#ANL252

Total Score:

Question

a)

b)

c)

d)

e)

f)

g)

h)

#TMA01

#B1972075

# **PART (a)**

import math

# **PART (b)**

while True:

#returns default mean=0 if no input

mean = input("Please enter a value of mean between –∞ and +∞: ") or float(0)

try:

mean = float(mean)

print(f"Mean set to {mean}.")

break

#error handling when user input mean cannot be converted to float

except ValueError:

print("This is not a number, please try again.")

while True:

#returns default var=1 if no input

var = input("Please enter a value of variance larger than 0: ") or float(1)

try:

var = float(var)

if var > 0:

print(f"Variance set to {var}.")

break

#when var <= 0, prompt user to enter valid value

else:

print("The number cannot be less than or equals 0, please try again.")

#error handling when user input var cannot be converted to float

except ValueError:

print("This is not a number, please try again.")

# **PART (c)**

while True:

X = input("Please enter a value of X between –∞ and +∞: ")

try:

X = float(X)

print(f"X set to {X}.")

break

#error handling

except ValueError:

print("This is not a number, please try again.")

# **PART (d)**

#values of mean and var based on user inputs in PART (b)

#value of X based on user inputs in PART (c)

def pdf\_fx(X):

num = math.exp(-(X - mean)\*\*2/(2\*var))

dem = math.sqrt(2\*math.pi\*var)

return num/dem

# **PART (e)**

#formatted printing of pdf function fx, rounded to 4 decimal places

print(f"The corresponding probability density fx is {round(pdf\_fx(X), 4)}.")

Text

Description automatically generated

# **PART (f)**

#value of k based on user input X, in PART (c)

#values of mean µ and var σ2 based on user inputs in PART (b)

def cdf\_PX(X):

α = 0.0001

a = -1000

list\_cdf = [pdf\_fx(a), pdf\_fx(X)] #list of pdf f(x) for x from a to X

while a < X:

a += α

list\_cdf.append(pdf\_fx(a))

return α\*sum(list\_cdf)

print(f"The cumulative distribution function PX is {round(cdf\_PX(X), 4)}.")

#check results for X = 0, 1.64, 1.96 with mean=0 and var=1

mean = 0

var = 1

print("\nThe mean and variance has been set to 0 and 1 respectively.")

X = 0

print(f"For k = 0, the cumulative distribution function PX is {round(cdf\_PX(X), 4)}.")

X = 1.64

print(f"For k = 1.64, the cumulative distribution function PX is {round(cdf\_PX(X), 4)}.")

X = 1.96

print(f"For k = 1.96, the cumulative distribution function PX is {round(cdf\_PX(X), 4)}.")

Text

Description automatically generated

**# PART (g)**

PART\_g = "\n\

In PART (f), first set k as the user input X, which was in PART (c).\n\

Create user defined function cdf\_PX() so that it can easily called later on.\n\

α can be a value as small as possible, as smaller step range approximates cdf more accurately. Thus, set to 0.0001.\n\

a can be a number close to negative infinity, –∞. Thus, set to -1000.\n\

Create list list\_cdf, which is a list of pdf function f(x) with x values from a to X, referenced to PART (d).\n\

Range (a, X) has step range of α=0.0001.\n\

Populate the list with the first and last value of pdf when x equals a and X respectively.\n\

While-loop is used to generate x values with increment of α=0.0001 for all a less than X.\n\

Append list\_cdf with pdf value of all the generated x values.\n\

Return the cdf value by multiplying α and the summation sum() of all values in list\_cdf, thereby obtaining weighted sum of pdf in range(a,X).\n\

Display cdf result using formatted printing f“{}” and rounding to 4 decimal places.\n\

Checking results of k=0,1.64,1.96 by first setting mean=0 and var=1.\n\

Set X=0, print cdf\_PX().\n\

Set X=1.64, print cdf\_PX().\n\

Set X=1.96, print cdf\_PX().\n\

Obtaining results 0.5001, 0.9495 and 0.975 for k=0,1.64,1.96 respectively."

print(PART\_g)

# **PART (h)**

#as given, mean=0 and var=1

mean = 0

var = 1

dict\_normdist = {}

x = -5

while True:

if x > 5:

break

str\_x = str(x)

dict\_normdist[str\_x] = round(cdf\_PX(x), 4)

x += 0.1

x = round(x, 1)

#print(dict\_normdist)

i = float(-2)

while True:

if i == 2.0:

break

i = round(i, 1)

stri = str(i)

print(f"{stri}: {dict\_normdist[stri]}")

i += 0.5

A picture containing text, electronics

Description automatically generated