**Introduction:**

In this project, you will work with Comma-Separated Value (CSV) files storing AP exam race and ethnicity data. Working with CSV files is an essential skill for data analysts due to the amount of data stored in this format.

**Data Science & Social Justice:**

Data science is a difficult concept to define; it can refer to anything from cleaning Excel sheets to creating machine learning models. The Oxford Reference defines social justice as *“The objective of creating a fair and equal society in which each individual matters, their rights are recognized and protected, and decisions are made in ways that are fair and honest”* (<https://bit.ly/3xZrlY2>). Data science can be used to further social justice by highlighting inequity and inequality in society.

**Data Description:**

One data source for this assignment is the [Advanced Placement (AP) exams for 2020](https://research.collegeboard.org/programs/ap/data/participation/ap-2020)**.** Secondary students can take AP courses for college credit and/or placement. We extracted the race and ethnicity data listing the number of exam takers for the Computer Science A exams. The columns of the data are of the different race and ethnicity groups that were measured, while the rows are of the different regions.

The other data is from the Census Bureau. We took the data from the American Community Survey. The exact query and resulting data table can be found [here](https://data.census.gov/cedsci/table?t=Race%20and%20Ethnicity&g=0400000US01,02,04,05,06,08,09,10,11,12,13,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,44,45,46,47,48,49,50,51,53,54,55,56,60,66,69,78,80,81,82,83&y=2019&tid=ACSDP1Y2019.DP05&hidePreview=true). We reduced the data to just the columns you need.

Each file is in CSV format and is already cleaned. The AP data can be found [here](https://drive.google.com/file/d/1x8cK4C_XEQ1U7wS99kauHXryBMa0Gn28/view?usp=sharing) and the Census data can be found [here](https://drive.google.com/file/d/1cPc46PyDggjwDNfbmxDMTlrFOVQcfSwH/view?usp=sharing). The data is also included when you clone the GitHub repository. The data dictionary for the data can be found [here](https://docs.google.com/document/d/1MBxTDn8uKUx0xFONA_br8L0VEOktt3q0tMWEJnhNQQw/edit?usp=sharing). The first row of data for both files is the header information.

**Assignment:**

You will create the following five functions, four corresponding test functions, and main to load, analyze, and store the data.

1. **load\_csv(“filename”) -> dict**

***load\_csv*** will take a filename to read from as a string. It will return a dictionary of dictionaries in which a region is a key. The inner dictionary will use the demographic categories as the keys and either the number of exam takers or number of people of that category in that region as the values. You will want to convert the numbers from strings to integers.

***test\_load\_csv*** will test load\_csv

**Example output:**

When run on the AP data it should produce a dictionary like this:

{“west”: {“AMERICAN INDIAN/ALASKA NATIVE”: 39, “ASIAN”: ...},...}

When run on the Census data it should produce a dictionary like this:

{“west”: {“AMERICAN INDIAN/ALASKA NATIVE”: 1253113, “ASIAN”: ...}

1. **get\_perc(dict) -> dict**

***get\_perc*** will take a dictionary of dictionaries. The function will iterate through that dictionary (dict) of dictionaries (dicts) and return a dict of dicts where the inner key values are the proportion that each demographic category is of the region’s population.

For the census data, include each demographic’s regional population percent. For the AP data, include each demographic’s regional test-taker percent. Remember that each inner dictionary value is the count of that demographic in that data set. We recommend that you round your percentages to two decimal points.

An example of this in general terms is:

(White Population of region/Region Totals) \* 100 = percentage of the region’s population that is white

***test\_get\_perc*** will test get\_perc

1. **get\_diff(dict1, dict2) -> dict**

***get\_diff*** will take two arguments. The first is a dictionary with the AP data while the second is a dictionary with the census data. For each demographic category in each region (excluding the region totals of course), you will calculate the absolute value of the difference between the two datasets by subtracting the second dict from the first dict. This will produce a doubly nested dictionary which contains the difference between each “cell” of the data. As a reminder, the AP data contains a column that won’t be found in the census data (“NO RESPONSE”) so you’ll have to ignore that. We recommend you round your percentages to two decimal points.

An example of this in general terms is:

Absolute value (% of population of region that is white - % of test takers of AP CS A exam that are white) = the percentage difference between the population and test takes for that demographic.

***test\_get\_diff*** willtest get\_diff

1. **write\_csv(dict, “filename”)**

***write\_csv*** will take two arguments. The first is the dictionary that was produced through **get\_diff** and the second is the name for the output file ("proj1-yourlastname.csv”). The function will write the data from the dictionary into a csv file. The first column should contain the regions and the rest of the columns the percentages for each respective demographic category separated by commas. The first line of the file should be the header information and each row of data should be on a new line.

1. **min\_max(dict)**

***min\_max*** will take the argument of a dictionary. The data has been mutated to make it easier for you to sort it. Your goal is to create a triple nested dictionary that will contain regions with the largest and smallest differences between their demographics and the demographics of AP exam test takers for each demographic. It will look like this:

{"max": {"demographic": {"region": value...}...} "min": {demographic: {"region": value}...}...}

You will print and return that dictionary (Hint: use sorted. It will make your life easier.) If you choose to do the extra credit, the print statement won’t be included in this function.

***test\_min\_max*** tests min\_max

1. **Questions:**

Once you’ve completed the coding portion of this assignment, think about the data and answer the following questions. Turn in your answers to these questions as well as your code.

* 1. Was there anything surprising in the data? Why or why not were you surprised?
  2. Think about how you might attempt to convey this information visually. What sort of graph would you use and why?
  3. Do you think the data is missing anything? Why or why not?
  4. Can you tell a broader story with this data? If so, what broader story can you tell with this data? If not, what limits you?

1. **Extra Credit:**

Create a pair of functions to calculate the national percentages for each dataset and compute the difference between them.

**nat\_perc(data\_dict, col\_list)**

***Note:*** *col\_list refers to the headers in the original spreadsheets loaded at the beginning of the homework. Most of these column names can be extracted from data\_dict.*

***nat\_perc*** will take in a data dict (ap\_data or census\_data) and then calculate the demographic percentages at the national level.

**nat\_diff(data\_dict1, data\_dict2)**

***nat\_diff*** will take in the two percentage dictionaries you created with the nat\_perc function and calculate the difference between each value in them.

**Rubric:**

| Item | Percentage |
| --- | --- |
| load\_csv + test\_load\_csv | 25% + 5% |
| write\_csv | 20% |
| get\_perc + test\_get\_perc | 8% + 2% |
| get\_diff + test\_get\_diff | 8% + 2% |
| min\_max + test\_min\_max | 12% + 3% |
| Reflection shows critical thought | 15% |
| Extra credit | 10% |