Clase 192 — Bayes intro: priors, posterior, MCMC con PyMC

Parte: 3 — Estadística Inferencial y Causal · Fuente: McElreath, Statistical Rethinking (2ª ed.) + Gelman et al., Bayesian Data Analysis (3ª ed.) + Martin, Bayesian Modeling and Computation in Python. Duración estimada: 95 min.

Clase 192 — Bayes intro: priors, posterior, MCMC con PyMC

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: McElreath, Statistical Rethinking (2ª ed.) + Gelman et al., Bayesian Data Analysis (3ª ed.) + Martin, Bayesian Modeling and Computation in Python. Duración estimada: 95 min.*

Objetivo

Entender la lógica bayesiana —prior + likelihood → posterior vía teorema de Bayes— y construir un modelo simple end-to-end con PyMC v5 (regresión bayesiana sobre datos reales), interpretar el posterior con ArviZ (trace plots, posterior intervals, posterior predictive checks), y conocer el stack moderno: PyMC v5 (post-Theano, sobre PyTensor), NumPyro (sobre JAX, GPU-friendly) y ArviZ (visualización + diagnóstico backend-agnóstico).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Escribir posterior likelihood × prior y aplicarlo a un caso conjugado (Beta-Binomial para una tasa de conversión).
- Construir un modelo lineal bayesiano con pymc.Model() as m: ... y muestrearlo con pm.sample() (NUTS).
- Inspeccionar trace con arviz.plot_trace, diagnosticar convergencia (r_hat ≤ 1.01, ess_bulk ≥ 400).
- Interpretar HDI (Highest Density Interval) como reemplazo del IC clásico — con interpretación directa de probabilidad.
- Hacer posterior predictive check (pm.sample_posterior_predictive) y entender por qué es la validación bayesiana fundamental.
- Conocer NumPyro y cuándo elegirlo sobre PyMC (modelos grandes, GPU/JAX, optimización stocástica via SVI).

Temas

- Teorema de Bayes: P(θ|D) = P(D|θ)·P(θ) / P(D).
- Conjugados: Beta–Binomial, Gamma–Poisson, Normal–Normal (intuición sin MCMC).
- MCMC: idea — muestrear de una distribución sin computarla analíticamente. NUTS (No U-Turn Sampler).
- HDI vs IC frecuentista: el HDI es interpretado directamente como P(θ HDI | datos) = 0.94.
- Posterior predictive: la distribución de datos futuros simulados desde el posterior. Test de modelo.
- Priors: no informativos (Uniform, HalfNormal con scale grande), débilmente informativos (recomendado), informativos (cuando hay expertise).
- Complemento moderno: PyMC v5 (PyTensor backend, ya estable post-Theano), NumPyro (JAX, GPU), ArviZ (diagnóstico estándar).

Versión profundizada — 2026

El tema moderno que vivía como complemento dentro de esta clase ahora tiene clase propia dedicada con

patrón completo, ejercicios y homework:

- Clase 158b — Stack bayesiano moderno: PyMC v5, NumPyro, ArviZ

Definiciones y características

- Prior P(θ): creencia sobre el parámetro antes de ver los datos.
- Likelihood P(D|θ): probabilidad de los datos dado el parámetro (misma que en MLE/frecuentista).
- Posterior P(θ|D): creencia sobre el parámetro después de los datos. Combina prior + likelihood vía Bayes.
- MCMC (Markov Chain Monte Carlo): técnica de muestreo de distribuciones complejas. NUTS es el algoritmo moderno default (extiende HMC).
- HDI (Highest Density Interval): intervalo de la distribución posterior que contiene la masa de probabilidad especificada Y tiene la mayor densidad. No confundir con IC — el HDI sí tiene interpretación de probabilidad directa.
- Posterior predictive distribution: $P(\tilde{y} \mid D) = \int P(\tilde{y}|\theta) \cdot P(\theta|D)\, d\theta$. Distribución de datos nuevos integrando sobre la incertidumbre del parámetro.
- r_hat: Gelman-Rubin convergence diagnostic. Compara varianza intra-cadena vs entre-cadenas.
- ESS (effective sample size): número de muestras independientes equivalentes. MCMC produce muestras correlacionadas, así que ESS < N total.
- Conjugate prior: prior cuya combinación con un likelihood específico da posterior de la misma familia. Beta-Binomial, Gamma-Poisson, Normal-Normal.
- SVI (Stochastic Variational Inference): aproxima el posterior con una familia paramétrica más simple (ej.: Normal) optimizando ELBO. Muchísimo más rápido que MCMC; menos exacto.

Dataset / recursos

- tips (regresión bayesiana).
- Tasa de conversión sintética (Beta-Binomial conjugado).
- McElreath's Howell1 (estatura vs peso) — el ejemplo canónico del libro.
- Librerías: pymc (≥ 5), arviz, numpyro, seaborn, matplotlib.

