will be processed using the template. The same will occur given a excel file with multiple sheets and a single template. For using multiple templates with multiple files there must be an equal number of files and templates.

You must run the program from the directory containing the application.py file. csv_filenames can be specified by relative path or using its absolute path

'Show Data' and the 'Back' links are currently not supported in the offline version. These are available through the online version at http://uwa.engineering/ in the Code Tools section.

Large files

For files larger than 300Mb we recommend splitting your data using a Csv spliiter. We recommend using one by Sopheap Ly from the fxfisherman forums, download here

Contributors

- Liam Jones
- Alastair Chin
- Jordan Hedges
- Kieran Richards
- Leighton Lilford
- Jan Villanueva
- Alastair Mory
- Csv Splitter by scorpion:

Functionality

Supported Data types

- Int
- Float
- Enumerated
- String
- Email
- Currency
- Boolean
- Scientific notation
- Identifier
- Date
- Day
- Time
- Hyperlink
- Character

Cleansing

- Inconsistent row lengths
- Inconsistent types within columns
- Blank values

Desired analysis

- Numerical
  - Minimum
  - Maximum
  - Mean
  - First Quartile
  - Median
  - Third Quartile

    - Standard deviation
    - Mode
    - Distribution type
    - Outliers
- Top 5 occurring results
- Bottom 5 occuring results
- Number of unique entries

Directory Structure

- The source code is all in the main directory
- csv_files contains test files used to evaluate the program
- Sphinx contains documentation of classes and methods of the program
- templates folder contains templates used with the test files in csv_files
- Detailed documentation is a pdf generated by the Sphinx Code