SI206 Project 1
Fall 2023

Introduction:
In this project, you will work with Comma-Separated Value (CSV) files and nested dictionaries. CSV files are very common and these foundational skills will enable more complex data analysis.

On data science and social justice: Data scientists use data to create actionable insights. Social justice is a broad term for several movements based on furthering equality and ending socioeconomic oppression. We can use data science in the pursuit of social justice by uncovering inequity so that we can act to correct it.

The instructions, starter code, data, and data dictionary are included when you clone the GitHub repository. These files can also be found on the Canvas site under Files > Projects > Project 1. The first row in each file is a header that describes the data.

Assignment:
The year is 1976, and you have been hired as a data scientist by a law firm specializing in civil rights and discrimination cases. Your most recent assignment involves gathering data to support the claim in the DeGraffenreid v. General Motors case. In this lawsuit, five Black women have filed a lawsuit against General Motors, alleging that the company's layoff policy disproportionately targeted Black women.

General Motors (GM) asserts that they did not engage in discrimination against the plaintiffs. They argue that their seniority-based layoff policy, which prioritizes laying off the most recently hired employees first, did not discriminate against individuals identifying as Black or as women at GM. Furthermore, GM contends that rewarding longer tenure with greater job security is a fair practice.

The Black women, on the other hand, argue that GM's seniority-based layoff policy had a disproportionate impact on employees who specifically identified as both Black *and* female. They point out that due to discriminatory hiring practices, GM did not hire any Black women until after 1964. Consequently, when the early 1970s recession-driven layoffs occurred, the recently hired Black female employees were the first to be let go under the "last hired, first fired" seniority system. This resulted in all Black women at GM losing their jobs while more senior employees were protected.

In the discovery for the case, GM provided historical employment data to you in a CSV format. It is up to you to validate your clients claims using data. You will create the following five functions and four corresponding test functions, and run them in *main()* in order to load, store, and analyze this simulated data. For the purposes of testing and validation, a truncated version of the dataset has also been provided.

1. ***load_csv("filename")***

   load_csv takes one argument: a string that represents the name of a file. The function returns a dictionary in which each key is the employee id and each value is another dictionary. Each inner dictionary will use the demographic categories or year hired as keys and their corresponding data as values. Hire year should be converted into an integer before it is added to the dictionary.

   Example output:
   When run on the GM data it should produce a dictionary like this:

```
{
  'employee_1': {'gender': 'Female', 'race': 'White', 'hire_year': 1960},
  'employee_2': {'gender': 'Male', 'race': 'Black', 'hire_year': 1965},
  ...
}
```

   test_load_csv()
   tests ***load_csv***
   Write a test case that checks for the length of the outer dictionary.
   Write a test case that checks for the length of the inner dictionary.

2. ***split_by_hire_year(employees, split_year)***

   GM's layoff policy is called "last hired, first fired." This is a common method of choosing people to lay off at a company that persists to this day. Due to the 1970s recession, the CEO decided to use this policy to shrink the company. So, we need to determine who was hired prior to 1964, and who was hired after 1964.

   split_by_hire_year takes two arguments: a dictionary of dictionaries, and an integer representing a hire year to split by. The function will iterate through that dictionary of dictionaries in order to return two dictionaries of dictionaries in a tuple: one in which the hire year is all before the split_year, and one in which the hire year is all after and including the split_year.

Example input and output:

```
split_by_hire_year(employees, 1964) → ({
    'employee_1': {'gender': 'Female', 'race': 'White', 'hire_year': 1960},
    'employee_3': {'gender': 'Male', 'race': 'Other', 'hire_year': 1955}
}, {
    'employee_2': {'gender': 'Male', 'race': 'Black', 'hire_year': 1965},
    'employee_4': {'gender': 'Male', 'race': 'White', 'hire_year': 1970}
})
```

test_split_by_hire_year()
Tests ***split_by_hire_year***
Test that the function correctly separates employees hired before 1964 from those hired in 1964 or later.

3. ***count_race_or_gender(employees)***

GM stated that their policy did not end up discriminating by race *or* by gender. We need to verify this claim with data.

count_race_or_gender takes one argument: a dictionary of dictionaries. The function should accurately count the number of employees belonging to each race and gender category. The output should be a dictionary containing two keys: 'race' and 'gender'. Under each key, there should be sub-dictionaries with race or gender categories as keys and their corresponding counts as values.

Example output:

```
{
    'race': {'White': 2, 'Black': 1, 'Other': 1},
    'gender': {'Female': 1, 'Male': 3}
}
```

test_count_race_or_gender()
Test that there are only two keys in the returned dictionary
Test that the function accurately counts the number of employees belonging to each race and gender category.

