Santiago Quiroz Upegui cc: 1036678350. Grupo 1

SIMULACIÓN DE SISTEMAS - 3007331

PRÁCTICA # 2 -

Problema de la reserva de un hotel: Un hotel tiene 100 habitaciones que alquila a 125 mil la noche. Existe un costo variable entre 10.000 y $30.000 por habitación (aseo, artículos de baño, etc.) por cada noche que está ocupada. Por cada reservación aceptada, existe una probabilidad del 5%, que el cliente no llegue. Si el hotel se excede en las reservaciones, hay un costo de 200.000 para compensar a los clientes cuyas reservaciones no pueden garantizarse.

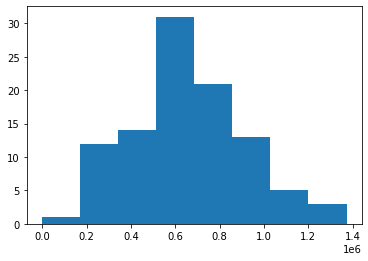
import random  
import math  
import matplotlib.pyplot as plt  
import numpy as np  
from scipy.stats import t

PASOS:

1. Definir que es temporada alta
2. Definir un rango para las variables de decision (#reservas a realizar para maximizar su utilidad diaria)
3. Simular 1000 veces la utilidad diaria del hotel con base en las variables de decision
4. declarar variables necesarias, que es utilidad(utilidad por habitacion = ingreso alquiler costos variables.... utilidad diaria... costo de indemnizacion)
5. Analizar si a raiz de una reserva el cliente llega o no ... (usar un numero aleatorio para poder asignar la probabilidad de que llegue o no)
6. Si el cliente llega y se tienen las habitaciones ocupadas -> calcular el costo de indemnizacion
7. si el cliente ocupo

El problema es un problema de maximizacion de utilidades, por lo tanto tenemos que encontrar el numeroReservacionesOptimo que maximice estas utilidades

# 1. ¿Cuántas reservas debería hacer el hotel en temporada alta si desea maximizar la utilidad diaria?

primero tomaremos el concepto de temporada alta como si todas las reservas se hacen por que hay mucha demanda

### Definimos las variables del problema:

siLlegan: Reservaciones aceptadas en donde el cliente termina llegando y hay habitaciones disponibles

siLleganPeroNoHayHabitacion: Reservaciones aceptadas en donde el cliente termina llegando y no hay habitaciones disponibles

### Nuestra funcion objetivo sera:

MAX z= costoPorNoche \* siLlegan - costoVariablePorNocheTotal - costoCompensacionReservacionesNoGarantizadas \* siLleganPeroNoHayHabitacion

### Sujeta a:

siLlegan >=0

siLlegan <=100

siLleganPeroNoHayHabitacion >= 0

## Hay un problema en este modelamiento, pues no podemos definir un buen limite del total de reservaciones dado al comportamiento aleatorio de la llegada de los clientes. Por lo anterior procederemos a estimar las medias poblacionales de las utilidades por medio de simulaciones variando el valor de las reservaciones hasta encontrar el primer valor que de un maximo de utilidades, esto lo haremos al final del documento.

# Actualizacion:

### nada de lo que intente me dio, asi que a ojo digo: el maximo ( en el mejor de los casos ) es 125 y por que?... pues por que lo digo (notece lo cansado que estoy)

### definimos una funcion que arroja las utilidades en una noche, asumiendo que no pagan antes de llegar

def noche(numeroReservaciones):  
  
 numeroHabitaciones = 100  
 costoPorNoche = 125000  
 rangoCostoVariablePorNoche = (10000,30000)  
 probabilidadClienteNoLlegue=0.05  
 costoCompensacionReservacionesNoGarantizadas=200000  
  
 costoVariablePorNocheTotal=0  
 siLlegan=0  
 siLleganPeroNoHayHabitacion=0  
 for i in range(0, numeroReservaciones):  
 # calculando si el cliente llega a la final  
 x= random.uniform(0, 1)  
 if(x<=probabilidadClienteNoLlegue):  
 if(siLlegan<numeroReservaciones):  
 siLlegan= siLlegan + 1  
 # calculando costo variable  
 costoVariablePorNoche = random.uniform(rangoCostoVariablePorNoche[0],rangoCostoVariablePorNoche[1])  
 costoVariablePorNocheTotal = costoVariablePorNocheTotal + costoVariablePorNoche  
 else:  
 siLleganPeroNoHayHabitacion = siLleganPeroNoHayHabitacion + 1  
 return (costoPorNoche\*siLlegan - costoVariablePorNocheTotal - costoCompensacionReservacionesNoGarantizadas \* siLleganPeroNoHayHabitacion)

