Problem Set 3: Estimators

Incremental Mean

```
#In class you wrote a function mean that computed the mean of a set of numbers
#Consider a case where you have already computed the mean of a set of data and
#get a single additional number. Given the number of observations in the
#existing data, the old mean and the new value, complete the function to return
#the correct mean

from __future__ import division

def mean(oldmean,n,x):
    #Insert your code here
    return (oldmean*n+x)/(n+1)

currentmean=10
currentcount=5
new=4

print mean(currentmean,currentcount,new) #Should print 9
```

Likelihood Challenge

```
#Compute the likelihood of observing a sequence of die rolls
#Likelihood is the probability of getting the specific set of rolls
#in the given order
#Given a multi-sided die whose labels and probabilities are
#given by a Python dictionary called dist and a sequence (list, tuple, string)
#of rolls called data, complete the function likelihood
#Note that an element of a dictionary can be retrieved by dist[key] where
#key is one of the dictionary's keys (e.g. 'A', 'Good').

def likelihood(dist,data):
    #Insert your answer here
    likelihood = 1
    for x in data:
        likelihood *= dist[x]
    return likelihood

tests= [(({'A':0.2,'B':0.2,'C':0.2,'D':0.2,'E':0.2},'ABCEDDECAB'), 1.024e-07),((

for t,l in tests:
    if abs(likelihood(*t)/l-1)<0.01: print 'Correct'
    else: print 'Incorrect'
```

tests= [(({'A':0.2,'B':0.2,'C':0.2,'D':0.2,'E':0.2},'ABCEDDECAB'), 1.024e-07),(({'Good':0.6,'Bad':0.2,'Indifferent':0.2},['Good','Bad','Indifferent','Good','Good','Bad']), 0.001728),(({'Z':0.6,'X':0.333,'Y':0.067},'ZXYYZXYXYZY'), 1.07686302456e-08),(({'Z':0.6,'X':0.233,'Y':0.067,'W':0.1},'WXYZYZZZZW'), 8.133206112e-07)]

Note:

Likelihood in statistics quantifies how well a specific set of parameter values explains observed data. **It's calculated by multiplying the probabilities of each data point given the chosen parameters.** The higher the likelihood, the better the model fits the data. In essence, it helps determine the most probable parameter values for a given dataset.