Estadística Bayesiana y Programación Probabilística

O cómo dejé de preocuparme y aprendí a amar la incertidumbre

Adolfo Martínez

2017/05/31

Reducir Incertidumbre

Históricamente:

- Oráculos
- Religión
- Empirismo
- Método Científico
- Análisis de Datos

Estadística Frecuentista

Estadística Frecuentista

Históricamente desarrollada a partir del estudio de **frecuencias**. Toma éstas como medida objetiva de la realidad y aproximación de la **probabilidad** "real"

- Prueba (contraste) de hipótesis
- Diseño de experimentos
- Predicción

Estadística Frecuentista

Algunos problemas

Probabilidad de eventos únicos y cantidades fijas pero desconocidas

By Sunny

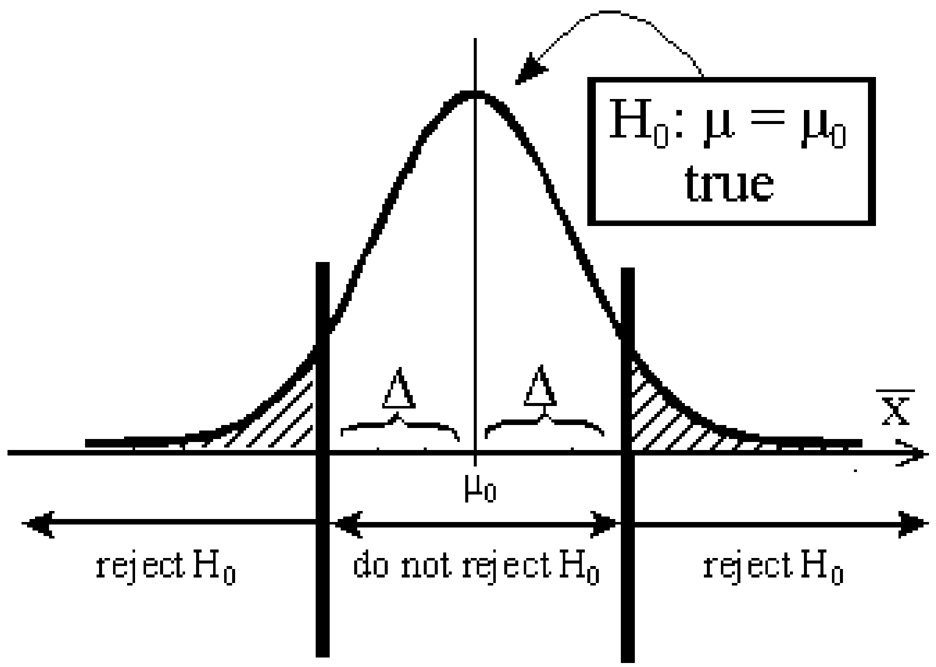
How people estimate the probabilities of unique events

Estadística Frecuentista

Algunos problemas

Técnicas con fundamento teórico débil (*e.g.* valores p)

- Juzga H basado en $P(D|H)$



Machine Learning

Machine Learning

Estudio y construcción de algoritmos que **aprenden** y realizan **predicciones** basados en datos

- Predicción
- Clustering
- Minado de Reglas

Machine Learning

Algunos problemas

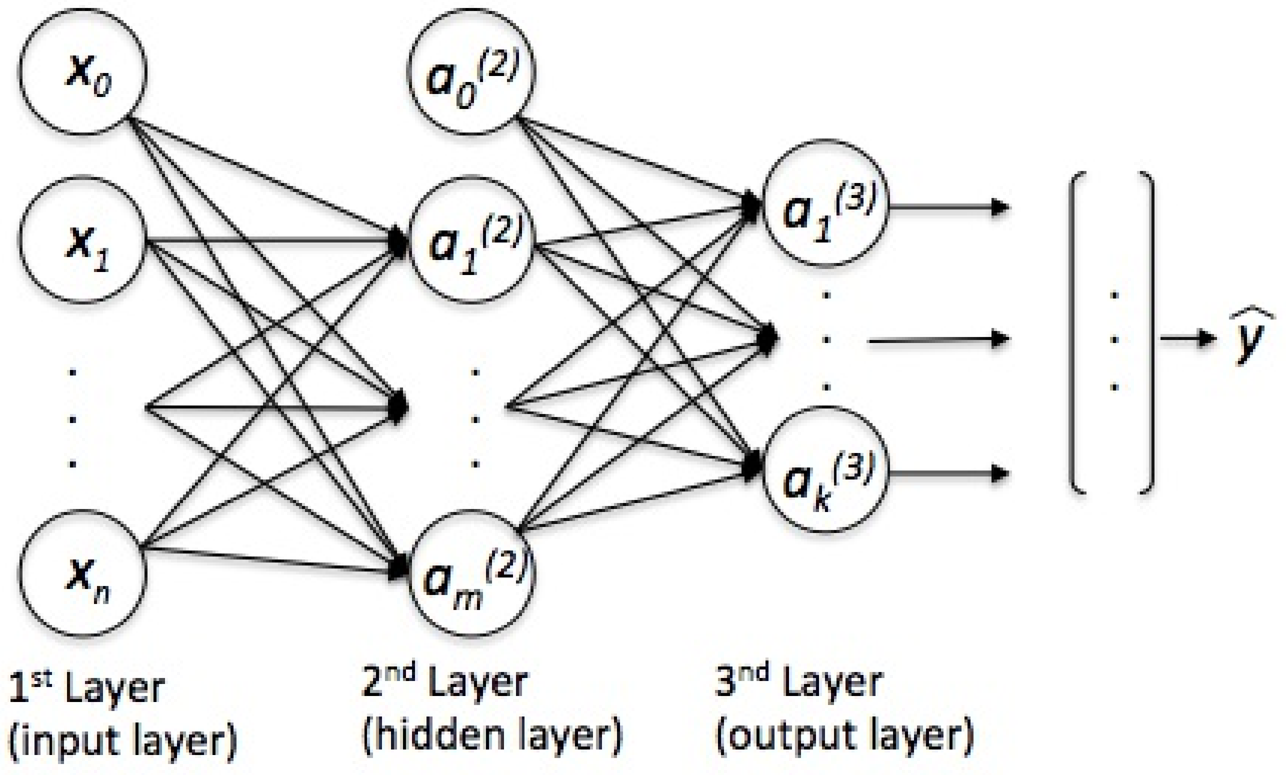
Responden una pregunta específica

- Por ejemplo, estimar una respuesta y dados los datos X, ***i.e.*** $y = \hat{f}(x)$
- Bajo pérdida cuadrática: $\hat{f}(x) = E[Y|X = x]$
- $P(Y|X = x) > \alpha$
- En muchos algoritmos, no hay una respuesta inmediata a esta pregunta

Machine Learning

Algunos problemas

Dificultad en la interpretación

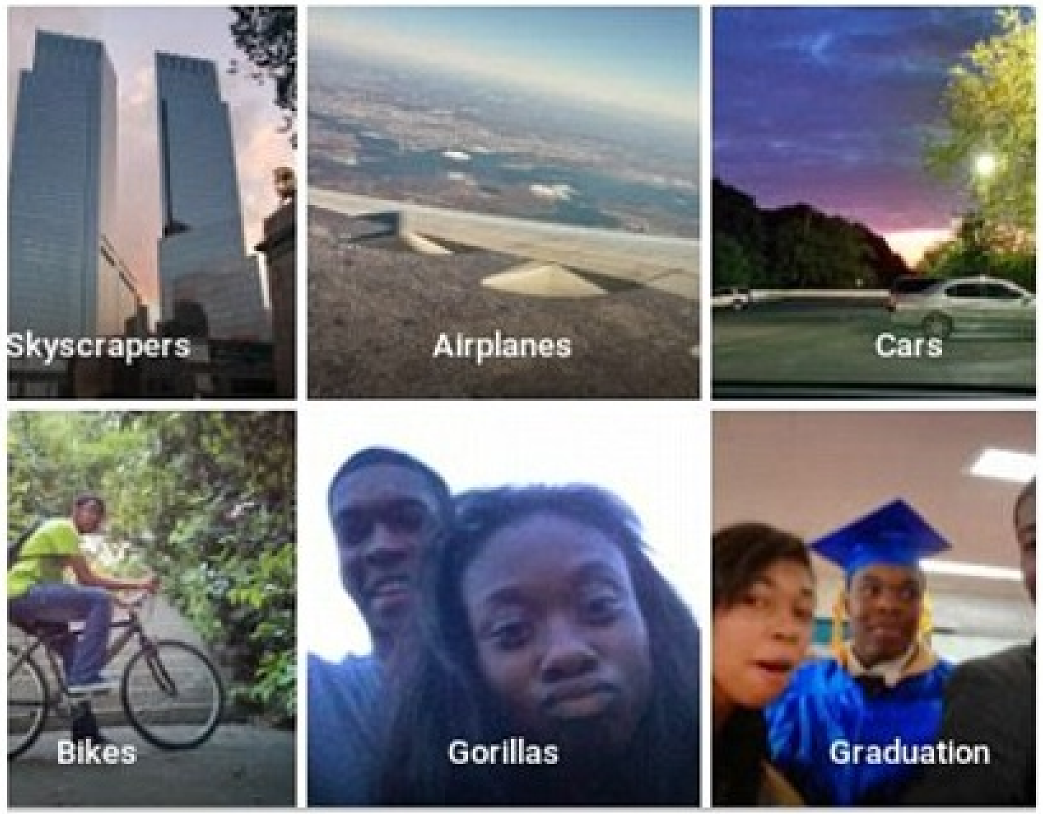


Schematic of a multi-layer perceptron.

