Monte Carlo - Unidad 2, Sesion 3 - Ejercicio

[Problema]: se desea estimar el volumen de una region R de [0,1]^6 definida por todos los puntos de la hiper-esfera de centro (0.45, 0.5, 0.6, 0.6, 0.5, 0.45) y radio 0.35 que ademas cumplen con las restricciones 3x1 + 7x4 <= 5; x3+x4 <= 1; x1-x2-x5+x6 >= 0

Entrega 2 - Ejercicio 3.1

Parte a:

[Letra] Implementar un programa que reciba como parametro la cantidad de replicaciones n a realizar, y emplee Monte Carlo para calcular (e imprimir) la estimaci´on del volumen de R, y la desviaci´on estandar de este estimador. Incluir codigo para calcular el tiempo de calculo empleado por el programa. Utilizar el programa con n = 104 y luego con n = 106 para estimar el volumen de R. Discutir si los dos valores obtenidos parecen consistentes. (en la sesi´on 5 se continuar´a este ejercicio).

In [2]:
```
import random
import math
import tabulate
import time
random.seed()

def sortearPuntoRN(dim=2):
    """
    Seortea un punto en R^N dentro del hiper-cubo [0,1]^N
    """
    punto = []
    for n in range(0, dim):
        punto.append(random.uniform(0.0, 1.0))
    # end for

    return punto
# end fun sortearPuntoRN

def puntoDentroVolumen(punto, restricciones=True):
    """
    Devuelve 0 o 1 si un punto esta fuera o dentro de un cierto volumen.
    Si restricciones es "false", el volumen es la hiperesfera en R6
    """

    # Para que este dentro del volumen tiene que estar dentro de la esfera
    # y ademas cumplir con las restricciones adicionales

    dentro = 1
    fuera = 0

    # chequeo 1 : dentro de esfera

    d = math.sqrt(
```

```
        (punto[0]-0.45)**2 +
        (punto[1]-0.5)**2 +
        (punto[2]-0.6)**2 +
        (punto[3]-0.6)**2 +
        (punto[4]-0.5)**2 +
        (punto[5]-0.45)**2
    )

    # si la distancia es mayor al radio, esta fuera
    if (d>=0.35) :
        return fuera

    if restricciones:
        # restriccion 1
        if 3*punto[0] + 7*punto[3] > 5:
            return fuera
        # restriccion 2
        if punto[2]+punto[3] > 1:
            return fuera
        # restriccion 3
        if punto[0]-punto[1]-punto[4]+punto[5] < 0:
            return fuera
    else:
        return dentro

    return dentro
# end fun punto dentro del volumen

# sortearPuntoRN(6)
```

In [10]:

```
# Implemento pseudocodigo Montecarlo

def MetodoMonteCarlo(N, FVolumen):
    """
    Implementa el pseudocodigo de MC
    N: cantidad de muestras
    FVolumen: funcion que define el volumen, devuelve 0 si el punto esta fuera, 1 si e
    """
    t0 = time.perf_counter()
    S = 0
    for j in range(0, N):
        punto = sortearPuntoRN(6)
        if FVolumen(punto):
            phi = 1
        else:
            phi = 0
        S = S + phi
    # end for
    VolR = S / N
    VarVorR = (S/N)*(1-S/N)/(N-1)
    return (VolR, VarVorR, time.perf_counter()-t0)
# end def

VolH = math.pi**3*(0.35**6)/6

(VolR, VarVolR, execTime) = MetodoMonteCarlo(10**6, lambda x: puntoDentroVolumen(x, Fa
```

Verificacion

Comparamos el volumen sin restricciones con el volumen calculado analiticamente de la hiperesfera en R6

In [12]:
```
print("Volumen hiper esfera por MMC = {:e}, Varianza = {:e}".format(VolR, VarVolR))
print(" ")
print("Volumen hiper esfera analitico = {:e}, diferencia MMC - analitico = {:.3f}%".fo
```

```
Volumen hiper esfera por MMC = 9.422000e-03, Varianza = 9.333235e-09

Volumen hiper esfera analitico = 9.499629e-03, diferencia MMC - analitico = 0.824%
```

Con un millon de muestras tenemos una diferencia de menos de 1% entre el volumen calculado de forma analitica y el volumen calculado por Montecarlo.

Ejecucion para diferentes tamanos de muestra

En esta seccion corremos MMC para calcular el volumen con restricciones para diferentes tamanos de muestra.

In [13]:
```
table = [ ['N', 'Vol hiperesfera+restricciones', 'Varianza', 'Tiempo'] ]

for n in [2, 4, 6]:
    (VolR, VarVolR, execTime) = MetodoMonteCarlo(10**n, lambda x: puntoDentroVolumen(x
    table.append( [10**n, VolR, VarVolR, execTime] )


tabulate.tabulate(table, tablefmt='html')
```

Out[13]:

N	Vol hiperesfera+restricciones	Varianza	Tiempo
100	0.0	0.0	0.0005960999988019466
10000	0.0003	2.999399939993999e-08	0.09121260000392795
1000000	0.000277	2.769235479235479e-10	6.50325769999472

Entre las corridas de 10mil y 1millon de muestras hay una diferencia de un 7.6% aproximadamente. Los resultados parecen coherentes en el sentido de que al aumentar el tamano de la muestra el resultado parece tender a un valor y no parece diverger. La varianza estimada tambien decrece al aumentar el tamano de la muestra, otro resultado esperable.