INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE FISICA Y MATEMÁTICAS

INTRODUCCIÓN A LOS MODELOS PyMC

SIMULACIÓN II

Nombre: Araceli González Mejía

Profesor: Ricardo Medel Esquivel

Grupo: 8MM1

Fecha: 04/09/2024

INTRODUCCION A PyMC3

El objetivo del video es demostrar la diferencia de la estadística frecuentista con la estadística bayesiana, para ello se aplicó un ejemplo donde se demostró como esta última nos permite utilizar herramientas como PyMC para generar modelos de manera sencilla en casos donde la parte matemática es muy rigurosa.

La estadística frecuentista es una rama de la estadística que se basa en la idea de que la probabilidad de un evento puede ser interpretada como la frecuencia relativa con la que dicho evento ocurre en un número grande de repeticiones o experimentos. O bien, de manera más sencilla, esta estadística piensa en la probabilidad como la frecuencia con la que ocurre un evento.

Algunos conceptos clave en la estadística frecuentista incluyen

1. Población y muestra
2. Estimación puntual y por intervalos
3. Pruebas de hipótesis
4. Disminución de probabilidad

Las características clave de esta probabilidad son las siguientes:

- La probabilidad está relacionada con la frecuencia de ocurrencia a largo plazo de los eventos.
- Se centra en muestras aleatorias extraídas de una población.
- No considera la probabilidad de hipótesis o modelos, solo se enfoca en los datos observados.

Por otra parte, la estadística bayesiana es una rama de la estadística que se basa en el teorema de Bayes para actualizar la probabilidad de una hipótesis a medida que se dispone de nueva evidencia o información. En palabras más simples, la estadística bayesiana piensa en la probabilidad como una forma de medir la incertidumbre.

Como su nombre lo dice, esta probabilidad utiliza como herramienta principal el teorema de Bayes.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

La cual, a manera de estimación, nos dice donde se concentran la mayoría de los datos sobre la ocurrencia de un evento.

Donde

- P(A|B) es la **probabilidad posterior**, es decir, la probabilidad de la hipótesis A dado que se han observado los datos B.
- P(B|A) es la **verosimilitud** (o likelihood), que representa la probabilidad de observar los datos B dado que la hipótesis A es verdadera.

- P(A) es la **probabilidad a priori** de la hipótesis A, que refleja nuestra creencia inicial antes de observar los datos.
- P(B) es la **probabilidad marginal** de los datos, que puede considerarse como un factor de normalización y se calcula como la suma ponderada de todas las probabilidades condicionales posibles para A.

Esta ecuación se puede ver de la siguiente manera

$$P(Parametro|Datos)\alpha P(Datos|Parametros)P(Parametros)$$

Donde, en la primera parte se describe lo que queremos calcular. Para ello, debemos aplicar los conocimientos sobre nuestro parámetro dado en análisis de los datos que obtuvimos.

Para $P(Datos|Parametros)$ también conocido como Likelihood nos responde la pregunta ¿cuál es la probabilidad de ver los datos que vimos distintos casos hipotéticos?

Y finalmente $P(Parametros)$ nos responde la pregunta de ¿cuál es la probabilidad de que sucedan los conocimientos anteriores?

Como mencionamos anteriormente, la podemos utilizar herramientas como PyMC donde la probabilidad bayesiana es más complicada, para la formula anterior $P(Datos|Parametros)P(Parametros)$ este término suele ser difícil de resolver matemáticamente.

Ahora, para aplicar la probabilidad bayesiana es necesario conocer los siguientes dos aspectos:

1. El modelo que genera los datos.
2. La información a priori de lo que queremos estimar.

Para aprender como es que esta herramienta funciona, en el video se nos plantea un ejemplo sobre la pandemia, en donde se dice que el gobierno quiere saber cuántas personas se han contagiado de COVID en Santiago, Chile. Esto con el objetivo de que el gobierno implemente una estrategia de inmunidad de rebaño.

La inmunidad de rebaño también conocida como **inmunidad colectiva**, es un fenómeno en el que una población se vuelve resistente a la propagación de una enfermedad infecciosa cuando una proporción significativa de sus miembros se ha vuelto inmune, ya sea por haber superado la infección o por haber sido vacunados.

Para aplicar este ejemplo, lo primero que debemos hacer es aleatorizar, para ello, dado que el ejemplo nos habla sobre si una persona esta contagiada o no, se supone que sigue una distribución binomial, ya que únicamente tiene dos opciones. Además, esta distribución es una de las más utilizadas en estadística. Para verlo de manera más apropiada podemos definirla como:

- Hay N “experimentos” que resultan en un éxito (un 1) o un fracaso (un 0) con probabilidad p.