Machine Learning

Algunos problemas

Sobrecertidumbre



Estadística Bayesiana

Estadística Bayesiana

Incorpora la **incertidumbre** subjetiva o **información incial** como información previa a la inferencia. La **probabilidad** es una medida de incertidumbre, no una frecuencia

- Prueba (contraste) de hipótesis
- Diseño de experimentos
- Predicción

Estadística Bayesiana

vs. Frecuentismo

Tiene una manera clara y bien fundamentada de asignar probabilidad a eventos únicos o cantidades desconocidas

`

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

`

- $P(\theta)$ se obtiene de la información inicial (*a priori*)
- De no existir, se pueden usar *a priori* no informativas

Estadística Bayesiana

vs. Frecuentismo

Tiene una manera clara y bien fundamentada de calcular la probabilidad de una hipótesis

- Juzga H basado en $P(H|D)$

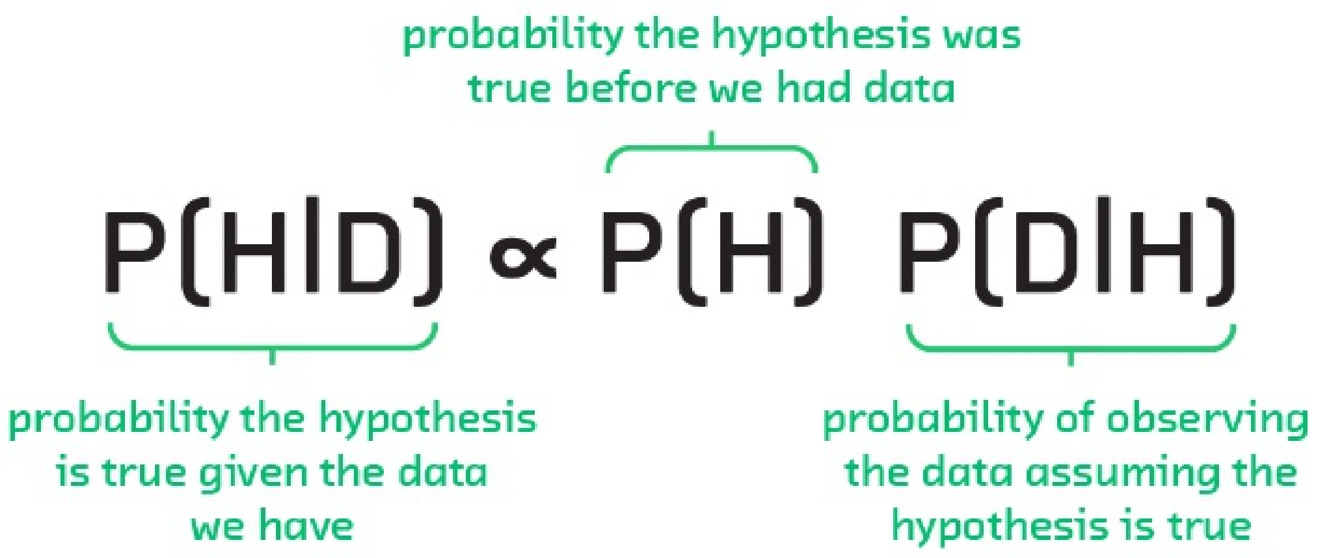
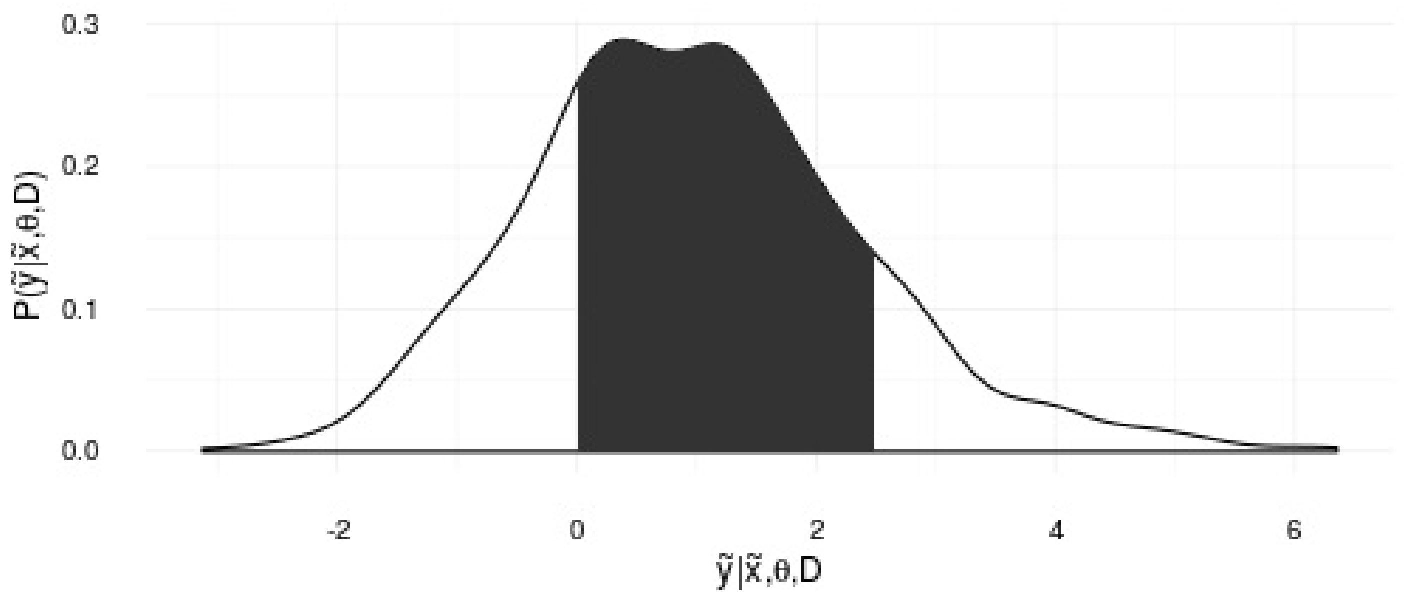


Imagen tomada de Fast Forward Labs #5

Estadística Bayesiana

vs. Machine Learning

Puede resolver una gran cantidad de preguntas acerca de la variable de respuesta



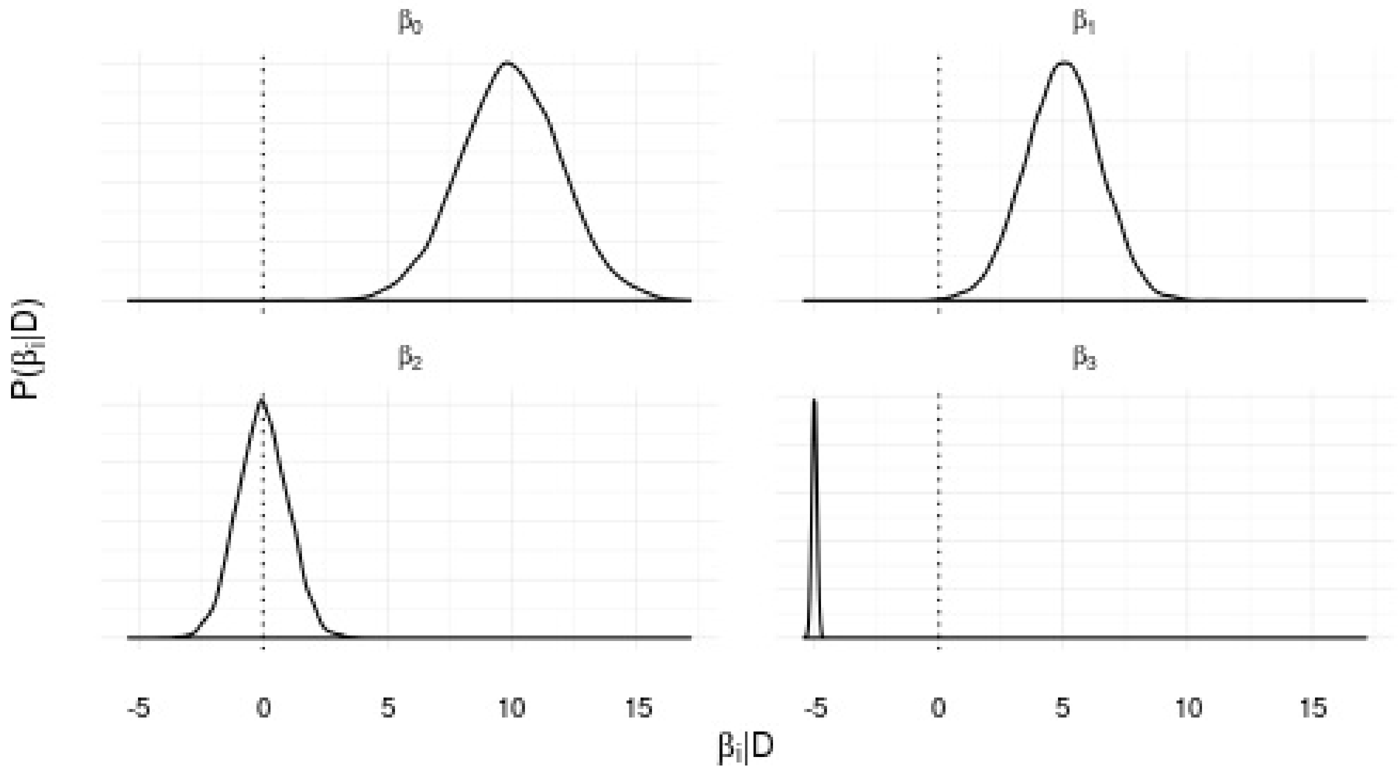
El área sombreada representa $P(0 < \tilde{y} < 2.5|\tilde{x}, X, \theta)$

