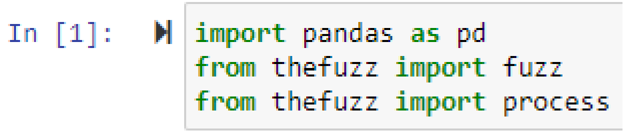
Fuzzy Matching Exercise

To get started with Python, we first need to install a Python shell. I won't go over installation details, but I recommend the Jupyter Notebook for beginners because of its friendly UI. After installation, the step-by-step process for running fuzzy matching is as follows.

1. Import required libraries:

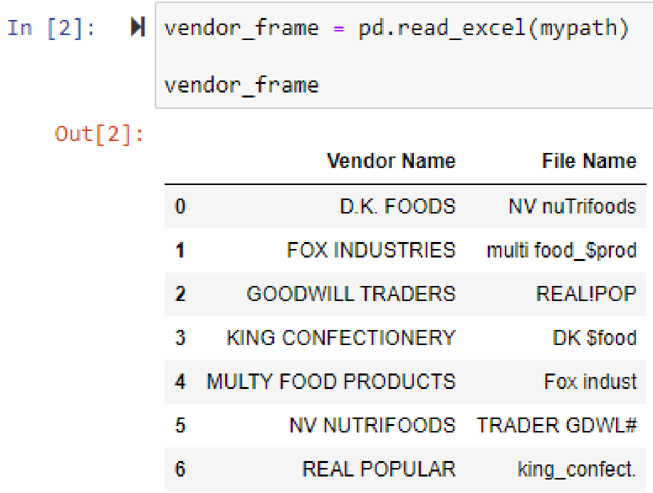
   I import the two essential libraries to complete this task: *pandas* and *thefuzz*. The first, pandas, is a data analysis library that is one of the most frequently used to read and write Excel tables. It is considered the "backbone" of business coding. The second, thefuzz, is the library we will use to perform fuzzy matching. Different approaches are suited for different fuzzy matching tasks and this library can help with most. I import pandas under the alias *pd* and two sub-modules from thefuzz, *process* and *fuzz* respectively. Refer to the detailed documentation (link above) for both libraries for more information.



```
In [1]:   import pandas as pd
          from thefuzz import fuzz
          from thefuzz import process
```

2. Import Excel file in Python:

   I use pandas to import the Excel file. I create a variable called *vendor_frame* and use the *read_excel* function containing the Excel file's path. I then call this variable to visualize my table.



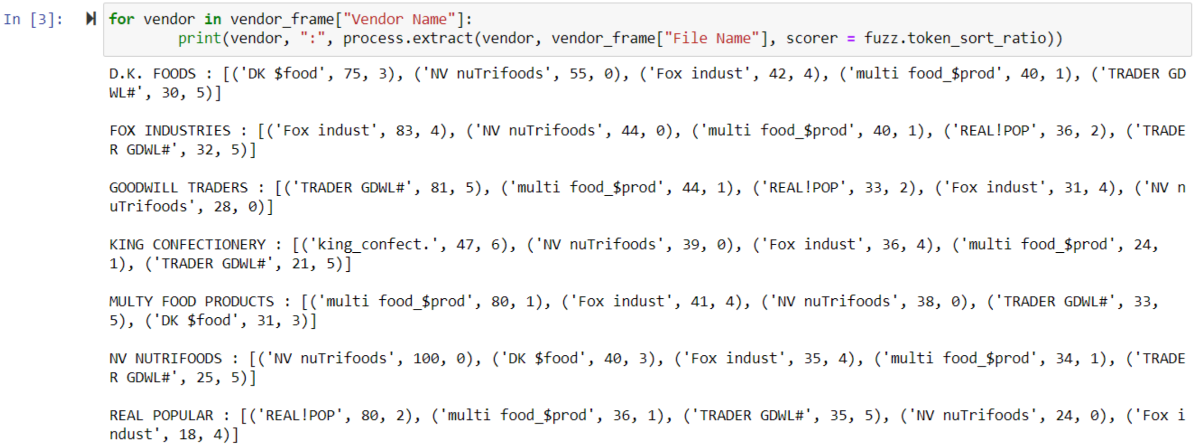
In [2]:
```
vendor_frame = pd.read_excel(mypath)

vendor_frame
```

Out[2]:

	Vendor Name	File Name
0	D.K. FOODS	NV nuTrifoods
1	FOX INDUSTRIES	multi food_$prod
2	GOODWILL TRADERS	REAL!POP
3	KING CONFECTIONERY	DK $food
4	MULTY FOOD PRODUCTS	Fox indust
5	NV NUTRIFOODS	TRADER GDWL#
6	REAL POPULAR	king_confect.

3. Run fuzzy matching:

   I run fuzzy matching on the data. I accomplish this by using a *for* loop in Python. For each vendor name, I run the LDFM algorithm with each file name. Since the file names are not always in the correct order (e.g., TRADER GDWL#), I use *token_sort_ratio* as the scorer. This approach first sorts the words in both strings and then compares them using LDFM. The *process.extract* function matches each vendor to each file name and returns scores sorted in descending order. I then print out each vendor name along with its respective fuzzy scores with all file names.