4. ***count_race_and_gender(employees)***

   The Black women claim that their policy discriminated by race *and* gender. We need to verify this claim with data.

   count_race_and_gender takes one argument: a dictionary of dictionaries. The function will return the number of employees within each combination of race and gender in a dictionary. The keys should be represented by the following format: “Race_Gender”.

   Example output:
   {'Black_Female': 4, 'Black_Male': 2, 'White_Male': 1, …}

   test_count_race_and_gender()
   Test that there are the correct number of keys in the dictionary representing each combination of race and gender in this dataset.
   Test that the function correctly counts the number of employees within each combination of race and gender.

5. ***write_csv(dict, “filename”)***
   write_csv will take two arguments. A dictionary that was produced by *count_race_and_gender()*, and a filename. The function will write the data from the dictionary into a csv file. The first column should contain the combinations of demographics and the second column should be integers representing the number of employees with that combination of demographics. The first line of the file should be the header information and each row of data should be on a new line.

6. **Questions:**
   Once you’ve completed the coding portion of this assignment, use the data you produced to answer the following questions. Data scientists think critically about how to turn data into actionable information – the programming and quantitative pieces are only part of the job. Turn in your answers to these questions as a PDF file in your Github repo along with your code and output. A few sentences for each question is fine.
   a. What are the potential advantages and disadvantages of seniority-based layoff policies, from the company owner’s perspective, the senior employee’s perspectives and from the perspectives of the employees

involved in the DeGraffenreid case? How might these stakeholders weigh these pros and cons differently?

b. This homework project is based on one of the law cases which Dr. Kimberlé Crenshaw analyzed in order to describe how the lived experiences of Black women differ from the lived experiences of White women and simultaneously differ from those of Black men. Black women exist in a space where the realities of race and gender overlap. Within the American social structure, it is at times a toxic place where racism *and* sexism exist simultaneously. Professor Crenshaw named the place "intersectionality" [cite]. In this case, we see that analyzing race and gender separately wasn't sufficient to capture the full story. What approaches can data scientists employ to ensure they're not just looking at isolated data points but are instead uncovering the many deeper narratives behind their datasets?

c. Data science is a powerful tool for uncovering information about the world, but it often grapples with imperfect data. In the context of this project, what limitations did you encounter regarding the data or analytical methods? Were there noticeable gaps in the data's representation of individuals' identities and experiences? How might these limitations impact the conclusions and insights derived from the data analysis?

7. **Extra Credit**

In the context of the DeGraffenreid v. General Motors case, consider the scenario where non-senior level employees (those with fewer years of service at GM) decide to form a coalition to protect their interests and prevent layoffs based on seniority. It may be helpful for the coalition to understand how many employees were hired each year. Your task is to iterate through the employee dataset, count the number of employees hired each year, and categorize them based on race and gender.

count_employees_by_year(employees): This function takes the employee dictionary as an input and returns a new dictionary where each key is a hiring year. The corresponding values to these keys are nested dictionaries. In the nested dictionaries, the keys are different racial categories, and they each have another nested dictionary as their values. This second-level nested dictionary has gender categories as its keys and the number of employees for each race-gender combination as the corresponding values.

Example output formatting:

```
{
   1965: {'White':{'Male':1, 'Female':0}, 'Black':{'Male':1, 'Female':0},...},
   1966: {'White':{'Male':0, 'Female':1}, 'Black':{'Male':1, 'Female':0},...},
   1967: {'White':{'Male':0, 'Female':1}, 'Black':{'Male':2, 'Female':0},...},
   1968: {'White':{'Male':1, 'Female':0}, 'Black':{'Male':1, 'Female':0},...},
   1969: {'White':{'Male':1, 'Female':0}, 'Black':{'Male':1, 'Female':0},...}
}
```

In addition to the code above, answer the following questions in the same PDF in addition to the questions in part 6. One or two sentences per question is fine.

a. Explain how the output of your function could help non-senior employees at GM understand how to build a coalition.
b. What potential actions could the employees take together to prevent discriminatory layoffs?
c. Analyze the potential advantages and limitations of different advocacy strategies - individual legal cases vs. collective bargaining (or a different idea you can think of)?
d. Provide your perspective on the most effective approach in this situation and why.

Item	Points
load_csv + ***test_load_csv***	50 + 10
write_csv	40
split_by_hire_year + ***test_split_by_hire_year***	14 + 6
count_race_or_gender + ***test_count_race_or_gender***	14 + 6
count_race_and_gender + ***test_count_race_and_gender***	24 + 6
Reflection shows critical thought	30
Extra credit	20