Para este ejemplo se tomó una muestra aleatoria de 50 personas en Santiago, donde de 50 pruebas, 40 resultaron ser negativas. En la siguiente imagen se muestran en verde las muestras que dieron positivo y en rojo las que fueron negativo.

Otro paso importante es incluir la tasa de incertidumbre, la que en algunos casos se tiene como dato, pero cuando no es así, se realiza un análisis independiente.

En este ejemplo se construyeron 3 modelos:

1. Modelo 1: asume que la prueba es perfecta.
2. Modelo 2: Se añade el caso en el que la prueba a veces da falsos positivos
3. Modelo 3: Se incluye la incertidumbre sobre la tasa de los falsos positivos.

MODELO 1

Para este modelo se asume que la prueba es perfecta, lo cual es un caso muy ideal y, por lo tanto, a simple vista, este podría ser el modelo que peores estimaciones nos devuelva.

Para aplicar estos modelos con PyMC es importante realizar dos pasos, el primero es definir los datos de la variable que deseamos estimar, es este caso, la probabilidad, por lo que podemos definir estas variables como una distribución uniforme, ya que esta esta definida de 0 a 1. Este podría ser un paso no informativo, dado que muchas veces buscamos que el modelo se alimente de los datos.

```
import pymc3 as pm
import arviz as az
```

Importamos las librerías

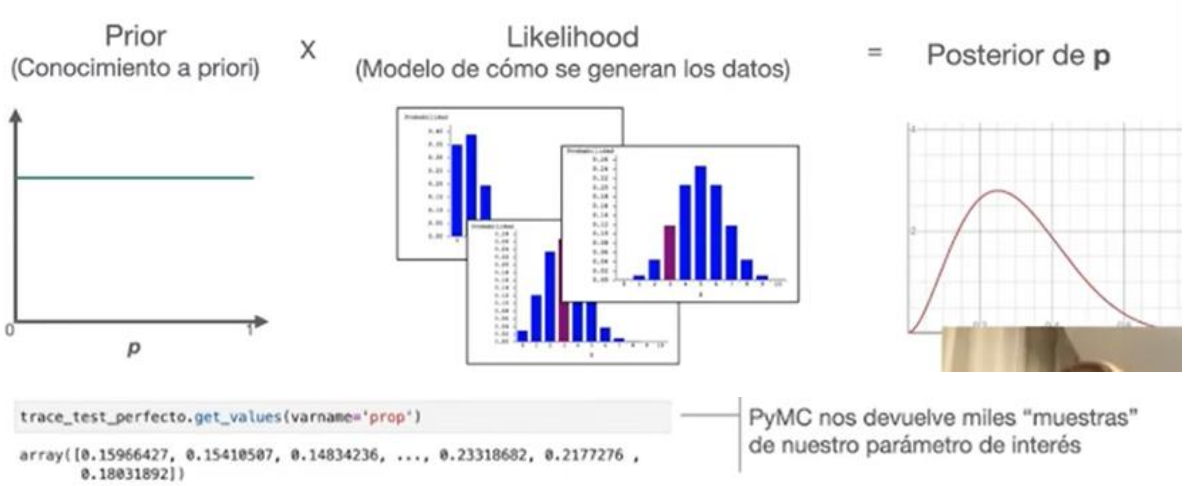
```
test_totales = 50
test_positivos = 10
```

Definimos los datos

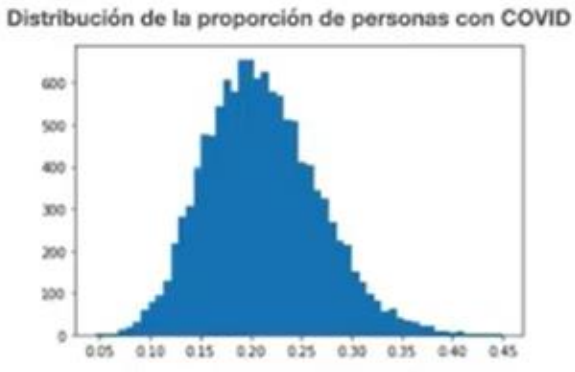
```
with pm.Model() as modelo_test_perfecto:
  prob = pm.Uniform(name = 'prob',
                    loer = 0,
                    upper = 1)
  casos_positivos = pm.Binomial(name = 'casos_positivos',
                                n = test_totales,
                                p = prob,
                                observed = test_positivos)
  trace_test_perfecto = pm.sample(3000)
```

Definimos la variable que se quiere estimar y dar una probabilidad a priori

Una forma de ver la formula antes mencionada, ya aplicando los conceptos es la siguiente:



Una vez ejecutado el programa se obtiene la siguiente gráfica:



MODELO 2

En el siguiente modelo consideramos las pruebas que nos dan falsos positivos, por lo que será importante aclarar que la proporción de pruebas positivas no son iguales que la proporción de personas con COVID.

Se ve que la probabilidad está afectada por dos factores importantes

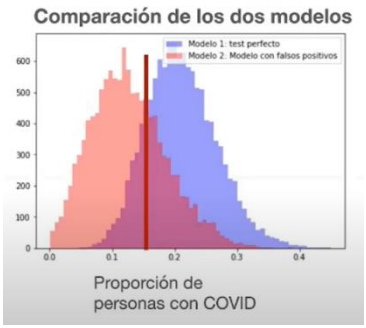
1. La probabilidad de tener COVID
2. La probabilidad de un falso positivo

Para la aplicación de este modelo en Python se genera el siguiente código, donde tomaremos en cuenta los falsos positivos dentro de los parámetros de la distribución que elegimos.

```
with pm.Model() as modelo_con_fp:
    prob_cov = pm.Uniform(name='prob_cov',
                          lower=0,
                          upper=1)
    prob_fp = 0.1
    prob_test_positivo = prob_cov + (1-prob_cov)*prob_fp
    casos_positivos = pm.Binomial(name='casos_positivos',
                                  p=prob_test_positivo,
                                  n=tests_totales,
                                  observed=tests_positivos)
    modelo_con_fp = pm.sample(3000)
```

Modelo tomando en cuenta falsos positivos

Del modelo anterior obtenemos la gráfica siguiente donde hace la comparación de la estimación de los modelos. En este segundo modelo se presenta una mejora ya que se alimentó de los datos y se aprendió que existen falsos positivos, por lo que, en la gráfica, al existir un desplazamiento se dice que en realidad hay menos casos positivos.



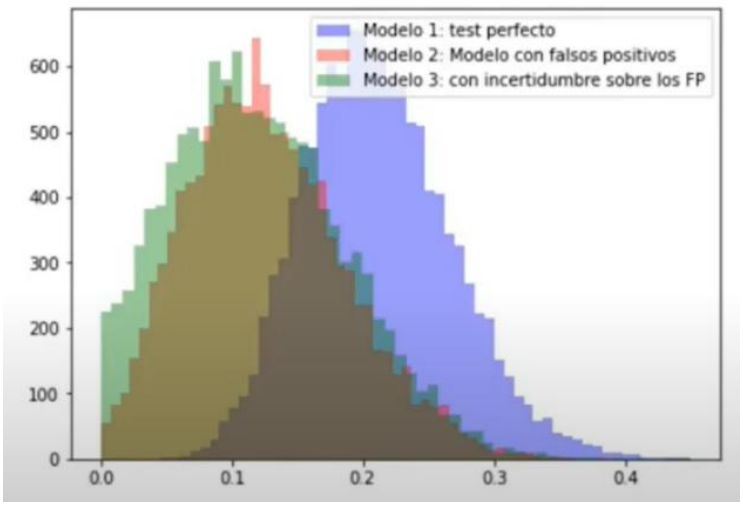
MODELO 2

```
lab_fp_observados = 10
lab_tests_hechos = 100

with pm.Model() as modelo_con_incertidumbre:
    # Modelo para estimar la tasa de falsos positivos
    prob_fp = pm.Uniform(name='prob_fp',
                         lower=0,
                         upper=1)

    test_de_falsos_positivos = pm.Binomial(name='test_de_falsos_positivos',
                                           p=prob_fp,
                                           n=lab_tests_hechos,
                                           observed=lab_fp_observados)

    # Modelo para calcular la proporción de personas con COVID
    prob_cov = pm.Uniform(name='prob_cov',
                          lower=0,
                          upper=1)
    prob_test_positivo = prob_cov + (1-prob_cov)*prob_fp
    casos_positivos = pm.Binomial(name='casos_positivos',
                                  p=prob_test_positivo,
                                  n=tests_totales,
                                  observed=tests_positivos)
    trace_modelo_con_incertidumbre = pm.sample(3000)
```



Para el tercer modelo se debe incluir la incertidumbre sobre la tasa de los falsos positivos, pero para ello fue necesario estimar este parámetro, por lo que se realizó el procedimiento anterior con una distribución de Bernoulli.

Para ello se alimenta el modelo con los datos de falsos positivos.

De este programa se obtienen la siguiente grafica, donde comparamos los tres modelos.

En conclusión, la estadística bayesiana junto con las herramientas de Python como PyMC nos ayudan a modelar eventos de manera sencilla, incluso aquellos cuyo cálculo matemático sea complicado, lo cual es una ventaja sobre la estadística frecuentista,