Estadística Bayesiana

vs. Machine Learning

Como la Frecuentista, los modelos básicos son fácilmente interpretables

- Por ejemplo, en un modelo lineal bayesiano, las posteriores de los coeficientes β_i , describen la incertidumbre acerca de su valor



Estadística Bayesiana

vs. Machine Learning

La incertidumbre se conserva en cada paso de la inferencia

- Por ejemplo, terminada la inferencia podemos calcular la probabilidad de que los parámetros sean cercanos a 0
- O bien, además del estimado puntual, calcular un intervalo de **probabilidad** para Y
- Conservar y conocer esta incertidumbre nos permite tomar mejores decisiones al predecir (por ejemplo, elegir no hacerlo)

Estadística Bayesiana

Algunos problemas

Cálculo de la posterior - Dificultad Matemática

- Para calcular la posterior de manera exacta, necesitamos resolver: $P(D) = \int_\theta P(D|\theta)P(\theta)d\theta$
- La distribución predictiva se obtiene a través de una integral similar: $P(\tilde{y}|\tilde{x},\theta,D) = \int_\theta P(\tilde{y}|\tilde{x},\theta)P(\theta|D)d\theta$
- Estos problemas no siempre tienen una solución analítica

Estadística Bayesiana

Algunos problemas

Cálculo de la posterior - Dificultad Computacional

- Las integrales mencionadas anteriormente se pueden calcular de manera numérica
- Los métodos más populares para esto son los de **Markov Chain Monte Carlo (MCMC)**
- Aunque estos métodos típicamente requieren ajuste para lograr una convergencia rápida, avances recientes hacen esto innecesario.

Programación Probabilística

Programación Probabilística

Una manera declarativa y sencilla de definir modelos jerárquicos Bayesianos, sin necesidad de resolver el problema matemático específico para cada modelo

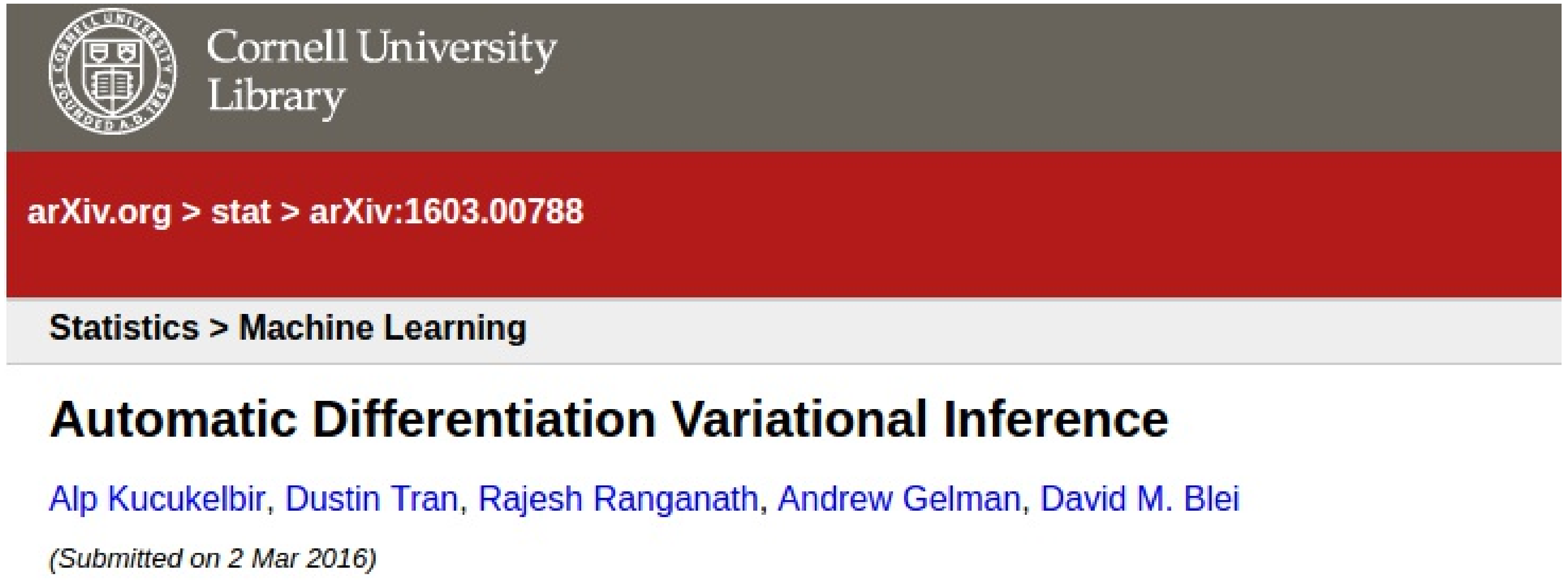
```python
import pymc3 as pm

with pm.Model() as model:
    # hyper-parameters
    alpha = 1/data.mean()

    # parameters
    tau = pm.DiscreteUniform("tau", lower=0, upper=n_obs)
    lambda1 = pm.Exponential("lambda1", alpha)
    lambda2 = pm.Exponential("lambda2", alpha)
    lambda_i = pm.math.switch(tau >= idx, lambda1, lambda2)

    # observations
    obs = pm.Poisson("obs", lambda_i, observed=data)
```

Programación Probabilística

Inferencia

Incluye los métodos computacionales más avanzados para el paso inferencial

```python
with model:
    # Do Inference
    step = pm.advi()
    trace = pm.sample(num_samples, step = step)
```



Programación Probabilística

PyMC3

```
model = pm.Model()
with model:
    # Create Model
    p_coef = Cauchy(0, 2.5)
    repaid = Bernoulli(logit(X * p_coef),
                       observed=logit(repaid_actual))

    # Do Inference
    start = pm.find_MAP()
    step = pm.NUTS()
    trace = pm.sample(num_samples, step, start,
                      progressbar=True)
```

Programación Probabilística

Stan

```
data {
    int<lower=0> N;
    int<lower=0> N_features;
    matrix[N, N_features] X;
    int<lower=0,upper=1> repaid[N];
}
parameters {
    vector[N_features] p_coef;
}
model {
    vector[N] p;
    p_coef ~ cauchy(0, 2.5);
    p = logit(X * p_coef);
    repaid ~ bernoulli(p);
}
```

Programación Probabilística

Anglican

```
;; Create Model
(defquery gaussian-model [data]
  (let [mu (sample (normal 1 (sqrt 5)))
        sigma (sqrt 2)]
    (doall (map (fn [x] (observe (normal mu sigma) x)) data))
    mu))

;; Do Inference
(def posterior
  ((conditional gaussian-model :smc :number-of-particles 10) dataset))

(def posterior-samples (repeatedly 20000 #(sample* posterior)))
```

¿Cómo Modelar?

- Tener una o varias preguntas
- Pensar cómo se pudieron haber **generado** los datos
- Escoger **distribuciones** que representen dichos datos
- Modelar **relaciones** entre variables (*e.g.* linealmente, árbol de decisión)
- Modelar parámetros con distribuciones que representen la **informacion inicial* adecuadamente

Modelo generativo de los datos

- No es necesario representar la generación de manera precisa
- No es necesario pensar en un proceso causal
- El modelo generativo debe de ser capaz de generar datos similares a los que se tienen

Ya casi viene el ejemplo...

Modelar relación entre variables

- Escoger una relación de acuerdo a la complejidad del fenómeno
- En caso de duda, escoger relaciones rígidas (al principio)
- La relación entre variables dicta la complejidad del modelo

Ejemplo: Lineal

$$y|x \sim N(\beta^T x, \sigma^2)$$

Ejemplo: Lineal con link logit

$$y|x \sim Bernoulli(sigmoid(\beta^T x))$$

$$sigmoid(x) = (1 + e^x)^{-1}$$

¿Cómo Escoger Priors?

Tres opciones:

1. Buscar representar la información inicial de la manera más precisa posible
2. Usar "no-informativas"
3. Usar "débilmente-informativas"

La elección de *prior* también afecta la complejidad del modelo, en el sentido de que afecta su flexibilidad.

