

**ANL252**

**Python for Business Analytics**

# **Tutor-Marked Assignment**

**July 2021**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Tia Kang Jun** | **B2070385** |

**Tutorial Group: ­­­­­­­­­­­­ T 09**

**Instructor’s Name: Mr. Kumar Manish**

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

1a) import math

1b)

#set mean and var to be a str

mean = ""

var = ""

#Create a while loop which will loop until mean is a float

while type(mean) != float:

try:

mean = input("Please enter a Mean value between minus infinity (–∞) and plus infinity (+∞): ")

#set mean = 0 if user just press enter when inputting the mean

if mean == "":

mean = 0

# make user input into a float as input function always returns user input as a string

mean = float(mean)

except ValueError:

print("Invalid input, please enter a number between minus infinity and plus infinity")

# Create a while loop which will loop until var is a float

while type(var) != float:

try:

var = input("Please enter the Variance (Value must be > 0): ")

#set var = 1 if user just press enter when inputting the variance

if var == "":

var = 1

# make user input into a float as input function always returns user input as a string

var = float(var)

#check if var is <= 0. If var <= 0, set var back to a string so that the loop will continue

if var <= 0:

print("Invalid input, please enter a number > 0")

var = ""

except ValueError:

print("Invalid input, please enter a number > 0")

1c)

#set x to be a string

x = ""

# Create a while loop which will loop until x is a float

while type(x) != float:

try:

x = input("Please enter the value of X (Value must be between minus infinity (–∞) and plus infinity (+∞)): ")

# make user input into a float as input function always returns user input as a string

x = float(x)

except ValueError:

print("Invalid input, please enter a number between minus infinity and plus infinity")

1d)

#set the math method pi, sqrt and exp as a variable for easier calling

pi = math.pi

sqrt = math.sqrt

exp = math.exp

#creating a function to calculate the probability density function

def pdf(mean,var,x):

result = (1/sqrt(2\*pi\*var)\*exp(-((x-mean)\*\*2)/(2\*var)))

return result

1e)

#set result be the result of the function pdf

result = pdf(mean,var,x)

#print the result using fstring formatting

print("The pdf of X = {} when Mean = {} and Variance = {} is {:.4f}".format(x,mean,var,result))

A computer screen shot

Description automatically generated with low confidence

1f)

#create a function to calculate the cdf

def cdf(mean,var,k):

#set a = -100, alpha = 0.01 and ccdf (the variable which will hold the value of cdf) = 0

a = -100

alpha = 0.01

ccdf = 0

#n is to calculate the number of iterations needed to make a = k

n = abs((a-k)/alpha)

#create a while loop that will run until n is smaller than k

while n >= k:

calc\_k = round(k - n\*alpha,2)

ccdf += pdf(mean,var,calc\_k)

n = n-1

ccdf = 0.01\*ccdf

return ccdf

print("The probability of X = {}, Mean = {} and Variance = {} is {:.4f}.".format(x,mean,var,cdf(mean,var,x)))

k = [0,1.64,1.96]

#print out the cdf for the corresponding cdf for each item in variable k using formatted printing

for num in k:

print("The probability of X = {}, Mean = 0 and Variance = 1 is {:.4f}.".format(num,cdf(0,1,num)))

A picture containing text, screenshot, computer

Description automatically generated

1g) Based on the equation 𝑃(𝑋≤𝑘)≈𝛼[𝑓𝑋(𝑎)+⋯+𝑓𝑋(𝑘−2𝛼)+𝑓𝑋(𝑘−𝛼)+𝑓𝑋(𝑘)],

First we have to find the number of times it will take for a to be equal to k. To find the number of times we have to solve the equation, a = k-n\*alpha. n = abs((a-k)/alpha). Once we have solved for n, we can integrate it into the program by creating a while loop that will loop until a > k.

For example if a = -10, alpha = 1 and k = 1. n will therefore be

abs((-10-1)/1) = 11. Substituting n into the equation, we get 𝑃(𝑋≤1)≈1[pdf(k-11\*alpha)+⋯+pdf(k-2\*alpha)+pdf(k-alpha)+𝑓𝑋(k)].

1h)

#set start = to -5 and step size to be 0.1

start = -5

step = 0.1

#create an empty list and dict

x\_list = []

x\_dict = {}

#while start is less than 5.1, it will loop and add each value of start into the list

while start < 5.1:

x\_list.append(start)

start = round(start + step,1)

#This loop will iterate through every item in the list to calculate its pdf and add object and its probability value into the dictionary

for i in x\_list:

prob = cdf(0,1,i)

x\_dict[i] = prob

#create a variable called keys

keys = -2

#create a while loop to iterate through and print out the key and its corresponding value with step size of 0.5

while keys <= 2:

print("Key: {} \t Probability: {:.4f}".format(keys,x\_dict[keys]))

keys = keys + 0.5

Graphical user interface, text

Description automatically generated