Ejercicios

1. Conjugado a mano: 100 visitas, 8 conversiones. Prior Beta(1,1). Posterior = Beta(9, 93). Graficá prior y posterior; calculá HDI 94 % con scipy.stats.beta.ppf.
2. Regresión bayesiana: ajustá el modelo PyMC del ejemplo sobre tips. Reportá summary y plot_trace. Verificá r_hat ≤ 1.01.
3. Comparación: compará los coeficientes bayesianos (mean del posterior) con OLS de statsmodels sobre el mismo dataset. Con priors débiles, deberían ser casi idénticos.
4. Posterior predictive check: ejecutá sample_posterior_predictive y az.plot_ppc. Discutí si el modelo captura la asimetría de tips (probablemente no — sugerir cambiar a likelihood Gamma o lognormal).
5. NumPyro: traducí el modelo a NumPyro, ajustá, convertí con az.from_numpyro y verificá que los resultados son equivalentes. Comparar tiempo de ejecución.

Homework verificable

Modelo bayesiano jerárquico simple sobre tips:

1. Modelar la propina como tip ~ Normal(α_día + β·total_bill, σ) donde α_día varía por día (efecto aleatorio jerárquico): α_día ~ Normal(μ_α, σ_α).
2. Ajustar con PyMC v5, 4 cadenas, 2 000 muestras.
3. Diagnosticar: r_hat, ess_bulk, trace plots, divergencias.
4. Reportar HDI 94 % de β (efecto del bill sobre tip) y de los 4 α_día.
5. Conclusión en 4 líneas: ¿qué día tiene mayor intercepto? ¿Cuán incierto es β?

Criterio de aceptación: el modelo converge (r_hat ≤ 1.01 para todos los parámetros), β HDI no incluye 0 (efecto positivo claro del bill sobre tip), y la conclusión menciona la jerarquía como ventaja sobre 4 regresiones separadas.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
r_hat = 1.3 y reporto el resultado igual	No convergió. Fix: más tune, target_accept
Prior Uniform(-10⁶, 10⁶) "para ser objetiv	Priors muy planos hacen MCMC inestable y n
Interpreto el HDI como "intervalo de predi	El HDI es del parámetro, no de futuras obs
MCMC se queda atascado con divergences > 1	Posterior con geometría difícil (embudos).
Comparar modelos por DIC	DIC tiene problemas conocidos. Fix: usar L

Preguntas frecuentes

¿Bayes o frecuentista?

No son enemigos — son herramientas distintas. Bayes brilla con n chico, modelos jerárquicos, interpretación directa de probabilidades. Frecuentista brilla con datasets enormes y modelos simples. Industria moderna: ambos, eligiendo por problema.

¿Cuánto tarda MCMC?

Para regresión lineal con 1 000 obs y 3 parámetros: segundos en PyMC v5. Para modelos jerárquicos con 100 grupos: 1-10 min. Para modelos con miles de parámetros: minutos a horas. NumPyro/JAX puede acelerar 10-50×.

¿Qué pasa si elijo un prior "malo"?

Con datos abundantes, el likelihood domina y el posterior es casi insensible al prior. Con datos escasos, el prior importa — y eso es bueno, refleja la realidad de que con n=10 no se puede aprender nada sin asumir algo. Hacé prior predictive check (pm.sample_prior_predictive) para verificar que los priors no generan datos absurdos.

¿Variational inference (VI) o MCMC?

MCMC es más exacto asintóticamente; VI es mucho más rápido. Para producción con modelos grandes, VI (en NumPyro o Pyro) es la opción. Para inferencia rigurosa, MCMC.

¿Cuál es la diferencia con regresión lineal "normal"?

OLS te da β̂ puntual + IC frecuentista. Bayes te da distribución completa de β, lo cual permite responder preguntas como P(β > 0.5 | datos), P(β [0.3, 0.7]) directamente — sin pirueta interpretativa.

Referencias

- McElreath, R. (2020), Statistical Rethinking (2ª ed.), CRC Press — el mejor libro para empezar.
- Gelman et al. (2013), Bayesian Data Analysis (3ª ed.) — la referencia técnica completa.
- Martin, O. (2024), Bayesian Modeling and Computation in Python — Python-first, con PyMC v5.
- PyMC v5 docs — sección Introductory Overview.
- NumPyro docs — JAX-based.
- ArviZ docs — diagnóstico backend-agnóstico.
- Hoffman & Gelman (2014), The No-U-Turn Sampler, JMLR — paper de NUTS.

Siguiente parte

Clase 193 — Stack bayesiano moderno: PyMC v5, NumPyro, ArviZ

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb