UPLUS EDUCATION
优加教育
Address: Lv1/868 Shepperton Rd, East Vic Park

Year 8

Science

Term 3 Week 7

Name: _____

Date: __/__/_____

Data Science Across the Disciplines

In today's lesson, we will focus on a large task to practise programming and learn new ideas.

Data Cleaning

Today's analysts need to pull information from many places. But working with multiple sources and preparing data for analysis can be time consuming and difficult to implement using standard tools like Excel or Access. We will be using python to accomplish this task. Some goals for data cleaning involve:

- Access, cleanse, and join data in any format from your hard drive, data warehouses, social media, and more.

- Prepare data for reports, presentations, visualization, or export to feed downstream processes

Data Acquisition

1. Open the directory where your python script is being stored. Click on the file unit-patterns.txt to inspect the file. The file contains all the different study plans of some students at some university.

2. This is a tab-separated values (tsv) file with two fields.

3. The first field contains a list of units studied (separated by plus signs), the second (after the tab character) contains a number and a percentage.

Task 1: Reading The File

Q1. Use the built-in function open() and the text file IO method readline() to read in and print out the first 5 lines of the file.

Your output should start like this:

```
 CITS2224-1 + CITS2023-2 + CITS3221-2 + CITS3224-1 + CITS3225-1     1 (2.0%)



 CITS2023-2 + ENSC3221-2 + ENSC3224-1 + ENSC3225-1 + ENSC3227-2 + ENSC3210-1 + ENS
C3221-1 + MECH3023-2      1 (2.0%)



 CITS1021-2 + CITS2021-1 + CITS2023-2 + ENSC2224-1 + ENSC3211-2 + ENSC3213-2 + MAT
H2221-1 + STAT1524-1      1 (2.0%)

...
```

Q2. Read the file again, but this time, as you read all the lines from the file, save them in a list called pattern-strings.

Printing the first 5 lines of your list should give:

```
[' CITS2224-1 + CITS2023-2 + CITS3221-2 + CITS3224-1 + CITS3225-1 \t 1 (2.0%) \n',
 ' CITS2023-2 + ENSC3221-2 + ENSC3224-1 + ENSC3225-1 + ENSC3227-2 + ENSC3210-1 + E
NSC3221-1 + MECH3023-2 \t 1 (2.0%) \n', ' CITS1021-2 + CITS2021-1 + CITS2023-2 + E
NSC2224-1 + ENSC3211-2 + ENSC3213-2 + MATH2221-1 + STAT1524-1 \t 1 (2.0%) \n', ' C
ITS2023-2 + CITS3220-2 + CITS3223-2 + CITS3021-1 + CITS3025-1 + MATH3224-1 + MATH3
221-1 + STAT3264-1 + STAT3260-2 \t 1 (2.0%) \n', ' BIOC3221-1 + BIOC3224-1 + BIOC3
225-2 + BIOC3225-2 + CITS1023-2 + CITS2021-1 + CITS2023-2 + MATH1211-1 \t 1 (2.0%)
 \n']
```

Tip: Rather than using readline(), use an iterator on the file directly to get each line in turn.

Task 2: Data Cleaning

Q1. We are only interested in the patterns to the left of the tab.Using the string partition() method, modify your code so that pattern_strings contains only the unit patterns. Printing the first 5 lines of your list (print(pattern_strings[:5])) should now give:

```
[' CITS2224-1 + CITS2023-2 + CITS3221-2 + CITS3224-1 + CITS3225-1 ', ' CITS2023-2
+ ENSC3221-2 + ENSC3224-1 + ENSC3225-1 + ENSC3227-2 + ENSC3210-1 + ENSC3221-1 + ME
CH3023-2 ', ' CITS1021-2 + CITS2021-1 + CITS2023-2 + ENSC2224-1 + ENSC3211-2 + ENS
C3213-2 + MATH2221-1 + STAT1524-1 ', ' CITS2023-2 + CITS3220-2 + CITS3223-2 + CITS
3021-1 + CITS3025-1 + MATH3224-1 + MATH3221-1 + STAT3264-1 + STAT3260-2 ', ' BIOC3
221-1 + BIOC3224-1 + BIOC3225-2 + BIOC3225-2 + CITS1023-2 + CITS2021-1 + CITS2023-
2 + MATH1211-1 ']
```

Q2. How many students are in this dataset?