Ejemplo: Coeficiente lineal

$$\beta \sim Cauchy(0, 2.5)$$

Ejemplo: Coeficiente lineal regularizado

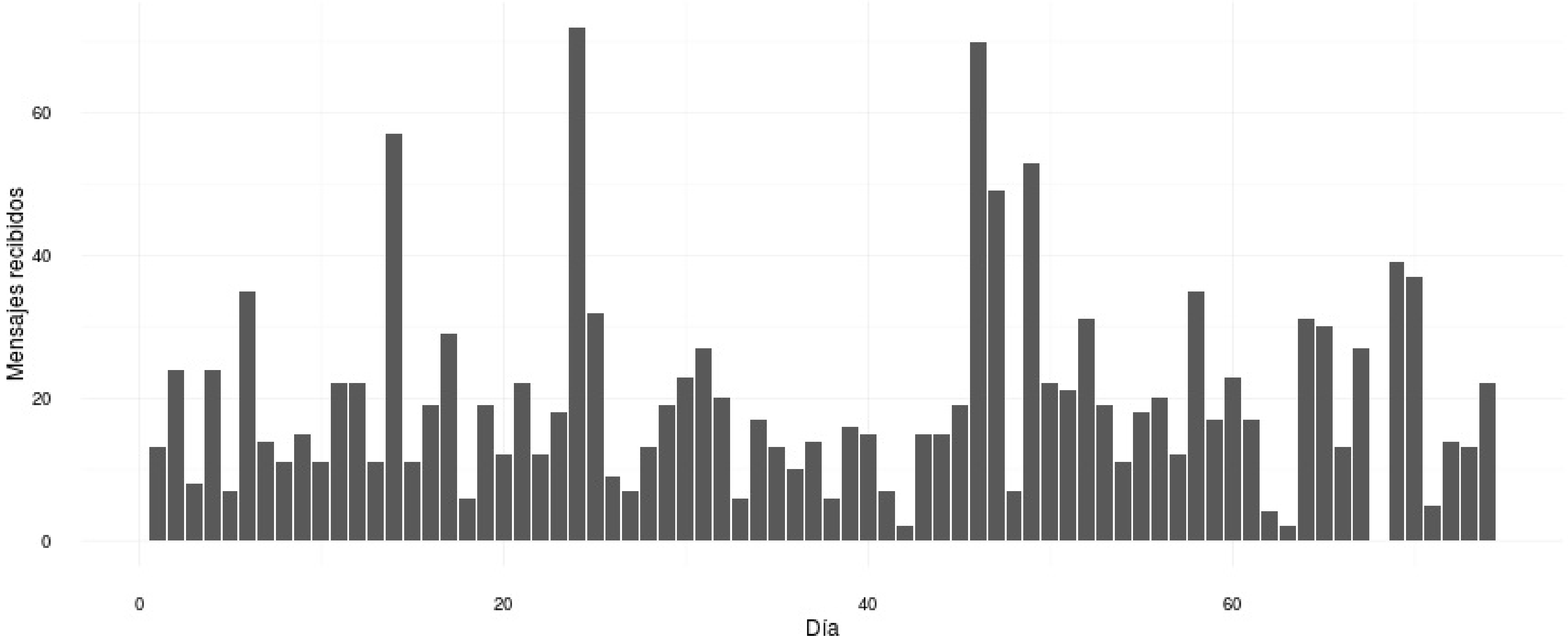
$$\beta \sim \ Laplace(0, \sigma)$$

Esto equivale a la regularización Lasso

Por ejemplo...

Ejemplo: Cambio de Media

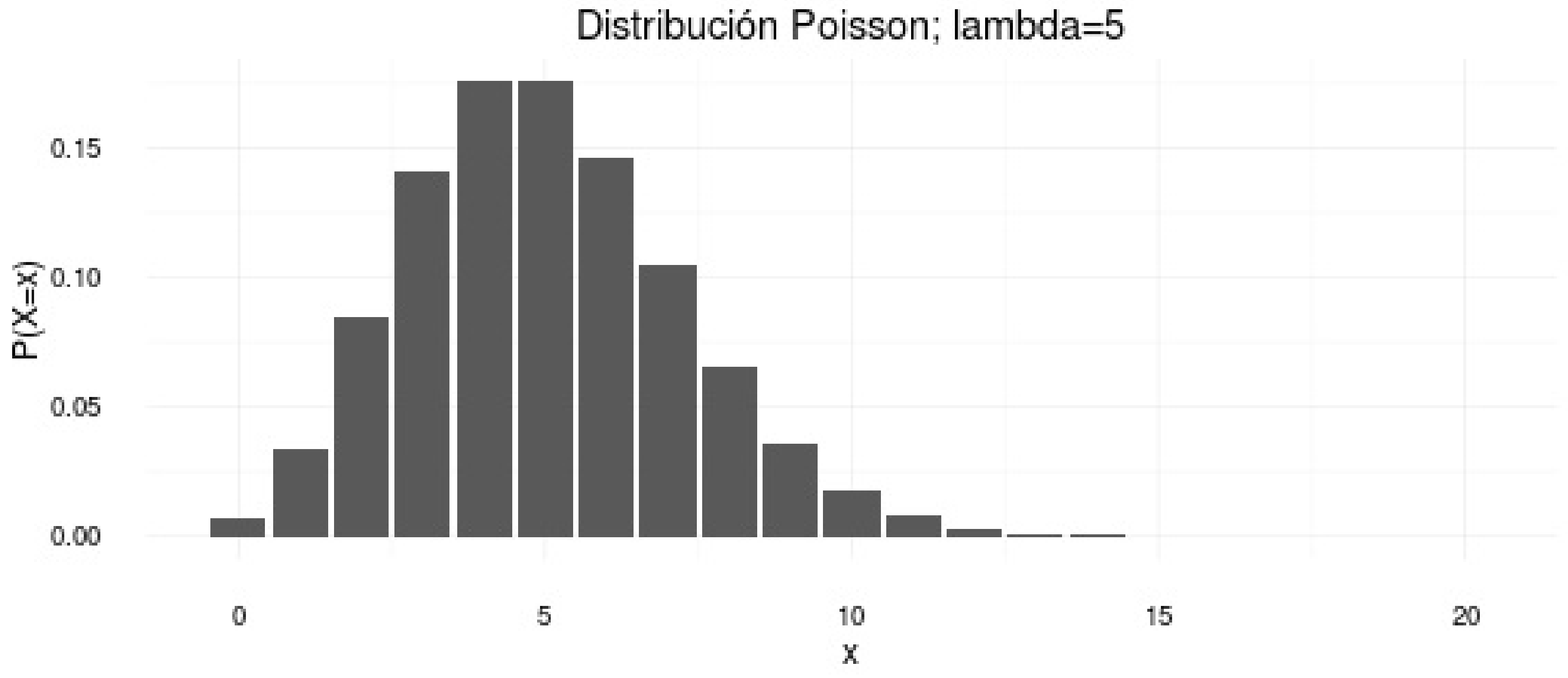
- *Data*: Número de mensajes de texto recibidos cada día
- ¿Existe un cambio súbito en esta variable?



Datos y ejemplo tomados de Bayesian Methods for Hackers

Ejemplo: Cambio de Media

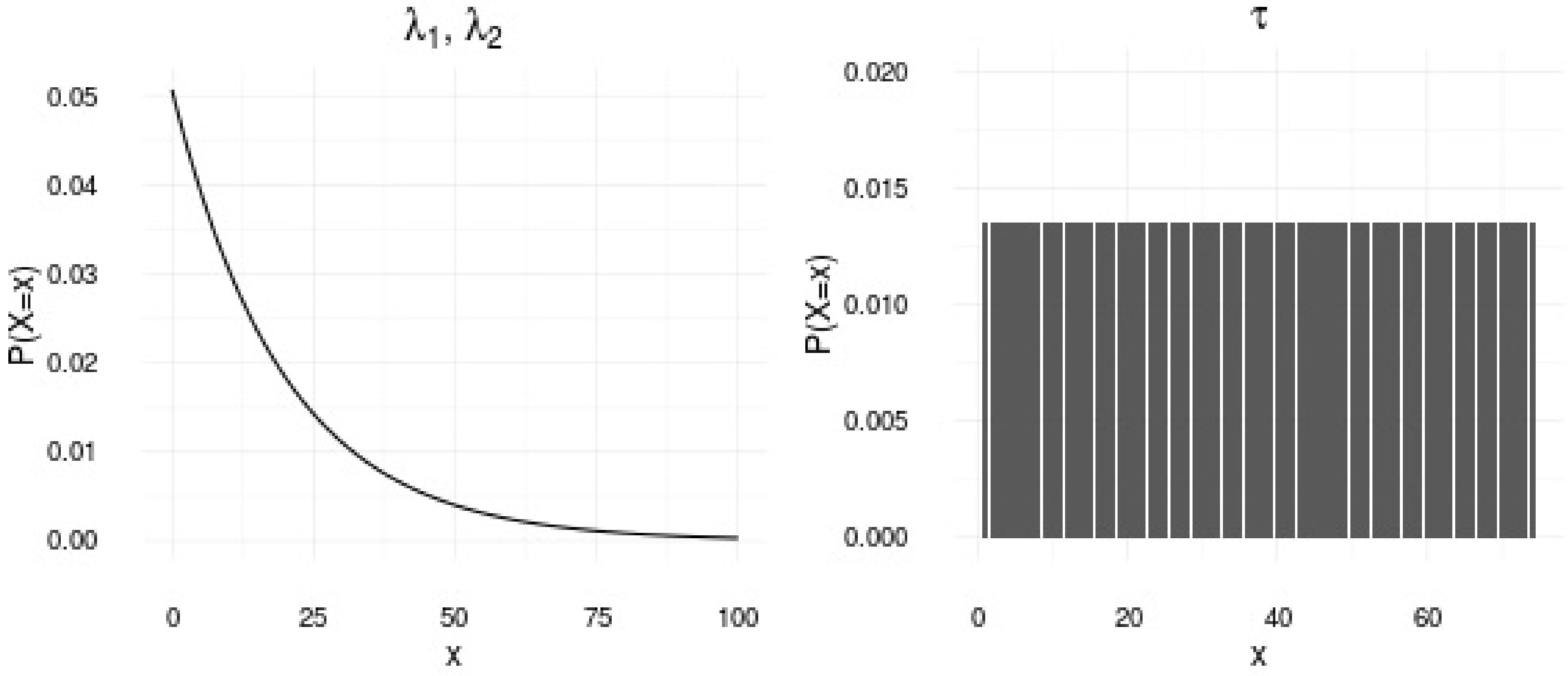
- Supongamos que los mensajes se generan de manera aleatoria, según el día
- Una buena distribución para representar esto es la Poisson



