

## CITS1401 Project 1 - Pseudocode

main() function:

- Prompt the user to input the name of the file containing the data
- Prompt the user to input the metric that is to be applied to the data. Either: min, mean, median, harmonic\_mean
- Prompt the user to choose an action that is to be performed on the data. The user can select either: list or correlation
- Check if the filename provided by user input exists in the current folder
- If it does not exist, print an error message to the user that the file doesn't exit. If the file does exist, continue executing the rest of the code

### Metric Selection

- If the user selects min, call the min() function
- Else if the user selects mean, call the mean() function
- Else if the user selects median, call the median() function
- Else if the user selects harmonic\_mean, call the harmonic\_mean() function
- Else the user has not selected one of the available metrics, print an error message and stop execution of the code

### Action

- If the user selects 'list', call the list\_countries() function
- Else if the user selects 'correlation' call the spearman\_rank\_correlation() function
- Else the user has not selected one of the available actions, print an error message and stop execution of the code
- Print an output, either a list of countries or the correlation coefficient, to the user depending on their chosen metric and action

generate\_lists() function:

- Check if the filename provided by user input exists in the current folder
- If it does not exist, print an error message to the user that the file doesn't exist. If the file does exist, open that file for reading
- Initialize empty lists for each of the columns of data
- After opening the file, create a variable to hold the data of that file, and then call the readlines() function
- Create a For Loop, which iterates over each line in the previously created opened file variable, starting at element position 1 (because element position 0 is the column header, which we don't need)
- Call the split function on each line, splitting at the comma (',') and assign it to a variable 'fields'
- Append to each of the lists with the values in that respective column. Eg, append the GDP\_per\_capita list with fields[2] – the third column
- Return each list for later use

normalize() function:

- For each list(column) apart from the first 2 (country, life\_ladder), calculate the smallest and largest values in each list using Python's min() and max() functions
- Initialise new lists for each column called '<column-header>\_normalised'
- Create a For Loop to be used on each list.
  - o For each element in the original list, append the new normalized list with ((<original list value> - minimum value) / (maximum value – minimum value))
- Return all of the new normalised lists

min() function:

- Create new variables to hold the normalized value lists returned from the normalize() function
- For each row (country), calculate the minimum value in the list of normalized values (excluding the first 2 columns) using the python min() function
- Return the minimum value

mean() function:

- Create new variables to hold the normalized value lists returned from the normalize() function
- For each row (country), calculate the mean by taking the sum of all values in the list of normalised values (excluding the first 2 columns) by using the sum() function, and divide that sum by the length of the list using the len() function
- Return the calculated mean

harmonic\_mean() function:

- Create new variables to hold the normalized value lists returned from the normalize() function
- For each row (country), use a For Loop to calculate the reciprocal of each normalized value in the list (excluding the first 2 columns)
- Calculate the average of those reciprocals:  $\text{sum}(\text{reciprocal values}) / \text{len}(\text{reciprocal values})$
- Calculate the Harmonic Mean by using the reciprocal of the previously calculated average:  $1/\text{Average} = \text{Harmonic Mean}$
- Return the calculated Harmonic Mean

median() function:

- Create new variables to hold the normalized value lists returned from the normalize() function
- For each row(country), sort the list of normalized values (excluding the first 2 columns)
- Find the middle 2 numbers of the list by using the length of the list divided by 2 minus 1 and plus 1 (these numbers are the index values)
- Then sum those 2 values from the list and divide by 2 to get the median
- Return the calculated median

list\_countries() function:

- Return a list of countries in the first column, followed by their corresponding computed happiness score according to the metric chosen by the user
- Sort the list in descending order using the python sort() function

spearman\_rank\_correlation() function:

- Order the values in the lists from greatest to smallest. The two lists are the score list that we computed using the specified metric and the life ladder list provided in the original data file
- Assign a rank to each score, rank 1 being the highest
- Find the difference between ranks in the two lists for each Country
- Calculate the square of each of those difference values
- Sum all of the squared values
- Use the following formula to calculate the Spearman Rank Correlation:  $1 - (6 * \text{sum of squared values}) / (\text{number of countries} (\text{number of countries squared} - 1))$
- Return the calculated Rank Correlation