SI 201: Discussion 6

Working with CSV files and Nested Structures

SCHOOL OF INFORMATION
UNIVERSITY OF MICHIGAN

CSV Format

- **CSV (comma separated values) files are a simple and lightweight way to store structured data**. They can be read by many different programs and are a common format for sharing datasets.
- A CSV file represents data as a **series of rows and columns**, much like an Excel spreadsheet or a table.

Movie Name	Director	Runtime (min)
Moonlight	Barry Jenkins	111
The Handmaiden	Park Chan-wook	168
Boyhood	Richard Linklater	165

Information in different formats

Movie Name	Director	Runtime (min)
Moonlight	Barry Jenkins	111
The Handmaiden	Park Chan-wook	168
Boyhood	Richard Linklater	165

…as a table

```
Movie Name,Director,Runtime (min)
Moonlight,Barry Jenkins,111
The Handmaiden,Park Chan-wook,168
Boyhood,Richard Linklater,165
```

…as a CSV

CSV Format & Example CSV data

First line of a CSV:

- This is called the **header row**
- Contains the name of each column
- Most CSVs have one, but it isn't required

Each line of a CSV:

- Represents a row
- Columns are separated from each other using commas (,)

```
month,date,sample,Harris result,Trump result
sept,19,1880 LV,0.51,0.45
sept,19,1880 LV,0.53,0.47
sept,17,810 LV,0.49,0.45
sept,17,820 RV,0.49,0.45
sept,17,1445 RV,0.49,0.45
sept,17,1000 LV,0.53,0.47
sept,16,1247 LV,0.50,0.46
sept,16,1306 LV,0.50,0.46
sept,16,1247 LV,0.51,0.49
sept,16,1306 LV,0.51,0.49
sept,16,1505 RV,0.50,0.45
```

Header

Note: the columns are only separated with commas.
There is no need for additional spaces between commas.

Nested Structures

How can we represent a table in python?

- We can take the CSV and turn it into a **nested dictionary**
- In each key-value pair, the **value** is another **dictionary**
- This maintains the hierarchy and order of the list
- Useful for the JSON file type and API responses (both of which we will explore later in the course)

Horse	Race 1 (s)	Race 2 (s)
Oguri Cap	65.4	71.2
Gold Ship	45.0	100.0

“outer key”

“inner key”

```
{‘Oguri Cap’:
        {‘Race 1 (s)’: 65.4,
         ‘Race 2 (s): 71.2},
 ‘Gold Ship’:
        {‘Race 1 (s)’: 45.0,
         ‘Race 2 (s)’: 100.0}
}
```

Discussion 6 Assignment

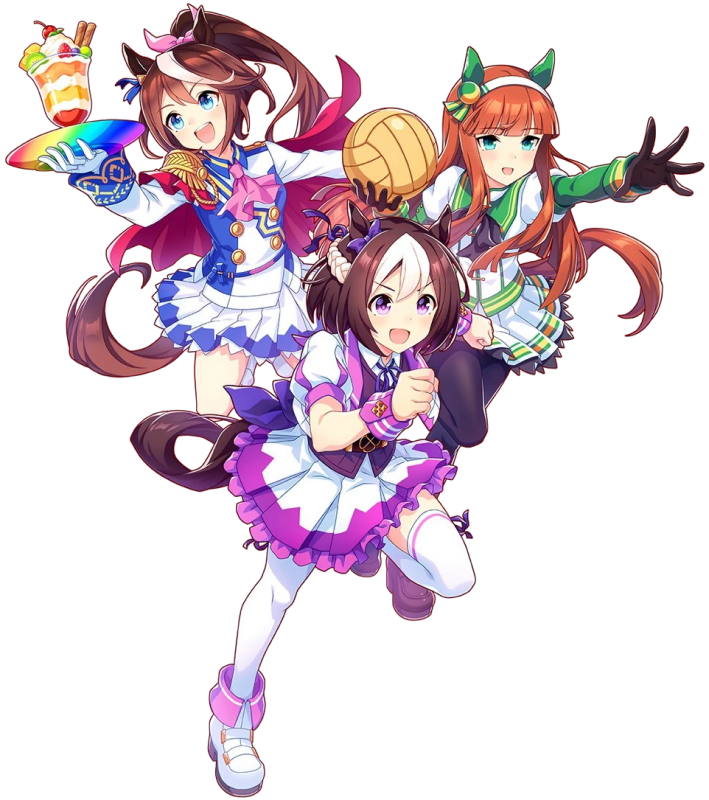
- Go to **Canvas Assignments > Discussion 6 Submission**
- Accept the GitHub Classroom assignment and clone the repo:
- **https://classroom.github.com/a/QzlqEWeE**
- **Commit at least 4 times and push to GitHub**
- **Submit the URL to your GitHub repository on Canvas**

Uma Musume CSV

This discussion you will be looking at horse racing data from a game called “Uma Musume.”

Your goal is to collect information to help the horses get better at racing.

The race data is has been given to you in a CSV file. You must open the file and process the data to help train the horses.



Tasks

TASK 1: Implement **load_results()**

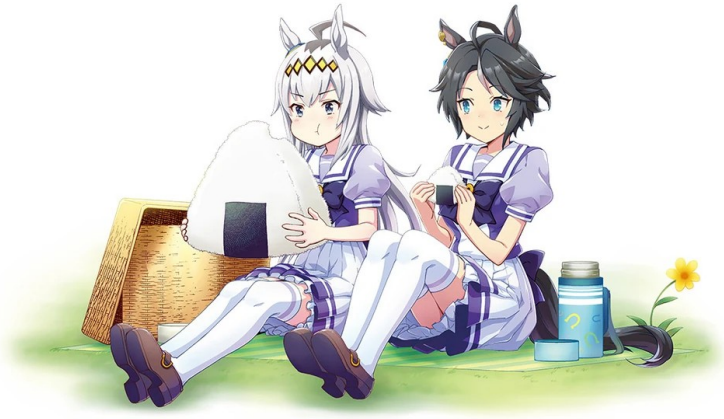
a. Takes in the processed CSV file as a list of lists

b. Return a **nested dictionary** of the horses, each with their time for each race

TASK 2: Implement **horse_fastest_race()**

c. Takes in the name of a horse

d. Return the **tuple** of the horse's fastest race and time



Tasks cont.

TASK 3: Implement **horse_personal_best()**

a. Returns a **dictionary of tuples** of each horse, with their fastest race and time

TASK 4: Implement **get_average_time()**

b. Calculate the average race time for each horse

c. Return a **dictionary** of each horse and their average time