- Como el parámetro λ indica el promedio, la pregunta puede formularse cómo ¿Existe un cambio de lambda?
- Podemos usar dos parámetros λ_1 y λ_2, para representar estos posibles dos estados
- Lo único que nos falta es un parámetro τ, el cual representa el día en el cuál ocurre el cambio

Ejemplo: Cambio de Media

- Hay que escoger distribuciones *a priori* para estos tres parámetros
- Suponiendo que no poseemos información previa, una buena idea es escoger la misma distribución para λ_1 y λ_2 y una no informativa para τ. Por ejemplo:



- Escogimos una distribución exponencial para λ_1, λ_2, con (hiper)parámetro α positivo
- La distribución *a priori* para τ es una discreta uniforme

Ejemplo: Cambio de Media

Modelado en PyMC3 (Parámetros)

```python
import numpy as np
import pymc3 as pm

data = np.loadtxt("data/txtdata.csv")
n_obs = len(data)
day = np.arange(n_obs)

# define model
with pm.Model() as model:
    # hyper parameters
    alpha = 1/data.mean()

    # parameters
    tau = pm.DiscreteUniform("tau", lower=0, upper=n_obs)
    lambda1 = pm.Exponential("lambda1", alpha)
    lambda2 = pm.Exponential("lambda2", alpha)
    lambda_i = pm.math.switch(day < tau, lambda1, lambda2)
```

Ejemplo: Cambio de Media

Modelado en PyMC3 (Observaciones y Predictiva)

```
with model:
    # observations
    obs = pm.Poisson("obs", lambda_i, observed=data)

    # predictive distributions:
    pred1 = pm.Poisson("pred1", lambda1)
    pred2 = pm.Poisson("pred2", lambda2)
```

- Después de la inferencia, `pred1` y `pred2` indicaran la distribución predictiva cuando $\lambda = \lambda_1$ y $\lambda = \lambda_2$, correspondientemente

Inferencia en PyMC3

```
# perform inference
with model:
    step = pm.Metropolis()
    trace = pm.sample(10000, tune = 5000, step = step)
```

Ejemplo: Cambio de Media

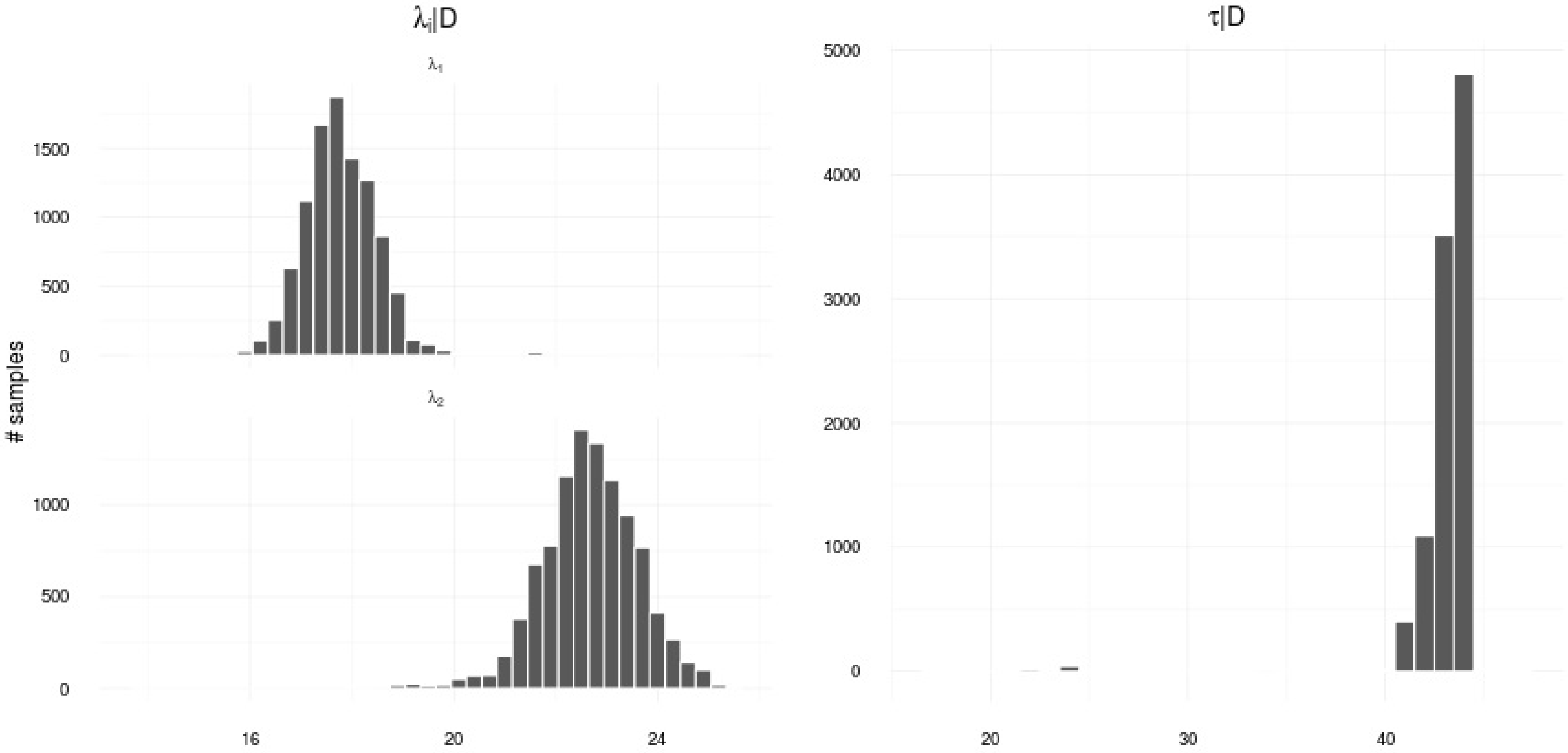
Preguntas específicas

- En la variable `trace` se encuentran las muestras tomadas de las distribuciones posteriores y de las predictivas.
- Esta muestra nos permite contestar una gran variedad de 'preguntas'

```
# Expected L1
print(trace["lambda1"].mean()) # 17.78916798570392
# Expected L2
print(trace["lambda2"].mean()) # 22.669546949779832
# Probability L2 > L1:
print((trace["lambda2"] > trace["lambda1"]).mean()) # 0.9914
# Probability tau = 44
print((trace["tau"] == 44).mean()) # 0.4808
# Probability 20+ messages before tau day
print((trace["pred1"] > 20).mean()) # 0.2478
# Probability 20+ messages after tau day
print((trace["pred2"] > 20).mean()) # 0.6715
```

