December 7, 2017
Data 520 Introduction to Programming
Judy Minichelli

Project Title: How to use SQLite3 and a GUI to analyze and display the most popular dog breeds

Introduction

An excerpt from a Kaggle submission "Largest Dog Breed Dataset"[1] was imported into a SQLite3 database table as a comma separated value (.csv) file. A graphical user interface (GUI) was created using Tkinter to access and display the top 10 breeds by count from the database table.

About the dataset

Pennsylvania state law requires registration of all dogs. There are millions of licensed dogs in Pennsylvania and 20,000+ in the greater Pittsburgh area.

The Kaggle Largest Dog Breed dataset contains the 2017 dog license renewals in the greater Pittsburgh area. An excerpt from the dataset above was imported into SQLite3 dogs89 database table Dog_Licc as a .csv file. Columns for Breed, Count, and Rank were included along with 212 aggregated rows, one for each unique breed.

Although 212 breeds appeared, as of December 2016 the American Kennel Club (AKC) only recognizes 189 breeds. The differential is due to inclusion of mixed breeds and rare breeds that are not yet recognized by the AKC. Only "popular" pure bred dogs, or dogs that have enough of a gene pool to create a breed standard against which they are judged in competition, are recognized by the AKC.

Despite the fact that this dataset is a microcosm, the top three breeds are the same as in the United States and most of the other breeds in the top 10 are in similar positions.

Importing a .csv file

Importing the .csv file proved to be more challenging than expected. Searches only yielded one-line commands used in Anaconda Jupyter Notebook or which buttons to push in software tools that handle administration of MySQL, for example, phpMyAdmin. Therefore, original code had to be written (see acknowledgement at the end of the document).

Once the code was written, the import ran for 20 minutes without success. It was assumed that the original data file containing seven columns and over 20,000 rows was too large to import. The data in the file were pared down and aggregated using Excel prior to successful import into the SQLite3 database table.

Using the database

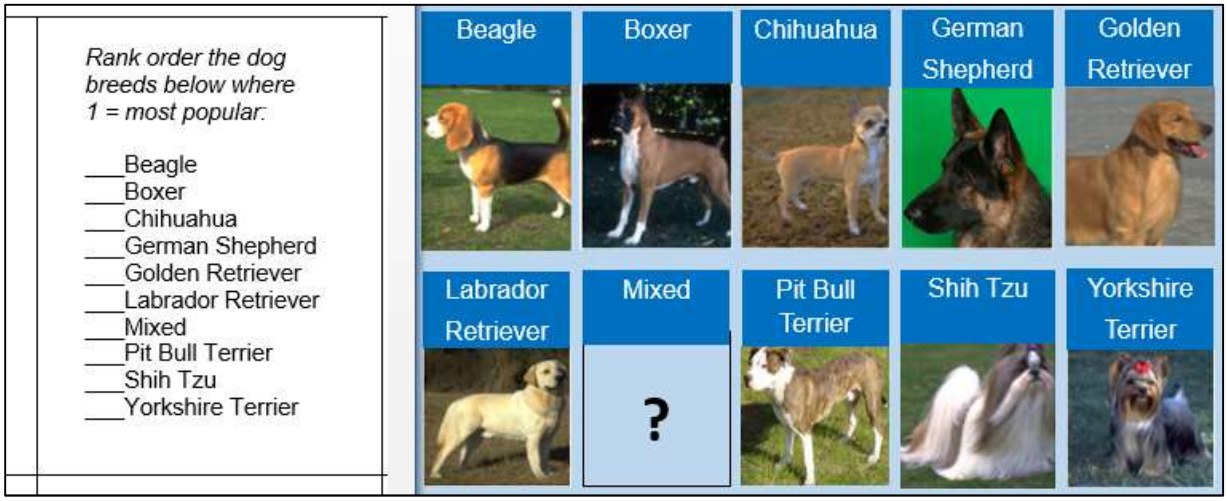
A lot of researching and experimenting was necessary to successfully query the database. And it was discovered that SQLite3 can function as a calculator. Sample queries and calculations appear in the code blocks in the appendix.

Creating a GUI

Although the GUI appeared to be simple, it was a challenging exercise. Returning output from the database table wasn't as difficult as expected; however, seemly simple things were; for example, centering the input in the entry box (successful) and reducing the size of the entry box (unsuccessful).

Presentation

To make the presentation interactive, a quiz[2] (see page two) was distributed where the top ten breeds were listed in alphabetical order. The audience ranked the breeds from most to least popular. The exercise was engaging but proved to be too difficult; the best score was 4 of 10 correct.



