

Celine Huang

INF1340 Programming for Data Science

Professor Maher Elshakankiri

18 October, 2024

## **List of Functions in Nutritional Analysis Program**

### **1. Function: input\_data (file)**

**Description:** This function reads a CSV file provided by the user and returns the contents as a pandas DataFrame. It is used to import data for analyzing or processing.

**Arguments:** The argument of the function is “file” (a string), and it represents the name of the CSV file.

**Returns:** A pandas DataFrame represented by *df* that contains the data from the input CSV file. Each column and row in the CSV is represented in the DataFrame.

### **2. Function: largest\_value (category)**

**Description:** This function finds the largest numeric value in a specified nutrient category from a CSV file and returns the value along with the corresponding food(s).

**Arguments:** The argument of the function is “category” (a string), and it represents the name of a nutrient category in a column whose largest value you would want to find. Examples of nutrient categories include “Calcium”, “Fat”, and “Protein.”

**Returns:** The largest value found in the specified nutrient category represented by *largest\_value*, which is a float. It also returns a list of food items from the “food” column that correspond to the largest nutrient value represented by *largest\_value\_row*.

### **3. Function: smallest\_value (category)**

**Description:** This function finds the smallest numeric value in a specified nutrient category from a CSV file and returns the value along with the corresponding food(s).

**Arguments:** The argument of the function is “category” (a string), and it represents the name of a nutrient category in a column whose smallest value you would want to find.

**Returns:** The smallest value found in the specified nutrient category represented by *smallest\_value*, which is a float. It also returns a list of food items from the “food” column that correspond to the smallest nutrient value represented by *smallest\_value\_row*.

#### 4. Function: `nutritient_ratios (category1, category2, food)`

**Description:** This function calculates the ratio between two nutrient categories for a specified food item from a CSV file. The specified pairs of nutrient categories to compare are specified and the reasoning behind using them is because they are considered key nutrient ratios to monitor one's health: Calcium and Magnesium, Potassium and Sodium, Zinc and Copper, and Iron and Copper.

**Arguments:** The first argument of the function is “category1” (a string) and it represents the first nutrient category (also the column name) used in the ratio calculation, serving as the numerator. The second argument is “category2” (a string) and it is the second nutrient category used in the ratio calculation, serving as the denominator. The third argument is “food” (a string), and it represents the food item (index value) for which the nutrient ratio is to be calculated.

**Returns:** The calculated ratio of the two nutrient values for the specified food item represented by *ratio* (a float).

#### 5. Function: `mode (category)`

**Description:** This function calculates the mode of a specified nutrient category from a CSV file. It keeps track of the occurrences of each value in the column and identifies the value that appears most frequently.

**Arguments:** The argument of the function is “category” (a string), and it represents the name of a nutrient category in a column for which the mode is to be calculated.

**Returns:** The mode of a specified nutrient category represented by *mode\_value* (a float).

#### 6. Function: `median (category)`

**Description:** This function calculates the median of a specified nutrient category from a CSV file. It reads the column data, sorts it, and then computes the median based on whether the total number of values is odd or even.

**Arguments:** The argument of the function is “category” (a string), and it represents the name of a nutrient category in a column for which the median is to be calculated.

**Returns:** The median of a specified nutrient category. If the number of values is odd, then the middle value is returned. If it is even, then the average of the two middle values is returned.

#### 7. Function: `mean (category)`

**Description:** This function calculates the mean(average) value of a specified nutrient category from a CSV file. It sums up the values in the column and divides it by the total number of values to find the mean.

**Arguments:** The argument of the function is “category” (a string) and it represents the name of a nutrient category in a column for which the mean is to be calculated.

**Returns:** The mean of a specified nutrient category represented by *mean* (a float) which is calculated as the total sum of the values divided by the number of values.

## 8. Function: total\_amount (category)

**Description:** This function calculates the total amount for a specified nutrient category from a CSV file. It sums up all the values in the column to provide the overall total for that nutrient.

**Arguments:** The argument for the function is “category” ( a string) and it represents the name of a nutrient category in a column for which the total amount is to be calculated.

**Returns:** The total amount of the specified nutrient category represented by *total* (a float), which is calculated as the sum of all its values.

## 9. Function: standard\_deviation (category)

**Description:** This function calculates the standard deviation of a specified nutrient category from a CSV file. It calculates the mean and then determines how dispersed the data is in relation to it.

**Arguments:** The argument for the function is “category” ( a string) and it represents the name of a nutrient category in a column for which the standard deviation is to be calculated.

**Returns:** The standard deviation of the specified nutrient category represented by *standard\_deviation* (a float). This indicates the amount of variation of the values within the category.

## 10. Function: output\_results (file)

**Description:** This function writes the results from a list to a specified output file. Each result is written to a new line, making it easier to read and organize the data.

**Arguments:** The argument for the function is “file” ( a string) and it represents the name of the file where the results will be saved. If the file already exists, then it will be overwritten.

**Returns:** This function does not return a value and instead writes data to a file.