IT 166 Lab 7

In this lab, we will practice how to cerate Python functions to calculate the same statistic parameters (e.g., mean, standard deviation) of a given population, as we did in Lab 3.

Objectives

- Be able to write Python functions to find minimal, maximal, mean, and standard deviation values of a collection of numerical data.
- Be able to write Python functions that conduct sampling on a given porpulation.
- Be able to write Python applications to use these functions.

Preparation

- Launch the Jupyter notebook.
- Rename the notebook page as “Lab7”

Please provide solutions to the problems below.

Background:

Statistic properties (e.g., mean, standard deviation) of a population is important for data analysis. In reality, we usually cannot collect the complete dataset of a whole population. Thus, we often take random samples of the population and estimate these statistic parameters based on the samples.

Given a sample of n observations from a population, we can calculate estimates of the population mean, standard deviation, and various other population characteristics (parameters). Prior to obtaining the complete dataset, there is uncertainty as to which of all possible samples will occur. Because of this, estimates such as $\bar{x}$, and s will vary from one sample to another. The behavior of such estimates in repeated sampling is described by what are called **sampling distributions**. Any particular sampling distribution will give an indication of how close the estimate is likely to be to the value of the parameter being estimated.

The **Central Limit Theorem** states that if you have a population with mean μ and standard deviation σ and take sufficiently large $n \geq 30$ random samples from the population with replacement, then the distribution of the sample means will be approximately normally distributed.

Sampling can be done with or without replacement. If with replacement, the same data point could appear more than once in a sample.

Problem 1

In this lab, we assume the following 100 weights to be a complete population, and thus can

accurately calculate some important parameters of this dataset:
[112.99,136.49,153.03,142.34,144.3,123.3,141.49,136.46,112.37,120.67,127.45,114.14,125.61,122.46,116.09,140,129.5,142.97,137.9,124.04,141.28,143.54,97.9,129.5,141.85,129.72,142.42,131.55,108.33,113.89,103.3,120.75,125.79,136.22,140.1,128.75,141.8,121.23,131.35,106.71,124.36,124.86,139.67,137.37,106.45,128.76,145.68,116.82,143.62,134.93,147.02,126.33,125.48,115.71,123.49,147.89,155.9,128.07,119.37,133.81,128.73,137.55,129.76,128.82,135.32,109.61,142.47,132.75,103.53,124.73,129.31,134.02,140.4,102.84,128.52,120.3,138.6,132.96,115.62,122.52,134.63,121.9,155.38,128.94,129.1,139.47,140.89,131.59,121.12,131.51,136.55,141.49,140.61,112.14,133.46,131.8,120.03,123.1,128.14,115.48]

Data source: http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_Dinov_020108_HeightsWeights

Your task:

- Create a Python List to store the population;
- Write four functions called **sumList()**, **meanList()**, **minMaxList()**, and **stdevList()** to calculate the summation, mean, miniumum, maximum, standard deviation of a given list. Each function has a list as an input, and returns the corresponding result (i.e., max, min, etc.).
- Write an application to apply the four functions above to the given weight list, and print out the results.

Please note: meanList() should use sumList() to calculate the summation of a list. minMaxList() should return both min and max values.

Mean: $\mu = \frac{\sum_{i=1}^{N} x_i}{N}$

Standard deviation: $\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}}$

(Here, x_i is a data point of the population)

Please note: in this lab, you still cannot directly use built-in functions max(); min(); sum(); and mean() from statistics module. You can use len() if needed.

The example of the expected output is shown below:

The summation of this population is 12920.86

The mean of this population is 129.21

The min and max of this population is 97.90 and 155.90

The standard deviation of this population is 12.06

Problem 2

- Write a function called **sampling()**, which can do sampling with and without replacement from the population.
- Write an application to use sampling() to conduct sampling **5** times with the sampling sizes 5, 10, 30, respectively for both with and without replacement. Calculate the mean value for each sampled list (You should print 30 mean values).

The expected outputs are shown below:

The mean of 5 samples with replacement is 133.21

.....(Omit 4)

The mean of 10 samples with replacement is 132.85

.....(Omit 4)

The mean of 30 samples with replacement is 131.93

.....(Omit 4)

The mean of 5 samples without replacement is 126.33

.....(Omit 4)

The mean of 10 samples without replacement is 124.41

.....(Omit 4)

The mean of 30 samples without replacement is 130.92

.....(Omit 4)

Here is a quick review about the two sampling methods:

- Sampling with replacement (the same data point could appear more than once in a sample)
- Sampling without replacement (the same data point could appear only once in a sample)

The function sampling() takes three arguments as input a list (type: List), sampling size (type: int), with/without replacement (type: boolean), which is **sampling(aList, sampleSize, withReplacement = True)**. sampling() returns the sampled list. Here, aList and sampleSize are positional arguments; withReplacement is a keyword argument with default as True.