Parte 3 — Estadística Inferencial y Causal

19 clases · Parte 3 del programa

Parte 3 — Estadística Inferencial y Causal

> *19 clases · bundle consolidado del currículo v3.*

Índice de clases

Clase 175 — Clase 175 — Distribuciones: normal, binomial, Poisson, exponencial

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 2 + Bruce & Bruce, cap. 2 Data and Sampling Distributions. Duración estimada: 70 min.*

Objetivo

Reconocer las cuatro distribuciones de probabilidad que aparecen en el 90 % de los problemas reales de data science —normal, binomial, Poisson, exponencial— sabiendo qué fenómeno modela cada una, cuáles son sus parámetros, cómo simularlas con scipy.stats / numpy.random, y cómo verificar empíricamente si los datos realmente siguen esa distribución antes de aplicar un test que la asuma.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Identificar la distribución apropiada para un fenómeno descrito en lenguaje natural (conteos raros → Poisson, éxitos/fracasos → binomial, tiempos entre eventos → exponencial, sumas/promedios → normal por TCL).

- Calcular media, varianza y cuantiles teóricos con scipy.stats.{norm, binom, poisson, expon} (.mean(), .var(), .ppf(), .pdf()/.pmf()).
- Simular muestras con rng = np.random.default_rng(seed) y comparar histograma vs PDF/PMF teórica.
- Aplicar un Q-Q plot (scipy.stats.probplot) y un Kolmogorov-Smirnov (scipy.stats.kstest) para validar normalidad.
- Reconocer cuándo el Teorema Central del Límite justifica usar normal aunque los datos crudos no lo sean.

Temas

#	Tema	Por qué importa
1	Distribución normal N(μ, σ²)	Base de t-test, ANOVA, intervalos de confi
2	Distribución binomial Bin(n, p)	Conversiones A/B, click-through rate, prop
3	Distribución de Poisson Poi(λ)	Eventos raros por unidad de tiempo/área (f
4	Distribución exponencial Exp(λ)	Tiempos entre eventos Poisson (churn, time
5	Teorema Central del Límite (TCL)	Por qué la normal aparece aunque los datos
6	Verificación empírica: Q-Q plot + KS test	Antes de asumir, mirá.

Definiciones y características

- PDF (Probability Density Function): para variables continuas. f(x) no es probabilidad; la probabilidad es $\int f(x)\,dx$ sobre un intervalo. f(x) puede ser > 1.
- PMF (Probability Mass Function): para variables discretas. P(X = k) directo. Siempre en [0, 1].
- CDF (Cumulative Distribution Function) F(x) = P(X ≤ x): en scipy .cdf(). Su inversa es .ppf() (quantile function), útil para construir intervalos.
- Normal N(μ, σ²): simétrica, soporte en . Regla 68-95-99.7 (1σ, 2σ, 3σ). Es la única distribución cuya suma de variables independientes sigue siendo de la misma familia exacta.
- Binomial Bin(n, p): suma de n ensayos Bernoulli independientes con éxito p. E[X]=np, Var[X]=np(1-p). Para n grande y p no extrema, ≈ N(np, np(1-p)) — esto es lo que justifica los z-tests de proporciones.
- Poisson Poi(λ): conteo de eventos raros independientes en un intervalo fijo. E[X]=Var[X]=λ (¡equidispersión!). Si tus datos tienen Var/Mean 1, hay sobre-dispersión y Poisson no aplica — considerar binomial negativa.
- Exponencial Exp(λ): tiempo entre eventos Poisson. Memoryless: P(X > s+t | X > s) = P(X > t). Por eso modela mal cosas con desgaste (un motor de 10 años no falla igual que uno nuevo).
- TCL: si X_1, …, X_n son i.i.d. con media μ y varianza finita σ², entonces $(\bar{X} - \mu) / (\sigma/\sqrt{n}) \to N(0, 1)$ cuando $n \to \infty$. Regla práctica: n ≥ 30 alcanza, salvo distribuciones muy asimétricas.
- Q-Q plot: scatter de cuantiles muestrales vs cuantiles teóricos. Si los puntos caen sobre la diagonal, los datos siguen la distribución.

Dataset / recursos

- Conteo de llamados a un call center por hora (sintético): rng.poisson(lam=4.2, size=10_000) → Poisson.
- Datos reales: seaborn.load_dataset('tips') para chequear normalidad de total_bill (no es normal, asimétrico positivo — buen contraejemplo).
- Librerías: numpy, scipy.stats, matplotlib, seaborn.

Ejercicios

1. Simulación y PDF/PMF: con rng = np.random.default_rng(42), generá 10 000 muestras de cada una de las 4 distribuciones con parámetros razonables. Para cada una: histograma con density=True superpuesto con la PDF/PMF teórica de scipy.stats.

2. Cuantiles: calculá scipy.stats.norm(loc=100, scale=15).ppf([0.025, 0.5, 0.975]) (IQ test → IC 95 % poblacional) y verificá que el 2.5 % y 97.5 % muestrales de una simulación con n=100_000 se acerquen.
3. TCL en acción: tomá Exp(λ=1) (claramente no normal). Generá 5 000 promedios de tamaños n {1, 5, 30, 100} y graficá los 4 histogramas lado a lado. Verificá cómo se va volviendo simétrico y campaniforme.
4. Q-Q plot: scipy.stats.probplot(tips.total_bill, dist='norm', plot=plt). Anotá qué muestra el extremo derecho (asimetría positiva → cola larga arriba de la diagonal).
5. ¿Poisson o no?: con los conteos por hora del dataset sintético, calculá mean() y var(). Si var/mean [0.8, 1.2], equidispersión → Poisson plausible. Probá con lam=4.2 (deberías ver ratio ≈ 1) y con datos contaminados (mezclá con rng.poisson(20, size=200) para ver overdispersión).

Homework verificable

Notebook que:

1. Carga tips.total_bill de seaborn.
2. Genera un Q-Q plot contra 'norm' y otro contra 'lognorm'.
3. Aplica scipy.stats.kstest contra ambas (estandarizando los datos).
4. Concluye por escrito qué distribución modela mejor total_bill y por qué (≤ 3 líneas).

Criterio de aceptación: el p-value del KS contra lognormal debe ser mayor que contra normal, y la conclusión debe mencionar que total_bill está acotado por abajo en 0 y tiene cola derecha — incompatible con normal.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
kstest da p=0.0 aunque los datos "se ven n	Probablemente no estandarizaste. KS contra
Histograma no coincide con la PDF teórica	Olvidaste density=True en plt.hist. Sin es
Aplico Poisson y la varianza es mucho mayo	Sobre-dispersión: hay heterogeneidad ocult
np.random.seed(42) no me da resultados rep	El estado global está deprecado para análi
Asumo normalidad con n=8 porque "el TCL lo	El TCL es asintótico. Con n chico y distri

Preguntas frecuentes

¿Cuándo uso scipy.stats.norm(loc, scale) vs np.random.normal(mu, sigma)?

Para simular muestras: ambos funcionan, pero rng = np.random.default_rng(seed); rng.normal(...) es el patrón moderno reproducible. Para calcular PDF, CDF, cuantiles: scipy.stats.norm, que es un objeto distribución con todos los métodos.

¿Poisson y binomial con n grande y p chica se parecen?

Sí: si n → ∞ y p → 0 con np = λ constante, Bin(n, p) → Poi(λ). Por eso "1 cada 1000" se modela igual con cualquiera de las dos. En la práctica, si n ≥ 100 y p ≤ 0.05, son intercambiables.

¿La distribución t-Student es lo mismo que la normal?

No, pero converge. t(ν) con ν → ∞ tiende a N(0,1). Para ν ≥ 30 son visualmente idénticas. La diferencia importa en muestras chicas (la t tiene colas más pesadas, lo que produce intervalos de confianza más anchos — correcto cuando estimás σ).

¿Mis datos tienen que ser normales para hacer un t-test?

No exactamente. Lo que tiene que ser ≈ normal es la distribución muestral de la media, y por TCL eso pasa con n ≥ 30 aunque los datos crudos no lo sean. Si n < 30 y los datos están sesgados, usá bootstrap (Clase 153) o Mann-Whitney (Clase 150).

¿Por qué exponencial es "sin memoria"?

Porque P(X > s + t | X > s) = P(X > t). Aplicado a un servidor que lleva 3 h sin caer: la probabilidad de aguantar otra hora es la misma que la de aguantar una hora desde 0. Para sistemas con desgaste (motores, humanos), usá Weibull o Gamma.

Referencias

- ISLP (James et al.), cap. 2 — Statistical Learning, sección sobre distribuciones.
- Bruce, P. & Bruce, A. Practical Statistics for Data Scientists (2ª ed., O'Reilly), cap. 2 Data and Sampling Distributions.
- scipy.stats reference — objetos norm, binom, poisson, expon.
- numpy.random.Generator — API moderna recomendada desde NumPy 1.17.
- 3Blue1Brown — Why π appears in the normal distribution (intuición visual del TCL).

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 176 — Clase 176 — Test t (una muestra, dos muestras, pareado)

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 13 + Bruce & Bruce, cap. 3 Statistical Experiments and Significance Testing. Duración estimada: 80 min.*

Objetivo

Que el alumno aplique correctamente las tres variantes del test t —una muestra, dos muestras independientes (Welch por default), pareado—, distinga hipótesis nula y alternativa, lea p-value e intervalo de confianza de la salida de scipy.stats y pingouin, y aprenda a reportar effect size (Cohen's d, Hedges' g) junto con el p-value para no caer en la trampa de "significativo pero irrelevante".

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Formular H y H (bilateral / unilateral) para un problema concreto y elegir la variante correcta del test t.
- Ejecutar scipy.stats.ttest_1samp, ttest_ind(equal_var=False) y ttest_rel, interpretando el statistic, pvalue y el atributo .confidence_interval() (scipy ≥ 1.10).
- Verificar supuestos: normalidad por grupo (Shapiro / Q-Q plot) o invocar TCL si n ≥ 30.
- Decidir entre test bilateral vs unilateral sin caer en p-hacking (la dirección debe estar fijada antes de mirar los datos).
- Reportar effect size: Cohen's d, Hedges' g corregido para muestras chicas, y su interpretación cualitativa (small/medium/large).

Temas

- H / H, errores tipo I (α) y tipo II (β).
- Test t de una muestra: $t = (\bar{x} - \mu) / (s/\sqrt{n})$, gl = n - 1.
- Test t de dos muestras independientes: Welch (varianzas distintas, default moderno) vs Student (varianzas iguales — supuesto fuerte, casi nunca correcto).
- Test t pareado (mismo sujeto antes/después).
- p-value: probabilidad bajo H de observar algo al menos tan extremo. NO es P(H | datos).
- Intervalo de confianza al 95 % como complemento del p-value.
- Complemento moderno: effect size (Cohen's d, Hedges' g, Cliff's δ) — la pregunta "¿es relevante?" que el p-value no responde.

Versión profundizada — 2026

El tema moderno que antes vivía como complemento dentro de esta clase ahora tiene su(s) clase(s) propia(s) con patrón completo, ejercicios y homework:

- Clase 147a — Effect size dedicado: Cohen's d, Hedges' g, Cliff's δ con pingouin

Definiciones y características

- Hipótesis nula H: la afirmación "conservadora" — no hay diferencia, o la diferencia es 0. Se busca rechazarla, no probarla.
- Hipótesis alternativa H: lo que querés demostrar. Bilateral (≠) o unilateral (>, <).
- p-value: P(estadístico al menos tan extremo como el observado | H verdadera). Si p < α, rechazo H. NO es la probabilidad de que H sea verdadera.
- α (alpha): probabilidad máxima de error tipo I (rechazar H siendo verdadera). Convencional: 0.05. Para problemas críticos: 0.01 o 0.001.
- β (beta): probabilidad de error tipo II (no rechazar H siendo falsa). Poder = 1 - β (Clase 154).
- Welch's t-test: variante que no asume varianzas iguales. Es el default razonable. equal_var=True (Student clásico) es legacy.
- Test pareado: cuando cada observación de un grupo tiene su "par" en el otro (mismo paciente antes/después, misma página web con/sin cambio). Reduce varianza al diferenciar dentro del par.
- Cohen's d: magnitud estandarizada de la diferencia. d = 0.2/0.5/0.8 → small/medium/large.
- alternative='greater': unilateral derecho. Solo legítimo si la dirección la fijaste antes de ver los datos.

Dataset / recursos

- seaborn.load_dataset('tips'): comparar tip entre sex='Male' vs 'Female', o time='Lunch' vs 'Dinner'.
- Para pareado: simular un dataset antes/después con numpy.random.default_rng (presión arterial pre/post fármaco).
- Librerías: scipy.stats, pingouin (pip install pingouin), seaborn.

Ejercicios

1. Una muestra: con tips.total_bill, testá H: μ = 20 vs H: μ ≠ 20 con scipy.stats.ttest_1samp(tips.total_bill, popmean=20). Reportá t, p y el IC95 % (.confidence_interval()).
2. Dos muestras (Welch): testá si tip difiere entre sex='Male' y sex='Female' con ttest_ind(equal_var=False). Calculá Cohen's d a mano y verificá contra pingouin.ttest.
3. Pareado: simulá presión arterial antes/después de un fármaco con rng = np.random.default_rng(0): antes = rng.normal(140, 12, 30), despues = antes - rng.normal(5, 3, 30). Aplicá ttest_rel(antes, despues) y comparalo contra hacer ttest_ind mal (verás cómo el pareado tiene mucho más poder).
4. Bilateral vs unilateral: para el ejercicio 2, repetí con alternative='greater' y 'less'. Observá cómo el p-value se divide ≈ 2.
5. Significativo vs relevante: generá grupo_a = rng.normal(100, 15, 10_000) y grupo_b =

rng.normal(100.5, 15, 10_000). El test va a dar p < 0.001; calculá Cohen's d y discutí en 2 líneas por qué el resultado "no importa".

Homework verificable

Sobre tips:

1. Hipótesis: la propina promedio es distinta para time='Lunch' y time='Dinner'.
2. Verificar normalidad de cada grupo con pingouin.normality (Shapiro).
3. Ejecutar pingouin.ttest y reportar: T, gl, p-value, IC95 %, Cohen's d, power.
4. Una conclusión de 3 líneas que mencione (a) si rechazás H, (b) la magnitud del efecto en palabras (small/medium/large), (c) si recomendarías ese hallazgo a un dueño de restaurante.

Criterio de aceptación: el reporte debe incluir effect size y la conclusión no puede limitarse a "p < 0.05". Si Cohen's d es < 0.2, la respuesta a (c) debería ser "no" aunque el p sea bajo.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Aplico ttest_ind con equal_var=True por co	Student clásico con varianzas desiguales y
p < 0.05 con n = 10^6 y declaro un hallazgo	"Significancia estadística" con muestra gi
Cambio alternative='greater' después de ve	Eso es p-hacking. La dirección se fija con
Aplico ttest_ind a observaciones pareadas	Pierde poder enormemente y puede dar p > 0
El test asume normalidad y mis datos son m	Welch es robusto pero no mágico. Fix: scip

Preguntas frecuentes

¿Welch o Student por default?

Welch siempre. No cuesta nada en poder cuando las varianzas son iguales, y protege contra el caso (muy común) de varianzas distintas. La literatura moderna (Delacre, Lakens & Leys 2017) recomienda abandonar el Student t-test.

¿Tengo que testear normalidad antes del t-test?

No con n ≥ 30 por grupo (TCL). Con n chico y datos visiblemente asimétricos, sí — y si Shapiro rechaza, mejor pasar a bootstrap o Mann-Whitney. Cuidado: con n enorme, Shapiro rechaza siempre por desviaciones triviales; mirá Q-Q plot como complemento.