Q3. Rather than the units in each line being part of a string, it is more useful to have them in a list. This will enable us to iterate through the list later. Use the split method to split each line into a list of units. Store the result in a variable pattern_lists.

This should result in a list of lists. Your output should start like this:

```
[[' CITS2224-1 ', ' CITS2023-2 ', ' CITS3221-2 ', ' CITS3224-1 ', ' CITS3225-1 '],
 [' CITS2023-2 ', ' ENSC3221-2 ', ' ENSC3224-1 ', ' ENSC3225-1 ', ' ENSC3227-2 ',
' ENSC3210-1 ', ' ENSC3221-1 ', ' MECH3023-2 '], [' CITS1021-2 ', ' CITS2021-1 ',
' CITS2023-2 ', ' ENSC2224-1 ', ' ENSC3211-2 ', ' ENSC3213-2 ', ' MATH2221-1 ', '
STAT1524-1 '], [' CITS2023-2 ', ' CITS3220-2 ', ' CITS3223-2 ', ' CITS3021-1 ', '
CITS3025-1 ', ' MATH3224-1 ', ' MATH3221-1 ', ' STAT3264-1 ', ' STAT3260-2 '], ['
BIOC3221-1 ', ' BIOC3224-1 ', ' BIOC3225-2 ', ' BIOC3225-2 ', ' CITS1023-2 ', ' CI
TS2021-1 ', ' CITS2023-2 ', ' MATH1211-1 ']]
```

Q4. Finally, notice that we've been left with unnecessary whitespace around each unit name.Modify your code to strip off the extra whitespace. Your first 5 patterns should now look like this:

```
[['CITS2224-1', 'CITS2023-2', 'CITS3221-2', 'CITS3224-1', 'CITS3225-1'], ['CITS202
3-2', 'ENSC3221-2', 'ENSC3224-1', 'ENSC3225-1', 'ENSC3227-2', 'ENSC3210-1', 'ENSC3
221-1', 'MECH3023-2'], ['CITS1021-2', 'CITS2021-1', 'CITS2023-2', 'ENSC2224-1', 'E
NSC3211-2', 'ENSC3213-2', 'MATH2221-1', 'STAT1524-1'], ['CITS2023-2', 'CITS3220-2'
, 'CITS3223-2', 'CITS3021-1', 'CITS3025-1', 'MATH3224-1', 'MATH3221-1', 'STAT3264-
1', 'STAT3260-2'], ['BIOC3221-1', 'BIOC3224-1', 'BIOC3225-2', 'BIOC3225-2', 'CITS1
023-2', 'CITS2021-1', 'CITS2023-2', 'MATH1211-1']]
```

Task 3. Putting it all together

Congratulations, you have now 'wrangled' your data from its 'raw' format in the file, into a very useable form. To consolidate this section, turn your code into a function get_patterns(filename) that takes a filename, and returns a list of lists containing all the patterns in the file.

Q1. print(get_patterns(DATAFILE)[:5]) should now print the following (same as above)

```
[['CITS2224-1', 'CITS2023-2', 'CITS3221-2', 'CITS3224-1', 'CITS3225-1'], ['CITS202
3-2', 'ENSC3221-2', 'ENSC3224-1', 'ENSC3225-1', 'ENSC3227-2', 'ENSC3210-1', 'ENSC3
221-1', 'MECH3023-2'], ['CITS1021-2', 'CITS2021-1', 'CITS2023-2', 'ENSC2224-1', 'E
NSC3211-2', 'ENSC3213-2', 'MATH2221-1', 'STAT1524-1'], ['CITS2023-2', 'CITS3220-2'
, 'CITS3223-2', 'CITS3021-1', 'CITS3025-1', 'MATH3224-1', 'MATH3221-1', 'STAT3264-
1', 'STAT3260-2'], ['BIOC3221-1', 'BIOC3224-1', 'BIOC3225-2', 'BIOC3225-2', 'CITS1
023-2', 'CITS2021-1', 'CITS2023-2', 'MATH1211-1']]
```

Task 4. Data Inspection and Interpretation

Now let's have a closer look at the data. The file contains all the different study plans of random students. Let's start by working out how many different or unique plans there are. To make it easier to see what is going on, start by defining a better print function.

