Lab Exp. 01

As a warm-up exercise, let's create a function that computes the mean of a list of numbers. The input L is the list a user inputs. Our job is to compute m, the mean of this list L and then return it (hint, think for loop).

```
def meanOfAList(L):
    ''' Input L: A list of numbers (float or ints)
        Output : The mean of this list (float)
    '''

m=0.


return m
```

Lab Exp. 02

The media outlet, FiveThirtyEight, published an article that discusses school fight songs.
They collected fight songs from 65 schools—those belonging to the power five conferences and Notre Dame. The data set includes variables such as whether the song includes the word "Rah", "Win", "Victory", etc, the beats per minute of the song and the durection of the song in seconds.

A

Let make a scatterplot of the song's duration and beats per minute, like FiveThirtyEight did, to verify we have the same dataset.

```
fightSongData = pd.read_csv('https://raw.githubusercontent.com/fivethirtyeight/
                                          data/master/fight-songs/fight-songs.csv
                                          ')

# Your awesome code goes here

plt.show()                         # Show us the plot
```

Please make on additional scatterplot of Beats per minute by year the song was written.

B

Define a Bernoulli-distributed random variable X that assigns the value 1 to songs including the word "fight" and 0 to songs excluding the word "fight".

Estimate the probability a song will include the word "fight" by counting the number of songs that include fight and dividing by the total number of songs.

C

To gain intuition about the prob. of a song including “fight”, lets create the following algorithm:

- Generate a list of 65 1’s and 0’s simulated from our r.v. X.
- Compute the proportion of 1’s in this list
- Store this proportion in a list called “props”
- Run the above steps 100 times (i.e. you will have 100 items in the list props)
- Plot props as a histogram.