¿Qué es BF10 que reporta pingouin?

Factor de Bayes contra H. BF10 > 3 es evidencia moderada a favor de H; > 10 fuerte. Es la versión bayesiana del p-value y no tiene los problemas del NHST (la testeás directamente en la Clase 158).

¿Por qué el IC95 % de la diferencia y el p-value siempre concuerdan?

Porque son la misma información presentada de otra forma. Si el IC95 % de μ_a - μ_b no incluye 0, entonces p < 0.05 (bilateral). Reportar el IC es más informativo: muestra magnitud y dirección.

Cohen's d me da 0.3, ¿cómo lo explico al cliente?

"Un efecto pequeño-mediano". Operacionalmente: si dibujás las dos distribuciones, el ≈ 55 % de los individuos del grupo tratamiento superan a la mediana del grupo control (vs 50 % bajo H). Para una conversión, traducilo a "incremento absoluto de X puntos porcentuales".

Referencias

- ISLP, cap. 13 — Multiple Testing, intro al p-value.
- Bruce & Bruce, cap. 3 — Statistical Experiments and Significance Testing.
- Delacre, Lakens & Leys (2017), Why Psychologists Should by Default Use Welch's t-test, IRSP.
- Cohen, J. (1988), Statistical Power Analysis for the Behavioral Sciences — referencia canónica de effect size.
- pingouin docs — Vallat, R. (2018), Pingouin: statistics in Python, JOSS.
- scipy.stats.ttest_ind — atención a equal_var y alternative.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 177 — Clase 177 — Effect size dedicado: Cohen's d, Hedges' g, Cliff's δ con pingouin

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Cohen (1988) + Lakens (2013) + Vallat (2018) pingouin. Duración estimada: 75 min.*

Objetivo

Dominar effect size —la métrica que el p-value no responde: "cuán grande es la diferencia"—. Cubrir 6 medidas: Cohen's d (means, varianzas similares), Hedges' g (bias-corrected para n chico), Glass's Δ (varianza del control como denominador), Cliff's δ (no paramétrico), r de correlación, odds ratio. Aplicar con pingouin en una sola llamada. Reportar correctamente: APA 7 lo exige.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Calcular Cohen's d a mano y con pingouin.compute_effsize.
- Aplicar la corrección de Hedges (recomendada cuando n < 50/grupo).
- Interpretar magnitudes (Cohen 1988): 0.2 / 0.5 / 0.8 = small / medium / large.
- Calcular Cliff's δ para datos ordinales / muy asimétricos.
- Reportar effect size con IC95 % bootstrap.
- Diseñar tabla APA-7 con mean ± SD, Cohen's d [95% CI], t, p.

Temas

- Cohen's d: $(\bar{x} - \bar{x})$ / s_pooled.
- Hedges' g: d · (1 - 3/(4·gl - 1)).
- Glass's Δ: usar s_control como denominator. Útil cuando control y treatment tienen varianza distinta.
- Cliff's δ: probabilistic dominance.
- Effect size para correlación: r (= Pearson) o R^2.
- Effect size para chi-cuadrado: Cramér's V, phi.
- IC95 % de effect size: bootstrap o fórmulas paramétricas.

Definiciones y características

- Cohen's d: estándar para 2 grupos independientes.
- s_pooled: sqrt(((n1-1)·$s1^2$ + (n2-1)·$s2^2$) / (n1+n2-2)).

- Hedges' g: corrección para muestras chicas; siempre menor que d.
- Cliff's δ: en [-1, 1]. Interpretación Romano (2006): < 0.147 negligible, < 0.33 small, < 0.474 medium, ≥ 0.474 large.
- CLES (Common Language Effect Size): P(rand X > rand Y). Más interpretable para no-técnicos.

Dataset / recursos

- seaborn.load_dataset('tips').
- Librerías: pingouin, scipy.stats, numpy.

Ejercicios

1. Cohen's d a mano: para tip por sex, calcular manualmente con s_pooled.
2. pingouin.compute_effsize: verificar contra cálculo manual. Probar eftype='cohen' | 'hedges' | 'glass' | 'CLES'.
3. Hedges' g: con n=10 por grupo, ver diferencia entre d y g (Hedges < d).
4. Cliff's δ: para datos Likert ordinales o muy asimétricos, calcular y interpretar.
5. Effect size + IC: bootstrap del Cohen's d → IC95 %.

Homework verificable

Análisis estadístico riguroso de tips:

1. Comparar tip por time (Lunch/Dinner) y por day.
2. Reportar Cohen's d (o Hedges' g si n < 30) con IC95 % bootstrap.
3. Para day (4 niveles), reportar partial η^2 del ANOVA.
4. Conclusión APA-7 estilo: "M ± SD, t(df) = X, p = Y, d [95% CI]".

Criterio de aceptación: tabla bien formada con todas las columnas; conclusión no usa solo p-value.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Reportar solo p < 0.05	Mala práctica desde 1999. Fix: agregar eff
Cohen's d sobre datos muy asimétricos	Sesgado. Fix: Cliff's δ.
Mismo Cohen's d en 2 estudios sin contexto	Magnitudes 0.2/0.5/0.8 son orientativas —
Effect size sin IC	Incompleto. Fix: bootstrap CI.
Confundir d con η^2	Distintos rangos y significados. Fix: d pa

Preguntas frecuentes

Cohen's d o Hedges' g?

Hedges siempre si n < 50/grupo. Para n grande son indistinguibles.

Magnitudes 0.2/0.5/0.8 son universales?

No. Son orientación. En medicina pueden ser muy distintas. Reporte tu d, dejá que el lector compare con su literatura.

CLES interpretable?

Sí — "65 % chance que una persona random del grupo A tenga mayor valor que una random del B". Comunicable a no-técnicos.

Effect size para A/B testing?

Sí — Cohen's h para proporciones, Cohen's d para continuous. Ver clase 154.

Effect size negativo?

Solo indica dirección. La magnitud |d| es lo que se interpreta.

Referencias

- Cohen (1988), Statistical Power Analysis for the Behavioral Sciences.
- Lakens (2013), Calculating and reporting effect sizes to facilitate cumulative science, Frontiers in Psychology.
- Romano et al. (2006), Cliff's δ interpretation.
- pingouin docs.
- APA Publication Manual 7th ed.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 178 — Clase 178 — Test chi-cuadrado de independencia y bondad de ajuste

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 4 + Bruce & Bruce, cap. 3 Chi-Square Test. Duración estimada: 70 min.*

Objetivo

Aplicar el test chi-cuadrado de Pearson en sus dos formas: (a) independencia en una tabla de contingencia de dos variables categóricas, y (b) bondad de ajuste entre una distribución observada y una teórica. Reconocer cuándo el test es válido (frecuencias esperadas ≥ 5 por celda) y cuándo hay que recurrir a Fisher exact o a la simulación de Monte Carlo.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Construir una tabla de contingencia con pd.crosstab y aplicar scipy.stats.chi2_contingency interpretando chi2, dof, pvalue y expected.
- Verificar el supuesto de frecuencias esperadas mínimas (regla de Cochran: ≥ 5 en ≥ 80 % de celdas).
- Decidir entre chi-cuadrado, Fisher exact (scipy.stats.fisher_exact, tablas 2×2 con conteos chicos) y chi-cuadrado con simulación (lambda_='log-likelihood' o montecarlo).
- Calcular Cramér's V como effect size para tablas r×c (análogo al Cohen's d categórico).
- Aplicar bondad de ajuste con scipy.stats.chisquare para validar dados, ruedas de roulette o conteos en bins.

Temas

#	Tema	Por qué importa
1	Tablas de contingencia	Estructura natural de datos categóricos cr
2	Estadístico $\chi^2 = \Sigma (O - E)^2 / E$	Lo que mide el test: distancia entre obser
3	Grados de libertad (r-1)·(c-1)	Determinan la distribución de referencia.

4	Supuesto de E ≥ 5 (Cochran)	Si se viola, el p-value asintótico es inco
5	Fisher exact para 2×2 con n chico	Alternativa exacta cuando chi² no sirve.
6	Cramér's V	Effect size — qué tan fuerte es la asociac
7	Bondad de ajuste vs independencia	Misma fórmula, distinto problema.

Definiciones y características

- Tabla de contingencia: matriz r×c donde O_{ij} es la frecuencia observada del cruce (fila i, columna j).
- Frecuencia esperada bajo independencia: E_{ij} = (row_i_total · col_j_total) / n. Es lo que esperaríamos si las dos variables fueran independientes.
- Estadístico χ^2: Σ (O_{ij} - E_{ij})² / E_{ij}. Sigue χ^2 con (r-1)·(c-1) gl si las E son suficientemente grandes.
- Test de independencia: H: las variables categóricas son independientes. Se aplica a una tabla cruzada de dos variables.
- Test de bondad de ajuste: H: la muestra proviene de una distribución teórica especificada. Las E vienen de la distribución teórica, no del producto de marginales.
- Cochran's rule: el test asintótico es válido si todas las E ≥ 1 y al menos 80 % de las celdas tienen E ≥ 5. Si no, usar Fisher (2×2) o Monte Carlo (>2×2).
- Fisher exact test: calcula el p-value exacto enumerando todas las tablas con las mismas marginales. Computacionalmente caro para n grande.
- Cramér's V: V = $\sqrt{(\chi^2 / (n \cdot \min(r-1, c-1)))}$. Va de 0 a 1. Interpretación cualitativa con min(r-1, c-1) = 1: 0.1 small, 0.3 medium, 0.5 large (Cohen 1988).
- Test de homogeneidad: matemáticamente idéntico al de independencia, pero el diseño muestral es distinto (se fijan los marginales de una variable). El cálculo y la interpretación práctica son los mismos.

Dataset / recursos

- seaborn.load_dataset('titanic'): cruzar survived × class o survived × sex.
- seaborn.load_dataset('tips'): smoker × day.
- Bondad de ajuste: simular tiradas de un dado posiblemente cargado y testear contra distribución uniforme.
- Librerías: pandas, scipy.stats, pingouin (que tiene pg.chi2_independence con Cramér's V incluido).

Ejercicios

1. Tabla cruzada: pd.crosstab(titanic.survived, titanic['class']). Aplicá chi2, p, dof, expected = scipy.stats.chi2_contingency(tabla). Reportá los cuatro valores e interpretá.
2. Effect size: calculá Cramér's V manualmente: V = sqrt(chi2 / (n * min(r-1, c-1))). Verificá contra pingouin.chi2_independence(titanic, x='survived', y='class').
3. Cochran check: imprimí la matriz expected y contá cuántas celdas tienen E < 5. Si supera el 20 %, recalculá con chi2_contingency(tabla, lambda_='log-likelihood') (G-test, mejor para celdas chicas).
4. Fisher exact (2×2): tomá la subtabla survived × sex y aplicá scipy.stats.fisher_exact(tabla_2x2). Comparalo con chi-cuadrado.
5. Bondad de ajuste: simulá rng = np.random.default_rng(7); tiros = rng.choice([1,2,3,4,5,6], size=600, p=[0.18, 0.16, 0.17, 0.17, 0.16, 0.16]). Hipótesis: el dado es justo (p uniforme). Aplicá scipy.stats.chisquare(observado, f_exp=esperado) con esperado = [100]*6. ¿Rechazás H?

Homework verificable

Notebook que sobre titanic:

1. Cruza survived × class y survived × sex por separado.
2. Para cada cruce: chi², gl, p-value, Cramér's V.

3. Identifica cuál de los dos tiene asociación más fuerte (mayor V) y cuál tiene evidencia estadística más fuerte (menor p).
4. En 3 líneas, explica por qué p y V pueden ordenar distinto cuando n cambia entre comparaciones.

Criterio de aceptación: ambos tests deben rechazar H al 5 %; Cramér's V debe ser mayor para sex que para class; la conclusión debe distinguir tamaño de efecto de evidencia.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
expected tiene celdas con valores < 5 y re	El p-value asintótico no es confiable. Fix
Aplico chi² sobre una columna numérica con	Chi² es solo para categóricas / conteos. F
Confundo "asociación" con "causalidad" por	Chi² detecta dependencia estadística, no r
Reporto solo p < 0.05 con n = 10^5 y declar	Con n gigante, asociaciones triviales son
Bondad de ajuste con f_exp proporcional pe	chisquare(obs, f_exp) requiere que f_exp.s

Preguntas frecuentes

¿Cuándo Fisher exact y cuándo chi-cuadrado?

Fisher cuando alguna E < 5 en una tabla 2×2. Para tablas mayores, Fisher es costoso; mejor usar Monte Carlo simulation (scipy.stats.chi2_contingency(..., method='monte-carlo') desde scipy 1.11) o G-test.

¿Qué es la corrección de Yates?

Para tablas 2×2, resta 0.5 al |O - E| antes de elevar al cuadrado. Hace el test más conservador. scipy.stats.chi2_contingency la aplica por default en 2×2 (correction=True). En la práctica, con n moderado, casi no cambia el p-value.

¿Por qué dof = (r-1)·(c-1)?

Porque al fijar los totales marginales (r+c-1 restricciones), solo (r-1)·(c-1) celdas son libres de variar — las otras quedan determinadas por aritmética.

¿G-test es mejor que chi-cuadrado?

Asintóticamente equivalentes, pero G-test tiene mejor comportamiento con celdas chicas y se generaliza mejor (es el log-likelihood ratio test). En scipy: chi2_contingency(tabla, lambda_='log-likelihood').

¿Puedo usar chi-cuadrado para validar la salida de un modelo de clasificación?

Sí — crosstab(y_true, y_pred) da la confusion matrix, y un chi² sobre esa tabla testea si la predicción es independiente de la verdad (H: clasificador trivial). Pero ojo: prefieres métricas específicas (accuracy, F1, Kappa de Cohen) — chi² no captura el balance de clases.

Referencias

- ISLP, cap. 4 — Classification, parte sobre datos categóricos.
- Bruce & Bruce, cap. 3 — sección Chi-Square Test.
- Cochran, W.G. (1954), Some Methods for Strengthening the Common χ² Tests, Biometrics.
- scipy.stats.chi2_contingency y fisher_exact.
- pingouin.chi2_independence — reporta Cramér's V automáticamente.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.

- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 179 — Clase 179 — ANOVA (one-way, two-way)

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 13 + Bruce & Bruce, cap. 3 ANOVA. Duración estimada: 80 min.*

Objetivo

Que el alumno aplique ANOVA de una vía (≥ 3 grupos, una variable categórica) y ANOVA de dos vías (dos factores categóricos + interacción), entienda por qué no se hacen "t-tests todos contra todos" (inflación de α) y sepa hacer post-hoc con Tukey HSD. Reconocer los supuestos (independencia, normalidad por grupo, homogeneidad de varianzas) y cuándo usar la alternativa robusta Welch ANOVA o el no paramétrico Kruskal-Wallis (Clase 150).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Plantear H: μ = μ = ... = μ_k vs H: al menos uno difiere y aplicar scipy.stats.f_oneway o pingouin.anova.
- Interpretar F = MS_between / MS_within y su relación con la F-distribution (F(k-1, n-k)).
- Aplicar Welch's ANOVA (pingouin.welch_anova) cuando se viola la homogeneidad de varianzas (Levene rechaza).
- Hacer Tukey HSD post-hoc con pingouin.pairwise_tukey y leer los IC ajustados.
- Distinguir efectos principales de interacción en ANOVA two-way y graficar interaction plots con seaborn.pointplot.
- Reportar η² o ω² como effect size de ANOVA.

