Data Wrangling

Flow and Utilization

In this document, we will present the workflow for a user of the Data Wrangling (DW) application. Basically, the goal is to load tabulated data into a spreadsheet-like GUI, apply transformation on these data, visualize statistics about these data, and save the *wrangled* data. Figure 1 presents an overview of the workflow. The user will go through an iterative process of "wrangling" until a point is reached where the user is satisfied with all the transformations applied. The goal is to transform the data to make it more amenable to machine learning (ML).

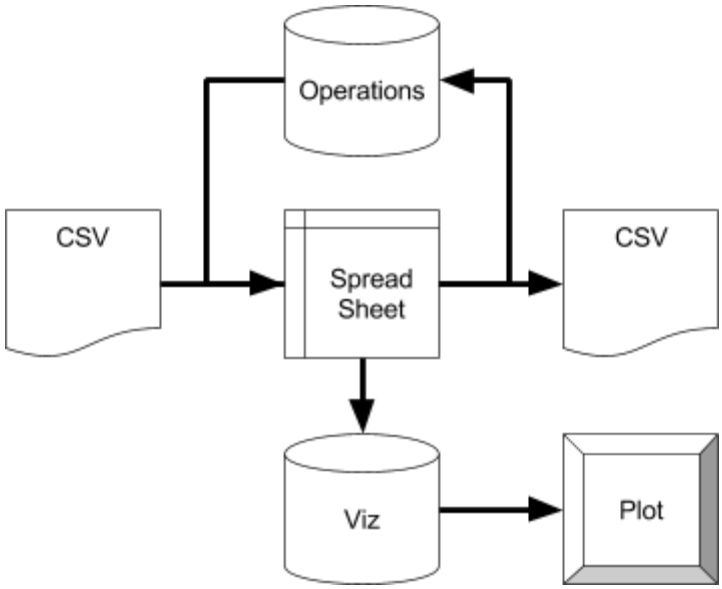

Figure 1: Workflow of the data wrangling application. The user loads the tabular data (comma separated values (CSV)). This data can potentially be displayed in a spreadsheet-like UI or a fixed set of transformations can be applied and the transformed data can then be saved to a file.

Import Data

The first step when *wrangling* data is to import the data into a UI. The project will start by considering CSV data. As the project moves forward, we might want to add some preprocessing modules that will generate tabulated data from unstructured data.

Spreadsheet-like GUI

The user interface will present the data in a spreadsheet-like format. This interface should offer the possibility of selecting any number of (non-contiguous) rows or columns. The default is to apply transformation to all rows based on the columns that are chosen. The tool should also offer the possibility to sort or filter rows using functions of multiple cells. Imported values in this spreadsheet will be read-only. However, any number of row/column can be added to the spreadsheet as result of the operations.

Operations

Many different operation can be applied to the data. These operation can apply to one column or a selection of columns. Some example of operations:

- One column:
  - **normalize**:
    - apply selected normalization to the column
    - store results in new columns
  - **values split**: for each *X* in column C
    - create a new column *isX*
    - $isX_i = (C_i == X)$

- Multiple columns:
  - **normalize**:
    - normalize each row for selected group of columns
    - store result in new columns
  - **PCA:**
    - Apply PCA to selected columns
    - creates N new columns where N is the number of principal components
  - **Decorrelation:**
    - Compute the decorrelated version of the selected columns

Visualization

To wrangle data properly, it is necessary to inspect these data, the visualization module will offer this possibility. Given a selection on the spreadsheet, the visualization module will provide a graphical representation of the selected data.
Some example of visualization:

- Columns:
  - **distribution**: display the distribution of each column (discrete or continuous)
  - **correlations:** display correlation in a graphical way
- Rows & Columns:
  - **histogram:** display histograms for selected rows and columns

NB1: The visualization module can require to perform operations on the data, the results of these operations could be saved in 'hidden' row/column of the spreadsheet...

Export Data

The last part of wrangling data is to save the result in a CSV. The user need to be able to select any columns and filter rows. The output format should be CSV at the start. Investigating other formats (numpy, HDF5, …) could be an interesting direction.

NB2: It might be interesting to provide a CLI and a scripting language for the data wrangler. It would enable to automate the wrangling of multiple dataset.
NB3: It might be interesting to look at multiple datasets simultaneously…
NB4: exporting ML ready datasets might be useful (separating features and target)