CIS*2250 (Winter 2023) Team Project Description

Introduction

In the first half of the course, the labs introduced you to working with Python and files with first names and their popularity. The files that you analyzed were from the province of Ontario and handled the years 1917 to 2019. They were acquired from

- https://data.ontario.ca/dataset/ontario-top-baby-names-female
- https://data.ontario.ca/dataset/ontario-top-baby-names-male

But you will notice if you download those files yourself that there are a number of errors and inconsistencies in the data as discussed in the pre-lab material on CourseLink (https://courselink.uoguelph.ca/d2l/le/content/793400/Home).

The team project in the second half of the course will concentrate on acquiring, organizing, and analyzing more data on first names from other provinces in Canada and other countries around the world. In the Team Project section of CourseLink you will find a set of links to sites in Canada and around the world that point to collections of first names. Each of these datasets will have different formats and will cover a different number of years.

Requirement #1

Collect datasets of first names from the sites listed on CourseLink. After downloading the data files, you will need to write Python code that will process the information and convert it to a common format. This requirement may need you to create multiple processing scripts; you should not try to fit all conversion into one Python script since many of the collections may differ greatly from each other. Part of this task will also include correcting errors (blank names, names such as NONAME, *etc.*) and looking at how you want to do the ranking when there are ties, *e.g.* for example, in the Ontario datasets that you have been looking at names that have the same frequency are arbitrarily ordered so if Fred has a frequency of 102 and George does to then Fred is rank 56 and George is rank 57 (frequencies and ranks are made up). In other datasets you will find that the ranking for names with the same frequencies are the same and the next rank after that is reflective of how many people had the same rank before it. For example:

- David, frequency = 102, rank = 50
- Edgar, frequency = 102, rank = 50
- Freddie, frequency = 102, rank = 50
- George, frequency = 100, rank = 53

Requirement #2

The common format for the data files will be designed by your team. For example, one collection of names may have all of the female names in one file and male names in another. A second collection may store all names, male and female, in one file with one of the fields indicating the name type (male or female). You must as a team decide how you want to store the data for all of the collections. You should be able to justify your design choices.

Requirement #3

Now that the data is all in a common format, you will need to design how you will store the data files. This may involve creating a directory hierarchy where you store the files for easy access. Or you may store all of the files in one directory and the file names contain the information needed to understand what is in the files. Or you may use a combination of the two approaches.

Requirement #4

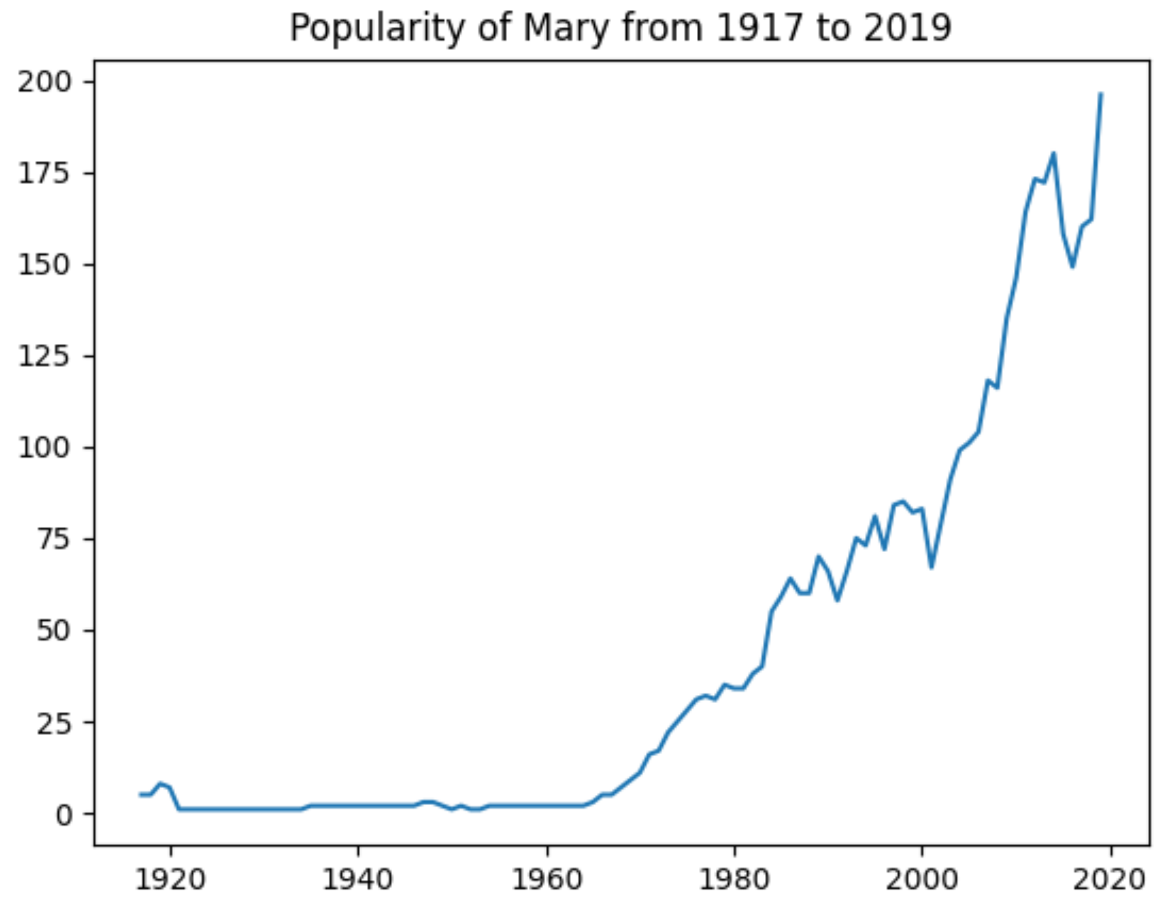
Now that you have the data formatted and stored, you will design an application that allows you to ask questions about the data. There are many different kinds of questions that you can ask such as:

- What is the most popular name for girls in the United States in the year 1930?
- If we compare the top 10 names for boys in Ontario and in Alberta in a given year, which names do they have in common?
- How many names are in both the female and male names lists in a given year in a given province or country and what are they?
- Given a name (male or female) where does it rank in various years in various provinces/countries?
- What is the all-time most popular male/female name in a given province/country?

and there are many more. Your team perform an analysis on all of the different types of questions that could be answered by your data and designing a way to ask the questions allowing for variables such as gender (male/female), time frame (1 year or a range of years), jurisdiction (province/country). You also need to think about how users will interact with your system and how you will present the answers.

Requirement #5

Visualization is a good way to show what is in the data. Decide on a number of data characteristics that would make good visualizations and design the code to input the variables such as gender, time frame, jurisdiction, *etc*. and the type of visualization requested. For example, you could show the rank of a name over a number of years in one jurisdiction (the female name Mary between 1917 and 2019 in Ontario) in a line graph.



This visualization was produced by the following code:

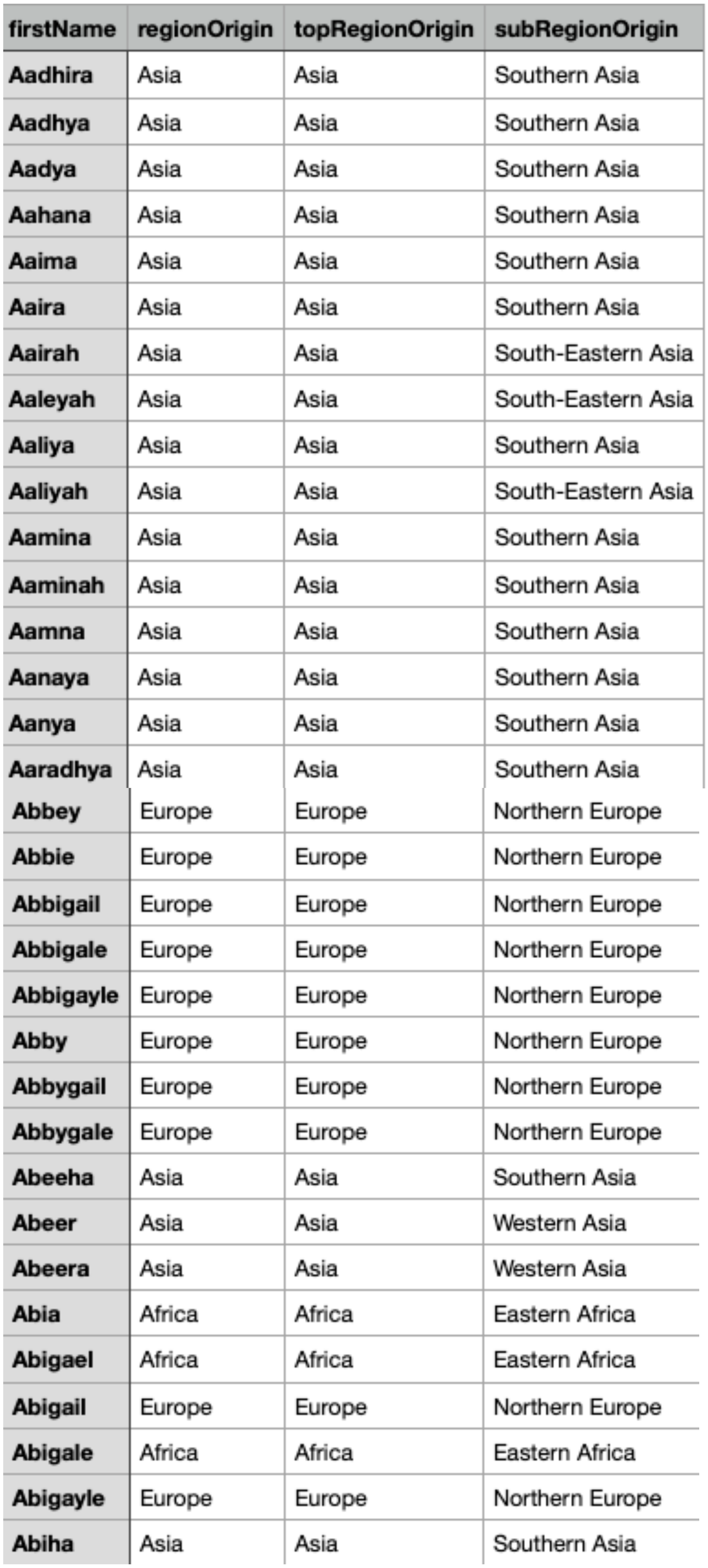
```
#!/usr/bin/env python3
import matplotlib.pyplot as plt
import numpy as np

# Define X and Y variable data
x = np.array([1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927,
1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941,
1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955,
1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969,
1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983,
1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997,
1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011,
2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019])
y = np.array([5, 5, 8, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 5, 5, 7, 9, 11, 16, 17, 22, 25, 28, 31, 32,
31, 35, 34, 34, 38, 40, 55, 59, 64, 60, 60, 70, 66, 58, 66, 75, 73, 81, 72, 84, 85, 82, 83,
67, 79, 91, 99, 101, 104, 118, 116, 135, 146, 164, 173, 172, 180, 158, 149, 160, 162,
196])

plt.plot(x, y)
plt.xlabel("X-axis")  # add X-axis label
plt.ylabel("Y-axis")  # add Y-axis label
plt.title("Popularity of Mary from 1917 to 2019")  # title
plt.show()
```