Temas

- ¿Por qué no t-tests múltiples? Si hacés 10 t-tests al α=0.05, la probabilidad de al menos un falso positivo es ≈ 40 %.
- Descomposición de varianza: SS_total = SS_between + SS_within.
- F-statistic: razón entre varianza explicada por los grupos y varianza residual.
- Supuestos: independencia, normalidad (Shapiro por grupo o residuos), homocedasticidad (Levene/Bartlett).
- Welch ANOVA — análogo a Welch's t-test para ≥ 3 grupos.
- Post-hoc: Tukey HSD (controla family-wise error rate), Bonferroni, Holm.
- Two-way ANOVA: efectos principales A, B, e interacción A×B.
- Effect size: η² (eta-squared), ω² (omega-squared, menos sesgado).

Definiciones y características

- One-way ANOVA: testea si la media de una variable continua difiere entre los niveles de una variable categórica.
- Two-way ANOVA: dos variables categóricas. Permite testear 3 cosas: efecto principal de A, efecto principal de B, e interacción (¿el efecto de A depende del nivel de B?).
- F-statistic: MS_between / MS_within. Si los grupos tienen la misma media, ambos son estimadores de σ² → F ≈ 1. Si difieren, MS_between crece.

- SS (sum of squares) within: variabilidad residual dentro de cada grupo. Σ (x_ij - x̄_i)².
- SS between: variabilidad de las medias grupales respecto a la gran media. Σ n_i · (x̄_i - x̄)².
- Homocedasticidad: igualdad de varianzas entre grupos. Test: Levene (robusto), Bartlett (sensible a normalidad).
- Welch ANOVA: no asume homocedasticidad. Es el default razonable moderno, igual que Welch's t-test.
- Tukey HSD (Honestly Significant Difference): compara todas las parejas controlando el family-wise error rate al α global.
- Interacción: cuando el efecto de un factor cambia según el nivel del otro. En el plot, las líneas no son paralelas.
- η² (eta-squared): SS_between / SS_total. Proporción de varianza explicada. Sesgado hacia arriba.
- ω² (omega-squared): corrige el sesgo de η². Recomendado para reportar.

Dataset / recursos

- seaborn.load_dataset('tips'): total_bill por day (one-way), o por day × time (two-way).
- seaborn.load_dataset('penguins'): body_mass_g por species (one-way claro).
- Librerías: scipy.stats, pingouin, statsmodels.api as sm, statsmodels.formula.api as smf.

Ejercicios

1. One-way: aplicá scipy.stats.f_oneway(*[grupo for grupo in penguins.groupby('species').body_mass_g]). Reportá F, p y dof_between, dof_within.
2. Supuestos: testá normalidad por grupo (pingouin.normality(penguins, dv='body_mass_g', group='species')) y homocedasticidad (pingouin.homoscedasticity). Si Levene rechaza, repetí con pingouin.welch_anova.
3. Post-hoc Tukey: pingouin.pairwise_tukey(data=penguins, dv='body_mass_g', between='species'). Identificá qué pares de especies difieren significativamente.
4. Two-way con interacción: pingouin.anova(data=tips, dv='total_bill', between=['day', 'time']). Mirá las tres filas de la tabla (day, time, day×time).
5. Interaction plot: sns.pointplot(data=tips, x='day', y='total_bill', hue='time'). Si las líneas se cruzan o no son paralelas → hay interacción visual; cruzala con el p-value del término day*time.

Homework verificable

Sobre penguins (filtrando filas con NA):

1. ANOVA one-way de flipper_length_mm por species.
2. Chequear Levene y Shapiro; decidir entre ANOVA clásico y Welch ANOVA, justificando.
3. Tukey HSD post-hoc.
4. Reportar ω² (pingouin.anova(... effsize='n2', y calcular ω² manualmente).
5. Conclusión en 3 líneas: qué pares difieren, magnitud del efecto general (η²/ω²), si hay alguna comparación dudosa.

Criterio de aceptación: el ANOVA debe rechazar H (p < 0.001), Tukey debe mostrar las tres parejas significativas, y η² debe ser > 0.5 (efecto grande — las tres especies tienen aletas claramente distintas).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Hago ttest_ind entre cada par de 4 grupos	Con 4 grupos son 6 comparaciones; α efecti
ANOVA da p < 0.05 pero ningún Tukey lo da	Puede pasar con varianzas muy distintas. F
Aplico ANOVA con varianzas muy distintas (	El F-test clásico es sensible a esto, espe
Los residuos no son normales y reporto ANO	Si los grupos son grandes (n ≥ 30 c/u), TC

Interpreto "efecto de A" sin mirar la inte	Si hay interacción, el efecto principal de

Preguntas frecuentes

¿ANOVA o regresión lineal con dummies?

Son matemáticamente equivalentes. ANOVA es la presentación tradicional; OLS con dummies (statsmodels.formula.api.ols('y ~ C(species)', data).fit()) da los mismos F y p. La regresión también te da los coeficientes de cada nivel respecto a la categoría de referencia, lo cual es más informativo.

¿Tukey o Bonferroni post-hoc?

Tukey es mejor para todas-las-parejas porque controla el family-wise error rate de manera específica para ANOVA (es uniformemente más poderoso que Bonferroni en ese caso). Bonferroni es más conservador (peor poder). Ver Clase 151 para alternativas modernas (Holm, BH).

¿Qué pasa si los grupos están desbalanceados (n distinto)?

ANOVA aguanta moderadamente, pero con varianzas distintas el F-test se rompe. Welch ANOVA lo maneja bien.

¿Two-way ANOVA cuando alguna celda tiene n=0?

Modelo no balanceado: usar Type III SS (statsmodels lo soporta vía sm.stats.anova_lm(model, typ=3)). Type I (default) depende del orden de los factores en la fórmula.

¿Y si tengo medidas repetidas (mismo sujeto en cada nivel)?

pingouin.rm_anova (repeated measures ANOVA). Modela la correlación intra-sujeto, lo cual ANOVA clásico ignora.

Referencias

- ISLP, cap. 13 — sección sobre ANOVA y multiple testing.
- Bruce & Bruce, cap. 3 — sección ANOVA.
- Welch, B.L. (1951), On the comparison of several mean values: an alternative approach, Biometrika.
- scipy.stats.f_oneway, pingouin.welch_anova, pingouin.pairwise_tukey.
- statsmodels — ANOVA (para Type II/III SS y modelos mixtos).

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 180 — Clase 180 — Tests no paramétricos: Mann-Whitney, Wilcoxon, Kruskal-Wallis

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Bruce & Bruce, cap. 3 Resampling and Non-parametric Tests + Conover, Practical Nonparametric Statistics. Duración estimada: 70 min.*

Objetivo

Aplicar las tres alternativas no paramétricas más usadas: Mann-Whitney U (= dos muestras independientes,

análogo a Welch's t), Wilcoxon signed-rank (= pareado, análogo a ttest_rel) y Kruskal-Wallis (= ≥ 3 grupos, análogo a ANOVA one-way). Saber cuándo elegirlos sobre los paramétricos: muestras chicas con datos visiblemente asimétricos, datos ordinales (Likert, ranks), o presencia de outliers extremos.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Reconocer las 3 situaciones en que un test no paramétrico es preferible al paramétrico (n chico + asimetría, ordinal, outliers).
- Aplicar scipy.stats.mannwhitneyu, wilcoxon, kruskal con los argumentos correctos (alternative, method='exact' vs 'asymptotic').
- Interpretar que los no paramétricos testean distribuciones (estocásticamente iguales) o medianas, no medias.
- Reportar effect size no paramétrico: rank-biserial correlation (Mann-Whitney) o ε^2 / η^2_H (Kruskal-Wallis).
- Hacer post-hoc no paramétrico tras Kruskal con Dunn's test (scikit-posthocs) y corrección por múltiples comparaciones.

Temas

#	Test	Reemplaza a	Para qué
1	Mann-Whitney U (Wilcoxon rank-s	ttest_ind Welch	2 grupos independientes
2	Wilcoxon signed-rank	ttest_rel	Pareado / 1 muestra contra mediana
3	Kruskal-Wallis H	ANOVA one-way	≥ 3 grupos independientes
4	Dunn's test (post-hoc)	Tukey HSD	Pares post Kruskal
5	Cliff's δ / rank-biserial r	Cohen's d	Effect size no paramétrico

Definiciones y características

- Test no paramétrico: no asume forma específica de la distribución (no requiere normalidad). Trabaja sobre rangos de los datos.
- Mann-Whitney U: para cada par (x_i, y_j) cuenta cuántas veces $x_i > y_j$. Bajo H de distribuciones iguales, U tiene distribución conocida. H: $P(X > Y) = P(X < Y) = 0.5$.
- Wilcoxon signed-rank: para datos pareados o una muestra contra mediana hipotética. Calcula la diferencia d_i, las rankea por $|d_i|$, suma rangos con signo. Asume simetría de la distribución de diferencias.
- Kruskal-Wallis H: extiende Mann-Whitney a k grupos. $H = (12 / (n(n+1))) \cdot \Sigma R_i^2/n_i - 3(n+1)$. Bajo H (todas las distribuciones iguales), $H \sim \chi^2(k-1)$ asintóticamente.
- Rank-biserial correlation $r_{rb} = 1 - 2U / (n \cdot n)$: effect size para Mann-Whitney. Va de -1 a 1.
- Cliff's δ: equivalente, $\delta = (\#(x_i > y_j) - \#(x_i < y_j)) / (n \cdot n)$. Interpretación: < 0.147 small, < 0.33 medium, ≥ 0.474 large (Romano et al. 2006).
- Dunn's test: comparaciones pareadas no paramétricas tras Kruskal-Wallis, basadas en la diferencia promedio de rangos. Se ajusta por múltiples tests (Bonferroni, BH).

Dataset / recursos

- seaborn.load_dataset('tips'): tip por sex o day.
- Datos con outliers: precios de Airbnb (Kaggle) — cola larga a la derecha por mansiones.
- Likert ordinal: simular respuestas 1–5 con rng.choice([1,2,3,4,5], p=...).
- Librerías: scipy.stats, pingouin, scikit-posthocs (pip install scikit-posthocs).

Ejercicios

1. Mann-Whitney: comparar tip entre sex con scipy.stats.mannwhitneyu(a, b, alternative='two-sided'). Compará el p con el del t-test del ejercicio 2 de la Clase 147. Calculá rank-biserial r.
2. Wilcoxon signed-rank: con el dataset simulado de presión arterial antes/después de la Clase 147, aplicá scipy.stats.wilcoxon(antes, despues). Verificá supuesto de simetría con un histograma de las diferencias.
3. Outliers: a un dataset normal rng.normal(50, 5, 100) agregale 3 outliers de valor 200. Compará Welch's t-test vs Mann-Whitney contra otro grupo normal — el Mann-Whitney es mucho más robusto.
4. Kruskal-Wallis: aplicalo a body_mass_g por species en penguins. Comparalo con el ANOVA de la Clase 149.
5. Post-hoc Dunn: con scikit_posthocs.posthoc_dunn(penguins, val_col='body_mass_g', group_col='species', p_adjust='holm') identificá qué pares difieren.

Homework verificable

Tomar el dataset de Airbnb por neighborhood (o un sintético equivalente con cola larga):

1. Verificar normalidad por grupo (Shapiro o KS). Mostrar que se rechaza.
2. Comparar precio entre 4 vecindarios con Kruskal-Wallis.
3. Post-hoc Dunn con corrección Holm.
4. Reportar mediana ± IQR por grupo (no mean ± SD, que es engañoso con asimetría).
5. Comparar conclusiones con las que daría un ANOVA clásico ingenuo.

Criterio de aceptación: el reporte debe usar mediana/IQR (no media/SD), identificar al menos un par significativo tras Dunn-Holm, y explicar en 2 líneas por qué ANOVA sería sospechoso aquí (cola larga inflando la varianza del grupo con outliers).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Aplico Mann-Whitney y reporto "la media di	El test no es sobre medias; es sobre la pr
Aplico Wilcoxon a diferencias muy asimétri	El signed-rank asume simetría de las difer
Uso Mann-Whitney con n=10^6 y se vuelve len	Es O(n log n) por el ranking, pero scipy l
Reporto Kruskal-Wallis sin post-hoc y digo	Kruskal solo te dice que al menos uno difi
Aplico no paramétrico "para ir a la segura	El t-test tiene más poder cuando sus supue

Preguntas frecuentes

¿Mann-Whitney testea medianas?

Solo si las distribuciones tienen la misma forma (mismo shape, distinto location). En general testea $P(X > Y) \neq 0.5$, que es una afirmación sobre superioridad estocástica, no sobre la mediana per se.

¿Wilcoxon o test de signos?

Wilcoxon usa magnitud de las diferencias (rangos) → más poderoso. Test de signos solo usa la dirección (positivo/negativo) → más robusto pero menos poderoso. Si dudás de la simetría, signos.

¿Cuánto poder pierdo usando no paramétrico cuando los supuestos se cumplen?

Para Mann-Whitney vs t-test con datos normales, la eficiencia asintótica relativa es $3/\pi \approx 0.955$ — perdés ≈ 5 % de poder. Si los datos son no normales, podés ganar mucho. Por eso "no paramétrico por default" no es una mala estrategia para n chico.

¿alternative='greater' significa lo mismo que en t-test?

Sí, pero referido a la dirección de la dominancia estocástica (no a la media). alternative='greater' en Mann-Whitney ↔ "la distribución de X tiende a producir valores mayores que la de Y".

¿Qué hago con datos ordinales (Likert 1–5)?

No paramétrico siempre. Ranks son la operación natural sobre escalas ordinales. Mann-Whitney para 2 grupos, Kruskal-Wallis para ≥ 3.

Referencias

- Bruce & Bruce, cap. 3 — Resampling and Non-parametric Tests.
- Conover, W.J. (1999), Practical Nonparametric Statistics (3rd ed.) — referencia canónica.
- Romano et al. (2006) — interpretación de Cliff's δ.
- scipy.stats.mannwhitneyu, wilcoxon, kruskal.
- scikit-posthocs — Dunn, Conover, Nemenyi.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 181 — Clase 181 — Corrección de comparaciones múltiples (Bonferroni, FDR)

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 13 Multiple Testing + Benjamini & Hochberg (1995). Duración estimada: 70 min.*

Objetivo

Entender por qué hacer 100 tests al α=0.05 produce ≈ 5 falsos positivos esperados aunque todas las H sean verdaderas, y aplicar las dos familias de corrección: family-wise error rate (FWER) con Bonferroni y Holm, y false discovery rate (FDR) con Benjamini-Hochberg (BH). Saber elegir entre ambas según el contexto (medicina/seguridad → FWER; screening exploratorio → FDR).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Cuantificar la inflación de α al hacer k tests independientes: 1 - (1-α)^k.
- Aplicar Bonferroni: α_corregido = α / m. Conservador pero simple.
- Aplicar Holm-Bonferroni (statsmodels.stats.multitest.multipletests(..., method='holm')) — uniformemente más poderoso que Bonferroni.
- Aplicar Benjamini-Hochberg (BH/FDR) y entender que controla la proporción esperada de falsos positivos entre los rechazos, no el FWER.
- Distinguir FWER (P[al menos 1 falso positivo] ≤ α) de FDR (E[V/R] ≤ q, donde V son falsos positivos y R rechazos totales).
- Reportar p-values ajustados (q-values) y umbrales claros.

Temas

- El problema: si m=20 tests independientes con H verdadera y α=0.05, P(al menos uno rechaza) = 1 -

$0.95^{20} \approx 64$ %.

