PROBABILITY METHODS IN ENGINEERING

ASSIGNMENT # 06

Spring 2023

CSE-209 Probability Methods In Engineering

Submitted by: **Shahzad Bangash, Suleman Shah , Ali Asghar**

Registration No. : **21PWCSE1980, 21PWCSE1983, 21PWCSE2059**

Class Section: **C**

“On my honor, as students of University of Engineering and Technology, We have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

Dr. Amaad Khalil

DATE: 20 / June / 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Probability Concepts and Implementation in Python

Introduction:

Probability theory is a fundamental branch of mathematics that deals with the study of uncertainty and randomness. It provides a framework for understanding and quantifying the likelihood of events occurring. In this report, we will explore various probability concepts and their implementation in Python.

1. Random Variables:

In probability theory, a random variable is a variable whose value depends on the outcome of a random event. Random variables can be discrete or continuous, and they are often used to model and analyze uncertain quantities. In Python, we can define and manipulate random variables using the SciPy library.

Example:

Let's consider a random variable X that represents the number of heads obtained when flipping a fair coin three times.

Code:

L13_BinomialRV.py > BinomialRV

```python
#Let X be the number of heads in three independent tosses
#of a coin. Find the pmf of X. (Binomial RV)

from math import *

def BinomialRV(n, k, p, q):
        binomial_coefficient = factorial(n) // (factorial(k) * factorial(n - k))
        probability = binomial_coefficient * pow(p, k) * pow(q, n - k)
        return probability

Sx =[0, 1, 2, 3]
probabilities = [] #Empty list for saving probabilities

print("Sx =", Sx)
for k in range(0,len(Sx)):
    probabilities.append(BinomialRV(3, k , 0.5, 0.5))
print("Px =", probabilities)
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS D:\UNi\PME\Python> python -u "d:\UNi\PME\Python\L13_BinomialRV.py"
Sx = [0, 1, 2, 3]
Px = [0.125, 0.375, 0.375, 0.125]
PS D:\UNi\PME\Python>
```

2. Expected Value of Random Variable:

The expected value, also known as the mean or average, is a fundamental concept in probability theory that measures the central tendency of a random variable. For a discrete random variable X with a probability mass function (PMF) f(x), the expected value is denoted as E(X) or μ.

Example:

Let X be the number of heads in three tosses of a fair coin. Find E[X].

Code:

```python
L14_ExpectedValue.py > ...
# Let X be the number of heads in three tosses of a fair coin. Find E[X].

from math import *
def BinomialRV(n, k, p, q):
        binomial_coefficient = factorial(n) // (factorial(k) * factorial(n - k))
        probability = binomial_coefficient * pow(p, k) * pow(q, n - k)
        return probability

def Calc_EX():
    EX=0
    for i in range(len(probabilities)):
        EX +=  Sx[i]*probabilities[i]

    return EX

Sx =[0, 1, 2, 3]
probabilities = [] #Empty list for saving probabilities

print("Sx =", Sx)
for k in range(0,len(Sx)):
    probabilities.append(BinomialRV(3, k , 0.5, 0.5))
print("Px =", probabilities)
print("E[X] =", Calc_EX())
```

Output:

```
PS D:\UNi\PME\Python> python -u "d:\UNi\PME\Python\L14_ExpectedValue.py"
Sx = [0, 1, 2, 3]
Px = [0.125, 0.375, 0.375, 0.125]
E[X] = 1.5
PS D:\UNi\PME\Python>
```

3. Variance of Random Variable:

The variance is a measure of the dispersion or spread of a random variable around its expected value. It quantifies how much the random variable deviates from its mean. For a random variable X, the variance is denoted as Var(X) or σ^2.

Example:

Let X be the number of heads in three tosses of a fair coin. Find VAR[X].

Code:

```
L10_Geometric.py   L2_RF.py   L13_BinomialRV.py   L14_ExpectedValue.py   L16_Variance.py ×   L7_
L16_Variance.py > ...
# Let X be the number of heads in three tosses of a fair coin.
#Find VAR[X].

from math import *
def BinomialRV(n, k, p, q):
        binomial_coefficient = factorial(n) // (factorial(k) * factorial(n - k))
        probability = binomial_coefficient * pow(p, k) * pow(q, n - k)
        return probability

def Calc_EX():
    EX=0
    for i in range(len(probabilities)):
        EX +=  Sx[i]*probabilities[i]

    return EX

def Calc_EX_2():
    EX_2 = 0
    for i in range(len(probabilities)):
        EX_2 +=  (Sx[i]*Sx[i])*probabilities[i]

    return EX_2
```

L16_Variance.py > ...

```python
    for i in range(len(probabilities)):
        EX_2 +=  (Sx[i]*Sx[i])*probabilities[i]

    return EX_2

Sx =[0, 1, 2, 3]
probabilities = [] #Empty list for saving probabilities

print("Sx =", Sx)
for k in range(0,len(Sx)):
    probabilities.append(BinomialRV(3, k , 0.5, 0.5))
print("Px =", probabilities)

E_X = Calc_EX()
E_X_2 = Calc_EX_2()
VAR_X = E_X_2 - (E_X * E_X)

print("E[X] =", E_X)
print("VAR[X] =", VAR_X)
```

Output:

```
PS D:\UNi\PME\Python> python -u "d:\UNi\PME\Python\L16_Variance.py"
Sx = [0, 1, 2, 3]
Px = [0.125, 0.375, 0.375, 0.125]
E[X] = 1.5
VAR[X] = 0.75
PS D:\UNi\PME\Python>
```