
STATISTICAL ANALYSIS

PURPOSE

In this exercise, usage of lists in Python is explored. Statistical analysis is performed on a mock dataset, and printed to the user.

OBJECTIVES

After completing this exercise, you should be able to:

- Create and populate lists in Python
- Modify and iterate over lists in Python

PROCEDURE

PREPARE SUBMISSION FILE

1. Create a copy of the submission template called COMP6060**INIT**Lab6.docx where **INIT** is replaced with your own initials. So if your name is John Smith, the document will be called COMP6060**JS**Lab6.docx

PREPARE PYTHON FILE

1. Create a Python file called COMP6060**INIT**Lab6.py where **INIT** is replaced with your own initials. So if your name is John Smith, the document will be called COMP6060**JS**Lab6.py
2. Print out the following to the console, replacing NAME with your name:
`Welcome to NAME's statistical analysis tool!`

CREATE AND POPULATE DATASET

1. Create a Python list called `dataset` and assign it the following values:
`56, 74, -10, 58, 4, 17, 26, 0, 13, 37`
2. Create a variable called `dataset_size`, and assign it the value of the list size
3. Print the dataset size using the following format:
`The dataset has size elements`

ADJUST LIST

1. Using a `for` loop, iterate through the dataset, and assign a value of 0 to any negative values. Refer to the following syntax:
`for i in range(dataset_size):`

SORT LIST

Python lists provide a useful member function called `sort`, which sorts the list in ascending numerical order.

1. Sort the dataset. Refer to the following syntax:
`dataset.sort()`

CALCULATE MEAN

The mean is calculated by summing up all the values in a dataset, then dividing by the number of values in the dataset.

$$\bar{x} = \frac{1}{n} \left(\sum_{i=1}^n x_i \right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

1. Create a variable called `dataset_sum`, and assign it the value 0
2. Create a `for` loop, with a body that does the following:
 - a. Assign the value at the index to a variable called `val`
 - b. Add `val` to `dataset_sum`
3. Outside the for loop, create a variable called `mean` and assign it the value of `int(dataset_sum / dataset_size)`
The reason we need to use the `int()` cast is because the result of division in Python results in a float value. Float values are not allowed to be used as list indices.
4. Print the variable `mean` to the user in the following format:
`The mean of the dataset = mean`

CALCULATE MEDIAN

The median is the middle value of a sorted dataset. If the dataset has an even number of values, the median is the mean of the two middle values in the dataset.

1. Create a variable called `median`
2. Using an `if` statement, check if the size of the dataset is odd or even
 - a. If the dataset size is odd:
 - i. Set the value of `median` to the dataset element at index `(dataset_size / 2) + 1`
 - b. Otherwise, if the dataset size is even:
 - i. Create a variable called `mid_val_1` and assign it the value of the dataset element at index `int(dataset_size / 2)`
 - ii. Create a variable called `mid_val_2` and assign it the value of the dataset element at index `(dataset_size / 2) + 1`
 - iii. Set the value of `median` to `int((mid_val_1 + mid_val_2) / 2)`
3. Outside the if statement, print the value of median using the following format:
`The median of the dataset = median`

EXPECTED OUTPUT

```
Welcome to Lynn's statistical analysis tool!
The dataset has 14 elements
The mean of the dataset = 40.57142857142857
The median of the dataset = 43.5
```

Show results to Instructor.

Student Name: _____

Instructor: _____

Date: _____