- FWER: probabilidad de al menos 1 falso positivo en toda la familia.
- FDR: proporción esperada de falsos positivos entre los rechazos (no entre todos los tests).
- Bonferroni: rechazar si $p_i \leq \alpha/m$. Controla FWER exactamente.
- Holm: ordenar p-values y comparar $p_{(i)} \leq \alpha/(m-i+1)$. Uniformemente más poderoso que Bonferroni.
- Benjamini-Hochberg (BH): ordenar $p_{(1)} \leq ... \leq p_{(m)}$; rechazar todos los $p_{(i)}$ tales que $p_{(i)} \leq (i/m)\cdot q$. Controla FDR a nivel q.
- Cuándo usar cada uno: FWER si un falso positivo es catastrófico (drug approval, security). FDR si esperás muchos descubrimientos verdaderos y querés tolerar algunos falsos (genómica, A/B testing masivo).

Definiciones y características

- Family-Wise Error Rate (FWER): P(rechazar al menos una H verdadera). Sin corrección, crece con m. Bonferroni y Holm lo acotan.
- False Discovery Rate (FDR): E[V / max(R, 1)] donde V = falsos positivos, R = total de rechazos. Concepto introducido por Benjamini & Hochberg (1995). Mucho menos restrictivo que FWER.
- Bonferroni: divide α por m. Si m=100 y α=0.05, cada test usa α_local=0.0005. Conservador, pierde poder con m grande.
- Šidák: α_corregido = 1 - (1-α)^(1/m). Marginalmente menos conservador que Bonferroni; asume independencia.
- Holm-Bonferroni (1979): step-down. Ordena p-values, compara el más chico contra α/m, el siguiente contra α/(m-1), etc. Uniformemente mejor que Bonferroni.
- BH (Benjamini-Hochberg): step-up. Ordena, encuentra el mayor i tal que $p_{(i)} \leq (i/m)\cdot q$, rechaza todos hasta ese. Controla FDR si los tests son independientes o tienen positive regression dependence (BY si la dependencia es arbitraria).
- q-value: p-value ajustado bajo FDR. Interpretación: "si rechazo todo con q ≤ 0.05, espero que ≤ 5 % de mis rechazos sean falsos".

Dataset / recursos

- Genómica sintética: simular m=1000 tests, m=950 nulos verdaderos y m=50 alternativos. Generar p-values y mostrar comportamiento de cada método.
- A/B testing real: 20 métricas testeadas a la vez → controlar familia.
- Librerías: statsmodels.stats.multitest, pingouin.multicomp, scipy.stats.

Ejercicios

1. Inflación de α: simulá 10 000 experimentos. En cada uno, hacé 20 tests con H verdadera (scipy.stats.ttest_ind entre dos grupos N(0,1), n=30). Contá en qué % al menos 1 da p<0.05. Verificá que ≈ 64 %.
2. Bonferroni: con un vector pvals de 20 p-values, calculá pvals_adj = np.minimum(pvals * 20, 1) y compará contra multipletests(pvals, method='bonferroni').
3. Holm: multipletests(pvals, alpha=0.05, method='holm'). Comparar cuántos rechaza vs Bonferroni con el mismo vector.
4. BH/FDR: genera 1000 p-values, 950 de Uniform(0,1) y 50 de Beta(0.5, 5) (concentrados cerca de 0 — alternativos). Aplicá multipletests(pvals, alpha=0.05, method='fdr_bh'). Contá cuántos rechaza y estimá el FDR empírico (rechazos del primer grupo / total rechazos).
5. Comparación: mismo vector del ej. 4, aplicar Bonferroni, Holm y BH. Tabla con: # rechazos, % de los 50 verdaderos descubiertos (recall), FDR empírico. Verificá que BH descubre mucho más con FDR controlado.

Homework verificable

Sobre un dataset con 30 features y un target binario (ej.: load_breast_cancer):

1. Para cada feature, t-test entre la clase 0 y la clase 1.
2. Aplicar tres correcciones: Bonferroni, Holm, BH (q=0.05).
3. Tabla comparativa: # features significativas según cada método.
4. Justificar en 3 líneas qué método elegirías si: (a) vas a publicar los hallazgos en un paper médico, (b) usás esto como screening para una etapa siguiente.

Criterio de aceptación: BH debe descubrir más features que Holm, que descubre ≥ que Bonferroni. La justificación debe mencionar que (a) → FWER (Bonferroni/Holm), (b) → FDR (BH).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar	
Hago 50 t-tests sin corrección y reporto l	Cherry-picking estadístico. Fix: BH como m	
Aplico Bonferroni con m=1000 y no rechaza	Demasiado conservador para screening. Fix:	
Aplico BH a tests no independientes (ej.:	BH clásico asume independencia o PRDS. Fi	
Reporto p-value ajustado como "probabilida	El p-value ajustado sigue siendo un p-valu	datos)`. Fix: usar lenguaje preciso ("cont
Decido cuál corrección usar después de ver	También es p-hacking. Fix: pre-especificar	

Preguntas frecuentes

¿Cuándo Bonferroni y cuándo BH?

Regla simple: Bonferroni cuando el costo de un falso positivo es alto (medicina, seguridad, decisiones binarias). BH cuando hacés screening y aceptás cierta proporción de FP entre los hallazgos para no perder verdaderos (genómica, A/B testing masivo, feature selection).

¿Por qué Holm es siempre mejor que Bonferroni?

Porque rechaza al menos los mismos tests y a veces más, sin perder control del FWER. La única razón para usar Bonferroni es simplicidad pedagógica.

¿BH controla FDR exactamente al 5 %?

Controla FDR ≤ q · (m/m), donde m es el número de H verdaderas. En la práctica, m ≤ m, así que FDR ≤ q. Si esperás muchos nulos, BH es un poco más conservador de lo que parece.

¿Storey's q-value es lo mismo que BH?

Es una versión adaptativa: estima π = m/m de los datos y corrige menos cuando hay muchos rechazos esperados. En genómica es estándar; en data science general, BH alcanza.

¿Tengo que corregir si los tests son sobre datasets distintos?

Si forman parte del mismo objetivo de inferencia (la misma "familia"), sí. Si son análisis independientes con conclusiones separadas, no. El límite es subjetivo; pre-registrar la familia ayuda.

Referencias

- ISLP, cap. 13 — Multiple Testing.
- Benjamini, Y. & Hochberg, Y. (1995), Controlling the False Discovery Rate, JRSS Series B.
- Holm, S. (1979), A Simple Sequentially Rejective Multiple Test Procedure, Scandinavian Journal of Statistics.

- statsmodels.stats.multitest.multipletests — todos los métodos en una sola función.
- Efron, B. (2010), Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 182 — Clase 182 — Intervalos de confianza

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 5 + Bruce & Bruce, cap. 2 Confidence Intervals. Duración estimada: 70 min.*

Objetivo

Construir e interpretar correctamente intervalos de confianza para media (t-based, z-based, bootstrap) y proporción (Wald, Wilson, Clopper-Pearson), entendiendo que un IC95 % NO significa "95 % de probabilidad de que el parámetro caiga en el intervalo" sino "si repitiéramos el experimento muchas veces, el 95 % de los intervalos construidos contendrían el parámetro". Saber elegir el método según n y la métrica.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Construir un IC para la media usando la distribución t: $\bar{x} \pm t_{\alpha/2, n-1} \cdot (s/\sqrt{n})$ con scipy.stats.t.interval.
- Construir un IC para la proporción con tres métodos y entender cuándo cada uno falla (Wald falla con p cerca de 0/1; Wilson y Clopper-Pearson son robustos).
- Usar scipy.stats.bootstrap (≥ 1.7) para IC sin supuestos paramétricos (anticipa Clase 153).
- Interpretar correctamente la frase "intervalo de confianza al 95 %" (es una propiedad del procedimiento, no del intervalo específico).
- Relacionar IC y test de hipótesis: si el IC95 % de la diferencia no incluye 0, el test bilateral al α=5 % rechaza H.

Temas

- IC para la media (varianza desconocida): t de Student con n-1 gl.
- IC para la media (varianza conocida o n grande): z.
- IC para proporción: Wald ($\hat{p} \pm z \cdot \sqrt{\hat{p}(1-\hat{p})/n}$) vs Wilson score (recomendado por Agresti & Coull 1998) vs Clopper-Pearson (exacto, conservador).
- IC bootstrap percentil (anticipa Clase 153).
- IC del odds ratio, riesgo relativo (medicina/epidemiología).
- Margen de error (ME = z·SE) y cómo determina n.

Definiciones y características

- Intervalo de confianza (IC) al 1-α: par (L, U) calculado a partir de la muestra tal que, sobre repeticiones del experimento, $P(L \le \theta \le U) = 1-\alpha$. El parámetro θ es fijo (no aleatorio); lo aleatorio son L y U.
- Cobertura nominal vs real: la cobertura "nominal" es 95 %; la "real" depende del método y la distribución. Wald infla error con n chico o p extrema.
- Standard error (SE) de la media: s / √n. Se reduce con √n — para reducir el SE a la mitad necesitás 4×

la muestra.
- t-distribution: para IC de la media cuando estimás σ. Tiene colas más anchas que la normal, lo que infla correctamente el IC con n chico.
- Wilson score interval: usa la fórmula $(\hat{p} + z^2/(2n) \pm z\cdot\sqrt{(\hat{p}(1-\hat{p})/n + z^2/(4n^2))}) / (1 + z^2/n)$. Mantiene cobertura ≈ nominal incluso con $\hat{p}$ cerca de 0 o 1.
- Clopper-Pearson: intervalo exacto basado en la distribución binomial. Garantiza cobertura ≥ nominal (puede ser conservador).
- Bootstrap percentile interval: cuantiles α/2 y 1-α/2 de la distribución bootstrap del estadístico. No requiere supuestos paramétricos.
- Margen de error (ME): mitad del ancho del IC. Para n requerido al planificar un estudio: $n = (z\cdot\sigma/ME)^2$.

Dataset / recursos

- seaborn.load_dataset('tips'): IC de la propina media.
- Encuesta sintética: simular n=500 respuestas binarias con p=0.03 (proporción chica → Wald falla).
- Librerías: scipy.stats, statsmodels.stats.proportion, pingouin.

Ejercicios

1. IC t para la media: con tips.total_bill, calculá el IC95 % con scipy.stats.t.interval(0.95, n-1, loc=mean, scale=sem). Verificá contra pingouin.compute_bootci.
2. IC para proporción extrema: con rng.binomial(1, 0.03, 100) (proporción de eventos raros), calculá IC con statsmodels.stats.proportion.proportion_confint(count, n, method='normal') (Wald), 'wilson' y 'beta' (Clopper-Pearson). Observá cómo Wald da límite inferior negativo (¡imposible!) y los otros dos no.
3. Cobertura empírica: simulá 5 000 muestras de tamaño 30 de N(50, 10). Para cada una, construí IC95 % t. Contá qué % contiene μ=50. Debería ser ≈ 95 %.
4. Bootstrap IC: con tips.total_bill, aplicá scipy.stats.bootstrap((tips.total_bill,), statistic=np.mean, n_resamples=10_000, method='percentile'). Compará con el IC t.
5. Sample size: querés estimar una proporción con margen de error de ±2 %, asumiendo $\hat{p} \approx 0.5$ (peor caso). Calculá el n requerido para 95 % de confianza.

Homework verificable

Diseñar un estudio para estimar la proporción de clientes satisfechos en una tienda:

1. Determinar n requerido para margen de error ±3 % con 95 % de confianza asumiendo $\hat{p} \approx 0.5$.
2. Simular el experimento con esa n y p_verdadera=0.78.
3. Construir los 3 IC (Wald, Wilson, Clopper-Pearson) y comparar anchos.
4. En 3 líneas: justificar cuál reportarías y por qué.

Criterio de aceptación: n ≈ 1068. Los 3 IC contienen 0.78. La justificación debe mencionar que Wilson tiene buen comportamiento general (cobertura ≈ nominal con menos ancho que Clopper-Pearson) y es el recomendado actual.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar	
"Hay 95 % de probabilidad de que μ esté en	Interpretación frecuentista incorrecta. Fi	datos) = 0.95").
Wald da IC con límite negativo para propor	Pasa con $\hat{p}$ cerca de 0/1 o n chico. Fix: W	
IC del 95 % se interpreta como "el dato ca	No, eso sería un intervalo de predicción (	
Construyo IC asumiendo normalidad con n=8	t.interval requiere normalidad o n grande.	
Comparo dos IC: "se solapan, no hay difere	Solapamiento de ICs no implica p > 0.05. P	

Preguntas frecuentes

¿Por qué a veces uso z y a veces t?

z cuando conocés σ poblacional (raro) o n ≥ 30 (TCL hace que la diferencia sea trivial). t cuando estimás σ con la muestra y n es chico. En la práctica moderna, siempre t (con n grande coincide con z, así que no perdés nada).

¿Wilson o Clopper-Pearson para proporciones?

Wilson por default (Agresti & Coull 1998, Brown et al. 2001 lo recomiendan). Clopper-Pearson si necesitás garantizar cobertura ≥ nominal (FDA, ensayos clínicos).

¿Bootstrap siempre es mejor?

No siempre. Si tus supuestos paramétricos se cumplen, t-based es más eficiente (intervalos un poco más cortos). Bootstrap brilla con n chico no normal, estadísticos no estándar (mediana, percentil, R^2), o estimadores complejos donde no hay fórmula cerrada.

¿IC95 % es siempre simétrico?

Para la media t-based, sí. Para proporciones y bootstrap, no — sobre todo cerca de los bordes. Eso es una característica, no un bug: refleja la asimetría real de la distribución muestral.

¿Cómo le explico el IC al cliente sin entrar en frecuentismo?

"Si repitiéramos el experimento muchas veces con muestras del mismo tamaño, el 95 % de los rangos que produciríamos contendrían el valor real. Este rango es uno de esos 95 % en promedio." O directamente: "el valor real está plausiblemente entre L y U; rangos más angostos requieren más datos".

Referencias

- ISLP, cap. 5 — Resampling Methods.
- Bruce & Bruce, cap. 2 — Confidence Intervals.
- Agresti, A. & Coull, B. (1998), Approximate is Better than 'Exact' for Interval Estimation of Binomial Proportions, American Statistician.
- Brown, Cai & DasGupta (2001), Interval Estimation for a Binomial Proportion, Statistical Science — review de métodos.
- statsmodels.stats.proportion.proportion_confint.
- scipy.stats.bootstrap — API moderna.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 183 — Clase 183 — Bootstrap y permutation tests

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: ISLP, cap. 5 Resampling Methods + Efron & Tibshirani, An Introduction to the Bootstrap. Duración estimada: 85 min.*

Objetivo

Sustituir los supuestos paramétricos (normalidad, homocedasticidad, fórmulas cerradas) por resampling: el bootstrap estima la distribución muestral de cualquier estadístico re-muestreando con reemplazo, y los permutation tests calculan un p-value re-mezclando etiquetas de tratamiento. Aprender a usar las APIs modernas de scipy (bootstrap, permutation_test, ≥ 1.9) y a interpretar las tres variantes de IC bootstrap (percentil, basic, BCa).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Implementar bootstrap a mano: B resamples con reemplazo del mismo tamaño que la muestra original, calcular el estadístico en cada uno, sacar cuantiles α/2 y 1-α/2.
- Usar scipy.stats.bootstrap(data, statistic, n_resamples=9_999, method='BCa') y entender por qué BCa corrige sesgo y asimetría.
- Diseñar un permutation test bilateral con scipy.stats.permutation_test((a, b), statistic, n_resamples=10_000, alternative='two-sided').
- Saber cuándo usar bootstrap (IC de estadísticos no estándar: mediana, R^2, AUC) vs permutación (p-value de comparación entre grupos sin supuestos).
- Reconocer las limitaciones del bootstrap (muestra muy chica n<20, dependencia temporal — usar block bootstrap).

