Go to a python command line, type “python RiskAssesmentBackEnd.py [questionnaire.csv]”

For the project-test, type “python project-test.py [questionnaire.csv]

def endometrialCancer(root):

\*\* This gets the information about pancreatic cancer in the CSV, adding it to a element tree which is then added to the larger parent tree of the whole XML file.

A SubElement is a child of a parent in our trees.

def pancreaticCancer(root):

\*\* This gets the information about pancreatic cancer in the CSV, adding it to a element tree which is then added to the larger parent tree of the whole XML file.

A SubElement is a child of a parent in our trees.

def get\_age(birthdate):

\*\*Gets the birthdate input from the user (formatted like 1/1/83) and separates it out to separate numbers in an array. The elements in the array (of size three) are in order of: *month, day,* and *year*. *Year* is only the last two digits of the actual birth year. Because they only input the last two digits, we have to assume that people filling out the questionnaire are less than 100 years old. If someone inputs “2/9/10,” we have to assume that they were born in 2010, not 1910.

if \_\_name\_\_ == '\_\_main\_\_':

\*\*We print the header of the xml string. This is the barebones needed for the output string to work in the Bayes Mendel Risk Assessment. The end product requires this content that we are printing out to std:out in a string format that has the quotation marks escaped but for testing purposes we are doing it this way.

After outputting the basic xml structure for each row, we loop through the columns of the xml document and extract the data. In our code, the only information that we were able to extract and successfully implement into our xml structure is the age of the person. We set up the code so that we can examine each category in every column and use that data to produce data for our xml string but we are currently having trouble finding how that data fits into the overall xml structure so that it complies with the Bayes Mendel requirements.