Q1. Write a function print_patterns (patterns, numlines) that prints the first numlines patterns each on a new line.

For example: print_patterns(get_patterns(DATA), 8)

```
['CITS2224-1', 'CITS2023-2', 'CITS3221-2', 'CITS3224-1', 'CITS3225-1']

['CITS2023-2', 'ENSC3221-2', 'ENSC3224-1', 'ENSC3225-1', 'ENSC3227-2', 'ENSC3210-1
', 'ENSC3221-1', 'MECH3023-2']

['CITS1021-2', 'CITS2021-1', 'CITS2023-2', 'ENSC2224-1', 'ENSC3211-2', 'ENSC3213-2
', 'MATH2221-1', 'STAT1524-1']

['CITS2023-2', 'CITS3220-2', 'CITS3223-2', 'CITS3021-1', 'CITS3025-1', 'MATH3224-1
', 'MATH3221-1', 'STAT3264-1', 'STAT3260-2']
```

```
['BIOC3221-1', 'BIOC3224-1', 'BIOC3225-2', 'BIOC3225-2', 'CITS1023-2', 'CITS2021-1
', 'CITS2023-2', 'MATH1211-1']

['CITS2021-1', 'CITS2023-2', 'GRMN2025-1', 'GRMN2020-2', 'MATH1214-1', 'MATH2521-2
', 'PHYS2221-1', 'PHYS2223-2']

['CITS1221-1', 'CITS1021-1', 'CITS1023-2', 'CITS2023-2', 'MGMT1135-1', 'MGMT1136-2
', 'STAT1024-1', 'STAT2023-2']

['CITS1221-2', 'CITS2023-2', 'CITS3223-2', 'CITS3021-1', 'CITS3025-1', 'CLAN3224-1
', 'STAT2021-1', 'STAT3260-2']
```

Q2. Write a function called sort_patterns(x) that takes a pattern list, and returns the list in which:

the units in each list (pattern) are sorted alpha-numerically

the lists (patterns) themselves are sorted alpha-numerically

The following test code:

```
print_patterns(get_patterns(DATA),3)

print()

print_patterns(sort_patterns(get_patterns(DATA)),3)
```

should produce the following output:

```
['CITS2224-1', 'CITS2023-2', 'CITS3221-2', 'CITS3224-1', 'CITS3225-1']

['CITS2023-2', 'ENSC3221-2', 'ENSC3224-1', 'ENSC3225-1', 'ENSC3227-2', 'ENSC3210-1
', 'ENSC3221-1', 'MECH3023-2']

['CITS1021-2', 'CITS2021-1', 'CITS2023-2', 'ENSC2224-1', 'ENSC3211-2', 'ENSC3213-2
', 'MATH2221-1', 'STAT1524-1']



['AACE1223-2AA', 'ACCT1121-1', 'CARS1223-2AA', 'CITS1021-1', 'CITS2023-2', 'ECON11
21-1', 'FINA1221-2', 'INDG1223-2AA', 'MGMT1135-2', 'MKTG1225-2', 'STAT1524-1']

['AACE1223-2AA', 'CARS1223-2AA', 'CITS1021-1', 'CITS1023-2', 'CITS2023-2', 'INDG12
23-2AA', 'LAWS1113-2', 'PSYC1121-1', 'PSYC1123-2', 'SCIE1121-1', 'STAT1024-1']

['AACE1223-2AA', 'CARS1223-2AA', 'CITS1021-2', 'CITS1023-2', 'COMM1223-2', 'INDG12
23-2AA', 'STAT1023-2']```
```

Q3. Write a function remove_duplicates (patterns) that returns a (sorted) list of patterns with any duplicate patterns removed.

Q4. How many unique study patterns are there in the class?

Task 5. Putting it all together

Q1. Write a function get_unique_patterns (patterns) that returns a pair (unique_patterns, num_patterns) where:

- unique_patterns contains a sorted list of sorted patterns, with any duplicate patterns removed (as above)

- num_patterns contains the number of unique patterns