Unit testing of the display\_filtered\_table function

The objective of the display\_filtered\_table function is to identify a “Product” column in the first row (header) of a csv file and then print out the rows containing a certain string provided as a parameter in the “Product”-column.

The display\_filtered\_table function returns nothing but takes two strings as input containing the csv-file name and a search phrase to filter by. Thus out of {int,float,string,list}, int, float and list are invalid. We do not create any tests for these since we did that for the function display\_csv\_as\_table.

Since we have no documentation, we do not know the valid inputs. The input domain was therefore explored and then 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. An empty string as search input
5. A csv file containing only 1 column
6. A csv file containing an empty row somewhere
7. A csv file containing different types
8. A csv file containing 4 columns
9. A csv file containing varying column amounts
10. A csv file not containing a 'Product' column
11. A csv file with 'Product' as the second column
12. A csv file containing varying column amounts with 'Product' as the second column
13. A non-existing product as search input
14. A csv file containing 3 columns (Product, Price, Units) as per the products.csv file and a product that exist within the csv as search input

We will create a test case for each of these equivalence classes. For the final equivalence class (EC14) we will test three different search inputs. To see the different test cases, their expected outcomes and the results, see the document “display\_filtered\_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.