

## INTRODUCCION A PyMC

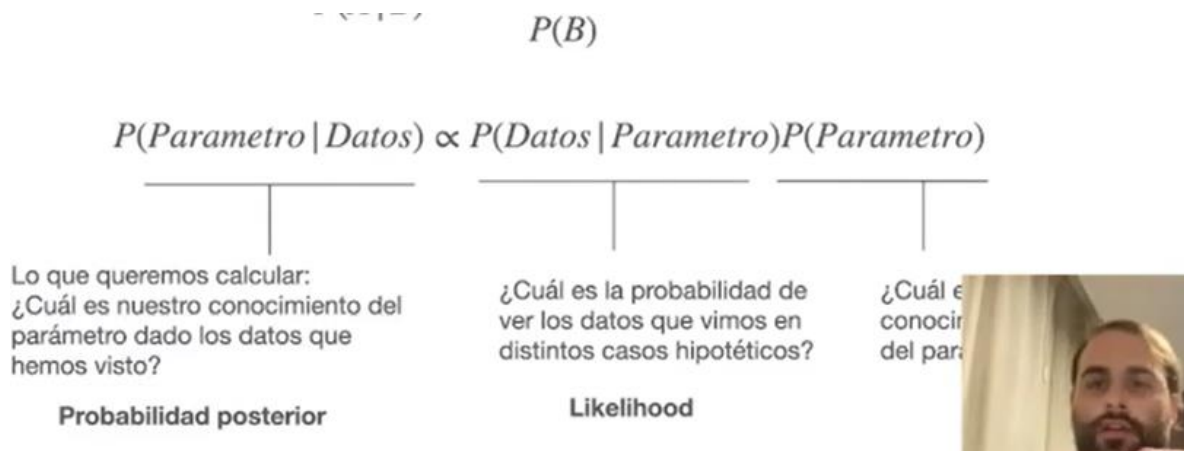
ESTADISTICA Frecuentista: se piensa a la probabilidad como la frecuencia con la que ocurre un evento.

Bayesian: se piensa en la probabilidad como una forma de medir incertidumbre.

TEOREMA DE BAYES:

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

Donde se concentran más los datos sobre la ocurrencia de un evento.



3. ¿Cuál es la probabilidad de los conocimientos anteriores?, puede ser uniforme.

Para aplicar la probabilidad bayesiana es necesario conocer dos datos:

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

$$P(\text{Parametro} | \text{Datos}) \propto \frac{P(\text{Datos} | \text{Parametro})P(\text{Parametro})}{\dots}$$

Esta parte es difícil de resolver matemáticamente

PyMC nos permite definir modelos bayesianos usando código

EJEMPLO:

Es mayo del 2020, hace unos meses empezó la pandemia... 

- Es mayo del 2020, acaba de empezar la pandemia, y **el gobierno quiere saber cuánta gente se ha contagiado de COVID** en Santiago, Chile.
- Dependiendo de este número el gobierno seguirá (o no) una estrategia de inmunidad de rebaño.
- Para esto usaremos tests que detectan anticuerpos de SARS-CoV-2.



#### 1. ALEATORIZAR

La distribución binomial es muy utilizada en estadística.

