Inferencia estadistica

- Con simulaciones podemos calcular probabilidades de eventos complejos basado en eventos simples

- Pero, si no conocemos probabilidad de eventos simples?

- Tecnicas de inferencia estadistica nos permiten inferir/concluir las propiedades de una poblacion a partir de muestra aleatoria

Inferencia estadistica: muestra aleatoria tiende a tener las mismas propiedades que la poblacion

LEY DE GRANDES NUMEROS

EN pruebas independientes repetidas con la misma probabilidad p del resultado

la fraccion de desviaciones de p (diferencia entre ellos) converge a 0 conforme

la cantidad de pruebas se acerca al infinito

Programa estocasticos utilizan distribuciones de probabilidad.

Ley complemento

P(A) + P(-A) = 1

Ley multiplicativa

P(A y B) = P(A) \* P(B)

Let aditiva

P(A o B) = P(A) + P(B) (eventos independientes, mutuamente exclusivos)

P(A o B) = P(A) + P(B) - P(A)\*P(B) (no exclusivos)

Dado, probabilidad de sacar 5 = 1/6

Probabilidad de que saque 1 o 2 1/6 + 1/6 = 2/6 = 1/3

Probabilidd de que no saque 1 = 1 - 1/6 = 5/6

Probabilidad de no sacar 1 en 3 tiros 5/6 \* 5/6 \* 5/6 = (5/6) ^ 3

class Drogado(Borracho):

def \_\_init\_\_(self, nombre):

super().\_\_init\_\_(nombre)

def camina(self):

return (

random.choice([

(random.random(), random.random() \* -1),

(random.random() \* -1, random.random()),

(random.random() \* -1, random.random() \* -1),

(random.random(), random.random()),

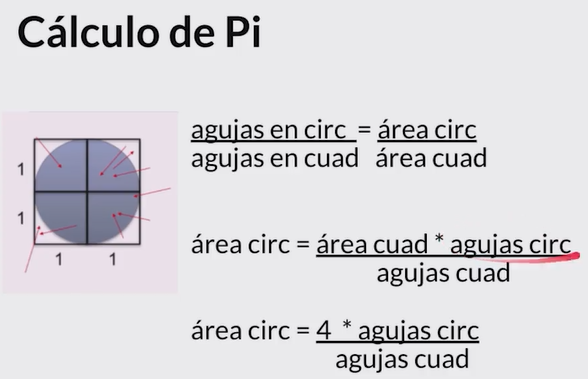
])

)

Simulacion montecarlo

- Crear simulacines para predecir

- COnvertir problemas deterministicos en estocasticos



SI la aguja esta con una hipotenusa mayor a 1, entonces esta afuera de circulo

Aventar\_agujas(num agujas) y el rango es x = -1..1 y = -1..1

Crear algoritmo que randon da x,y

Import random

X = random.random() \* random.choice([-1,1])

Y = random.random() \* random.choice([-1,1])

Si la hipotenusa es mayor a 1 entonces esta fuera de circulo

hipotenusa = math.sqr(x\*\*2 + y\*\*2)

if (hipotenusa <=1)

circ++;

else

fuera++

Se suma agujas afuera y dentro de circulo

De ahi se calcula el area del circulo

secuencia = []

dados = random.rand\_int(1,7)

dados = random.cloice([1,2,3,4,5,6])

secuencia.append(dados)

edades = [random.randint(1,35) for i in range (20)]

import numpy

speed = [86,87,88,86,87,85,86]

x = numpy.std(speed)

Teorema del limite central

- Muestras aleatorias de CUALQUIER distribucion van a tender a distribucion normal

Si sacamos muestras, y de estas la media

la distribucion de sus medias tendera a la normal

Lanzamientos de datos tienen una distribución uniforme, debido a que cada número tiene las mismas posibilidades que salir que los otros 5 => (1/6).

Entonces lo que hice fue crear un script donde se realiza una cantidad de lanzamientos de dados y se obtenga la cantidad de veces que salio cada número, luego seleccionar muestras de 10 lanzamientos y hallar su media.  
Con 1000 lanzamientos:

Si tomamos muestras aleatorias y las promediamos, es como lógico pensar que lo mas típico sea que el promedio sea igual a 3,5 (promedio entre valores tipicos del dado: 1,2,3,4,5,6). Luego, obvio van a haber casos “raros” donde los valores que se tomen den como promedio 2 o 5 (como casos extremos) y estos serán menos probables. Y bueno lo interesante es que distribuye normal…

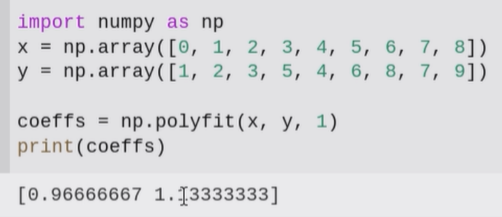
DATOS EXPERIMENTALES

- COMIENZA CON TEORIA/HIPOTESIS

- LUEGO SE CREA EXPERIMENTO

Algunas librerías de visualización de Python:

* [Bokeh](https://bokeh.pydata.org/en/latest/)
* [Matplotlib](https://matplotlib.org/)
* [Seaborn](http://web.stanford.edu/~mwaskom/software/seaborn/index.html)
* [plotly](https://plot.ly/python/)
* [Altair](https://altair-viz.github.io/)



Coeficientes lineales

Y = 1.3333X + 0.96666

