Project Report for “How diverse is this group?” – An Analytic Tool

Introduction

For any group, given their demographic information such as gender, race/ethnicity, age, and similar, this program can calculate what percentage is M/F, White/Black/Hispanic /Asian/etc, 20-30/30-40/40-50/etc years old. This analysis can be used in hiring decisions or bias investigations in companies, as well as looking at data from other groups, like a census.

Data sets

Data input will come from a .txt file. This file will have each person’s demographic information in a separate line, each answer separated by a comma. The first line will be the categories of questions. The first three lines of an example file would be:

Gender, Race-Ethnicity, Age, Wage

Female, Black, 41 to 50, 21 to 30/hr

Male, White, 31 to 40, 31 to 40/hr

The name of the data file will be entered by the user. It will need to be in the same directory as the program. The output file, named by the user, will be saved in the same directory as well.

Methodologies and Analysis

**Approach:**

* The program prints the following: "This program analyzes the answers from a demographic survey to investigate the diversity of the group. Please enter the file name of the database to analyze:”
* The string entered + “.txt” is the file name of the input database. The first line is read, split, stripped, and made into a list categorieslist – this stores the names of the categories.
  + categorieslist: [“Gender”, “Age”, “Race-Ethnicity”, “Wage”]
* Similarly, with the use of the function makelistofdata(file), the rest of the file is read, split, stripped, and each answer is placed into a list restofdata which stores every single answer (with duplicates).
  + restofdata: [“Male”, “White”, “Female”, “Male”, “White”, …]
* The list restofdata is made into a dictionary where key is the category name, and the value is a set of subcategories – categoryrefdic.
  + categoryrefdic: {“Gender”: [Male, Female], “Race”: [White, Black, Asian] …}
* The function makedicofperc(listofdata) is then called. This takes the list restofdata and creates a new dictionary percentdic where the key is the subcategory and the value is the percentage of all occurrences.
  + percentdic: {“White”: 50.0, “Male”: 60.0, “Female”: 40.0 …}
* outputlist is a list created by putting the (key, value) of percentdic into a tuple. This is then sorted by the percentage of occurrence (value).
  + outputlist: [(“White”, 50.0), (“Male”, 60.0), (“Female”, 40.0) …]
* The results are printed using the function showresults(finallist). The printed sentence varies depending on the type of the category; i.e. for subcategories in “Wage” the statement changes to “earns” instead of “is”.
  + “ \_\_ percent of this group is \_\_\_” or “ \_\_ percent of this group earns \_\_\_”
* A nearly identical function (writetofile) is called if the user enters an output file name, and writes the same statements into a file.

**Limitations/Future optimizations:**

Since this data is assumed to come from a demographic survey, the categories such as “age” and “wage” will be ranges of values. The program currently cannot convert integer answers into ranges of data.

The input must also include “age” and “wage” as categories. A way to move past this would be to introduce a subcategory “not disclosed.“

The data printed is not grouped by category; it is simply sorted (descending) by occurrence. categoryrefdic could be used to group the data – the author found it too complex and did not have the time to figure it out.

Summary

The program was tested with two different sets of data, one blank text file, one text file with unusable data, and a .jpg file. The results were as expected; generating the statistics asked for the first 2, error messages for the latter 3.