Temas

- Intuición del bootstrap: "tratá la muestra como si fuera la población y resampleá".
- Tres intervalos bootstrap: percentile, basic (reflejado), BCa (bias-corrected + accelerated).
- ¿Cuántos resamples? B = 10_000 para IC95 % (los percentiles 2.5 y 97.5 se estabilizan).
- Permutation test: intercambiar etiquetas de tratamiento bajo H de "no diferencia".
- Diferencia conceptual: bootstrap estima variabilidad del estadístico; permutación produce un p-value exacto condicional a los datos.
- Block bootstrap para series temporales (preserva autocorrelación).
- Complemento moderno: APIs scipy.stats.bootstrap y permutation_test desde scipy 1.9, vectorizadas y con BCa por default.

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 153b — BCa bootstrap y APIs modernas de scipy

Definiciones y características

- Bootstrap (Efron 1979): re-muestreo con reemplazo de la muestra original, del mismo tamaño n. Cada resample produce un valor del estadístico; el conjunto aproxima la distribución muestral.
- B (número de resamples): 1 000 alcanza para SE; 10 000 para IC95 %; 100 000 para colas o p-values pequeños.
- Percentile IC: cuantiles (α/2, 1-α/2) de la distribución bootstrap.
- Basic IC: $(2\cdot\hat{\theta} - q_{1-\alpha/2}, 2\cdot\hat{\theta} - q_{\alpha/2})$. Refleja respecto al estimador puntual.
- BCa IC: percentile ajustado por sesgo (z) y aceleración (a). Default moderno.
- Permutation test: bajo H de no efecto, las etiquetas son intercambiables. Re-mezclar las etiquetas y recalcular el estadístico genera la distribución exacta bajo H.
- p-value de permutación: (# permutations con |stat| ≥ |stat_obs| + 1) / (n_resamples + 1). El +1 se llama corrección de continuidad y evita p=0.

- Block bootstrap: para series temporales, resamplea bloques contiguos en vez de observaciones individuales. Preserva autocorrelación.

Dataset / recursos

- seaborn.load_dataset('diamonds'): bootstrap del mediana del precio por cut.
- Modelos: AUC de un clasificador entrenado — IC bootstrap sobre la AUC en test.
- Librerías: scipy.stats (≥ 1.9), numpy, sklearn.

Ejercicios

1. Bootstrap a mano: para tips.tip, hacé B=10_000 resamples con rng.choice(x, size=len(x), replace=True), calculá la media, sacá los cuantiles 2.5 y 97.5. Verificá contra scipy.stats.bootstrap(..., method='percentile').
2. BCa vs percentile: con datos lognormales rng.lognormal(0, 1, 50), calculá IC de la mediana con method='percentile' y con method='BCa'. Comprobá que BCa es asimétrico hacia la cola derecha (refleja la asimetría real).
3. IC para AUC: entrenar un LogisticRegression en breast cancer, calcular AUC en test. Bootstrap n_resamples=2_000 sobre (y_true_test, y_proba_test) con un statistic que devuelva roc_auc_score. Reportar IC95 % BCa.
4. Permutation test bilateral: con tips.tip por sex, ejecutá scipy.stats.permutation_test y comparalo con el mannwhitneyu de la Clase 150.
5. Cobertura: simulá 1 000 datasets de Exp(1) con n=25. Para cada uno, calculá IC95 % de la mediana con BCa y con percentile. Contá la cobertura empírica. BCa debería estar más cerca de 95 % que percentile.

Homework verificable

Sobre diamonds:

1. Bootstrap BCa (B=10 000) para la mediana de price global.
2. Bootstrap BCa para la diferencia de medianas entre cut='Ideal' y cut='Fair'.
3. Permutation test sobre la misma diferencia, p-value.
4. Reportar mediana ± IC95 % BCa por categoría de cut (un gráfico con 5 puntos + bigotes).
5. Conclusión en 4 líneas: relacionar el p-value de permutación con el hecho de que el IC de la diferencia no incluye 0.

Criterio de aceptación: la diferencia de medianas tiene IC95 % BCa positivo (Ideal < Fair en precio mediano, contraintuitivo — los diamantes Fair son más grandes), p por permutación < 0.001, y la conclusión menciona que IC y p coinciden cualitativamente.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Bootstrap con n_resamples=100 y conclusion	Demasiado poco para IC. Fix: 10 000 mínimo
Aplico bootstrap a una serie temporal sin	Asume independencia → IC demasiado angosto
Reporto p=0 en un permutation test	Significa que ninguna permutación produjo
Uso bootstrap para n=8	Sesga el SE hacia abajo. Fix: bootstrap fu
Bootstrap del R^2 da IC negativo	Pasa cuando hay un mal fit; significa que

Preguntas frecuentes

¿Bootstrap o cross-validation?

Distintos objetivos. CV estima el error de generalización de un modelo. Bootstrap estima la variabilidad de un estimador (cualquiera). Para "qué tan bueno es mi modelo en datos nuevos", CV. Para "qué IC tiene el AUC reportado", bootstrap.

¿Por qué BCa no es siempre el default?

Porque requiere jackknife (n recálculos del estadístico), lo cual es caro para modelos costosos. Para estadísticos baratos (media, mediana), siempre BCa. Para estadísticos caros (AUC de un modelo), percentile o basic puede ser un compromiso aceptable.

¿Permutation test es exacto?

Es exacto condicional a los datos observados: si hicieras todas las permutaciones (no una muestra de 10 000), el p-value sería exacto. Con 10 000 permutaciones, el SE del p-value es pequeño (≈ $\sqrt{(p(1-p)/n)}$).

¿Bootstrap funciona para cualquier estadístico?

No para todos. Falla con estadísticos no suaves (máximo, percentiles extremos en datasets chicos). En esos casos, subsampling o jackknife son alternativas. Para estadísticos suaves (media, mediana, percentiles centrales, regresión), funciona excelente.

¿Y el out-of-bag de Random Forest es bootstrap?

Sí — Random Forest hace bootstrap sobre filas para cada árbol, y las muestras no incluidas (out-of-bag, ≈ 37 % por árbol) se usan para estimar error sin necesidad de CV. Es bootstrap aplicado a ensembles.

Referencias

- ISLP, cap. 5 — Resampling Methods.
- Efron, B. & Tibshirani, R. (1993), An Introduction to the Bootstrap, Chapman & Hall.
- Efron, B. (1987), Better Bootstrap Confidence Intervals, JASA — paper original de BCa.
- DiCiccio & Efron (1996), Bootstrap Confidence Intervals, Statistical Science.
- scipy.stats.bootstrap y permutation_test.
- arch.bootstrap — block bootstrap para series temporales.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 184 — Clase 184 — BCa bootstrap y APIs modernas de scipy

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Efron (1987) BCa + DiCiccio & Efron (1996) + scipy.stats.bootstrap docs. Duración estimada: 75 min.*

Objetivo

Profundizar el BCa (Bias-Corrected and accelerated) bootstrap —el default moderno (Efron 1987)— y las APIs modernas de scipy (scipy.stats.bootstrap ≥ 1.9, scipy.stats.permutation_test ≥ 1.8). Cubrir las correcciones que BCa hace sobre percentile clásico: bias correction (z) y acceleration (a) vía jackknife.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Diferenciar bootstrap percentile vs basic vs BCa.
- Calcular z (bias correction) y a (acceleration) manualmente.
- Aplicar scipy.stats.bootstrap((data,), statistic, method='BCa', n_resamples=10_000).
- Aplicar permutation_test para p-value de comparación de 2 grupos sin paramétrica.
- Reconocer cuándo BCa importa: estadísticos no lineales, distribuciones asimétricas.

Temas

- Percentile bootstrap: cuantiles directos. Sub-cubre con asimetría.
- Basic bootstrap: reflexión 2θ̂ - q_{1-α/2}.
- BCa: corrige bias (z) y aceleración (a vía jackknife).
- Studentized bootstrap: estandariza con SE bootstrap del SE.
- Scipy.stats.bootstrap API.
- Permutation test exacto.

Definiciones y características

- z: Φ¹(P(θ*_b ≤ θ̂)) — fracción de bootstraps por debajo del estimador.
- a (acceleration): estima cómo SE depende del parámetro. Calculado con jackknife.
- BCa interval: percentiles ajustados α, α función de z, a, z_{α/2}.
- scipy.stats.bootstrap desde 1.9: vectorizado, BCa default, IC + SE + bias estimate.
- permutation_test: re-mezcla labels bajo H, calcula stat distribution.

Dataset / recursos

- Datos lognormales sintéticos.
- seaborn.load_dataset('diamonds') para mediana de price.
- Librerías: scipy.stats, numpy, matplotlib.

Ejercicios

1. Tres ICs: para mediana de x = rng.lognormal(0, 1, 100), calcular IC con percentile, basic, BCa. Comparar.
2. z a mano: implementar z = ppf((B_below_θ̂) / B). Verificar contra scipy.
3. a con jackknife: implementar leave-one-out para cada θ̂_(i). Calcular a.
4. Cobertura empírica: 1000 datasets Exp(1), n=25; cobertura percentile vs BCa. BCa más cerca de 95 %.
5. Permutation_test: comparar dos lognormales con tamaño efecto chico. P-value exacto.

Homework verificable

IC del AUC de un clasificador binario:

1. LogisticRegression en breast cancer. AUC en test.
2. Bootstrap BCa de (y_test, y_proba): 5000 resamples.
3. Reportar AUC [BCa 95% CI].
4. Comparar con percentile bootstrap (más estrecho, sub-cubre).

Criterio de aceptación: BCa CI asimétrico (refleja asimetría de AUC cerca de 1.0); más amplio que percentile.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
BCa con n_resamples=100	Inestable. Fix: ≥ 5000, idealmente 10000.

Estadístico no vectorizable lento	Bootstrap es O(B). Fix: vectorized=False e
Bootstrap sobre serie temporal	Asume independencia. Fix: block bootstrap
permutation_test n_resamples=999	Resolución del p-value 1/(n+1). Fix: 10_00
Reportar percentile vs BCa indistintamente	BCa tiene cobertura nominal. Fix: document

Preguntas frecuentes

Cuándo BCa importa?

Con estadísticos sesgados (mediana en asimetría) o n chico. Para media + n grande, percentile basta.

Studentized bootstrap mejor que BCa?

A veces. Requiere SE del SE → bootstrap doble → costoso. BCa es el compromiso pragmático.

Para IC de proporciones?

Wilson o Clopper-Pearson son específicos y mejores que bootstrap genérico.

vectorized=True en scipy?

Si tu statistic acepta axis=, sí — 100× más rápido.

Block bootstrap para series?

scipy no lo tiene; arch.bootstrap.MovingBlockBootstrap sí.

Referencias

- Efron (1987), Better Bootstrap Confidence Intervals, JASA.
- DiCiccio & Efron (1996), Bootstrap Confidence Intervals, Statistical Science.
- scipy.stats.bootstrap.
- scipy.stats.permutation_test.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 185 — Clase 185 — A/B testing: tamaño de muestra, poder estadístico

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Bruce & Bruce, cap. 3 A/B Testing + Kohavi, Tang & Xu, Trustworthy Online Controlled Experiments (2020). Duración estimada: 90 min.*

Objetivo

Diseñar y analizar un A/B test end-to-end: definir hipótesis y métrica primaria, calcular tamaño de muestra con el poder estadístico deseado, randomizar correctamente, analizar resultados sin peeking y reportar con effect size + IC. Conocer tres herramientas modernas que reducen muestra requerida o eliminan el problema de peeking: CUPED, sequential testing (always-valid p-values) y A/B bayesiano.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Calcular n requerido con statsmodels.stats.power.TTestIndPower o NormalIndPower para una MDE (minimum detectable effect) dada, α y poder.
- Implementar el análisis: t-test (continua) o z-test de proporciones (binaria), con effect size + IC95 %.
- Identificar y evitar 5 errores clásicos: peeking, p-hacking, no estratificación, SRM (sample ratio mismatch), Simpson's paradox.
- Aplicar CUPED para reducir varianza usando una covariable pre-experimento.
- Diseñar un test secuencial con always-valid p-values (Howard et al. 2021) o GST (group sequential testing) que permita parar antes sin inflar α.
- Comparar A/B clásico (frecuentista) con A/B bayesiano (PyMC o bayesab) y entender ventajas (interpretación directa, parar cuando alcance precisión).

Temas

- Hipótesis nula vs alternativa en A/B; métrica primaria, guardrails (no degradar latencia, error rate).
- Poder estadístico: P(rechazar H | H verdadera). Convención: 80 %.
- Sample size: depende de α (0.05), poder (0.8), σ y MDE.
- Aleatorización a nivel correcto (usuario vs sesión vs request).
- Peeking problem: mirar el resultado intermedio e inflar α.
- SRM (Sample Ratio Mismatch): si el ratio observado A/B se aleja del esperado 50/50, hay bug de asignación.
- Simpson's paradox: la tendencia global se invierte al estratificar.
- Complemento moderno: CUPED, sequential testing, A/B bayesiano.

Versión profundizada — 2026

El tema moderno que antes vivía como complemento dentro de esta clase ahora tiene su(s) clase(s) propia(s) con patrón completo, ejercicios y homework:

- Clase 154a — CUPED, sequential testing, always-valid p-values

Definiciones y características

- MDE (Minimum Detectable Effect): el efecto más chico que considerás relevante de detectar. Lo elegís antes del experimento.
- Poder estadístico (1-β): probabilidad de detectar el MDE si es real. Convención: 0.80.
- α: error tipo I. Para A/B testing, 0.05 es estándar; 0.01 para decisiones críticas.
- SRM (Sample Ratio Mismatch): cuando el ratio de asignación observado se desvía significativamente del esperado. Indica bug en la randomización. Test: chi² sobre conteos A vs B.
- Stratificación: balancear covariables (país, plataforma) entre A y B para evitar Simpson's paradox.
- CUPED: reducción de varianza usando covariable pre.
- Always-valid p-value: válido bajo cualquier tiempo de parada; permite peeking sin inflación de α.
- Novelty effect: los usuarios reaccionan al cambio en sí, no al diseño. Confirma con análisis por cohortes/tiempo.

Dataset / recursos

- Simular A/B: rng.binomial(1, 0.10, n) vs rng.binomial(1, 0.12, n) → MDE de 2 pp absoluto.
- Para CUPED: simular X (pre) y Y = 0.5*X + ε + δ·tratamiento.
- Librerías: statsmodels.stats.power, scipy.stats, confseq, pingouin.

Ejercicios

1. Sample size: querés detectar un uplift de tasa de conversión de 10 % → 11 % con poder 0.8 y α=0.05. Usá statsmodels.stats.proportion.samplesize_proportions_2indep_onetail o

power.NormalIndPower().solve_power. ¿Cuánto necesitás por grupo?

2. Análisis clásico: simulá el experimento (n=8 000 por grupo, p_A=0.10, p_B=0.108), aplicá z-test de proporciones, reportá p, IC95 % de la diferencia y poder post-hoc.
3. CUPED: simulá X = rng.normal(50, 10, 2000) y Y = X + ε + 2·tratamiento con ε ~ N(0,5). Calculá n requerido con y sin CUPED para detectar el efecto de 2.
4. Peeking simulado: bajo H verdadera, simulá 1 000 experimentos donde "parás temprano" si p < 0.05 mirando cada 100 obs hasta 5 000. Mostrá cómo el α real se infla a ≈ 0.25.
5. A/B bayesiano: con A=(1000, 80), B=(1000, 100), calculá P(p_B > p_A) con priors Beta(1,1). Interpretá.