Results were successfully shown one at a time using the GUI, then displayed in the presentation for "color commentary" and discussion:

Rank order the dog breeds below where 1 = most popular:

8 Beagle
10 Boxer
6 Chihuahua
3 German Shepherd
4 Golden Retriever
2 Labrador Retriever – 10% of total count, (was Poodle, now #12)
1 Mixed – 29% of total count, includes "designer dogs", ex. Labradoodle
7 Pit Bull Terrier – no longer recognized by AKC
5 Shih Tzu
9 Yorkshire Terrier

The Labrador Retriever is the top pure bred dog. Up until the 70s, Poodles occupied the top spot. The trend is attributed to people migrating to the suburbs and home size doubling (that gives the bigger dogs more room to run and more space to co-habitate with their owners). It is suspected that temperament is also a factor; Labs are typically calm and affable while Poodles can be less so.

Included in mixed breeds are "designer dogs" like the Labradoodle; a cross between a Labrador Retriever and Poodle for gentler temperament without shedding.

In the 1970s, the American Pit Bull Terrier was removed the list of AKC recognize breeds because of breeding practices; they were and still are being bred to fight other dogs. Aside from the ethical consideration, they no longer fit the breed category of "Terrier" which is reserved for "ratters" (meaning they hunt vermin). The version of the breed that is now recognized is the Staffordshire Bull Terrier.

Lastly, SQLite3 database queries and calculations were successfully performed and all commented code blocks were reviewed.

Conclusion

The simplicity of the project doesn't correlate to the amount of time spent or knowledge gained! I've now analyzed this same dataset but in several different ways using Excel, Anaconda Jupyter Notebook and Python. So I've expanded my toolbox and understand some of the pros and cons of each application. The project reinforced the importance of knowing the dataset and objectives before committing to an application, method, or type of visualization.

Acknowledgement

Justin Minsk wrote the .csv file import code and provided two tweak to the GUI code that I wrote from scratch.

References

1. Kaggle dataset: https://www.kaggle.com/kingburrito666/largest-dog-breed-data-set
2. Pictures of dogs shown during the quiz: http://www.akc.org/dog-breeds/

Appendix

Five code blocks are shown on pages three – five. Comments appear on the right, in **BLUE.**

Code 1: import .csv file into SQLite3 database table

```
import csv
from sqlite3 import *

dataframe = connect('dogs89.db')          Create database, database table and empty table (placeholder)
df = dataframe.cursor()
df.execute('CREATE TABLE Dog_Licc(Breed TEXT, Count INTEGER, Rank INTEGER)')
table = []
with open('Dogbreedsimport.csv', newline='') as f:          Open and read .csv file
    reader = csv.reader(f)
    skip = next(reader)                                     Skip column headings
    for row in reader:
        table.append(tuple(row))                Add each row in .csv file to empty table as a tuple
print(table)
for line in table:                                  Add each row in table to database table
    df.execute('INSERT INTO dog_Licc VALUES (?, ?, ?)', line)
dataframe.commit()                                          Save database table
```

Code 2: SQLite3 database table/queries, Part 1

```
>>> import sqlite3
>>> con = sqlite3.connect('dogs89.db')          Connect to database
>>> cur = con.cursor()                          Connect the cursor
>>> cur.execute('SELECT * FROM Dog_Licc')       View contents of table
<sqlite3.Cursor object at 0x03886720>
>>> cur.fetchall()
[('MIXED', 5959, 1), ('LABRADOR RETRIEVER', 1379, 2), ('GERMAN SHEPHERD', 748, 3), ('GOLDEN
RETRIEVER', 694, 4), ('SHIH TZU', 636, 5), ('CHIHUAHUA', 626, 6), ('PIT BULL TERRIER', 595,
7), ('BEAGLE', 590, 8), ('YORKSHIRE TERRIER', 495, 9), ('BOXER', 447, 10), ('DACHSHUND', 444
, 11), ('POODLE ', 427, 12), ('BULLDOG', 333, 13), ('BICHON FRISE', 273, 14), ('MALTESE', 27
1, 15), ('POMERANIAN', 252, 16), ('GOLDENDOODLE', 245, 17), ('PUG', 239, 18), ('COCKER SPANI
EL', 223, 19), ('SCHNAUZER', 198, 20), ('ROTTWEILER', 196, 21), ('SIBERIAN HUSKY', 189, 22),
('BOSTON TERRIER', 188, 23), ('SHETLAND SHEEPDOG', 184, 24), ('PARSON RUSSELL TERRIER', 171,
25), ('AUSTRALIAN SHEPHERD', 157, 26), ('BORDER COLLIE', 157, 26), ('LABRADOODLE', 148, 27),
('WEST HIGHLAND WHITE TERRIER', 142, 28), ('COCKAPOO', 140, 29), ('CAVALIER KING CHARLES SPA
NIEL', 127, 30), ('MINIATURE PINSCHER', 123, 31), ('ENGLISH SPRINGER SPANIEL', 113, 32), ('H
AVANESE', 109, 33), ('JACK RUSSEL TERRIER', 107, 34), ('PUGGLE', 106, 35), ('LHASA APSO', 10
5, 36), ('DOBERMAN PINSCHER', 102, 37), ('GERMAN SHORTHAIR POINTER', 99, 38), ('MORKIE', 97,
39), ('GREAT DANE', 92, 40), ('GREYHOUND', 89, 41), ('CAIRN TERRIER', 87, 42), ('WEIMARANER'
, 81, 43), ('COLLIE', 80, 44), ('PEEKAPOO', 80, 44), ('BASSET HOUND', 79, 45), ('AUSTRALIAN
CATTLE DOG', 72, 46), ('SAINT BERNARD', 68, 47), ('BRITTANY SPANIEL', 66, 48), ('PAPILLON',
65, 49), ('VIZSLA', 64, 50), ('PEMBROKE WELSH CORGI', 52, 51), ('BERNESE MOUNTAIN DOG', 51,
52), ('ENGLISH MASTIFF', 51, 52), ('PEKINGESE', 51, 52), ('RAT TERRIER', 51, 52), ('BULLMAST
```