We will discuss this more in class.

Requirement #6

Now you have a wonderful way of exploring the names that are in your set of collections but can we do better? This requirement needs you to explore one (or more) of the following:

- Can you trace the change in demographics with regards to ethnicity with this data? Can you find a way to assign a type of ethnicity to each name (Europe, Asia, Africa, etc.) and trace how the distribution of these ethnicity changes over time? Using a "free" service, it is possible to get such information.



firstName	regionOrigin	topRegionOrigin	subRegionOrigin
Aadhira	Asia	Asia	Southern Asia
Aadhya	Asia	Asia	Southern Asia
Aadya	Asia	Asia	Southern Asia
Aahana	Asia	Asia	Southern Asia
Aaima	Asia	Asia	Southern Asia
Aaira	Asia	Asia	Southern Asia
Aairah	Asia	Asia	South-Eastern Asia
Aaleyah	Asia	Asia	South-Eastern Asia
Aaliya	Asia	Asia	Southern Asia
Aaliyah	Asia	Asia	South-Eastern Asia
Aamina	Asia	Asia	Southern Asia
Aaminah	Asia	Asia	Southern Asia
Aamna	Asia	Asia	Southern Asia
Aanaya	Asia	Asia	Southern Asia
Aanya	Asia	Asia	Southern Asia
Aaradhya	Asia	Asia	Southern Asia
Abbey	Europe	Europe	Northern Europe
Abbie	Europe	Europe	Northern Europe
Abbigail	Europe	Europe	Northern Europe
Abbigale	Europe	Europe	Northern Europe
Abbigayle	Europe	Europe	Northern Europe
Abby	Europe	Europe	Northern Europe
Abbygail	Europe	Europe	Northern Europe
Abbygale	Europe	Europe	Northern Europe
Abeeha	Asia	Asia	Southern Asia
Abeer	Asia	Asia	Western Asia
Abeera	Asia	Asia	Western Asia
Abia	Africa	Africa	Eastern Africa
Abigael	Africa	Africa	Eastern Africa
Abigail	Europe	Europe	Northern Europe
Abigale	Africa	Africa	Eastern Africa
Abigayle	Europe	Europe	Northern Europe
Abiha	Asia	Asia	Southern Asia

- Can you detect name spelling variations so that you can group them together and count them as the same name? This is very difficult to do and you may only be able to do a subset of names in a small number of collections?
- Make up your own explorations!