### Simulamos 100 noches y fijamos el numero de reservaciones en 125 para tantear un valor

datosSimulacion = []  
numeroSimulaciones = 100  
numeroReservacionesOptimo=125  
utilidadesNoche=0  
for i in range(0, numeroSimulaciones):  
 utilidadesNoche=noche(numeroReservacionesOptimo)  
 datosSimulacion.append(utilidadesNoche)

# 2. Realice un histograma de las utilidades diarias y analícelo.

### calculando el numero k de intervalos(clases):

k=roundup(log2(n)+1)

k = math.ceil(math.log2(numeroSimulaciones) + 1)  
print("este es el numero de intervalos (clases)", k)

este es el numero de intervalos (clases) 8

## encontrando los extremos del intervalo a graficar

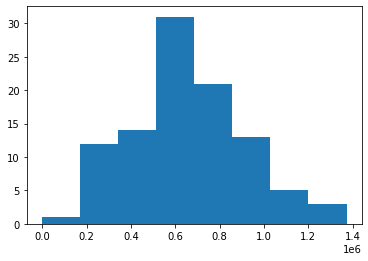
minimo = np.min(datosSimulacion)  
maximo = np.max(datosSimulacion)  
print("este es el valor minimo de la muestra: ", minimo)  
print("este es el valor maximo de la muestra: ", maximo)

este es el valor minimo de la muestra: 0.0  
este es el valor maximo de la muestra: 1372313.5631919175

### graficando un histograma de las utilidades dadas de la muestra

plt.hist(datosSimulacion , bins=k)

(array([ 1., 12., 14., 31., 21., 13., 5., 3.]),  
 array([ 0. , 171539.19539899, 343078.39079798,  
 514617.58619697, 686156.78159596, 857695.97699495,  
 1029235.17239394, 1200774.36779293, 1372313.56319192]),  
 <BarContainer object of 8 artists>)

svg

### analisis:

parece que los datos de utilidades de la muestra tienen un comportamiento aproximado a una distribución normal, lo que por el teorema central del limite podemos asumir ya que el numero de muestras es lo suficientemente grande.

# 3. Calcule estadística descriptiva a las utilidades diarias obtenidas como: mínimo, máximo, media, mediana y desviación estándar e interprételas.

#### procedemos con un analisis descriptivo de la simulacion sobre parametros como la media, el valor maximo y minimo de la utilidad

media = np.mean(datosSimulacion)  
desviacionEstandar = np.std(datosSimulacion)  
print('esta es la media estimada de la muestra: ', media, 'esta es la desviacion estandar',desviacionEstandar,', este es el valor maximo de la muestra: ', maximo, ', este es el valor minimo de la muestra: ', minimo)

esta es la media estimada de la muestra: 649880.8531489752 esta es la desviacion estandar 264095.571457473 , este es el valor maximo de la muestra: 1372313.5631919175 , este es el valor minimo de la muestra: 0.0

### analisis:

Segun el valor minimo de la muestra podemos ver que almenos en esta muestra la utilidad no llega a ser negativa utilidades fue de con este numero de reservas, en el resto de datos obtenidos no se les puede dar ningun analisis aparte de decir que tienen tal valor segun lo estimado desde la muestra. Para poder concluir algo certero de la simulacion, debemos encontrar un itervalo donde realemente encontremos el valor de los estadisticos que vamos a analizar.

# 4. Calcule un intervalo de confianza para el promedio de las utilidades diarias considerando un valor de significancia de 5%.

### CALCULO DE UN INTERVALO DE CONFIANZA DEL 95% PARA LA MEDIA DE LAS UTILIDADES: para el calculo de este intervalo tomaremos un valor de α = 0.05 y los parametos estimados hasta el momento son:

alfa=0.05  
print("media estimada: ",media)   
print("desviacion estandar:", desviacionEstandar)  
print("α: ", alfa)

media estimada: 649880.8531489752  
desviacion estandar: 264095.571457473  
α: 0.05

#calculando el intervalo de confianza  
t\_tabla= t.ppf(1-(alfa/2),(numeroSimulaciones-1))  
print("valor t con grados de libertad α/2 y n-1 ", t\_tabla)  
derecha =media+(t\_tabla)\*(desviacionEstandar/(numeroSimulaciones\*\*1/2))  
izquierda = media-(t\_tabla)\*(desviacionEstandar/(numeroSimulaciones\*\*1/2))

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

### CALCULO DE UN INTERVALO DE CONFIANZA DEL 95% PARA LA MEDIA DE LAS UTILIDADES, CON UN NUMERO DE RESERVACIONES MAXIMO DE 125:

print("(",izquierda,",",derecha,")")

( 639400.3949548893 , 660361.311343061 )

### Analisis:

Con esto podemos decir que la media poblacional de las utilidades esta con un 95% de confiabilidad dentro de este intervalo. Pero, este intervalo es para el caso donde elegimos un maximo de reservaciones maximas de 125 reservaciones, pero este valor fue elegido sin ninguna justificacion y no estamos seguros de que este sea el valor de reservaciones que maximiza las utilidades.

