

Total Score:

Question

a)

b)

c)

d)

e)

f)

g)

h)



**ANL252 (Online)**

**PYTHON FOR DATA ANALYTICS**

# **Tutor-Marked Assignment**

**July 2021 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Michelle Tan Ming Hui** | **J2110299** |

**Tutorial Group: ­­­­­­­­­­­­ T09**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 15/08/2021**

**Question(a)**

**Codes:**

# ANL252\_TMA01\_J2110299\_michelletan013\_MichelleTanMingHui\_15AUG2021

# Michelle Tan

# question(a)

# importing math functions

import math

print("Welcome to Python....")

print("\n")

print("In this program you will learn about how to calculate some math probabilities using mean, variance and other variables. Once you have entered all the valid inputs, you may be expected to receive two outputs. One which is pertaining to the probability density function (pdf) and the other would be the cumulative distribution function (cdf).")

print("\n")

**Description:**

The math module is a built-in Python module that is always accessible. To utilize the mathematical functions in this module, you must first import it using import math.

**Question(b)**

**Codes:**

# question(b)

def questionB():

while True:

try:

\_mean = input("Please enter the mean with any value between minus infinity (–∞) and plus infinity (+∞): ")

if \_mean == "":

\_mean = 0

varianceDistribution = 1

\_mean = int(\_mean)

print(f"Mean ", \_mean)

varianceDistribution = input("Enter a value larger than 0 for the variance: ")

if varianceDistribution == "":

\_mean = 0

varianceDistribution = 1

1

varianceDistribution = int(varianceDistribution)

print(f"Variance ", varianceDistribution)

except:

print ("Not a valid input, please try again. Please ensure your mean or variance is numeric :)")

else:

if varianceDistribution <= 0:

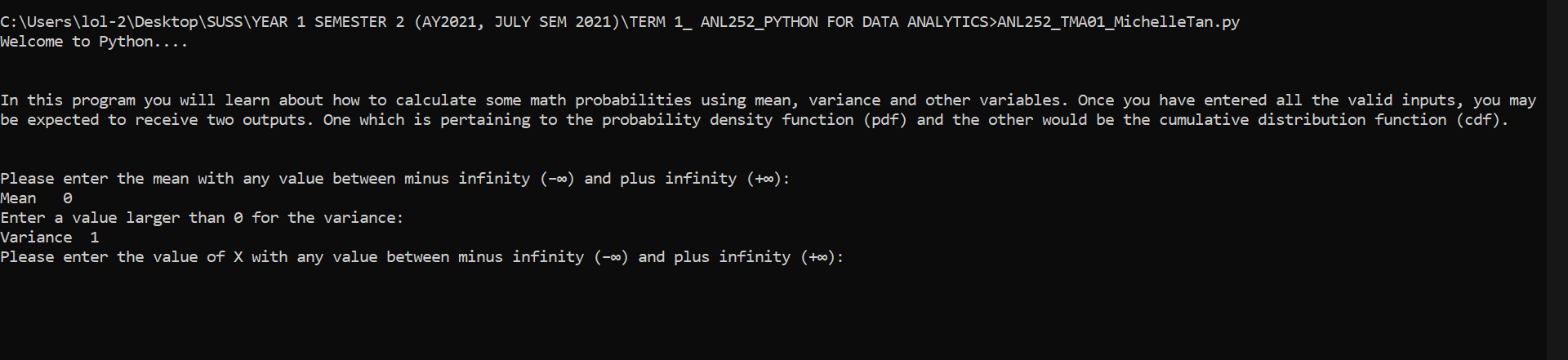
print ("Please try again, the variance have to be larger than 0!")

else:

\_mean = float(\_mean)

varianceDistribution = float(varianceDistribution)

return \_mean, varianceDistribution



**Description:**

Ques(b) is to ask the user to input the mean and variance with any value between negative infinity and positive infinity. Moreover, the variance must meet certain criteria, it must be larger than 0. Furthermore, the inputs of mean and variance must be numeric too. Lastly, if user press ENTER (did not input any valid values), the mean will automatically set to 0 and variance will automatically set to 1.

