Logo, company name

Description automatically generated

|  |  |
| --- | --- |
| Course: | ANL252 |
| Assignment: | TMA01 |
| Name: | Tan Jing Jie |
| PI Number: | Q1882541 |
| Tutorial Group: | T09 |
| Submission Date: | 15 August 2021 |

a)

# import of math package  
import math  
  
# getting the value of Pi  
pi = math.pi  
print("Value of pi is", pi)

Print Screen:

Text

Description automatically generated

b)

# Create a Python program to allow user to enter the mean and variance of the distribution.  
# Users need to know that the mean can be any value between minus infinity and plus infinity  
# However, variance must be a value larger than 0.  
# Both input must be numeric.  
# If user press ENTER without providing any values, the program will automatically set mean to 0  
# and variance to 1.  
  
print("\nPlease enter the mean of the distribution.\nYou can enter any values from negative infinity to positive infinity.")  
  
while True:  
 mean = input("\nPlease enter the mean: ")  
 if mean == "":  
 mean = 0  
 break  
 else:  
 try:  
 mean = int(mean)  
 except ValueError:  
 print("Please enter a vaild number")  
 continue  
 else:  
 break  
  
print("\nPlease enter the variance of the distribution. \nValue of variance must be more than 0")  
  
while True:  
 variance = input("\nPlease enter the variance: ")  
 if variance == "":  
 variance = 1  
 break  
 else:  
 try:  
 variance = int(variance)  
 except ValueError:  
 print("Please enter a vaild number")  
 continue  
 else:  
 if variance < 1:  
 print("Please enter a value larger than 0")  
 continue  
 else:  
 break  
  
print("\nYou have entered mean value of", mean, "and the variance of", variance)

c)

# Design an input screen for the user to enter the value of X.  
# User needs to know that the value they he enter can range from negative infinity to positive infinity.  
# Must integrate control mechanism to ensure that the input is numeric.  
  
print("\nPlease enter the value of X.\nYou can enter any values from negative infinity to positive infinity.")  
  
while True:  
 try:  
 x = int(input("\nPlease enter value of X: "))  
 except ValueError:  
 print("Please enter a vaild number")  
 continue  
 else:  
 break

d)

# Construct a user-defined function using the formula of the probability density function.  
  
def pdf(x, mean, variance):  
 return (1.0 / (math.sqrt(2\*math.pi\*variance))) \* math.exp(-((x-mean)\*\*2)/(2\*variance))

e)

# Formatted printing to display the results.  
  
print(f"\nBased on your input values of mean = {mean}, variance = {variance}, x = {x}.")  
print(f"\nThe probability of a normally distributed random variable is {round(pdf(x, mean, variance),4)}.")

Print Screen of formatted printing:

Text

Description automatically generated

f)

def cdf(k, mean, variance):  
 start = -100  
 step = 0.01  
 current = start  
 result = 0;  
  
 while current < k:  
 result = result + pdf(current, mean, variance)  
 current = current + step  
 # print("Current:", current)  
  
 return result \* step  
  
print("The probability of X less than or equal to k is", round(cdf(x, mean, variance),4))

Print Screen of x = 0:

Text

Description automatically generated

Print Screen of x = 1.64:

Text

Description automatically generated

Print Screen of x = 1.96:

Text

Description automatically generated

g)

At the first part of the program, I employed a user define function for CDF(k, mean, variance) with the variables of start = -100 (default) as a; step = 0.01 (default) as the alpha; current = start; and result = 0. This is to define the variable required for the while loop.

A while loop is employed in the CDF function to compute the sum of cumulative distribution function for each step. In the equation of result = result + pdf function, the initial is 0 and as we go down the steps, the equation will sum up all the probabilities within the [ ] of the equation:

[𝑓x (𝑎𝑎)+⋯+𝑓𝑓 (𝑘𝑘−2𝛼𝛼)+𝑓𝑓 (𝑘𝑘−𝛼𝛼)+𝑓𝑓 (𝑘𝑘)]

Once the current value increased to equal or larger than the value of k, the loop will then stop the computation. The return function will then compute the weighted sum multiply by alpha to get the CDF value as shown in (return result \* step).

After the user define function, a formatted printing of the result is used to display the CDF value to the user.

h)

#Create a dictionary to store the probabilities of the normal distribtion range from -5 to 5.  
  
