Unit testing of the display\_csv\_as\_table function

The objective of the display\_csv\_as\_table function is to print every row in a csv-file in the terminal. It identifies the first row as the header.

The display\_csv\_as\_table function returns nothing but takes a string input as the name of the csv file. Thus out of {int,float,string,list}, int, float and list are invalid types. We create one test for each of these and assert that the display\_csv\_as\_table function is runnable.

Since we have no documentation we do not know the valid inputs. The input domain was therefore explored divided into the following equivalence classes (EC):

1. A non existing csv file name
2. An empty csv file
3. An empty string as csv file name
4. A csv file containing only 1 column
5. A csv file containing an empty row somewhere
6. A csv file containing different types
7. A csv file containing 4 columns
8. A csv file containing varying column amounts
9. A csv file containing 3 columns (Product, Price, Units) as per the products.csv file

We will create a test case for each of these equivalence classes. To see the different test cases, their expected outcomes and the results, see the document “display\_csv\_as\_table function testing document”.

In order to create different test cases we need to create different csv files that can be used to test the equivalence classes.