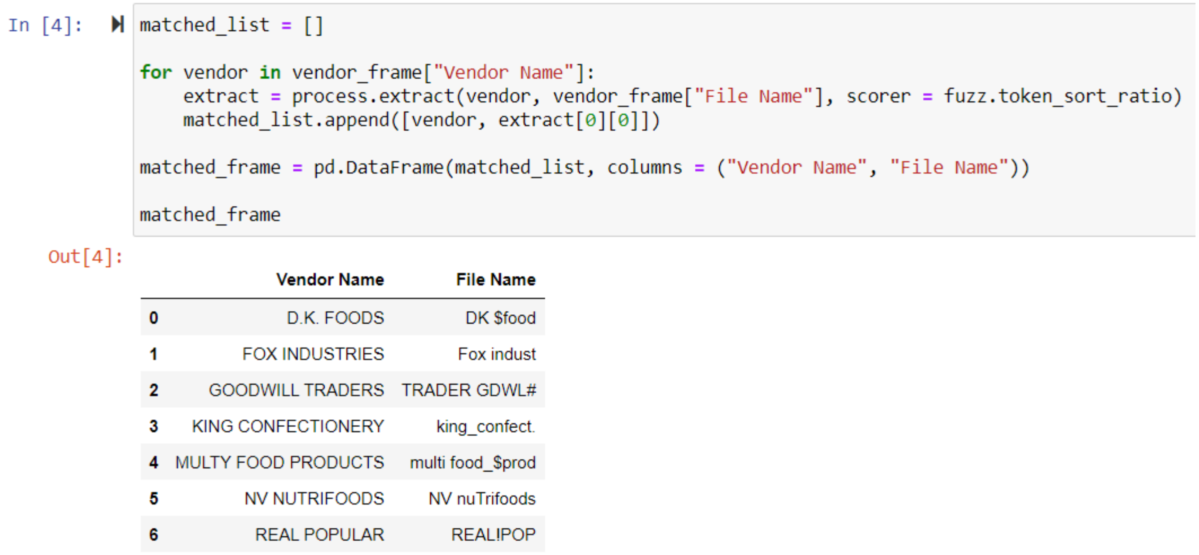
In [3]:
```
for vendor in vendor_frame["Vendor Name"]:
        print(vendor, ":", process.extract(vendor, vendor_frame["File Name"], scorer = fuzz.token_sort_ratio))
```

```
D.K. FOODS : [('DK $food', 75, 3), ('NV nuTrifoods', 55, 0), ('Fox indust', 42, 4), ('multi food_$prod', 40, 1), ('TRADER GD
WL#', 30, 5)]

FOX INDUSTRIES : [('Fox indust', 83, 4), ('NV nuTrifoods', 44, 0), ('multi food_$prod', 40, 1), ('REAL!POP', 36, 2), ('TRADE
R GDWL#', 32, 5)]

GOODWILL TRADERS : [('TRADER GDWL#', 81, 5), ('multi food_$prod', 44, 1), ('REAL!POP', 33, 2), ('Fox indust', 31, 4), ('NV n
uTrifoods', 28, 0)]

KING CONFECTIONERY : [('king_confect.', 47, 6), ('NV nuTrifoods', 39, 0), ('Fox indust', 36, 4), ('multi food_$prod', 24,
1), ('TRADER GDWL#', 21, 5)]

MULTY FOOD PRODUCTS : [('multi food_$prod', 80, 1), ('Fox indust', 41, 4), ('NV nuTrifoods', 38, 0), ('TRADER GDWL#', 33,
5), ('DK $food', 31, 3)]

NV NUTRIFOODS : [('NV nuTrifoods', 100, 0), ('DK $food', 40, 3), ('Fox indust', 35, 4), ('multi food_$prod', 34, 1), ('TRADE
R GDWL#', 25, 5)]

REAL POPULAR : [('REAL!POP', 80, 2), ('multi food_$prod', 36, 1), ('TRADER GDWL#', 35, 5), ('NV nuTrifoods', 24, 0), ('Fox i
ndust', 18, 4)]
```

4. Create new table with matched data:

I create a new Python list called *matched_list* and append to it a pair for each vendor and their best matched file name. I then create a new pandas table called *matched_frame* from the list and visualize it.

As seen below, each vendor is now matched with their respective file. It is also possible to now rename each file to its vendor’s name, but I will not delve into that code here.



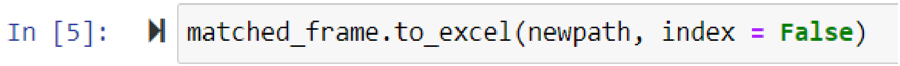
```
In [4]:  matched_list = []

         for vendor in vendor_frame["Vendor Name"]:
             extract = process.extract(vendor, vendor_frame["File Name"], scorer = fuzz.token_sort_ratio)
             matched_list.append([vendor, extract[0][0]])

         matched_frame = pd.DataFrame(matched_list, columns = ("Vendor Name", "File Name"))

         matched_frame
```

Out[4]:

	Vendor Name	File Name
0	D.K. FOODS	DK $food
1	FOX INDUSTRIES	Fox indust
2	GOODWILL TRADERS	TRADER GDWL#
3	KING CONFECTIONERY	king_confect.
4	MULTY FOOD PRODUCTS	multi food_$prod
5	NV NUTRIFOODS	NV nuTrifoods
6	REAL POPULAR	REAL!POP

5. Export pandas table as an Excel file:

   Finally, I export the new table as an Excel file. I exclude the index (0 – 9 in the table above) from my export.



```
In [5]:  matched_frame.to_excel(newpath, index = False)
```