Santiago Quiroz Upegui CC.: 1036678350. Grupo 1

SIMULACIÓN DE SISTEMAS - 3007331

PRÁCTICA # 1 - SIMULACIÓN DE MONTE CARLO

SIMULACION DEL JUEGO DE LA MONEDA:

Le proponen lanzar una moneda (no cargada) y debe contar el número de caras y sellos que va obteniendo. El juego se termina sólo cuando la diferencia entre caras y sellos sea tres, no interesa cual sea mayor. En ese instante usted recibe $8 por el juego, pero tiene que pagar $1 por cada lanzamiento de la moneda que usted haya hecho.

Definimos los siguientes pasos para simular esta situación:

# paso 1: lanzar una moneda (generando resultados aleatorios procedentes de una distribución uniforme)

# paso 2: contar cuantos caras y sellos

# paso 3: calcular la diferencia absoluta entre caras y sellos

# paso 4: calcular estadísticas para tomar decisión

Los pasos del 1 al 3 están representados en la siguiente función (algoritmo en Python que representa la situación):

def jugar():

    resultado\_dinero= 8

    numero\_lanzamientos = 0

    numero\_caras = 0

    numero\_sellos = 0

    while True:

        # paso 1: lanzar una moneda

        lado = "cara"

        x = random.uniform(0, 1)

        numero\_lanzamientos = numero\_lanzamientos + 1

        # paso 2: contar cuantos caras y sellos

        if x > 0.5:

            lado = "sello"

        if lado == "cara":

            numero\_caras = numero\_caras + 1

        else:

            numero\_sellos = numero\_sellos + 1

        # paso 3: calcular la diferencia absoluta entre caras y sellos

        if abs(numero\_caras-numero\_sellos) == 3:

            break

    resultado\_dinero = resultado\_dinero - numero\_lanzamientos

    return [resultado\_dinero, numero\_lanzamientos, numero\_caras, numero\_sellos], resultado\_dinero >= 0

Luego realizamos una simulación con 100 de ejemplos:

datos\_simulacion = []

numero\_ganadores = 0

numero\_juegos = 100

for i in range(0,numero\_juegos):

    resultado, gano = jugar()

    datos\_simulacion.append(resultado)

    if gano:

        numero\_ganadores = numero\_ganadores + 1

Con un análisis descriptivo estimamos de la muestra la media de esta y podemos explorar si es una buena idea participar en el juego según la muestra de datos que obtuvimos de la simulación

Para esto utilizamos el estimador insesgado de la media (el cual esta implementado en la librería Python con numpy.mean)



Y como dato adicional a analizar tenemos el total de ganadores en la simulación.

El paso 4 esta representado en el siguiente fragmento del algoritmo:

# paso 4: calcular estadisticas para tomar decision

medias = np.mean(datos\_simulacion,axis=0)

media\_pago = medias[0]

print("el numero de ganadores fue: ", numero\_ganadores," con respecto a el total de ", numero\_juegos," jugadores")

if media\_pago>0:

    print("Es factible participar en el juego pues la media de la muestra en la simulacion gano algo dinero, la media perdio: ", media\_pago)

else:

    print("no es factible participar en el juego pues la media de la muestra en la simulacion no gano dinero, la media gano: ", media\_pago)

En la ejecución de la simulación obtuvimos lo siguiente:

el número de ganadores fue: 52 con respecto al total de 100 jugadores

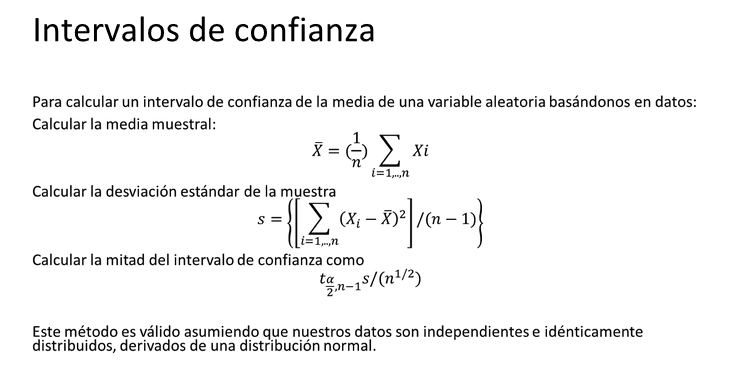
la ganancia media de pagos de la muestra fue de: -2.7

RESULTADO 1:

Con el análisis exploratorio anterior se sospecha que este juego no está muy a favor del jugador, pues, aunque hay una probabilidad de ganar, por lo general se terminaría ganando poco pero cuando se pierde se terminaría perdiendo mucho más. Por lo tanto, no es conveniente participar en este juego.

EXPERIMENTACION:

Como el estimador para la media muestral que fue utilizado es un estimador insesgado y el numero de muestras n arrojadas por la simulación es muy grande, podemos aplicar el teorema central del límite, donde podemos aproximar a una distribución normal de las medias muestrales y podemos utilizar los siguientes conceptos de intervalos de confianza:



CALCULO DE UN INTERVALO DE CONFIANZA DEL 95% PARA LA MEDIA DE LOS PAGOS: para el cálculo de este intervalo tomaremos un valor de α = 0.05 y los parámetros estimados hasta el momento son:

media estimada: -2.7

desviación estándar: 8.145550932871267

α: 0.05

calculamos el intervalo de confianza para la media con:



#calculando el intervalo de confianza

t\_tabla= t.ppf(1-(alfa/2),(numero\_juegos-1))

print("valor t con grados de libertad α/2 y n-1 ", t\_tabla)

Valor t con grados de libertad α/2 y n-1 : 1.9842169515086827

#calculando los extremos del intervalo

derecha =media\_pago+(t\_tabla)\*(desviacion\_standart\_pago/(numero\_juegos\*\*1/2))

izquierda = media\_pago-(t\_tabla)\*(desviacion\_standart\_pago/(numero\_juegos\*\*1/2))

Obtenemos este intervalo:

( -3.0232508048076108, -2.3767491951923896)

CONCLUSION:

Con esto podemos decir que la media poblacional de los pagos esta con un 95% de confiabilidad dentro de este intervalo.

Y en este caso sería mejor llamarlo perdidas, pues el intervalo siempre es negativo y por ende no es recomendable jugar, pues la media de jugadores siempre tiende a perder dinero cuando juega.