L4 PROBLEM 3 (5/5 points)

Write a function, `stdDevOfLengths(L)` that takes in a list of strings, `L`, and outputs the standard deviation of the lengths of the strings. Return `float('NaN')` if `L` is empty.

Recall that the standard deviation is computed by this equation:

$$\sqrt{\frac{\sum_{t \text{ in } X}(t-\mu)^2}{N}}$$

where:

- μ is the mean of the elements in *X*.
- $\sum_{t \text{ in } X}(t-\mu)^2$ means the sum of the quantity $(t-\mu)^2$ for *t* in *X*.

  That is, for each element (that we name *t*) in the set *X*, we compute the quantity $(t-\mu)^2$. We then sum up all those computed quantities.
- *N* is the number of elements in *X*.

1. Test case: If `L = ['a', 'z', 'p']`, `stdDevOfLengths(L)` should return 0.
2. Test case: If `L = ['apples', 'oranges', 'kiwis', 'pineapples']`, `stdDevOfLengths(L)` should return 1.8708.

```
    # compute mean first
    sumVals = 0
    for s in L:
        sumVals += len(s)
    meanVals = sumVals / float(len(L))

    # compute variance (average squared deviation from mean)
    sumDevSquared = 0
    for s in L:
        sumDevSquared += (len(s) - meanVals)**2
    variance = sumDevSquared / float(len(L))

    # standard deviation is the square root of the variance
    stdDev = variance**(.5)

    return stdDev
```

Correct

```
def stdDevOfLengths(L):
    """
    L: a list of strings

    returns: float, the standard deviation of the lengths of the strings,
      or NaN if L is empty.
    """
    if (len(L) == 0):
        return float('NaN')

    # compute mean first
    sumVals = 0
    for s in L:
        sumVals += len(s)
    meanVals = sumVals / float(len(L))

    # compute variance (average squared deviation from mean)
    sumDevSquared = 0
    for s in L:
        sumDevSquared += (len(s) - meanVals)**2
    variance = sumDevSquared / float(len(L))

    # standard deviation is the square root of the variance
    stdDev = variance**(.5)

    return stdDev

# using listcomps
def stdDevOfLengths2(L):
    n = float(len(L))
    if (n == 0):
        return float('NaN')
    lengths    = [len(s) for s in L]
    mean       = sum(lengths) / n
    squaredDev = [(l-mean)**2 for l in lengths]
    variance   = sum(squaredDev) / n
    return variance**(.5)


# using a separate function for std dev from lecture video
def stdDev(X):
    mean = sum(X) / float(len(X))
    tot = 0.0
    for x in X:
        tot += (x - mean)**2
    return (tot/len(X))**0.5

def stdDevOfLengths3(L):
    n = len(L)
    if (n == 0):
        return float('NaN')
    X = []
    for s in L:
        X.append(len(s))
    return stdDev(X)
```

Help

Test results

See full output

CORRECT

See full output

Check | Hide Answer

Show Discussion

New Post

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

About & Company Info

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+