

National University of Singapore  
School of Computing  
CS1010S: Programming Methodology  
Semester I, 2024/2025

**Tutorial 8**  
**Data Analysis**

Release date: 21<sup>st</sup> October 2024

**Due: 27<sup>th</sup> October 2024, 23:59**

## General Restrictions

- No importing other packages unless explicitly allowed to do so.

## Required Files

- crypto.csv
- tut08-template.py

## Introduction

This tutorial is adapted from Practical Exam AY 2017/2018 Semester 1. For your convenience, a template file has been provided.

You are provided with a data file crypto.csv for this question for testing, but your code should work correctly for *any* data file with the same format. You should also use the following code to read from csv.

```
import csv
def read_csv(csvfilename):
    """
    Reads a csv file and returns a list of list
    containing rows in the csv file and its entries.
    """
    rows = []
    with open(csvfilename) as csvfile:
        file_reader = csv.reader(csvfile)
        for row in file_reader:
            rows.append(row)
    return rows
```

However, take note that you should **only be using the data structures taught so far** (`tuple`, `list`, `dict`). You are still **not allowed** to use additional packages beyond those provided.

## Question

Your friend recently made a lot of money buying Bitcoins, only to lose it all during the recent China ICO panic. Being wary of the returns, you obtained a price history the top few cryptocurrencies to perform some analysis. The first line of the data file is a header which describes each column of data.

Year	Month	Day	Currency	Components...
------	-------	-----	----------	---------------

You should only assume that **the first four columns are fixed** and that the rows are unique. The remaining columns are the components, which are not fixed, i.e. different data files can have different components. Your code should take this into account.

Hint: The method `List.index(item)` returns the index of the first matching item in the list.

1. Implement the function `monthly_avg`, which takes as inputs a filename (str), currency (str), year (int) and component (str). It returns a dictionary where the keys are months, and the values are the monthly average of the given component of the given currency in the given year, rounded to 4 decimal places. You may assume that all the values in the requested component are floats.

Note, if there are months with no data for the given inputs, then the month is not included in the returned dictionary.

Hint: You can use the function `round(n, d)` to round  $n$  to  $d$  decimal places.

```
>>> monthly_avg('crypto.csv', 'ETH', 2017, 'Close')
{'Nov': 296.4443, 'Oct': 306.2474, 'Sep': 293.0473,
 'Aug': 301.6094, 'Jul': 224.1239, 'Jun': 313.7343,
 'May': 125.7494, 'Apr': 50.3367, 'Mar': 34.7916,
 'Feb': 12.3711, 'Jan': 10.2013}

>>> monthly_avg('crypto.csv', 'BTC', 2013, 'High')
{'Dec': 856.4419, 'Nov': 569.307, 'Oct': 161.9442,
 'Sep': 134.164, 'Aug': 116.0023, 'Jul': 93.869,
 'Jun': 111.3007, 'May': 123.949, 'Apr': 143.4667}
```

2. We are now interested in computing the gain for each component in a month. The gain is calculated by taking the highest value of the component in the month and dividing it by the lowest value in the month. Since the result should be displayed as a percentage gain, it should be subtracted by 1 then multiplied by 100.

For each month in a given year, we want to know which currency had the highest gain for a particular component amongst all the currencies.

Implement the function `highest_gain`, which takes as input a filename (str), a year (int), and a component (str), and returns a dictionary where the keys are the months and the values are a tuple of two elements: the currency and the gain (rounded to 2 decimal places).

You may assume that the values for the components are either integers or floats. Note that it is possible for some values to be missing, in which case it is denoted by a '-'. Such rows should be ignored.

```
>>> highest_gain('crypto.csv', 2017, "Volume")
{'Nov': ('XPR', 695.17), 'Oct': ('XPR', 3485.19),
 'Sep': ('LTC', 1792.14), 'Aug': ('XPR', 5524.07),
 'Jul': ('LTC', 1354.52), 'Jun': ('XPR', 1071.13),
 'May': ('ETH', 2302.35), 'Apr': ('XPR', 4537.85),
 'Mar': ('XPR', 7003.02), 'Feb': ('ETH', 1049.67),
 'Jan': ('XPR', 1975.93)}

>>> highest_gain('crypto.csv', 2013, "Market Cap")
{'Dec': ('XPR', 272.37), 'Nov': ('LTC', 1862.94),
 'Oct': ('BTC', 88.94), 'Sep': ('XPR', 171.23),
 'Aug': ('XPR', 109.9), 'Jul': ('BTC', 59.08),
 'Jun': ('LTC', 52.22), 'May': ('LTC', 48.27),
 'Apr': ('BTC', 7.15)}
```