**Question(c)**

**Codes:**

# question(c)

# enter the value of X and make sure x is numeric

def questionC():

while True:

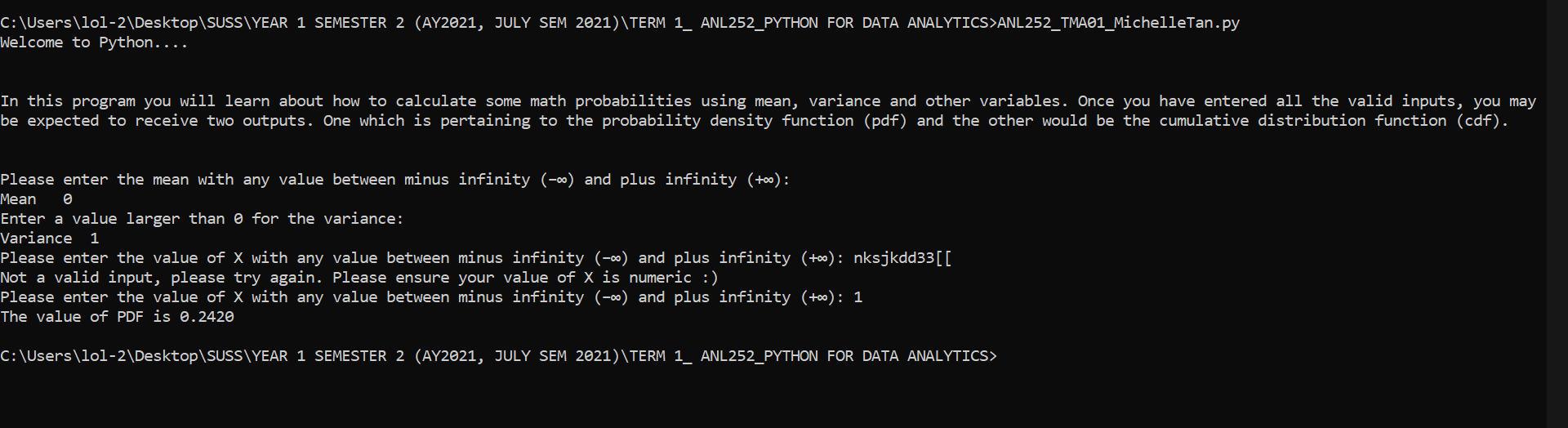
try:

randomX = int(input("Please enter the value of X with any value between minus infinity (–∞) and plus infinity (+∞): "))

return randomX

except:

print ("Not a valid input, please try again. Please ensure your value of X is numeric :)")



**Description:**

Ques(c) is to design an input screen to ask the user to enter the value of X with any value between negative infinity and positive infinity. Moreover, the input of X must meet certain criteria, it must be numeric.

**Question(d)**

**Codes:**

# question(d)

# construct a user-defined function using the formula of pdf to compute the value of pdf.

# the inputs are based on user inputs in (b) and (c)

def questionD(varianceDistribution, randomX, \_mean):

ProbabilityDensityFunction = 1 / math.sqrt(2 \* math.pi \* varianceDistribution) \* math.exp(- (randomX - \_mean)\*\*2 / 2 \* varianceDistribution)

return ProbabilityDensityFunction

**Description:**

Ques(d) is to design a user-defined function using the formula of pdf to compute the corresponding probability density fX(x) based upon the user inputs in (b) (mean and variance) and (c) (value of X).

**Question(e)**

**Codes:**

# question(e)

def questionE():

pass

# displaying the final value of probability density function (pdf), using formatted printing.

def pdfcomputation():

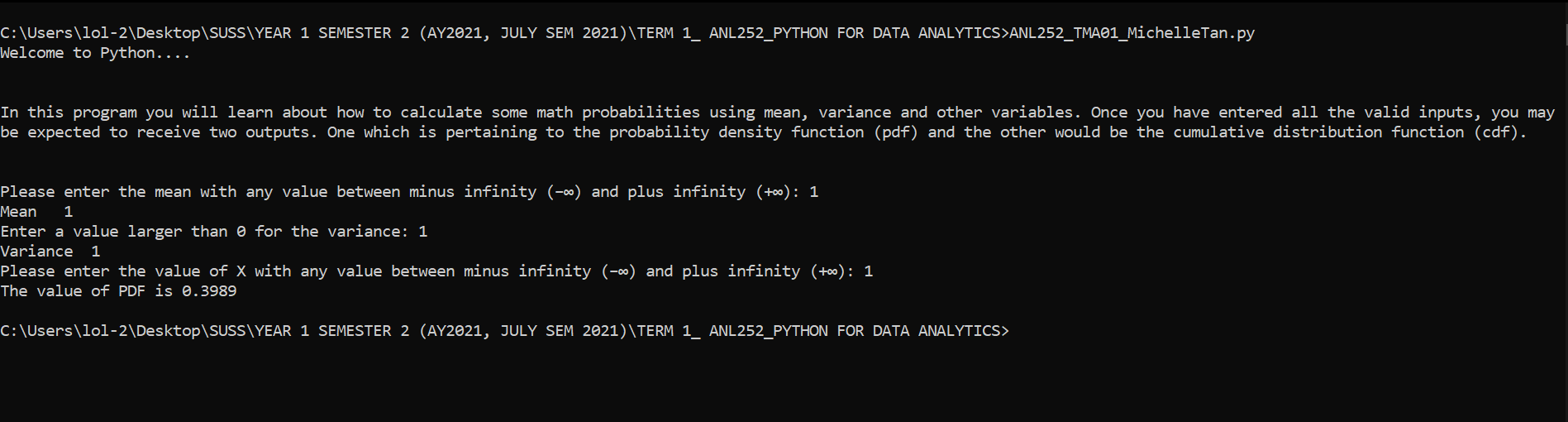
\_mean, varianceDistribution = questionB()