#define print of the dictionary.  
def print\_part\_h(dict):  
 current = -2.0  
 step = 0.5  
 end = 2.0  
  
 while current <= end:  
 print("x = ", current, "Probability = ", dict.get(current))  
 current = current + step  
  
#Create empty dictionary  
  
probability\_dict = {}  
mean = 0  
variance = 1  
step = 0.1  
  
#import of numpy to np.divide function.  
  
import numpy as np  
  
#Create of List with x-values (-5,5,0.1)  
x\_key = []  
  
for i in range(-50, 51, 1):  
 x\_key.append(i)  
  
#divide the value by 10 to and convert into a new list  
new\_list = np.divide(x\_key, 10)  
new\_x\_key = list(new\_list)  
  
#Create list for p\_values:  
p\_values = []  
  
for i in range(len(new\_x\_key)):  
 x = new\_x\_key[i]  
 mean = 0  
 variance = 1  
 step = 0.1  
 p\_values.append(cdf(x, mean, variance))  
  
#Insert list into dictionary.   
probability\_dict = {new\_x\_key[i]: round(p\_values[i],4) for i in range(len(new\_x\_key))}

#Printing of answer  
print\_part\_h(probability\_dict)

Print Screen of Dictionary {x : probability}:

Text

Description automatically generated

Appendix A: PY File Code

# ANL252 TMA01

# Tan Jing Jie Q1882541

# Dated 04 August 2021

# ------------------------------------(a)------------------------------

# import of math package

import math

# getting the value of Pi

pi = math.pi

print("Value of pi is", pi)

# ------------------------------------(b)------------------------------

# Create a Python program to allow user to enter the mean and variance of the distribution.

# Users need to know that the mean can be any value between minus infinity and plus infinity

# However, variance must be a value larger than 0.

# Both input must be numeric.

# If user press ENTER without providing any values, the program will automatically set mean to 0

# and variance to 1.

print("\nPlease enter the mean of the distribution.\nYou can enter any values from negative infinity to positive infinity.")

while True:

mean = input("\nPlease enter the mean: ")

if mean == "":

mean = 0

break

else:

try:

mean = float(mean)

except ValueError:

print("Please enter a vaild number")

continue

else:

break

print("\nPlease enter the variance of the distribution. \nValue of variance must be more than 0")

while True:

variance = input("\nPlease enter the variance: ")

if variance == "":

variance = 1

break

else:

try:

variance = float(variance)

except ValueError:

print("Please enter a vaild number")

continue

else:

if variance <=0:

print("Please enter a value larger than 0")

continue

else:

break

print("\nYou have entered mean value of", mean, "and the variance of", variance)

# ------------------------------------(c)---------------------------------

# Design an input screen for the user to enter the value of X.

# User needs to know that the value they he enter can range from negative infinity to positive infinity.

# Must integrate control mechanism to ensure that the input is numeric.

print("\nPlease enter the value of X.\nYou can enter any values from negative infinity to positive infinity.")

while True:

try:

x = float(input("\nPlease enter value of X: "))

except ValueError:

print("Please enter a vaild number")

continue

else:

break

# ------------------------------------(d)------------------------------

# Construct a user-defined function using the formula of the probability density function.

def pdf(x, mean, variance):

return (1.0 / (math.sqrt(2\*math.pi\*variance))) \* math.exp(-((x-mean)\*\*2)/(2\*variance))

# ------------------------------------(e)------------------------------

# Formatted printing to display the results.

print(f"\nBased on your input values of mean = {mean}, variance = {variance}, x = {x}.")

print(f"\nThe probability of a normally distributed random variable is {round(pdf(x, mean, variance),4)}.")

# ------------------------------------(f)------------------------------

def cdf(k, mean, variance):

start = -100

step = 0.01

current = start

result = 0;

while current < k:

result = result + pdf(current, mean, variance)

current = current + step

# print("Current:", current)

return result \* step

print(f"\nThe probability of X less than or equal to k is {round(cdf(x, mean, variance),4)}.\n")

#-------------------------------(h)------------------------------------

#Create a dictionary to store the probabilities of the normal distribtion range from -5 to 5.

#define print of the dictionary.

def print\_part\_h(dict):

current = -2.0

step = 0.5

end = 2.0

while current <= end:

print("x = ", current, "Probability = ", dict.get(current))

current = current + step

#Create empty dictionary

probability\_dict = {}

mean = 0

variance = 1

step = 0.1

#import of numpy to np.divide function.

import numpy as np

#Create of List with x-values (-5,5,0.1)

x\_key = []

for i in range(-50, 51, 1):

x\_key.append(i)

#divide the value by 10 to and convert into a new list

new\_list = np.divide(x\_key, 10)

new\_x\_key = list(new\_list)

#Create list for p\_values:

p\_values = []

for i in range(len(new\_x\_key)):

x = new\_x\_key[i]

mean = 0

variance = 1

step = 0.1

p\_values.append(cdf(x, mean, variance))

#Insert list into dictionary.

probability\_dict = {new\_x\_key[i]: round(p\_values[i],4) for i in range(len(new\_x\_key))}

#Printing of answer

print\_part\_h(probability\_dict)