CS 2316 **Individual Homework 5 – Joint Probability**
Out of 100 points

Files to submit: **1. HW5.py**

<u>This is an INDIVIDUAL Assignment:</u>
Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 2316, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:
- TA Helpdesk – Schedule posted on class website.
- Email TA's or use Piazza

Notes:
- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- ***Do not wait until the last minute*** *to do this assignment in case you run into* problems.
- **Read the entire specifications document before starting this assignment.**

Introduction
In this assignment, you will be tasked with reading data from files in CSV format. The CSV file will contain a data for a joint probability table. Your task will be to verify that this is a valid joint probability table and to find the expected value of the two represented variables. This information will then be displayed on a GUI. Your GUI will also allow the user to select the file from which you should read data.

Your ENTIRE program must be built as a single Python class. Name the class HW5. At the very end of your code file, you will start the TK windowing system and instantiate your object. That may look something like this:

```
rootWin = Tk()
app = HW5( rootWin )
rootWin.mainloop()
```

Although you are free to organize your class in any way you would like, we *strongly recommend* that you break your code up into several manageable methods as outlined below. If you choose not to follow this suggested program organization, please place a comment in your code that uses the following names to illustrate where the similar functionality exists in your code:

The recommended breakdown of functionality within your class is the following methods:

1. __init__()
2. clicked()
3. readData(fileName)
4. convertData()
5. checkValid()
6. calcExpected()

Background Information

Often, multiple variables interact together. Consider the example of grades on CS 2316 homework and the final grade in CS 2316. These could be considered as two different random variables, X and Y.

If X and Y are discrete, the joint probability mass function of X and Y can be written as

$$f(x,y) = P(X = x, Y = y).$$

This gives way to

$$\sum_x \sum_y f(x,y) = 1.$$

The marginal pmf (probability mass function) of X is written as:

$$f_X(x) = P(X = x) = \sum_y f(x,y)$$

The marginal pmf of Y is written as:

$$f_Y(y) = P(Y = y) = \sum_x f(x,y)$$

Often, the joint pmf of $f(x,y)$ is shown in a joint probability table, which shows the joint probabilities in addition to the accompanying marginal probabilities. Note that $F_y(y)$ is calculated by summing along the row, and $F_x(x)$ is calculated by summing along the columns. Also note that both the marginal probabilities of X and Y sum to 1.

f(x,y)	**X = 1**	**X = 2**	**X = 3**	**$f_y(y)$**
Y = 1	0.3	0.2	0.1	0.6
Y = 2	0.1	0.2	0.1	0.4
$f_x(x)$	0.4	0.4	0.2	**1**

To calculate the expected value of X, each value of X should be multiplied by its marginal probability. For example:

$$E(X) = \sum_{a} a * f_X(a)$$

In the example above, E(X) is: 1 * 0.4 + 2 * 0.4 + 3 *0.2 = 1.8
and E(Y) is: 1 * 0.6 + 2 * 0.4 = 1.4

More information about joint probability distributions, marginal probabilities, and expected value can be found at the following links:
http://en.wikipedia.org/wiki/Joint_probability_distribution
http://en.wikipedia.org/wiki/Marginal_distribution
http://en.wikipedia.org/wiki/Expected_value

File Format Information

For this assignment, you will be given two comma separated value (CSV) files. These files will have different numbers of rows/columns, but the format will generally be the same.

The data in the CSV file mimics the joint probability table pictured above, without the last row and column, which represent the marginal probabilities. The values on the first row are the X values, and the values down the first column are the Y values. The interior values represent the joint probabilities. See the table1.csv and table2.csv files for examples.

Your program must work with and produce the correct answers for any similarly formatted data file, regardless of size. You are allowed to use the csv module to read this data, or you may write your own function to read the data.

Here is what we recommend for each of your object methods:

__init__()

This method will be called automatically when your object is initiated. It is responsible for setting up your GUI. It should create labels and text Entry widgets, along with a button to produce a GUI that looks like the following:

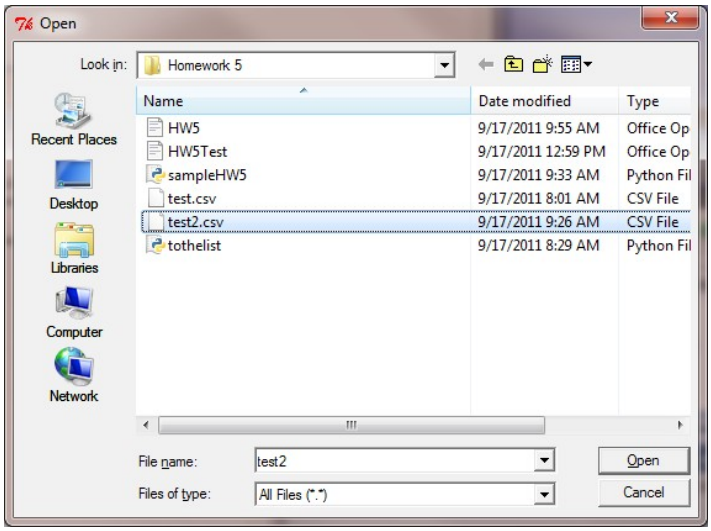
Note that the labels and text entry widgets are carefully arranged. The window is titled. The labels are right ("East") justified. The "Select File" button is close to the filename entry but has some padding space in between. The filename display text entry widget starts out with "…" in it, and is wide enough to show plenty of the file path (its width is 60). We suggest you use a grid layout to achieve this effect, although if you are able to do so by using multiple frames and the "pack" layout manager you may use that instead.

Note that all of the text entry widgets are set to "readonly" at this point.

The "Select File" button should trigger (call) the clicked() method.

clicked()

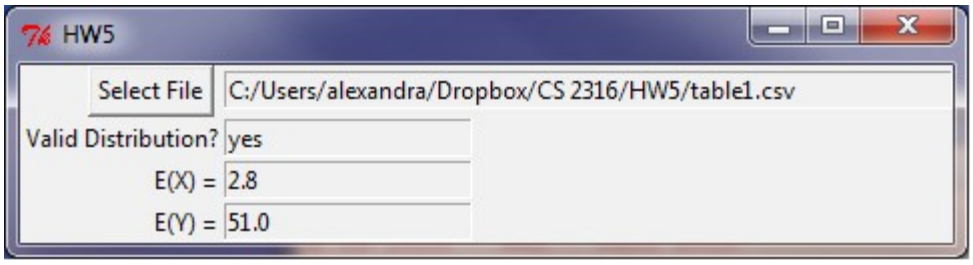
When the user clicks the "Select File" button, it executes the clicked() method. This method will present the user with a file dialog box. If the user clicks "cancel" your program must gracefully wait for the user to again click the "Select a File" button. (Also, if the entries have data left over in them from a previous run, they should be set back to blank/empty if the user hits cancel.) If the user actually selects a file and clicks "Open", the clicked method will do a lot more work.



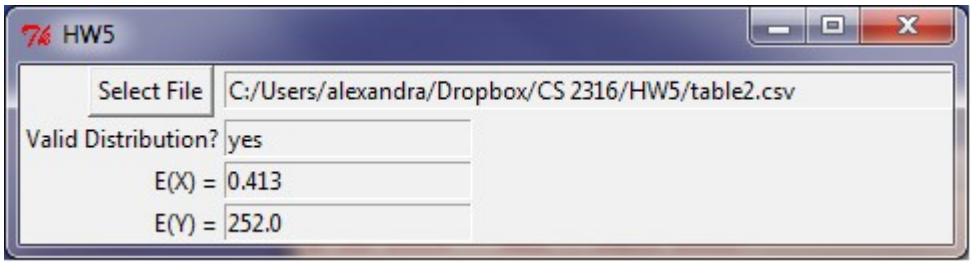
Specifically, it must cause the CSV data to be read in and stored, the textual representation of numbers converted to floats, the joint probability table validated, and, if valid, the expected values of X and Y calculated. It will use the readData, convertData,

checkValid, and calcExpected methods to do most of the heavy lifting for it. It also displays the file name in the entry that used to have "..." in it.

Here is an example of the data filled in on the GUI after reading the table1.csv file:



After loading one file, the user must be able to select the "Select File" button and load in any other file. The GUI must update itself appropriately. Here is an example of the GUI after reading in the table2.csv file:



Note that to change an Entry widget that has been set to "readonly", you must first set the status to NORMAL, cut the existing text out, insert new text, and then return the status to "readonly". (It may be helpful to create an additional method to take care of this, or use a StringVar). We use Entry widgets so that the user can copy and paste the data out of our program.

You may assume that the user will select a valid file, so do not have to do any error checking on the format or contents of the file.

readData(fileName)

This function will read in CSV data from the file name passed in. You may assume that the user will only select "valid" CSV files, and do not have to do error checking. You may use the CSV module if you would like. We suggest that you use an attribute of your HW5 object (such as self.CSVData = []) to store your data. This will allow all methods to access the data via the "self" reference. If an entry has leading or trailing spaces, you *should strip the extra spaces before storing it*. (This will prevent leading spaces from showing up in your GUI).

convertData()

This method will go through the self.CSVData variable, and convert all of the numbers from string representations to floats. This will allow you to do math on the numbers in the CSV files. You can either create a different data structure that will be used by the checkValid method, or you can replace the strings with floats in-place.

checkValid()

This method will check to see if the joint probability distribution is valid and update the GUI to reflect whether or not it is valid ("yes" or "no" should be added to the appropriate Entry box). That is, all the marginal probabilities of X and marginal probabilities of Y should sum to 1. Note that the first row entry and first column entry are not probabilities, so you should only sum starting with the 2nd entry of each. It will be useful if you create lists such as self.xMarginalProb and self.yMarginalProb to store the marginal probabilities for use in the calcExpected method. If the joint probability distribution is valid, the expected value of both X and Y should be calculated by the calcExpected method. Note that your actual answer may not be exactly 1.0 due to loss of precision with Python floating point arithmetic. You may accept any number between 0.9999 and 1.0001 as "valid".

calcExpected()

This method will determine the expected value for X and Y and update the GUI to reflect the calculated values. The expected value is found by the sum of each marginal probability multiplied by its corresponding value. Remember that the expected values should only be calculated if the joint probability distribution is valid (as demonstrated in the checkValid method). If the joint probability distribution is invalid, this method should not be employed. In that case, change those entries on the GUI to be blank.

Grading:

You will earn points as follows for each piece of functionality that works correctly according to the specifications.

The GUI	**40**
Has all labels, text entry fields, and "Select File" button	10
Properly allows the user to specify a file to load, and displays the file name when selected.	10
Text Entry areas are "readonly", allowing selection and copying, but not modification by the user	10
The data displayed is updated correctly when each file is loaded.	10
Data Loading / Calculations	**60**
Properly loads data from the CSV file	10
Correctly converts the string representation of numbers into floats that can be used for calculations	10
Correctly calculates the marginal probabilities of X and Y	10
Correctly identifies and displays whether the joint probability distribution is valid	10
Correctly calculates and displays the expected value of X	10
Correctly calculates and displays the expected value of Y	10