randomX = questionC()

pdf = questionD(varianceDistribution, randomX, \_mean)

print(f"The value of PDF is {pdf:.4f}")

pdfcomputation()



**Description:**

Ques(e) is to use formatted printing to display the results of (d). There are several ways to present an output of a program. f” formats and inserts the specified value into the string’s placeholder.

**Question(f)**

**Codes:**

# question(f)

# # # ..................................PART2...............................

# question(f)

# i program to ask mean, variance and k again....

# importing math functions

import math

# alpha = 0.01

# variableA = -100

print("Welcome to Python....")

print("\n")

print("In this program you will learn about how to calculate some math probabilities using mean, variance and other variables. Once you have entered all the valid inputs, you may be expected to receive two outputs. One which is pertaining to the probability density function (pdf) and the other would be the cumulative distribution function (cdf).")

print("\n")

def questionB():

while True:

try:

\_mean = input("Please enter the mean with any value between minus infinity (–∞) and plus infinity (+∞): ")

if \_mean == "":

\_mean = 0

varianceDistribution = 1

\_mean = int(\_mean)

print(f"Mean ", \_mean)

varianceDistribution = input("Enter a value larger than 0 for the variance: ")

if varianceDistribution == "":

\_mean = 0

varianceDistribution = 1

varianceDistribution = int(varianceDistribution)

print(f"Variance ", varianceDistribution)

except:

print ("Not a valid input, please try again. Please ensure your mean or variance is numeric :)")

else:

if varianceDistribution <= 0:

print ("Please try again, the variance have to be larger than 0!")

else:

\_mean = float(\_mean)

varianceDistribution = float(varianceDistribution)

return \_mean, varianceDistribution

def questionC():

while True:

try:

randomK = int(input("Please enter the value of k with any value between minus infinity (–∞) and plus infinity (+∞): "))

return randomK

except:

print ("Not a valid input, please try again. Please ensure your value of k is numeric :)")

def questionD(varianceDistribution, randomK, \_mean):

ProbabilityDensityFunction = 1 / math.sqrt(2 \* math.pi \* varianceDistribution) \* math.exp(- (randomK - \_mean)\*\*2 / 2 \* varianceDistribution)

return ProbabilityDensityFunction

def questionE():

pass

def cdfcomputation():

\_mean, varianceDistribution = questionB()

randomK = questionC()

cdf = questionD(varianceDistribution, randomK, \_mean)

print(f"The value of CDF is {cdf:.4f}")

cdfcomputation()

alpha = 0.01

variableA = -100

randomK = 0

def alpha():

alpha = 0.01

return alpha

alpha()

def variableA():

variableA = -100

return variableA

variableA()

def randomK():

random\_K = float(randomK)

return random\_K

randomK()

cDF = alpha()

variableA = float(variableA)

print("Value of a: ", variableA)

alpha = float(alpha)

print ("Value of Alpha: ", alpha)

\_number = variableA

result = 0.0

**Question(g)**

Ques(f) asked us to design a program to compute the value of cdf where the value of k is from the input in (c). I had changed the value of x to named it as “k”, and ask the user to input the value of k. Thereafter, after validating all the inputs for mean, variance and k, I tried using a while loop to generate the cdf. Next, declaring values of “alpha”, “a” and “k”. The smaller the step width is, the more accurate the probability is. I named the “a” as “variableA”. Using values from -100 to 0 (value of k), it has 10,000 steps. I am trying to compute out the “a” from -100, and it will stop when it reaches to the value of k. The output will be the cdf using formatted printing.

**Question(h)**

**Codes:**

# question (h)

# the probabilities would be from the (d).

d\_list = dict()

while \_number <=randomK:

result += questionD(\_number)

result = alpha \* result

d\_list[round(\_number, 4)] = round (result, 4)

\_number += alpha

print("The cdf is ", result)

for key in d\_list:

print (key, end = " ")

print(d\_list[key])