Homework verificable

Diseñar y analizar un A/B test simulado:

1. Calcular n por grupo para MDE=1 pp absoluto, baseline 10 %, α=0.05, poder=0.8.
2. Simular el experimento con n calculado y true uplift de 1.2 pp.
3. Reportar: z-test (p, IC), Cohen's h (effect size para proporciones), análisis bayesiano (P(B > A), expected uplift).
4. Repetir simulando peeking cada 1 000 obs sin corrección → mostrar inflación de α.
5. Concluir en 4 líneas: recomendación para producción (frecuentista clásico vs bayesiano vs always-valid).

Criterio de aceptación: n ≈ 14 800 por grupo. El z-test debe rechazar H con p < 0.05, P(B > A) debe ser > 0.95 en bayesiano, y el ejercicio de peeking debe mostrar α real entre 0.20 y 0.30.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Pareo temprano cuando vi p < 0.05	Peeking → α inflado. Fix: pre-registrar fe
El test A/B muestra p < 0.05 pero el negoc	Novelty effect, sesgo de selección, no est
Conteos A vs B son 48/52 con n=10⁶ y "es c	SRM serio. Fix: chi² sobre conteos; si p <
Resultado positivo global, negativo por ca	Simpson's paradox por estratificación disp
Reporto p < 0.05 sin effect size ni IC	Inutilizable para decisión de negocio. Fix

Preguntas frecuentes

¿Cuánto dura un A/B test?

Hasta alcanzar el n calculado y cubrir al menos un ciclo completo del negocio (típicamente 1-2 semanas para capturar weekday/weekend effects, holidays, etc.). Parar antes solo con sequential testing válido.

¿Si mi MDE es muy chico, qué hago?

Necesitás más muestra (1/MDE²). Si no podés conseguirla: (a) usá CUPED para reducir varianza, (b) re-evaluá si ese MDE realmente importa para el negocio (un 0.5 % de uplift puede no compensar el costo de desplegar).

¿A/B bayesiano necesita pre-registro?

Conceptualmente menos: la decisión bayesiana (P(B > A) > threshold) es coherente con cualquier tiempo de parada si el prior es honesto. En práctica industrial igual conviene pre-registrar el threshold para evitar racionalizaciones.

¿Aleatorizo a nivel de usuario o de sesión?

A nivel de unidad de tratamiento: si el cambio afecta al usuario (ej.: redesign de homepage), por usuario. Si es un cambio que el usuario puede experimentar múltiples veces sin "aprenderlo" (ej.: ranking de búsqueda), por sesión o request — pero con cuidado por correlación intra-usuario.

¿Qué hago si n calculado es prohibitivo?

Opciones: (a) aumentar MDE (¿es realista el efecto que esperás?), (b) reducir varianza con CUPED, (c) reducir α a 0.10 si el costo de un falso positivo es bajo, (d) hacer un quasi-experiment con synthetic controls (Clase 157).

Referencias

- Bruce & Bruce, cap. 3 — A/B Testing.
- Kohavi, R., Tang, D. & Xu, Y. (2020), Trustworthy Online Controlled Experiments, Cambridge University Press — la biblia industrial.
- Deng, A. et al. (2013), Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data, WSDM (CUPED original).
- Howard, Ramdas, McAuliffe & Sekhon (2021), Time-Uniform, Nonparametric, Nonasymptotic Confidence Sequences, Annals of Statistics.
- statsmodels.stats.power.
- confseq — always-valid CIs.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 186 — Clase 186 — CUPED, sequential testing, always-valid p-values

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Deng et al. (2013) CUPED + Howard et al. (2021) always-valid + Kohavi et al. (2020). Duración estimada: 85 min.*

Objetivo

Aplicar las 3 técnicas modernas que la industria (Microsoft, Netflix, Booking, Spotify) usa para hacer A/B testing más eficientemente: CUPED (variance reduction con covariable pre-experiment), Sequential Testing (mirar el resultado durante el experimento sin inflar α), y always-valid p-values / confidence sequences (Howard et al. 2021, decisión correcta en cualquier momento).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Implementar CUPED: ajustar Y_cuped = Y - θ·(X - E[X]) con θ = Cov(Y,X)/Var(X).
- Calcular la reducción de varianza esperada: 1 - ρ².
- Configurar Group Sequential Testing con O'Brien-Fleming boundaries.
- Aplicar always-valid CIs con la librería confseq.
- Decidir entre frequentist clásico, GST, y mSPRT según contexto.

Temas

- CUPED math: θ óptimo minimiza varianza.

- Implementación: con Y_pre (mismo usuario en período previo) o X (covariable).
- GST: K looks, boundaries pre-definidas.
- O'Brien-Fleming (gasta poco α al principio) vs Pocock (constante).
- mSPRT: ratio test que provee p-value válido siempre.
- Confidence sequences: IC válido en cualquier t.

Definiciones y características

- CUPED: Controlled-experiment Using Pre-Experiment Data.
- Variance reduction: nuevo Var(Y_cuped) = Var(Y) · (1 - ρ²).
- Sequential testing: tomar decisión antes de N planificado.
- GST: predefine K mira-instantes, pasa α budget según schedule.
- Always-valid p-value: válido bajo cualquier optional stopping rule.
- confseq library: implementación de Howard et al.

Dataset / recursos

- Simulación A/B con numpy.random.default_rng.
- Librerías: numpy, scipy, confseq (pip install confseq).

Ejercicios

1. CUPED implementation: simular X_pre, Y_post = α·X_pre + tratamiento + ε. Calcular θ. Comparar Var(Y) vs Var(Y_cuped).
2. Variance reduction: con ρ=0.7, calcular reducción esperada (= 51 %); verificar con simulación.
3. Peeking inflado: bajo H, simular 1000 experimentos con 5 looks naïve, contar % de rejects. Debería ser ≈ 18 %.
4. O'Brien-Fleming: implementar boundaries con rpy2 + gsDesign (o aproximación). Verificar α controlled.
5. Always-valid CI: confseq.bounds.normal_log_mixture_bound sobre stream simulado. Plotear CI a lo largo del tiempo.

Homework verificable

Comparar 4 enfoques sobre simulación A/B realista:

1. Frequentist clásico (fixed N).
2. CUPED + frequentist.
3. Naïve peeking cada 1k samples (α inflated).
4. Always-valid (confseq).

Reportar para cada uno: tipo I error (bajo H), poder (bajo H), promedio sample size hasta decisión.

Criterio de aceptación: CUPED reduce sample requerido ~30 % con poder igual; naïve peeking infla α a 0.15-0.25; always-valid mantiene α=0.05 con +20 % sample.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
CUPED reduce varianza solo si ρ > 0	Si X no correlaciona, no ayuda. Fix: elegi
Peeking sin corrección	α inflado. Fix: GST o always-valid.
GST con boundaries mal calibradas	Engineers re-implementan mal. Fix: librerí
Always-valid muy conservador con pocos sam	Inherente. Fix: aceptarlo o usar GST.
Reportar GST como "p-value normal"	Distinto significado. Fix: documentar que

Preguntas frecuentes

CUPED siempre vale la pena?

Sí, si tenés X correlated y es free de computar. Microsoft reporta 50 % menos samples en muchos casos.

GST o always-valid?

GST: K looks fijos, simple, comunidad estadística. Always-valid: mirá cuando querás, más conservador. Para casos modernos (peeking continuo), always-valid.

Implementations open source?

- GST: gsDesign (R), pyabtest o reimplementación.
- Always-valid: confseq (Python), cpsequential (R).

Bayesian A/B con stopping?

También válido para optional stopping, pero requiere prior honesto. Decisión: P(B > A | data) > 0.95.

Industria realmente lo usa?

Sí: Microsoft (CUPED), Netflix (sequential), Optimizely (always-valid). Buscar "Trustworthy Online Controlled Experiments" book.

Referencias

- Deng, Xu, Kohavi & Walker (2013), Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data, WSDM.
- Howard, Ramdas, McAuliffe & Sekhon (2021), Time-uniform, nonparametric, nonasymptotic confidence sequences, Annals of Statistics.
- Kohavi, Tang & Xu (2020), Trustworthy Online Controlled Experiments, Cambridge.
- confseq.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 187 — Clase 187 — Diseño experimental

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Montgomery, Design and Analysis of Experiments (8ª ed.) + Kohavi, Tang & Xu (cap. 4-5). Duración estimada: 80 min.*

Objetivo

Pasar del A/B simple a diseños más ricos: bloques aleatorizados, factorial completo / fraccional, diseños cruzados (cross-over), switchback para experimentos con interferencia, y cluster randomization cuando la unidad de análisis no coincide con la unidad de tratamiento. Saber qué problema resuelve cada diseño y leer las consideraciones de SUTVA (Stable Unit Treatment Value Assumption).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Distinguir diseño completamente aleatorizado (CRD), bloques aleatorizados (RBD), factorial y fraccional 2^(k-p).
- Detectar cuándo SUTVA se viola (efectos de red, interferencia entre usuarios, fila/competencia) y aplicar el diseño correcto: cluster randomization, switchback, marketplace experiments.
- Diseñar un factorial 2^2 o 2^3 con pyDOE2 / statsmodels y descomponer efectos principales + interacciones.
- Saber cuándo usar fraccional (2^(k-p)) para reducir corridas y qué se sacrifica (confounding de interacciones de alto orden).
- Aplicar cross-over para experimentos pareados dentro de sujeto, con análisis vía test pareado o modelo mixto.

Temas

- CRD: el A/B clásico. Asume SUTVA (no interferencia entre unidades).
- RBD (bloques): bloquear por variable nuisance (ej.: día de semana, país) para reducir varianza dentro del bloque.
- Factorial 2^k: testear k factores simultáneamente. Captura interacciones; mucho más eficiente que A/B por factor.
- Fraccional 2^(k-p): corridas reducidas. Se confunden ("aliasing") efectos de alto orden con principales.
- Cross-over: cada sujeto recibe ambos tratamientos en períodos distintos. Análisis pareado, controla variabilidad inter-sujeto. Riesgo: carry-over effect.
- Cluster randomization: aleatorizar grupos (clases, ciudades) en lugar de individuos cuando hay contaminación social.
- Switchback: alternar tratamiento global en bloques de tiempo (típico de marketplaces de dos lados — Uber, DoorDash).
- SUTVA: cada unidad solo recibe una versión del tratamiento; los efectos no se propagan entre unidades.

Definiciones y características

- Aleatorización: asignación al azar a tratamiento; protege contra confounders observados y no observados.
- Bloqueo: agrupar unidades similares en bloques homogéneos; aleatorizar dentro del bloque. Reduce varianza si la variable de bloqueo importa.
- Efecto principal: efecto promedio de un factor sobre la respuesta.
- Interacción: efecto que un factor tiene sobre el efecto de otro (no aditividad).
- Confounding (en fraccional): la imposibilidad estructural de distinguir un efecto de otro debido al diseño. Se acepta confundir interacciones triples con efectos principales (asumimos triples ≈ 0).
- SUTVA: no interference + no hidden variations of treatment. Es la asunción callada de todo A/B clásico.
- Carry-over: efecto residual de un tratamiento previo en cross-over. Se mitiga con washout periods y diseños equilibrados (Latin square).
- ICC (Intra-Cluster Correlation): correlación dentro del cluster. Si $\rho > 0$, el n efectivo es menor; corregir con $n_{effective} = n / (1 + (m-1)\cdot\rho)$ donde m es tamaño del cluster.

Dataset / recursos

- Sintéticos para factorial 2^2 (e.g., A=color botón, B=texto botón → CTR).
- Iris / penguins para análisis ANOVA tipo factorial.
- Librerías: pyDOE2 (pip install pyDOE2), statsmodels.formula.api, pingouin.

Ejercicios

1. Factorial 2²: simulá CTR con ctr = 0.10 + 0.02·A + 0.015·B + 0.005·A·B + ε. Hacé el experimento con 1 000 obs por celda. Ajustá ols('ctr ~ A * B', data).fit() y reportá los 4 coeficientes (intercepto, A, B, A:B). Verificá contra el verdadero.
2. Bloqueo: simulá un experimento de uplift en tasa de retención por país con paises = ['AR','BR','MX'] con baselines distintos (p0 {0.5, 0.3, 0.4}). Compará: A/B sin estratificar vs bloqueado por país. Mostrá cómo el SE de δ cae con bloqueo.
3. Fraccional 2^(4-1): usá pyDOE2.fracfact('a b c d') y discutí qué interacciones quedan aliased. ¿Cuántas corridas vs full factorial?
4. Cluster randomization: simulá 50 escuelas con 30 alumnos c/u, ICC=0.10, efecto verdadero 0.3. Compará t-test ingenuo (n=1500) vs análisis correcto a nivel cluster (n=50). El primero infla α; el segundo es correcto.
5. Switchback: simulá precio dinámico en una ciudad con bloques de 1 h alternando A y B durante 7 días. Análisis: comparar bloques A vs B con tests pareados por hora-del-día.

Homework verificable

Sobre un dataset simulado de factorial 2³ (3 factores binarios, ej.: color × texto × posición sobre conversion):

1. Generar datos con efectos principales no nulos y una interacción A:B.
2. Ajustar OLS con todos los términos hasta triple.
3. Reportar la tabla ANOVA y identificar los términos significativos.
4. Verificar gráficamente la interacción A:B con sns.pointplot(x='A', y='conv', hue='B').
5. Discutir en 3 líneas qué pasaría si hubieras hecho 3 A/B tests separados en vez del factorial.

Criterio de aceptación: el OLS recupera los efectos verdaderos; la ANOVA marca A, B y A:B como significativos; el análisis muestra que el factorial requiere menos corridas totales que 3 A/B independientes (típicamente la mitad) y además detecta la interacción.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Hago A/B en una red social y "el grupo con	SUTVA violada por contaminación (el contro
Análisis a nivel individual de experimento	α inflado por correlación intra-cluster. F
Asumo interacción nula y la interpretación	Si A solo funciona cuando B está activo, l
Cross-over sin washout y carry-over inflad	El efecto del tratamiento 1 contamina la m
Hago RBD pero no bloqueo lo que importa	Si bloqueás por país pero el efecto import

Preguntas frecuentes

¿Cuándo factorial vs A/B múltiple?

Casi siempre factorial. A/B múltiple (1 factor a la vez) requiere más muestra y no detecta interacciones. Factorial 2² con n por celda tiene la misma precisión que dos A/B independientes con ~n total.

¿Cuántos factores antes de necesitar fraccional?

Con 2^k = 16 (k=4) ya tenés 16 celdas → si querés ~1 000 por celda son 16 000 obs. Manejable. A partir de k=5 (32 celdas) considerar fraccional, sobre todo si esperás interacciones de alto orden insignificantes.

¿Cuándo cluster randomization?

Cuando hay contaminación natural: educación (alumnos en la misma aula), salud pública (familias), marketplaces (oferta + demanda compartidas). Costo: n efectivo cae con ICC; necesitás muchos clusters.

¿Switchback es válido si hay tendencia temporal?

Sí pero requiere modelar la tendencia (ej.: incluir hora del día como covariable). Si tu métrica tiene fuerte estacionalidad y bloques son largos, mejor diseño Latin square en el tiempo.

¿Diseño antes o análisis primero?

Diseño primero, siempre. El análisis sin diseño correcto produce conclusiones sospechosas (Simpson, confounders, peeking). "You can't analyze your way out of a bad design" (Tukey).