# Experimentacion para poder encontrar el valor optimo del numero de reservaciones

por los resultados anteriores esperamos (o sospechamos) que al iniciar el numero de reservaciones maximas igual a el maximo de habitaciones que las utilidades sean crecientes a medida de que aumentamos el numero de reservaciones hasta que las utilidades comiencen a decrecer pues llegamos a un punto donde se encuentra y por la naturaleza del problema sabemops que a partir de que la funcion comience a decrecer va a continuar decreciendo, pues seguiran y seguiran apareciendo cliente a los que se le deba pagar por que llegaron y no encontraron la habitacion que habian reservado.

## ALERTA: lo siguiente puede que tenga muchos errores, pero es la forma que encontre logica para acercarme mas al maximo

para encontrar el maximo verifico si el limite superior del nuevo intervalo de confianza es inferior al minimo del anterior intervalo de confianza

### para lograr esto de manera facil, podemos declarar el proceso que hicimos en el anterior punto como funciones reutilizables

#funcion que calcula los estimadores necesarios para el intervalo de confianza de la media y retorna una tupla (media, desviacion)  
def calcularEstimadores(datos):  
 media = np.mean(datos)  
 desviacionEstandar = np.std(datos)  
 return (media, desviacionEstandar)  
  
#funcion que retorna un intervalo de confianza para la media   
def calcularIntervaloConfianzaMedia(media, desviacion, alfa, n):  
 #calculando el intervalo de confianza  
 t\_tabla= t.ppf(1-(alfa/2),(n-1))  
 derecha =media+(t\_tabla)\*(desviacion/(n\*\*1/2))  
 izquierda = media-(t\_tabla)\*(desviacion/(n\*\*1/2))  
 return (izquierda, derecha)

### continuamos simulando hasta que encontremos el maximo

reservas = 100  
alfa=0.05  
numeroSimulaciones = 100  
intervaloAnterior = (0,0)  
datosRecorrido =[]  
while True:  
 datosSimulacion = []  
 utilidadesNoche=0  
 for i in range(0, numeroSimulaciones):  
 utilidadesNoche=noche(reservas)  
 datosSimulacion.append(utilidadesNoche)  
 media, desviacion = calcularEstimadores(datosSimulacion)  
 intervalo = calcularIntervaloConfianzaMedia(media, desviacion, alfa, numeroSimulaciones)  
  
 # calculando si comenzo a decrecer  
  
 if reservas > 100:  
 print("intervalo anterior: ",intervaloAnterior)  
 print("intervalo nuevo: ",intervalo)  
 if intervaloAnterior[0] > intervalo[1] :  
 break  
 intervaloAnterior =intervalo  
 reservas += 1  
print("el numero de reservas que maximiza la utilidad es: ", reservas -1)

intervalo anterior: (530086.8180501573, 549352.296173624)  
intervalo nuevo: (508147.0483516519, 525421.2834049802)  
el numero de reservas que maximiza la utilidad es: 100

## ESTO COMO QUE NO FUNCIONO, ASI QUE ES HORA DEL PLAN C: graficar todos los intervalos y decir a dedo cual es el que maximiza

reservas = 100  
alfa=0.05  
numeroSimulaciones = 100  
x =[]  
y = []  
for i in range(1,1000):  
 datosSimulacion = []  
 utilidadesNoche=0  
 for i in range(0, numeroSimulaciones):  
 utilidadesNoche=noche(reservas)  
 datosSimulacion.append(utilidadesNoche)  
 media, desviacion = calcularEstimadores(datosSimulacion)  
 intervalo = calcularIntervaloConfianzaMedia(media, desviacion, alfa, numeroSimulaciones)  
  
 # calculando si comenzo a decrecer  
  
 x.append(reservas)  
 y.append(intervalo[1])  
 reservas += 1  
fig, ax = plt.subplots()  
ax.plot(x, y)  
plt.savefig('grafica\_lineal\_extremos\_superiores.png')  
plt.show()

svg

svg