

**ANL252**

Total Score:

Question

a)

b)

c)

d)

e)

f)

g)

h)

**Python for Data Analytics**

# **Tutor-Marked Assignment**

**July 2021 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Tan Mei Ping** | **Q1872131** |

**Tutorial Group: ­­­­­­­­­­­­ T 09**

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

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

#Question (a)

#import math package

import math

pi = math.pi

exp = math.exp

sqrt = math.sqrt

#Question (b)

#user to enter the mean of the distribution

#any value between +/- infinity

#automatically sets to 0 without any value

while True:

try:

mean = float(input("Enter mean: ") or "0")

break

except ValueError:

print("Value of mean can only be numeric, please try again")

print ("Mean =", mean)

#user to enter the variance of the distribution

#value must be larger than 0

#automatically sets to 1 without any value

while True:

variance = (input("Enter value of variance: ") or "1")

try:

variance = float(variance)

if variance > 0:

break

else:

print("Value of variance must be more than 0, try again")

except ValueError:

print("Value must be numerical, try again")

print("Variance =", variance)

![Text

Description automatically generated]()

#Question (c)

#user to enter the value of X

#any value between +/- infinity, must be numeric

while True:

try:

x = float(input("Enter the value of x: "))

break

except ValueError:

print("Value of x can only be numeric, please try again")

print ("x = ", x)

#Question (d)

#Construct user-defined function of PDF formula

def PDF(pi, exp, sqrt, mean, variance, X):

return (1 / sqrt(2 \* pi \* variance) \* exp(-((x - mean)\*\*2) / 2 \* variance))

print("The probability density is: ", PDF(pi, exp, sqrt, mean, variance, x))

#Question (e)

#Use formatted printing to display the result

print(f"Given that the distribution mean is {mean} and variance is {variance}, the probability density of {x} will be {PDF(pi, exp, sqrt, mean, variance, x)}")

#Question (f)

alpha = 0.01

a = -100

while True:

try:

k = float(input("Enter the value of k: "))

break

except ValueError:

print("Value of k can only be numeric, please try again")

print("k = ", k)

sum = 0

while a <= k: # value should be = k

PDF = (1 / math.sqrt(2 \* math.pi \* variance) \* exp(-((a - mean)\*\*2) / 2 \* variance))

sum += PDF

a += alpha

CDF = alpha \* sum

print("The CDF is: ", round(CDF,4))

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

#Question (g)

Assuming that alpha = 0.01 and a = -100, we allowed the user to input the value of k using a while-loop to ensure that the value of k is numeric. It prompts the user to enter the value again if it is not a numerical value.

The cumulative distribution function is derived from the weighted sum of the density function, with the step range being the value of alpha. In this case, alpha is 0.01, so every step range will decrease the value of a by a value of 0.01 until it reaches the value of k.

Therefore, a while-loop is being used to establish that as long as a is less than k, it will calculate the probability density function using the PDF formula given.

When a is less than or equals to k, the while-loop breaks, and the weighted sum of the probability density is calculated. Then this sum is multiplied by the value of alpha to derive the cumulated density function to give the probability of X being less than or equal to k.

#Question (h)

#create dictionary to store probabilities

# x are keys, proababilities are values

def xrange (start, stop , step):

while start < stop:

yield start

start = start + step

seq = xrange (-5.00,5.00,0.01)

keys = []

for z in seq:

keys.append(round(z,2))

values = []

for z in keys:

values.append()

dictionary = dict(zip(keys,values))

print(dictionary)

for key <= 2 and >= -2 in dictionary.items():

print(key, ":", value)