## Partamos con un recordatorio de la **distribución binomial**!

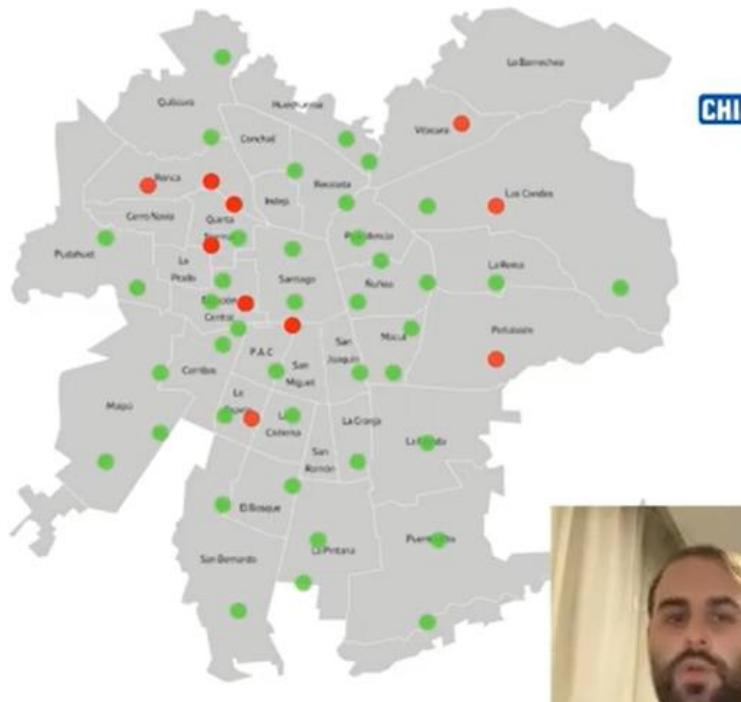
CHILE 2021

- Hay  $N$  “experimentos” que resultan en un éxito (un 1) o un fracaso (un 0) con probabilidad  $p$
- Por ejemplo, si **tiramos una moneda 10 veces** que da **cara con probabilidad 0.5** el modelo binomial nos dice lo siguiente



Tomaremos una **muestra aleatoria de 50 personas** en Santiago

De los 50 tests:  
- **40 son negativas**  
(nuestro test no detecta anticuerpos)

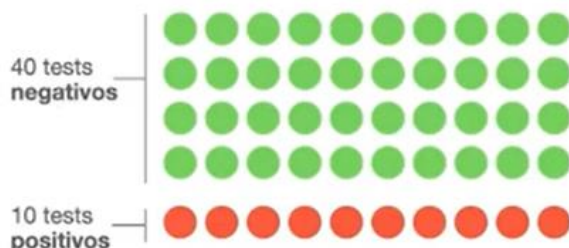


3- Tasa de incertidumbre

# Construiremos 3 modelos

CHILE

- **Modelo 1:** asume que el test es perfecto
- **Modelo 2:** incluiremos que el test a veces da falsos positivos
- **Modelo 3:** incluiremos la incertidumbre sobre la tasa de falsos positivos



## Paso 1: Asumiremos que el test es perfecto

CHILE 21

Un diagrama que muestra los resultados de 50 tests. Hay una fila superior de 40 círculos verdes, etiquetada como '40 tests negativos'. Debajo de ella hay una fila de 10 círculos rojos, etiquetada como '10 tests positivos'. Los círculos están agrupados en 5 columnas de 10.

¿Cuánta gente tuvo COVID-19?

20% (10 de 50)

¿Qué tan seguros estamos de este resultado?

Mmmm...  
preguntémosle a la estadística

Es necesario hacernos las siguientes preguntas:

## Creando este modelo en PyMC

CHILE 20

- Lo primero que siempre hacemos es preguntarnos, ¿cómo podemos modelar los datos?
- En este caso tenemos el clásico ejemplo de una **distribución Binomial**.
- Recordemos que en la Binomial tenemos **N** "experimentos" que pueden tener "éxito" (test positivo) o "fracaso" (test negativo) con probabilidad **p**

-Definimos los datos de la variable que deseamos estimar como una distribución uniforme porque la probabilidad esta entre cero y uno. (priori)

-Como muchas veces queremos que el modelo se alimente de los datos, no es muy importante, no es informativo

