**Probability Distribution**

**Problem Description**

Suppose paperweights produced at XYZ Company have weights that are normally distributed with a mean of M grams and variance of V grams. What is the probability that a randomly chosen paperweight weighs more than **W** grams?  
  
**Input Format**  
Mean of distribtion : M (float)  
Variance of distribtion : V (float)  
Given Weight : W (float)  
  
**Output Format**  
A float upto two decimal places  
  
**Example Input**

42  
19.5  
46

**Example Output**  
0.18  
  
**Hint**  
  
use large enough sample size (ex: 1000000) for generalization

SOLUTION:-

def calculate\_probability(M, V, W):

    siz= 1000000

    sd=np.sqrt(V)

    d=np.random.normal(M,sd,size=siz)

    c=(d>W).sum()

    pb=c/siz

    return round(pb,2)

# Covid probability (bayes)

**Problem Description**

As you know, Covid-19 tests are common nowadays, but some results of tests are not true. Let’s assume a diagnostic test has 90% accuracy and 60% of all people have Covid-19. If a patient tests positive, what is the probability that they actually have Covid Positive? Write a program that calculates the required probability.

**Input Format**

prior\_probability (float)

probability of positive marked (float)

probability of negative marked (float)

All the inputs are in string format kindly typecast them into required formats.

**Output Format**

A float

**Example Input**

0.7

0.95

0.05

**Example Output**

0.97

**Example Explanation**

Used the concept of bayes theorem

Solution:-

def solve(A,B,C):

    prior = float(A)

    p\_g = float(B)

    p\_ng = float(C)

    return (prior\*p\_g)/((prior\*p\_g) + (p\_ng\*(1-prior)))