Referencias

- Montgomery, D.C. (2017), Design and Analysis of Experiments (8ª ed.) — referencia clásica completa.
- Kohavi, R., Tang, D. & Xu, Y. (2020), Trustworthy Online Controlled Experiments, caps. 4-5 (advanced designs, interference).
- Imbens & Rubin (2015), Causal Inference for Statistics, Social, and Biomedical Sciences.
- pyDOE2 — generación de matrices de diseño.
- Athey, Eckles & Imbens (2018), Exact p-Values for Network Interference, JASA.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 188 — Clase 188 — Inferencia causal: DAGs, confounders, instrumentos

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Pearl, The Book of Why + Hernán & Robins, Causal Inference: What If (libro gratuito, 2024) + Imbens & Rubin. Duración estimada: 95 min.*

Objetivo

Distinguir correlación de causalidad con rigor: dibujar DAGs (Directed Acyclic Graphs), identificar confounders, colliders y mediators, aplicar el backdoor criterion para decidir qué variables controlar, y usar variables instrumentales (IV) cuando la randomización no es posible. Conocer la herramienta moderna Double Machine Learning (DoubleML / EconML) para estimar ATE/CATE con ML como nuisance estimator.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Dibujar un DAG para un problema de negocio e identificar los tres tipos de estructura: chain (X → M → Y), fork (X ← Z → Y, confounder), collider (X → C ← Y).
- Aplicar el backdoor criterion de Pearl: encontrar el conjunto mínimo de variables a controlar para identificar el efecto causal.
- Reconocer que controlar por un collider o un mediator introduce sesgo, NO lo elimina.
- Estimar ATE (Average Treatment Effect) con regresión + controles, IPW (Inverse Probability Weighting) y matching.
- Usar 2SLS (Two-Stage Least Squares) con linearmodels.iv cuando hay un instrumento válido.
- Aplicar Double Machine Learning con doubleml / econml para estimar ATE/CATE con ML como nuisance (sin asumir linealidad).

Temas

- Correlación ≠ causalidad: el clásico ejemplo "helado y ahogamientos" — confounder: temperatura.
- DAGs: nodos = variables, flechas = relación causal.
- 3 estructuras canónicas: chain, fork, collider.
- Backdoor criterion: bloquear todos los caminos no causales de X a Y; no abrir colliders.
- ATE = E[Y | do(T=1)] - E[Y | do(T=0)]. El "do" indica intervención, no observación.
- Identificación: ¿se puede expresar E[Y | do(T)] con datos observacionales? Si sí → estimar.
- IV: variable Z que afecta T pero NO a Y excepto vía T. Permite identificar el efecto cuando hay confounders no observados.
- Complemento moderno: Double Machine Learning (Chernozhukov et al. 2018) — usa ML para estimar las "nuisance functions" y separa la inferencia causal de la complejidad del fit.

Versión profundizada — 2026

El tema moderno que antes vivía como complemento dentro de esta clase ahora tiene su(s) clase(s) propia(s) con patrón completo, ejercicios y homework:

- Clase 156a — DoubleML / EconML: Machine Learning para causalidad

Definiciones y características

- DAG: grafo dirigido acíclico que representa relaciones causales. Cada flecha es una hipótesis causal.
- Confounder (fork): T ← Z → Y. Z causa tanto T como Y; controlando Z se identifica el efecto.
- Collider: T → C ← Y. Controlar C introduce asociación espuria (sesgo de selección). Ejemplo clásico: Berkson's bias.
- Mediator (chain): T → M → Y. Controlar M bloquea el efecto causal indirecto y solo deja el directo (a veces útil, a veces no).
- Backdoor criterion (Pearl): un set Z de variables identifica el efecto causal de T sobre Y si: (a) bloquea todos los caminos backdoor (que empiezan con flecha entrando a T), y (b) no contiene descendientes de T.
- ATE (Average Treatment Effect): E[Y(1) - Y(0)] — efecto promedio si todos vs nadie recibiera tratamiento.
- CATE (Conditional ATE): E[Y(1) - Y(0) | X=x] — efecto para individuos con características X=x.
- Instrumento (IV): variable Z que satisface (a) relevancia (afecta a T), (b) exclusión (no afecta Y excepto vía T), (c) independencia (no comparte confounders con Y).
- 2SLS: estima IV ajustando T = αZ + ε (1ª etapa), reemplazando T̂ en Y = θT̂ + ε (2ª etapa).
- DML: ML para nuisance + Neyman-orthogonal score + cross-fitting → inferencia causal robusta.

Dataset / recursos

- Ejemplos simulados de Pearl: smoking-cancer con tar como mediator.
- econml.tests.dgps para datos sintéticos con efectos heterogéneos conocidos.
- Lalonde 1986 (NSW job training program) — clásico de causal inference.
- Librerías: doubleml, econml, linearmodels, pgmpy (DAG inference), dowhy (Microsoft, framework completo).

Ejercicios

1. DAG en código: con pgmpy (o networkx), definí un DAG con T, Y, Z (confounder), C (collider). Identificá visualmente paths y aplicá dowhy para encontrar el adjustment set.
2. Sesgo del collider: simulá T ~ N(0,1), Y ~ N(T, 1), C = T + Y + ε. Estimá Y ~ T sin controlar C y controlando C. Mostrá que controlar el collider destruye la relación causal.
3. Backdoor ajustando confounder: simulá Z, T = f(Z) + ε, Y = 2T + 3Z + δ. OLS Y ~ T sesgado. OLS Y ~ T + Z recupera el 2.

4. 2SLS: simular un IV Z → T → Y con confounder no observado entre T y Y. Aplicar linearmodels.iv.IV2SLS.from_formula('Y ~ 1 + [T ~ Z]', data).fit(). Recuperar el efecto verdadero.
5. DML con random forest: dataset sintético con confounders no lineales. Comparar OLS ingenuo vs OLS con polinomios vs DoubleMLPLR(ml_g=RF, ml_m=RF). Verificar que DML es el menos sesgado.

Homework verificable

Sobre un dataset simulado de "efecto de un programa de capacitación sobre salario":

1. Generar X (edad, educación, experiencia) como confounders. Generar T (participa) con P(T|X) no trivial. Generar Y con efecto causal θ_true = 2_000.
2. Estimar el efecto con: (a) diferencia ingenua de medias, (b) OLS con controles lineales, (c) DoubleML con RF.
3. Comparar contra θ_true. Reportar bias y IC95 %.
4. Dibujar el DAG (puede ser un comentario con notación o un grafo simple).
5. Discutir en 3 líneas qué pasaría si hubieras controlado por un mediator (ej.: "horas trabajadas").

Criterio de aceptación: DML debe recuperar θ_true ± 200. OLS lineal puede estar sesgado si la relación X→Y no es lineal. La diferencia ingenua debe estar fuertemente sesgada. La discusión debe mencionar que controlar mediators sesga hacia 0.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Controlo "todo lo que tengo" en la regresi	Si entre esos hay colliders o mediators, i
Asumo que el coeficiente OLS es causal cua	No lo es. Fix: IV (si tenés instrumento),
Uso un instrumento débil (correlación con	2SLS con IV débil tiene sesgo y SE inflado
doubleml con n_folds=2 y muestra chica	Cross-fitting con pocos folds no estabiliz
Interpreto un coeficiente OLS como ATE sin	OLS = ATE solo bajo unconfoundedness. Fix:

Preguntas frecuentes

¿Cómo sé si dibujé bien el DAG?

No hay método estadístico para validarlo completamente — el DAG codifica supuestos sustantivos (de dominio). Lo que sí podés hacer: falsificación condicional — el DAG implica ciertas independencias condicionales; testealas con los datos y si fallan, el DAG está mal. dowhy.refute_estimate automatiza muchos refutation tests.

¿IV o DML cuando tengo ambos?

Si tenés un IV válido y confiable, IV es identificación más fuerte (resiste confounders no observados). DML solo aguanta confounders observados. Lo ideal: triangular con ambos.

¿Causal forest vs random forest clásico?

Causal forest no minimiza error de predicción; minimiza heterogeneidad del efecto causal entre hojas. Cada hoja contiene unidades con efecto causal similar.

¿Qué tan robusto es DML a especificación errónea?

DML es doubly robust: si o el modelo de g o el de m está bien especificado, el estimador del efecto es consistente. Cero modelos correctos → sesgo. Es la mejor garantía actual sin randomización.

¿Inferencia causal con datos observacionales puede reemplazar un RCT?

No completamente. RCT randomiza el tratamiento → corta todas las flechas backdoor por diseño. Observacional siempre depende de supuestos no testables (unconfoundedness, IV exclusion). El estándar es: RCT cuando se puede; cuasi-experimental (DiD, IV, RDD) cuando no; y reportar sensitivity analyses.

Referencias

- Pearl, J. (2018), The Book of Why — intuición sin matemática pesada.
- Pearl, J., Glymour, M. & Jewell, N. (2016), Causal Inference in Statistics: A Primer — el libro técnico corto.
- Hernán, M. & Robins, J. (2024), Causal Inference: What If. Libro gratuito.
- Chernozhukov et al. (2018), Double/Debiased Machine Learning for Treatment and Structural Parameters, Econometrics Journal.
- doubleml docs (Python + R).
- EconML docs — Microsoft Research.
- DoWhy — framework de identificación + estimación + refutación.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 189 — Clase 189 — DoubleML / EconML: Machine Learning para causalidad

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Chernozhukov et al. (2018) DML + docs DoubleML + EconML. Duración estimada: 95 min.*

Objetivo

Aplicar Double Machine Learning (Chernozhukov 2018) y EconML (Microsoft Research) para estimar ATE y CATE (Conditional ATE — heterogeneidad del efecto) usando cualquier ML como nuisance estimator (Random Forest, XGBoost, neural net). Inference válida (CI, p-value) sin asumir linearidad de los confounders.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Explicar Neyman-orthogonal score y por qué DML es doubly robust.
- Aplicar DoubleMLPLR para ATE con ML nuisance.
- Usar CausalForestDML de EconML para CATE personalizado.
- Cross-fitting: K-fold para evitar overfitting del nuisance.
- Inspeccionar policy óptimo: policy_tree para árbol de decisión de tratamiento.

Temas

- Marco PLR (Partially Linear Regression): $Y = \theta T + g(X) + \varepsilon$, $T = m(X) + v$.
- Score orthogonal: derivada respecto a nuisances = 0 en expectation.
- Cross-fitting: estimar nuisances en fold A, evaluar en B.
- CATE: efecto por subgroup.
- Heterogeneity tests.
- Policy learning: decidir a quién tratar.

Definiciones y características

- ATE (Average Treatment Effect): E[Y(1) - Y(0)].
- CATE: E[Y(1) - Y(0) | X=x].
- Nuisance functions: g(X) = E[Y|X], m(X) = E[T|X].
- Doubly robust: consistente si o g o m es correctamente especificado.
- Cross-fitting: K folds, estimar nuisance en train fold, evaluar score en test fold.
- CausalForestDML: random forest entrenado para minimizar heterogeneidad del effect (no error de predicción).

Dataset / recursos

- Sintético con efectos conocidos (de econml.tests.dgps).
- Lalonde 1986 (NSW training program) — clásico.
- IHDP (Infant Health and Development).
- Librerías: doubleml, econml, scikit-learn, xgboost.

Ejercicios

1. Sintético: simular $Y = 2 \cdot T + 3 \cdot X1 + X2^2 + \varepsilon$, T = P(...|X). ATE verdadero = 2.
2. DML básico: DoubleMLPLR(data, ml_g=RF, ml_m=RF, n_folds=5). Reportar $\hat{\theta}$.
3. Comparar OLS vs DML: OLS lineal con Y ~ T + X1 + X2 sesgado por X2 no lineal. DML lo recupera.
4. CausalForestDML: con dataset heterogéneo, estimar CATE. Mapa por X1.
5. Policy tree: econml.policy.PolicyTree para decidir a quién tratar.

Homework verificable

Estimar efecto causal sobre dataset con confounders no lineales:

1. Generar dataset sintético (econml.dgps.ihdp_surface_B o custom).
2. Estimar ATE con: (a) diferencia ingenua, (b) OLS lineal, (c) DML RF, (d) DML XGBoost.
3. Reportar bias contra ground truth.
4. CATE con CausalForestDML; visualizar heterogeneidad.

Criterio de aceptación: DML recupera ATE con bias < 5 %; OLS lineal sesgado; CATE muestra variación por subgrupo.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
DML con n_folds=2 inestable	Fix: 5-10 folds.
Confounders no observados → DML sesgado	DML asume unconfoundedness. Fix: si hay IV
ml_g muy expresivo overfittea	Cross-fitting ayuda pero no inmune. Fix: r
Reportar CATE sin IC	Necesario para inference. Fix: est.effect_
Asumir SUTVA cuando hay interferencia	Fix: ajustar diseño experimental.

Preguntas frecuentes

DML o regresión clásica?

Si confounders few + relación cerca-lineal: regresión OK. Si muchos / no lineales / interacciones: DML.

DoubleML vs EconML?

DoubleML: más académico, claro framework. EconML: más features (instrumental variables, dynamic

treatment, policy). En la práctica complementarios.

Random Forest mejor que XGBoost como nuisance?

Suele dar similar. Probar ambos y elegir el mejor en out-of-fold prediction.

Causal Forest = Random Forest?

NO. Causal Forest divide para minimizar heterogeneidad del efecto causal, no error de Y.

Cuántos samples?

Mínimo 1000-5000 para CATE confiable. Para ATE, menos.

Referencias

- Chernozhukov, Chetverikov, Demirer et al. (2018), Double/Debiased Machine Learning, Econometrics Journal.
- Athey & Wager (2019), Estimating Treatment Effects with Causal Forests.
- Imbens (2020), Potential Outcome and Directed Acyclic Graph Approaches to Causality.
- DoubleML docs, EconML docs.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 190 — Clase 190 — Uplift modeling, DiD (difference-in-differences)

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Gutierrez & Gerardy (2017), Causal Inference and Uplift Modeling + Abadie, Diamond & Hainmueller (2010), Synthetic Control Methods. Duración estimada: 90 min.*

Objetivo

Dominar las dos técnicas causales más usadas en industria cuando hay datos panel/observacionales: DiD (Difference-in-Differences) —comparar la evolución antes/después en grupo tratado vs control— y uplift modeling —predecir a quién conviene tratar (heterogeneidad del efecto causal a nivel individuo). Conocer el complemento moderno Synthetic Control Method para cuando no hay grupo de control natural y solo se trata una unidad (una ciudad, un país).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Aplicar DiD clásico con OLS: $Y = \beta + \beta \cdot \text{tratado} + \beta \cdot \text{post} + \beta \cdot (\text{tratado} \times \text{post}) + \varepsilon$. El coeficiente β es el efecto causal bajo parallel trends.
- Diagnosticar la asunción de parallel trends con un gráfico antes/después y un placebo test.
- Construir modelos de uplift: T-learner, S-learner, X-learner (Künzel et al. 2019), Causal Forest (econml).
- Evaluar uplift con Qini curve y uplift@k (no con AUC clásico — uplift es individual, no global).
- Aplicar Synthetic Control con pysyncon o SparseSC cuando una sola unidad recibe tratamiento (estudio de caso).

Temas

- DiD: comparación dos-por-dos en panel (2 grupos × 2 tiempos).
- Asunción crítica: parallel trends — sin tratamiento, ambos grupos hubieran evolucionado paralelos.
- Generalización: DiD con muchos tiempos, two-way fixed effects (TWFE), event study designs.
- Uplift = CATE individual = E[Y(1) - Y(0) | X=x].
- 4 cuadrantes de uplift: persuadables, sure things, lost causes, do-not-disturb (no tocarlos).
- Métricas: Qini, uplift@k, AUUC (area under uplift curve).
- Complemento moderno: Synthetic Control Method (Abadie et al.) — construye un "país sintético" como combinación convexa de unidades no tratadas que replica la trayectoria pre-tratamiento.