- 2. Como se generan los datos, para este caso es una distribución binomial.

```
import pymc3 as pm
import arviz as az

tests_totales = 50
tests_positivos = 10

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

Importamos pymc y Arviz


Definimos nuestros datos

Definimos la variable que queremos estimar y le damos un **probabilidad a priori**

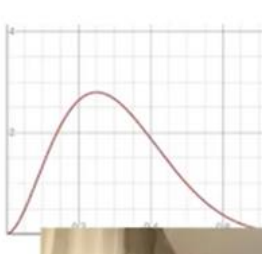
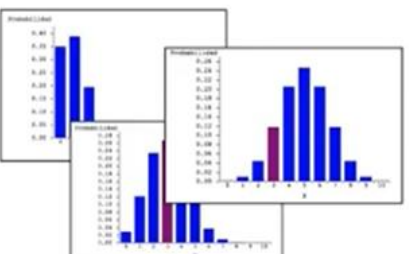
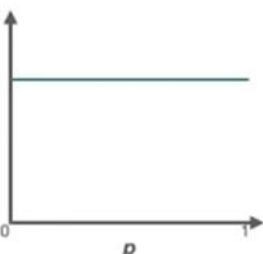
Nuestra creencia, antes de ver los datos, de  $p$

Definimos la distribución de los datos (Likelihood)

Magia de PyMC



Prior (Conocimiento a priori)  $\times$  Likelihood (Modelo de cómo se generan los datos) = Posterior de  $p$



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

Los modelos bayesianos muchas veces no se pueden resolver con una formula matemática.

CHIL

pm.sample() hace toda la matemática por nosotros!

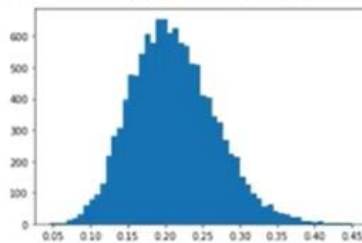
```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [prog]
100.00% [16000/16000 02:02<00:00 Sampling 4 chains, 0 divergences]
```

```
trace_test_perfecto.get_values(varname='prob')
```

```
array([0.15966427, 0.15410507, 0.14834236, ..., 0.23318682, 0.2177276 ,
       0.18031892])
```

PyMC nos devuelve miles "muestras" de nuestro parámetro de interés

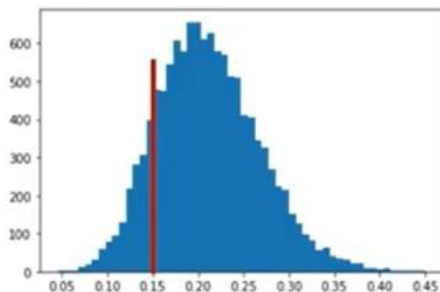
Distribución de la proporción de personas con COVID



## Las muestras son muy poderosas

CHILE 2

Distribución de la proporción de personas con COVID



Ministra de Salud: ¿Cuál es la probabilidad de que menos de 15% de la población se haya infectado?

Bayesiano: Deme un segundo...

```
muestras_prop = trace_test_perfecto.get_values(varname='prob')
len(muestras_prop[muestras_prop<0.15])/len(muestras_prop)
```

```
0.13433333333333333
```



Si fuese probabilidad frecuentista no se podría realizar.



## En un modelo bayesiano esta información es fácil de incorporar

- Ahora le diremos al modelo que la proporción de tests positivos no es lo mismo que la proporción de gente con COVID
- Ahora la probabilidad de un test positivo es afectada por dos factores:
  - La probabilidad de tener COVID (lo llamaremos **prob\_cov**)
  - La probabilidad de un falso positivo (lo llamaremos **prob\_fp**)
- **$\text{prob\_test\_positivo} = \text{prob\_cov} + (1 - \text{prob\_cov}) * \text{prob\_fp}$**



## El nuevo modelo

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

Modelo test  
perfecto



```
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)
```

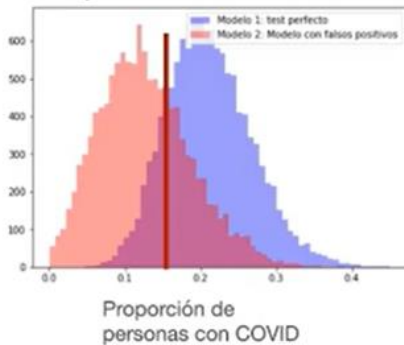
Modelo  
tomando  
en cuenta  
falsos  
positivos

Mejoro el modelo, ya que aprendió que existen falsos positivos, por lo que en la grafica se explica que en realidad hay menos casos de covid.

## El nuevo modelo

CHILE 2021

Comparación de los dos modelos



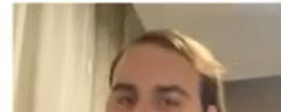
Ministra de Salud: ¿Cuál es la probabilidad de que menos de 15% de la población se haya infectado?

Bayesiano: Deme un segundo...