Code 2: SQLite3 database table/queries, Part 2

```
>>> import sqlite3
>>> con = sqlite3.connect('dogs89.db')
>>> cur = con.cursor()
>>> cur.execute('SELECT Breed FROM Dog_Licc WHERE Rank <11')
<sqlite3.Cursor object at 0x033F6720>
>>> cur.fetchall()
[('MIXED',), ('LABRADOR RETRIEVER',), ('GERMAN SHEPHERD',), ('GOLDEN RETRIEVER',),
ER',), ('BOXER',)]
>>> cur.execute('SELECT SUM (Count) FROM Dog_Licc WHERE Rank <11')
<sqlite3.Cursor object at 0x033F6720>                    Count for top ten breeds
>>> cur.fetchall()
[(12169,)]
>>> cur.execute('SELECT SUM (Count) FROM Dog_Licc')
<sqlite3.Cursor object at 0x033F6720>                    Count for all breeds
>>> cur.fetchall()
[(20691,)]
>>> round((12169/20691)*100,1)
58.8          Calculate % of total for top ten breeds to 1 decimal place
>>> cur.execute('SELECT Rank FROM Dog_Licc WHERE Breed = "GOLDENDOODLE"')
<sqlite3.Cursor object at 0x033F6720>
>>> cur.fetchall()
[(17,)]
>>> cur.execute('SELECT Count FROM Dog_Licc WHERE Breed = "GOLDENDOODLE"')
<sqlite3.Cursor object at 0x033F6720>
>>> cur.fetchall()
[(245,)]
>>> round((245/20691)*100,1)
                    Same exercise as above but for single breed
```

Code 3: Tkinter GUI, Part 1

```
import tkinter
import sqlite3

def convert(out_breed, enter_rank):                                    Function
    """ Find the rank in Sqlite3 dogs89.db table Dog_Licc and return the breed. """
    num = enter_rank.get()                                          Read user input
    num_tup=(num,)
    con = sqlite3.connect('dogs89.db')                          Connect to database
    cur = con.cursor()
    cur.execute('''SELECT Breed FROM Dog_Licc WHERE Rank = ?''',num_tup)   Find input in tuple in table
    output=cur.fetchone()
    output = output[0]
    out_breed.set(output)                          Return breed that corresponds to input (rank)
```

Code 3: Tkinter GUI, Part 2

```
window = tkinter.Tk()                                          Create root window (visible container)
frame = tkinter.Frame(window)                                  Create frame (invisible container for widgets)
frame.pack()                                                   Call method "pack" for all widgets, to place them in the parent widget

out_breed = tkinter.StringVar()                                Variables change with each input
enter_rank = tkinter.IntVar()

tkinter.Label(frame, text='Enter Breed Rank 1-10').pack()

enter_rank = tkinter.StringVar()
text = tkinter.Entry(frame, textvar=enter_rank, justify='center')   Entry box for rank input (centered)
text.pack()

out_breed = tkinter.StringVar()                                Output label for breed type
label = tkinter.Label(frame, textvar=out_breed)
label.pack()
                                                               Create two buttons
button = tkinter.Button(frame, text='Find Breed', command=lambda: convert(out_breed, enter_rank))
button.pack()

button2 = tkinter.Button(frame, text='Quit', command=lambda: window.destroy())
button2.pack()

window.mainloop()                                              Allows the code to run multiple times
```