Versión profundizada — 2026

El tema moderno que antes vivía como complemento dentro de esta clase ahora tiene su(s) clase(s) propia(s) con patrón completo, ejercicios y homework:

- Clase 157a — Synthetic Control Method dedicado (pysyncon, SparseSC)

Definiciones y características

- DiD: estima (Y_tratado,post - Y_tratado,pre) - (Y_control,post - Y_control,pre). Equivale al coeficiente de la interacción tratado × post en OLS.
- Parallel trends: sin el tratamiento, ambos grupos hubieran tenido la misma trayectoria. NO testeable directamente; sí en el pre.
- TWFE (Two-Way Fixed Effects): regresión con dummies de unidad + dummies de tiempo + tratamiento. Generaliza DiD a panel.
- Event study: grafica los coeficientes de dummy × (t - t_tratamiento) para cada t. Permite ver dinámica del efecto y testear pre-tendencias.
- Uplift / CATE: τ(x) = E[Y(1) - Y(0) | X=x]. Diferencia entre lo que pasa con tratamiento vs sin.
- T-learner: ajustar dos modelos separados, uno por grupo (μ(x) = E[Y|X, T=1], μ(x) = E[Y|X, T=0]); restar.
- S-learner: un solo modelo con T como feature; predecir con T=1 y T=0 y restar.
- X-learner (Künzel et al. 2019): combina T y S, usando propensity para ponderar; mejor con grupos desbalanceados.
- Causal Forest (econml.CausalForestDML): random forest entrenado para minimizar heterogeneidad del efecto, con IC válidos.
- Qini curve: ranking individuos por uplift predicho; eje x = % tratado; eje y = ganancia incremental acumulada vs random.

Dataset / recursos

- DiD clásico: Card & Krueger 1994 (mínimum wage en NJ vs PA).
- Uplift: Hillstrom email dataset (criteo), Lenta uplift dataset.
- Synthetic Control: California Prop 99 (smoking) — clásico de Abadie.
- Librerías: linearmodels (PanelOLS), econml, causalml (Uber), pysyncon, SparseSC.

Ejercicios

1. DiD ingenuo: simulá panel con 2 grupos y 2 períodos. Aplicá DiD con OLS y verificá que β recupera el efecto verdadero. Probá violar parallel trends y ver el sesgo.
2. Event study: con panel 10 períodos (5 pre, 5 post), graficá coeficientes por período. Si los pre son ≈ 0 → parallel trends plausible.
3. T-learner: en Hillstrom (binario tratamiento email), entrená dos RandomForestClassifier y predecí

uplift = p(x) - p(x).

4. Qini curve: con las predicciones del ej. 3, calculá Qini con sklift.metrics.qini_score o a mano. Compará contra "tratar al azar".
5. Synthetic Control: con un dataset panel simulado (10 estados × 20 años, tratamiento en California año 11), ajustá pesos con pysyncon y graficá path_plot + gaps_plot. Aplicá placebo_test.

Homework verificable

Sobre el dataset Card & Krueger (mínimum wage NJ vs PA, 1992):

1. Cargar datos, calcular promedio de empleo por estado en pre (Feb 1992) y post (Nov 1992).
2. Aplicar DiD con OLS y reportar β con IC95 %.
3. Hacer un event study si hay más de 2 períodos disponibles, o el gráfico 2×2 si no.
4. Discutir en 4 líneas: ¿se cumple parallel trends? ¿Cómo se relaciona el β con el debate clásico (Card vs Neumark)?

Criterio de aceptación: el coeficiente DiD debe ser positivo y ≈ +2.7 empleados (consistente con el paper original); la discusión menciona parallel trends y un riesgo concreto (sesgo de selección de stores, attrition).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar	
Aplico DiD sin verificar parallel trends	Si los grupos tenían trayectorias divergen	
Uplift evaluado con AUC	AUC mide clasificación, no uplift. Un mode	X)` puede ser pésimo para uplift. Fix: Qin
T-learner con un grupo mucho más chico que	El modelo del grupo chico overfittea. Fix:	
Synthetic control con pre-período de 2 año	Pesos no convergen a algo confiable. Fix:	
Interpreto el sintético como "lo que hubie	Sin placebo, no podés saber si tu "efecto"	

Preguntas frecuentes

¿DiD funciona con un solo período pre y un solo post?

Sí (es el "2×2 DiD"), pero no podés testear parallel trends — solo asumirlo. Con más períodos, podés hacer event study y verificarlo.

¿Uplift modeling vs causal forest?

Causal forest es uplift modeling — produce CATE por instancia con IC. Las otras (T/S/X-learner) son métodos clásicos. En la práctica, X-learner y causal forest tienen el mejor performance en benchmarks.

¿Cuántas unidades de control mínimas para synthetic control?

10-20 idealmente; con menos los pesos saturan en 1 o 2 unidades y pierde robustez. Si pocas unidades pero muchos períodos: synthetic DiD o factor models.

¿Negative weights en synthetic control?

El método clásico fuerza w ≥ 0 (combinación convexa). Variantes modernas (Doudchenko & Imbens 2016, SparseSC) relajan esto y permiten negativos con regularización — más flexible pero menos interpretable.

¿DiD con TWFE es siempre correcto?

No con tratamientos escalonados (unidades tratadas en momentos distintos). TWFE puede pesar con signo negativo períodos donde unidades "ya tratadas" sirven de control de "recién tratadas". Fix: Callaway & Sant'Anna 2021, did package en R o differences en Python.

Referencias

- Angrist & Pischke (2009), Mostly Harmless Econometrics, cap. 5 (DiD).
- Künzel, Sekhon, Bickel & Yu (2019), Metalearners for estimating heterogeneous treatment effects using ML, PNAS.
- Abadie, Diamond & Hainmueller (2010), Synthetic Control Methods, JASA.
- Callaway & Sant'Anna (2021), Difference-in-Differences with Multiple Time Periods, J. of Econometrics.
- Gutierrez & Gerardy (2017), Causal Inference and Uplift Modeling, JMLR Workshop.
- econml, causalml (Uber), pysyncon, SparseSC.

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 191 — Clase 191 — Synthetic Control Method dedicado (pysyncon, SparseSC)

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: Abadie, Diamond & Hainmueller (2010) + Doudchenko & Imbens (2016) + Arkhangelsky et al. (2021) Synthetic DiD. Duración estimada: 80 min.*

Objetivo

Aplicar Synthetic Control Method (Abadie et al. 2010) — el estándar para evaluar políticas o intervenciones aplicadas a una única unidad (un país, un estado, una ciudad) sin grupo control natural. Construir un "control sintético" como combinación ponderada de donors. Conocer variantes modernas: Synthetic DiD (Arkhangelsky 2021), Generalized SC (Xu 2017), SparseSC (Microsoft Research).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Construir un Synthetic Control con pysyncon: tratado, donors, predictors, períodos pre/post.
- Interpretar pesos W (combinación convexa) y path plot.
- Aplicar placebo test (in-time y in-space) como inference informal.
- Conocer Synthetic DiD que combina lo mejor de DiD y SCM.
- Reconocer cuándo SCM no aplica (pocos donors, fit pre malo).

Temas

- Setup: 1 tratado + N donors + features predictoras + período pre/post.
- Optimización: pesos minimizan $||Y\_treat\_pre - W \cdot Y\_donors\_pre||^2$.
- Constraint: $w\_i \geq 0$, $\Sigma\, w\_i = 1$ (combinación convexa) — clásico.
- Placebo test in-space: aplicar SCM a cada donor; comparar effect real vs distribución de placebos.
- Placebo in-time: aplicar antes del tratamiento real → debería dar 0.
- Synthetic DiD: relax constraints + agregar pesos temporales.

Definiciones y características

- Tratado: la única unidad que recibió intervención.
- Donors: pool de unidades no tratadas, idealmente similares pre.
- Predictores: covariables usadas para hacer matching pre.
- Dataprep: clase pysyncon que estructura input.

- Synth: optimizador que encuentra pesos.
- Pre-RMSPE: error de matching pre-período. Si alto, malo.
- Post-RMSPE / Pre-RMSPE ratio: indicador informal de magnitud del efecto.

Dataset / recursos

- California Prop 99 smoking (Abadie's dataset clásico).
- Cualquier panel de países × años con una intervención.
- Librerías: pysyncon, SparseSC (Microsoft), numpy, pandas.

Ejercicios

1. California Prop 99: cargar dataset, definir tratado California, donors otros estados, pre 1970-1988, post 1989-2000.
2. Path plot: synth.path_plot() — California real vs sintética. Visualizar gap post-1989.
3. Pesos: imprimir synth.weights. Verificar que solo few estados tienen peso > 0.
4. Placebo in-space: aplicar a cada otro estado; plot de gaps. California debe destacar.
5. Placebo in-time: tratamiento artificial en 1980 (5 años antes del real). Gap debería ser ≈ 0.

Homework verificable

Estudio de caso: efecto de una política aplicada a 1 unidad (ej.: Brexit en UK, COVID lockdowns en una ciudad):

1. Dataset panel, 10+ donors, ≥ 5 años pre.
2. SCM con pysyncon.
3. Path + gap plots.
4. Placebo in-space.
5. Interpretación: ¿el efecto observado está en el extremo de la distribución placebo?

Criterio de aceptación: pre-RMSPE bajo (good fit); diagnóstico de placebo informativo; conclusión justificada.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Pre-RMSPE alto	Donors no comparables. Fix: filtrar donors
Pocos donors (<10)	Pesos no robustos. Fix: ampliar donor pool
Predictores correlated → pesos inestables	Fix: SparseSC con regularización.
Reportar p-value clásico	No existe en SCM. Fix: placebo test ranks
Aplicar SCM a tratamiento gradual	SCM asume tratamiento discreto. Fix: Synth

Preguntas frecuentes

SCM vs DiD?

DiD necesita "parallel trends"; SCM construye control sintético. SCM mejor cuando 1 tratado; DiD mejor con many.

Synthetic DiD cuándo?

Combina virtudes: usable con many tratados + permite no-parallel trends. Default 2024+ en muchos casos.

Cuántos períodos pre necesarios?

5-10 mínimo. Más es mejor para fit confiable.

SparseSC?

Microsoft Research lib que extiende SCM a paneles grandes con regularización L2.

Bayesian SCM?

Existe (Bayesian Structural Time Series — CausalImpact de Google). Otra alternativa.

Referencias

- Abadie, Diamond & Hainmueller (2010), Synthetic Control Methods for Comparative Case Studies, JASA.
- Doudchenko & Imbens (2016), Balancing, Regression, Difference-in-Differences and Synthetic Control Methods.
- Arkhangelsky et al. (2021), Synthetic Difference-in-Differences, AER.
- pysyncon.
- SparseSC (Microsoft).
- Google CausalImpact (R / Python ports).

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Clase 192 — 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(ỹ | D) = ∫ P(ỹ|θ) · P(θ|D) dθ. 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 $\hat{\beta}$ 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

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

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

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: PyMC v5 docs + NumPyro docs + ArviZ docs. Duración estimada: 90 min.*

Objetivo

Aprender el stack bayesiano moderno post-Theano —PyMC v5 (PyTensor), NumPyro (JAX), ArviZ (visualización backend-agnóstica)— a nivel de poder construir modelos jerárquicos serios, diagnosticar convergencia (r_hat, ess_bulk, divergences), comparar modelos con LOO-CV y WAIC, y elegir backend según escala.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Construir modelo jerárquico no-centered en PyMC v5.
- Diagnosticar: r_hat ≤ 1.01, ess_bulk ≥ 400, divergences = 0.
- Aplicar non-centered parameterization para evitar funnel posteriors.
- Comparar modelos con az.compare([m1, m2], ic='loo').
- Migrar modelo de PyMC a NumPyro para 10-50× speedup en CPU.
- Aplicar SVI (Stochastic Variational Inference) en NumPyro como alternativa rápida a MCMC.

Temas

- PyMC v5: PyTensor backend, sintaxis estable.
- NumPyro: JAX backend, JIT + autograd + GPU/TPU.
- Centered vs non-centered parametrization.
- Posterior predictive check con ArviZ.
- LOO-CV y WAIC para comparación.
- SVI con AutoNormal / AutoMultivariateNormal.

Definiciones y características

- PyTensor: sucesor de Theano (archivado 2017). Backend nativo de PyMC.
- NumPyro: probabilistic programming en JAX. 5-50× más rápido en CPU, GPU/TPU nativo.
- r_hat: Gelman-Rubin convergence. ≤ 1.01 → OK.
- ess_bulk / ess_tail: effective sample size. ≥ 400 bulk para inferencia.
- Non-centered: x = μ + σ·z, z ~ N(0,1) en vez de x ~ N(μ, σ). Evita funnel.
- LOO-CV (Vehtari): leave-one-out con Pareto smoothed importance sampling.
- SVI: aproximación variational; cierra el gap MCMC con velocidad.

Dataset / recursos

- tips (regresión jerárquica por day).
- McElreath's Howell1 o chimpanzees (modelos clásicos).
- Librerías: pymc (≥ 5), numpyro, arviz, jax.

Ejercicios

1. PyMC v5 hierarchical: tip ~ Normal(α_day + β·bill, σ); α_day ~ Normal(μα, σα).
2. Non-centered: re-parametrizar el modelo con α_day = μα + σα · z_day, z_day ~ N(0,1). Comparar divergences.
3. PPC: pm.sample_posterior_predictive + az.plot_ppc. Decidir si modelo razonable.
4. NumPyro version: traducir, comparar tiempo.
5. LOO compare: 3 modelos (intercepto solo, + slope, + jerárquico). az.compare.

Homework verificable

Modelo bayesiano completo sobre tips:

1. PyMC v5 jerárquico con day como nivel.
2. NumPyro mismo modelo.
3. SVI en NumPyro como alternativa rápida.
4. Comparar 3 enfoques: MCMC PyMC, MCMC NumPyro, SVI NumPyro.
5. az.plot_trace, az.summary, az.compare para 2 variantes del modelo.

Criterio de aceptación: convergencia (r_hat ≤ 1.01); SVI cierra gap con MCMC en < 30 s; ArviZ plots producidos.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Divergences > 100	Funnel posterior. Fix: non-centered parame
r_hat = 1.3	No convergió. Fix: más tune, target_accept
Prior Uniform(-1e6, 1e6)	Plano no es "no informativo". Fix: weakly
SVI con resultados raros	Posterior multimodal o AutoNormal no aplic
Comparar modelos con DIC	Deprecated. Fix: LOO o WAIC.

Preguntas frecuentes

PyMC v5 o NumPyro?

PyMC v5 si tu modelo ya está en PyMC3/v4 (migración fácil). NumPyro si necesitás velocidad o GPU/TPU.

SVI o MCMC?

MCMC para inferencia rigurosa. SVI para producción donde latencia importa.

Stan?

Sí, alternativa madura. cmdstanpy. Sintaxis Stan propia. ArviZ integra.

Edward2 / TFP?

TensorFlow Probability. Menos comunidad que PyMC/NumPyro. Bueno si ya en TF.

Modelos jerárquicos cuándo non-centered?

Casi siempre. Centered solo si hay mucho data por nivel.

Referencias

- Salvatier, Wiecki & Fonnesbeck (2016), PyMC3 — original paper.
- Phan, Pradhan & Jankowiak (2019), NumPyro.
- Vehtari et al. (2017), Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC.
- McElreath (2020), Statistical Rethinking — best book intro bayesiano.

Siguiente parte

Clase 194 — Versionado de datos con DVC

Material descargable

- Guía explicativa (PDF) — versión imprimible con todo el contenido de la clase.
- Presentación (PPTX) — deck PowerPoint listo para proyectar en clase.
- Notebook ejecutable (.ipynb) — abrilo desde el laboratorio del programa o desde Jupyter.

Cierre de la parte

Fin del bundle consolidado de Parte 3 — Estadística Inferencial y Causal · 19 clases.