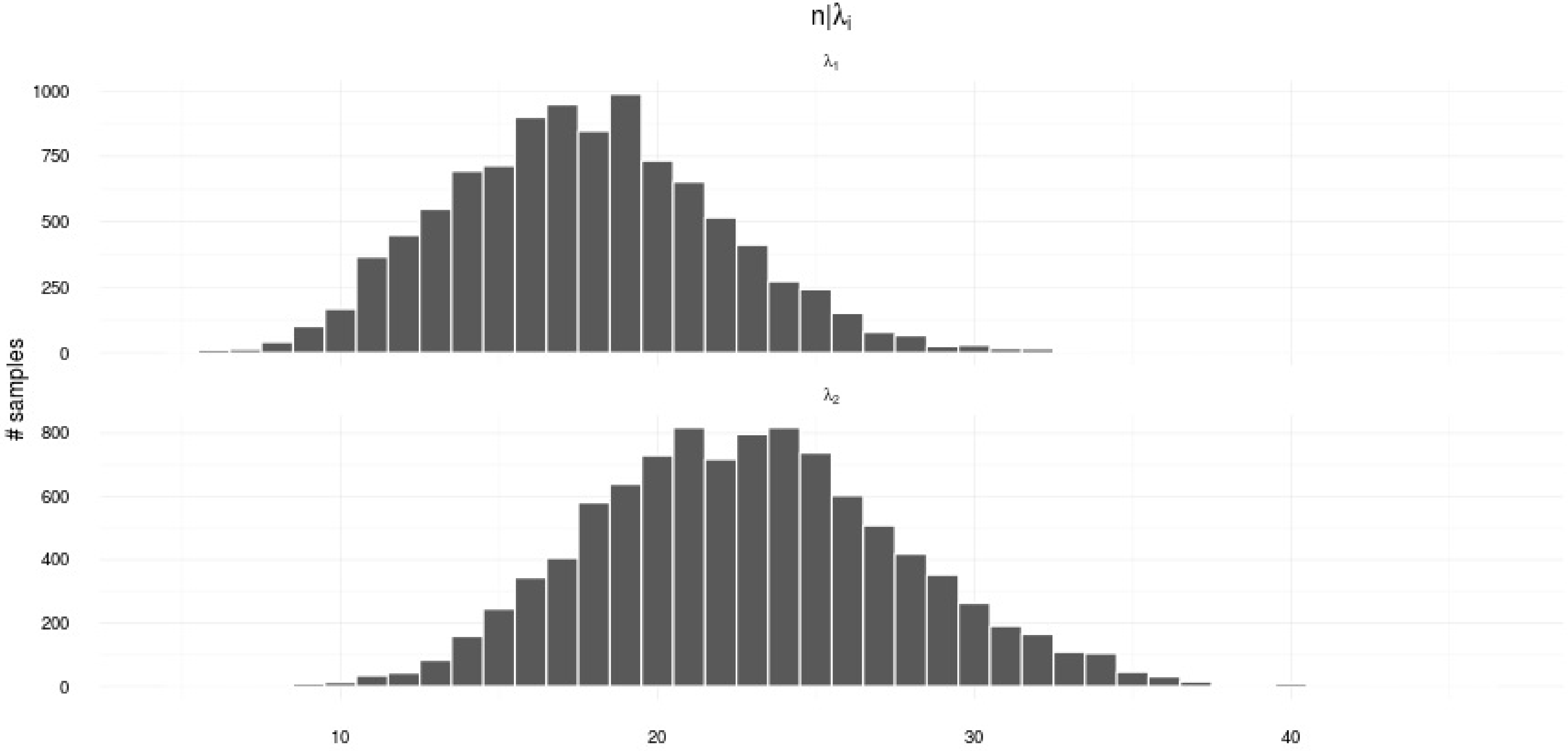
Ejemplo: Cambio de Media

Posteriores



Ejemplo: Cambio de Media

Predictiva(s)



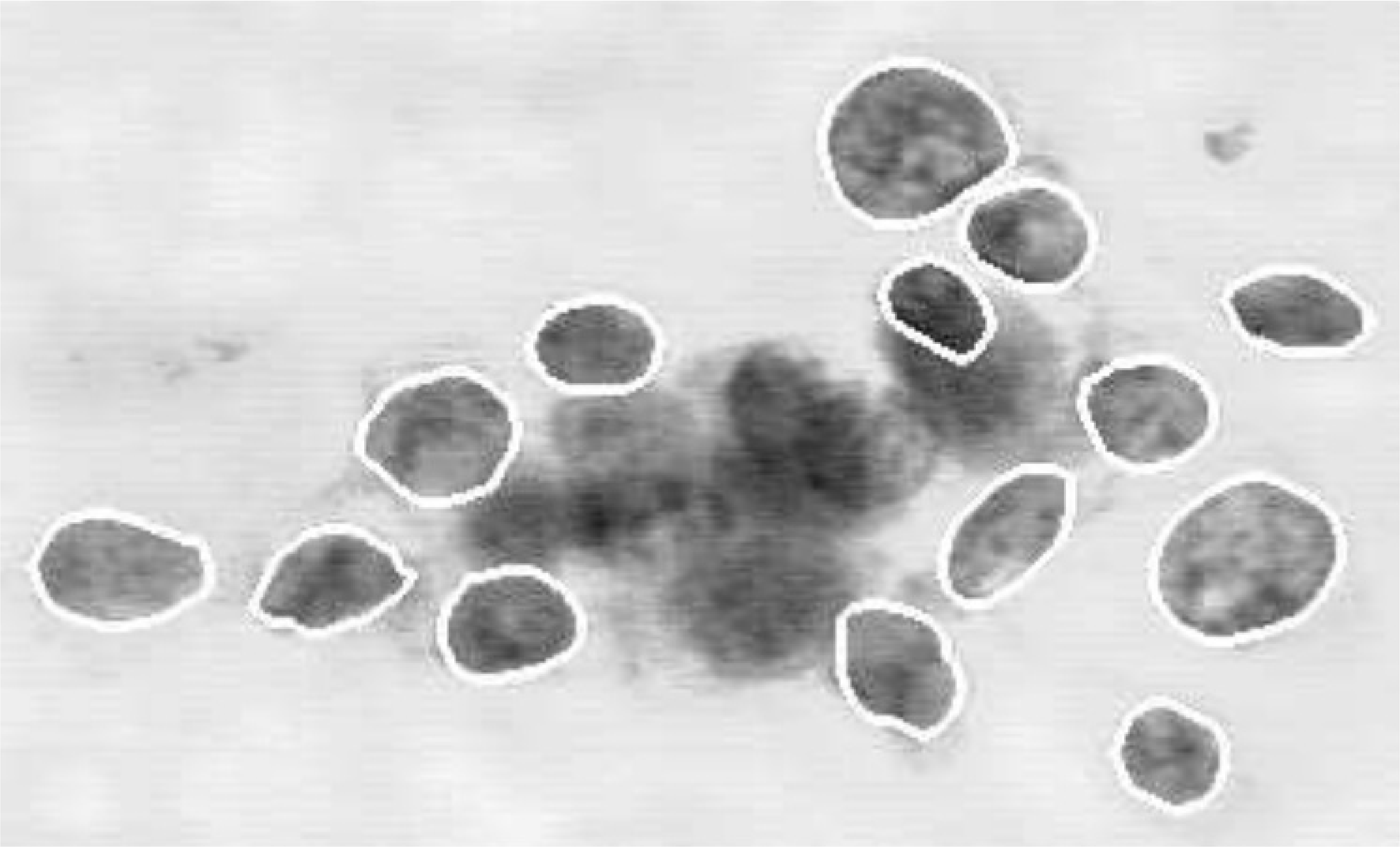
Pero sólo era una variable...

Otro ejemplo!

Ejemplo: Cáncer de Mama

Data: Features del núcleo de las células de un tumor de pecho, tomadas con una jeringa

Objetivo: Clasificarlo como benigno o maligno



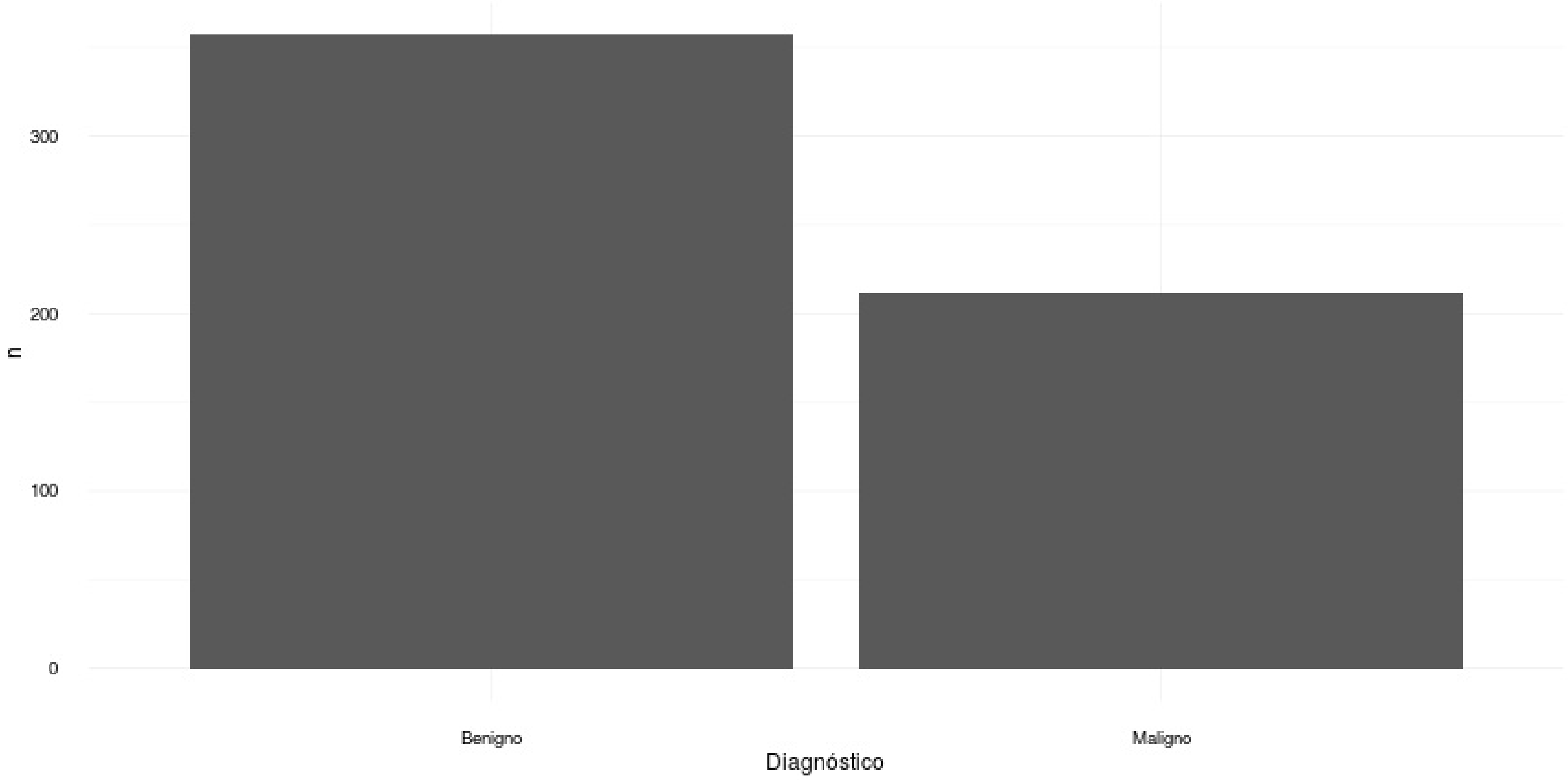
Ejemplo: Cáncer de Mama

Promedio, desviación estándar y "peor valor" de:

- Radio
- Textura
- Perímetro
- Área
- Suavidad
- Compacidad
- Concavidad
- Simetría
- Estructura Fractal

Ejemplo: Cáncer de Mama

Como nuestra variable de respuesta tiene sólo dos clases, podemos usar una Bernoulli para representarla



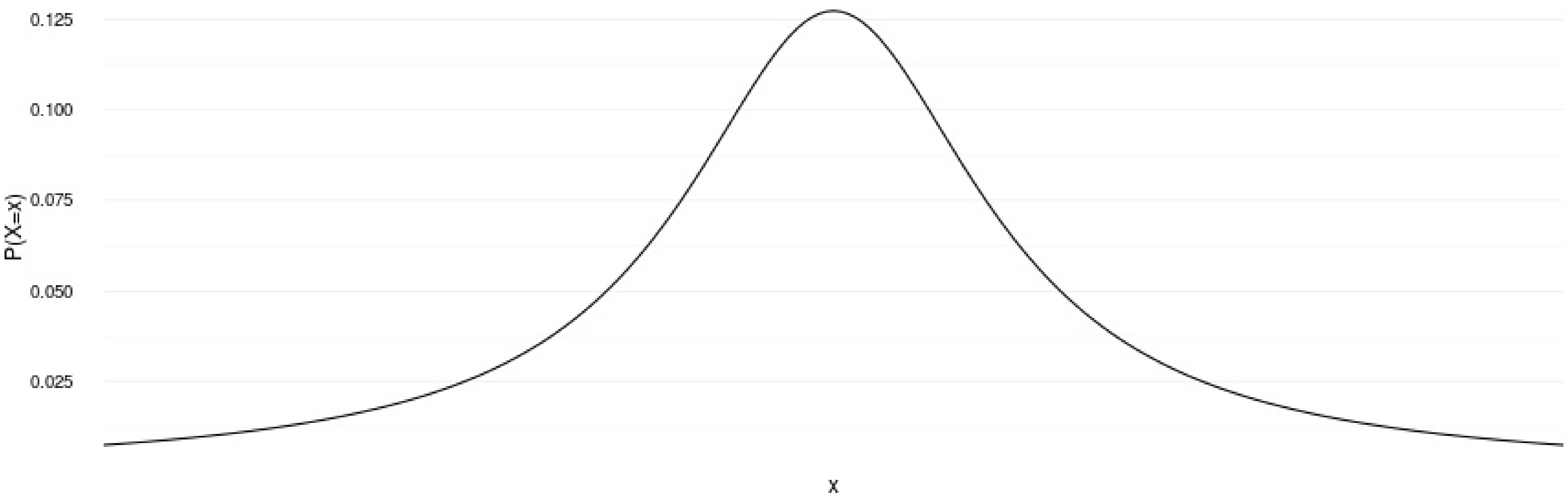
Ejemplo: Cáncer de Mama

Dado nuestro (mi) escaso conocimiento de cómo éstas variables son importantes para un diagnóstico de cáncer, escogemos un modelo lineal general con priors débilmente informativas.

$$y|x \sim Bernoulli(p)$$

$$p = (1 + e^{\beta^t x})^{-1}$$

El prior en este caso es una Cauchy con escala 2.5



Ejemplo: Cáncer de Mama

Preprocesamiento

```
import numpy as np
import pymc3 as pm

# Read
data = pn.read_csv("../data/cancer.csv")

# Remove malformed column
data = data.drop("Unnamed: 32", axis=1)

# Train and test
np.random.seed(42)
is_train = np.random.rand(len(data)) < 0.8
train = data[is_train]
```

Ejemplo: Cáncer de Mama

Preprocesamiento

```
# Standardize the predictors
X = np.array(train.iloc[:, 2:])
mX, sX = X.mean(axis=0), X.std(axis=0)
scaled = (X - mX)/sX

# Add bias term
predictors = np.c_[np.ones(scaled.shape[0]), scaled]

# Transform response to boolean
y = np.array(train.diagnosis == "M")
```

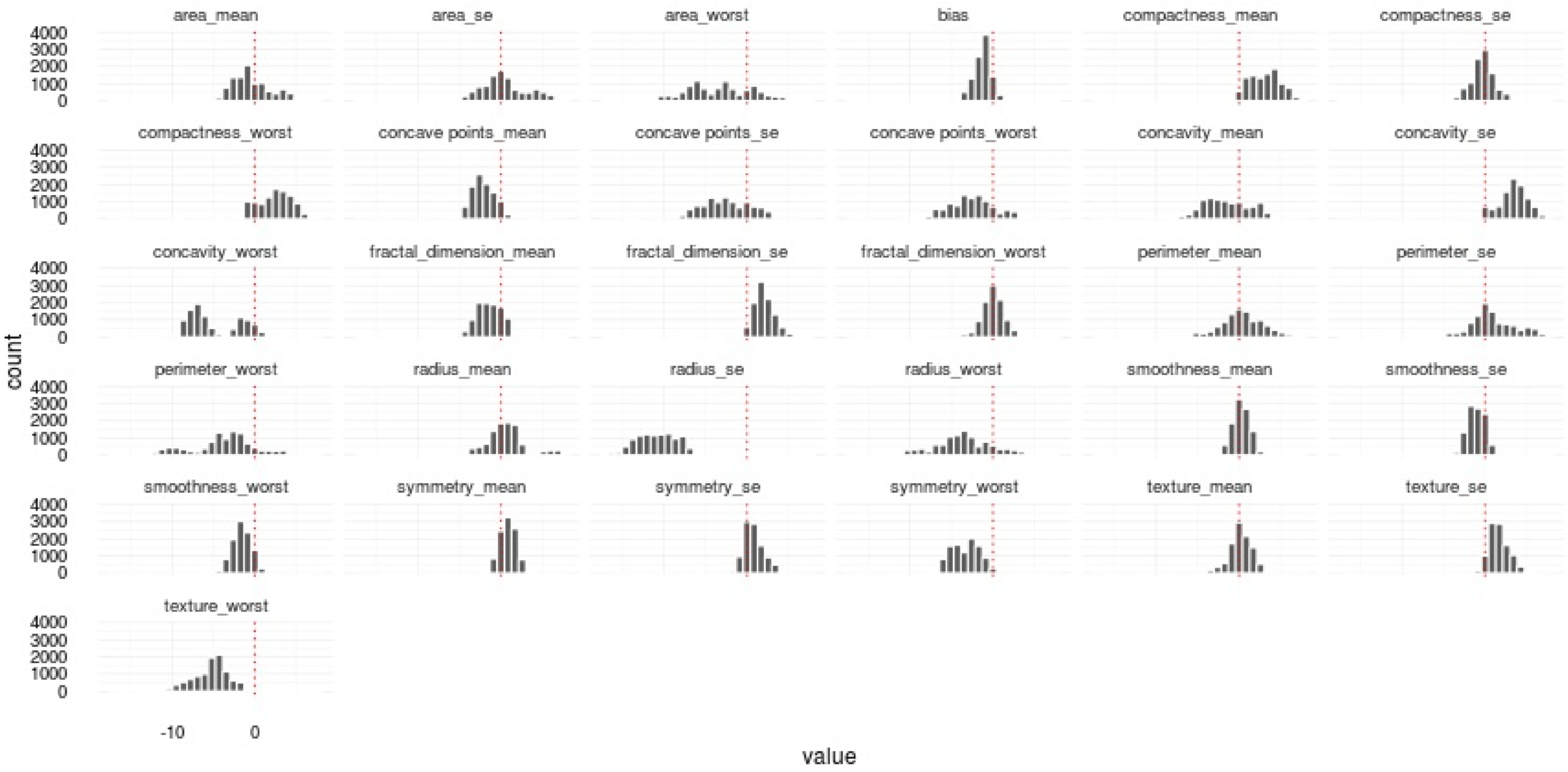
Ejemplo: Cáncer de Mama

Modelo en PyMC3 e Inferencia

```
n_predictors = predictors.shape[1]
# Esto nos permitirá muestrear de la predictiva más adelante
predictors = shared(predictors)

with pm.Model() as model:
    beta = pm.Cauchy("beta", 0, 2.5, shape=n_predictors)
    p = 1/(1 + tt.exp(tt.dot(predictors, beta)))
    obs = pm.Bernoulli("obs", p, observed=y)

    step = pm.Metropolis()
    trace = pm.sample(50000, step=step, random_seed=42)
    burned = trace[40000:]
```