```
muestras_prop = modelo_con_fp.get_values(varname='prob_cov')  
len(muestras_prop[muestras_prop<0.15])/len(muestras_prop)
```

0.6703333333333333

Bayesiano: 67%!



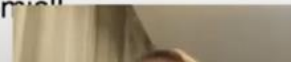
PASO: NO SABEMOS CUAL ES LA TASA DE FALSO POSITIVO.

Pero no sabemos exactamente  cuál es la tasa de falsos positivos: ...

CHILE 2021

El laboratorio que creó el test hizo 100 pruebas y nos informa que la tasa de falsos positivos es de 10%

- O sea, hay incertidumbre acerca de la verdadera tasa de falsos positivos.
- Podemos pensar que la tasa de falsos positivos también es un parámetro que tenemos que estimar
- ¿Qué distribución usamos? También podemos usar la Binomial





## Modelo 3: agregamos la incertidumbre a los falsos positivos

CHILE 20

```
lab_fp_observados = 10
lab_tests_hechos = 100

with pm.Model() as modelo_con_incetidumbre:
    # 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)

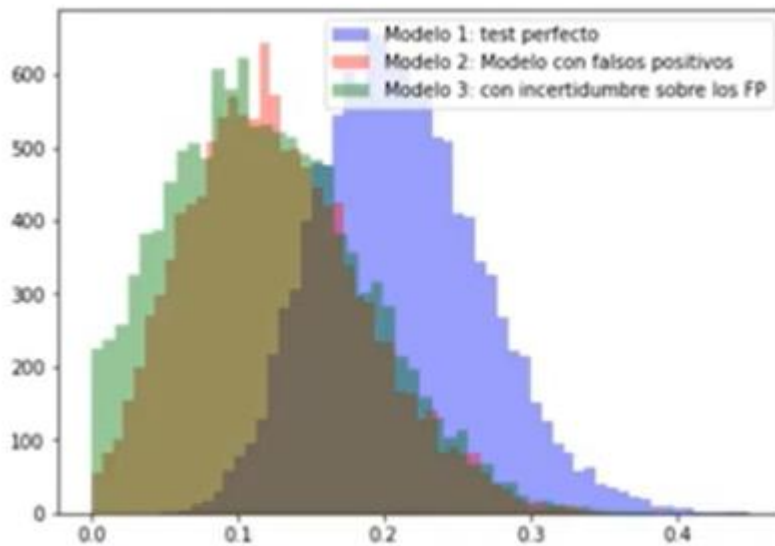
    prob_test_positivo = prob_cov + (1-prob_cov)*prob_fp
    casos_positivos = pm.Binomial(name='casos_positivos',
                                  p=prob_test_positivo,
                                  n=tests_totales,
                                  observed=tests_positivos)

    trace_modelo_con_incetidumbre = pm.sample(3000)
```

Ahora también modelamos la tasa de falso positivos con una distribución Binomial



Lo modelamos pero con los datos de falsos positivos.



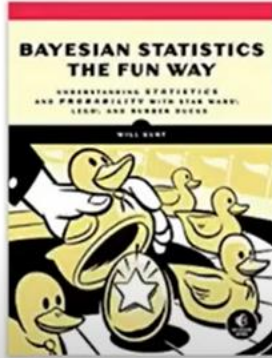
## Otras extensiones

- Agregar falsos negativos (buen ejercicio!)
- Nosotros usamos los datos del censo en Etiopía para mejorar nuestros estimados de la población (post estratificación)
- En Brazil le hicimos 2 tests de anticuerpos a cada persona, y eso también se puede incluir en el modelo

# Materiales recomendados



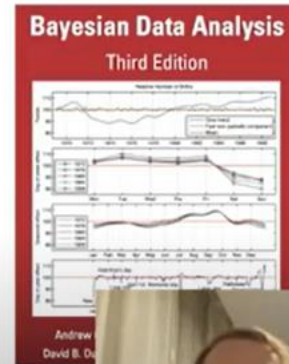
CHILE 202



Muy poca matemática, muy bueno para partir de cero



Ejemplos aplicados en Python



La biblia del cálculo, t  
av

ye4y