

**ANL252**

**Python For Data Analytics**

# **Tutor-Marked Assignment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Ng Yi Jian** | **E2211293** |

**Submission Date: 14/09/2023**

1(a)

Plagiarism in coding occurs mostly in an academic context. This happens when students, who are expecting to meet deadline for their coding-related assignment, will go to online forum to seek for reuse codes to help expedite on their completion of their assignment (Lee, 2020). However, there are many ways to mitigate this issue.

* Using Plagiarism Checkers
* On the internet, there are sophisticated programs that can detect cloned lines of code. These similarity checkers are capable of detecting both genuine and partial plagiarism even if there are different variables (Carmil, 2021).
* Giving Clear Guidelines
* If students are allowed to use outside code or have a restricted source list for their assignment, urge students to comment on their code to clarify any sections where code has been taken from outside source
* Promote original and unique codes
* By showing students the benefit of coming up with solutions on their own can be rewarding, it prevents students to look up on online sources. Although there will be many errors that will occur, Students will comprehend more knowledge in programming

(178 words)

1(b)

import math

dataset = [2, 3, 4, 1, 2, 5]

sm=0

for i in range(len(dataset)):

sm+=dataset[i]

mean = sm/len(dataset)

deviation\_sum = 0

for i in range(len(dataset)):

deviation\_sum+=(dataset[i]- mean)\*\*2

psd = math.sqrt((deviation\_sum)/len(dataset))

ssd = math.sqrt((deviation\_sum)/len(dataset) - 1)

print("Population standard deviation of the dataset is", psd)

print("Sample standard deviation of the dataset is", ssd)

***code sourced from*** [***https://www.tutorialspoint.com/python-program-to-calculate-standard-deviation#***](https://www.tutorialspoint.com/python-program-to-calculate-standard-deviation)

In this code, we are trying to calculate the standard deviation of a dataset. We need to **import math** to get the square root function.

Firstly, we need to find the mean of the data set. We use **“for” loop** to iterate the dataset. **Len(dataset)** returns the number of elements in the dataset. To get the mean, we will get the sum of the dataset divided with the **len(dataset)**.

Secondly, we need to find the population standard deviation (PSD) and sample standard deviation (SSD). To derive the PSD, we will need to find the variance of the dataset while using **deviation\_sum** as a variable and pluck in the variance formula. From there, we will need to square root the variance then divided by **len(dataset)** get the standard deviation and use **psd** as a variable. The SSD is the same procedure except for the denominator where instead of dividing by **len(dataset)**, we will divide it by **len(dataset)-1** due to the degree of freedom.

Lastly, we will print out the result and the output would be the PSD and SSD.

(178 words)

1c)

import math

dataset = []

while True:

number = float(input("Enter a number:"))

dataset.append(number)

choice = input("Do you want to enter another number? (yes/no):")

if choice.casefold() == "no":

break

sm=0

for i in range(len(dataset)):

sm+=dataset[i]

mean = sm/len(dataset)

deviation\_sum = 0

for i in range(len(dataset)):

deviation\_sum+=(dataset[i]- mean)\*\*2

psd = math.sqrt((deviation\_sum)/len(dataset))

ssd = math.sqrt((deviation\_sum)/len(dataset) - 1)

print (dataset)

print ("The mean of the dataset is", "%.2f" % mean)

print("The PSD of dataset is", "%.2f" % psd)

print("The SSD of dataset is", "%.2f" % ssd)

In this code, there are few things that has been amended and added

* Firstly, the program now allows the user to input their numbers into the dataset such that it is much easier for user to store the number inputted instead of changing the codes, making it more flexible
* The code also gives user a choice to either input another number into the data set by saying “yes” or stop putting anymore input by saying “no”
* The program`s choice command is case insensitive by using **“choice.casefold()”** so that if user accidentally put capitals on their input, the program is still able to understand the input
* The program now includes the output of the dataset and the mean so that
* User is able to have a visual aid of the dataset so that they can spot any wrong input by them
* User will be able to compare their mean with their standard deviation to make a conclusion
* The number that the code churn out is much more concise as we used the modulus function to return the number as two decimal places

The output will be as such

Enter a number: 25.6

Do you want to enter another number? (yes/no): yes

Enter a number: 49.2

Do you want to enter another number? (yes/no): yes

Enter a number: 29

Do you want to enter another number? (yes/no): no

[25.6, 49.2, 29.0]

The mean of the dataset is 34.60

The PSD of dataset is 10.42

The SSD of dataset is 10.37

(183 words)

2)

The program code below is the improved code that was curated with the reference of Appendix 1.

products = ['laptop', 'mouse', 'webcam', 'keyboard', 'speaker']

print(f'We have a list of products here: {products}.')

def valid\_product():

while True:

item = input("Hello! What do you want to buy?").casefold()

if item in products:

return item

else:

print('Wrong product! Please try again.')

def product\_price():

while True:

try:

price = float(input("How much is it (in SGD)?"))

return price

except ValueError:

print("Invalid input. Pleaser enter a valid price.")

updated\_items = []

while True:

item = valid\_product()

price\_of\_item = product\_price()

entered\_input = [item, price\_of\_item]

updated\_items.append(entered\_input)

query = input("Would you like to continue? (yes/no) ").lower()

if query != 'yes':

break

print(f'This is our updated shopping list: {updated\_items}'

Firstly, we included the two define function. The first function is to check the validity of the product, i.e., to see whether our user input is within the shopping list and the second function is the price of the item. This is to help decrease the opportunities for syntax error by simplify the program code and debugging process making the program maintability(Wu &Zhu,2023)

Secondly, we use a while loop around a try-block for the second function to prevent any error from value error from showing User will receive an invalid input and prompt the user to try again. This is to make the price input more reliable.

Thirdly, all the inputs by user were made to be case insensitive using **.lower()** and **.casefold()** to prevent any errors occurring should user input anything that is capitalise.

Lastly, we create a condition for the query to break when user input anything other than yes. This is to improve the overall quality of the program.

(179 words)

**Reference**

Lee, C. (2020, July 28). *What is programming plagiarism? why is it on the rise?*. Turnitin. https://www.turnitin.com/blog/what-is-programming-plagiarism-why-is-it-on-the-rise

Carmil. (2021, July 9). *All you need to know about plagiarism in coding*. Copyleaks. <https://copyleaks.com/blog/all-you-need-to-know-about-plagiarism-in-coding>

Wu, K. Y., & Zhu, S. (2023). ANL252 Python for data analytics. Singapore University of Social Sciences.