Ejemplo: Cáncer de Mama

Posteriores



Ejemplo: Cáncer de Mama

Predicciones en testing

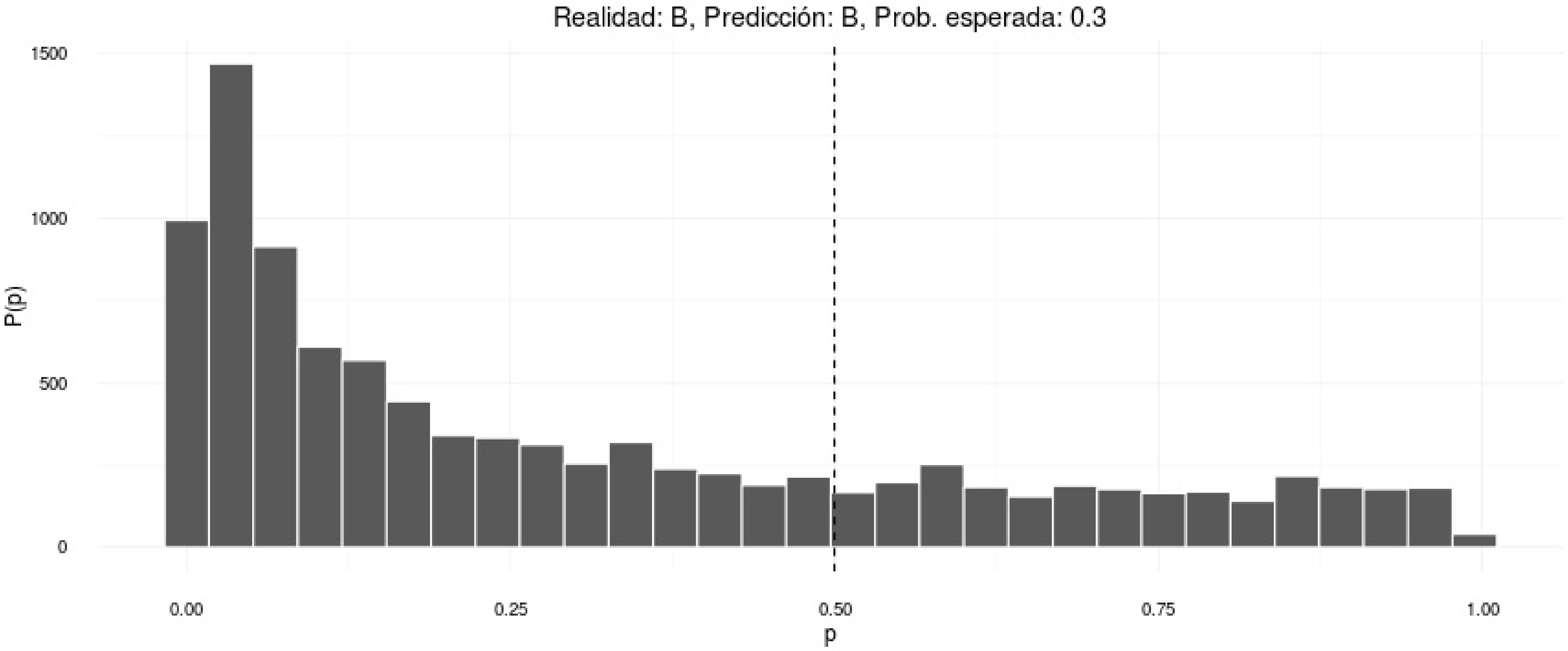
```
# Standardize testing set
X_test = np.array(test.iloc[:, 2:])
scaled_test = (X_test - mX)/sX
# Add bias
predictors_test = np.c_[np.ones(scaled_test.shape[0]), scaled_test]

# Set theano shared object to testing predictors and sample from the predictive distribution
predictors.set_value(predictors_test)
pred_samples = pm.sample_ppc(burned, model=model, samples=10000, progressbar=True)
```

```
##
##          FALSE TRUE
##   FALSE     83    0
##   TRUE       2   42
```

Ejemplo: Cáncer de Mama

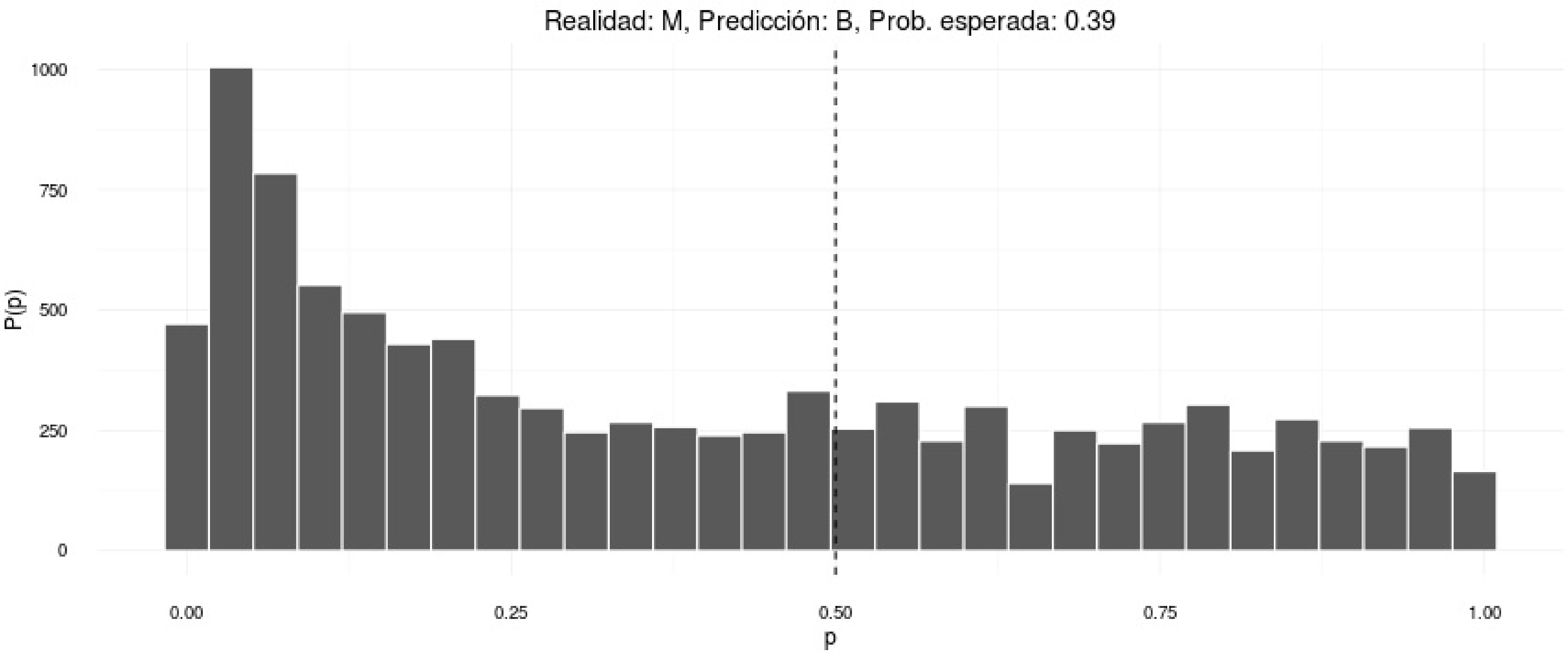
Importancia de la Incertidumbre

Chequemos un par de casos específicos



Ejemplo: Cáncer de Mama

Importancia de la Incertidumbre



Con una incertidumbre así, ¿Confiarían en no tener cáncer?

Ejemplo: Cáncer de Mama

Función de pérdida

Una manera de resolver el problema anterior es especificando bien lás **pérdidas** o penalizaciones en las que incurre el modelo al predecir.

Cuándo tomamos 0.5 como límite para predecir la clase positiva, estamos asumiendo (y suponiendo) tácitamente una matriz de pérdida como:

$$L = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

En nuestro caso en particular, una mejor matriz podría ser:

$$L = \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Ejemplo: Cáncer de Mama

Función de pérdida

Al cambiar la matriz de pérdida, cambiamos la regla para predecir. En general, la clase óptima a predecir está dada por:

$$argmin_l E_L = argmin_l \sum_{k \in K} L(k,l) Pr(k|x)$$

Con dos clases y la matriz de pérdida como en el slide anterior, esto se reduce a:

$$argmin\{2P(M|x), P(B|x)\}$$

Que implica un límite de $p = 1/3$ para predecir la clase positiva (tumor maligno).

En general, establecer funciones de pérdidas adecuadas (y pesarlas de acuerdo a la incertidumbre poseída) nos puede ayudar a tomar mejores decisiones de predicción.

Conclusiones

La **Estadística Bayesiana** provee un *framework* de análisis de datos que propone soluciones para problemas prevalentes en otras técnicas como el *Machine Learning* y la Estadística frecuentista.

La **Programación Probabilística** resuelve algunos de los problemas prácticos relacionados con la inferencia Bayesiana, poniendo al alcance del Científico de Datos estas técnicas, sin necesidad de resolver problemas específicos relacionados a cada modelo.

Tomados juntos, proveen una manera práctica y poderosa de resolver preguntas acerca de incertidumbre y causalidad. En particular, ayudan a disminuir el problema de **sobrecertidumbre** en las predicciones, que puede tener consecuencias catastróficas según la aplicación.

¡Gracias!

twitter: @arinarmo, github: arinarmo

Referencias

Bayesian Methods for Hackers - Cameron Davidson-Pilon

Probabilistic Programming - Fast Forward Labs

Bayesian Reasoning and Machine Learning - David Barber

Weakly Informative Priors - Andrew Gelman, *et al.*

Bayesian Lasso - Trevor Park, *et al.*

Nuclear Feature Extraction for Breast Tumor Diagnosis - W. Nick Street, *et al.*

Repositorio con la plática: arinarmo/love_uncertainty