Kapitel 2: Monte Carlo Schätzung

Hinweis: Berichten Sie stets: Schätzer $\hat{\mu}$, geschätzten Standardfehler (SE), 95%-CI, verwendetes N, RNG/Seed und ggf. Laufzeit. Visualisieren Sie, wo sinnvoll (Histogramme/Fehlerbalken). Notieren Sie Annahmen und Limitierungen.

Aufgabe 5 *MC-Integration & Vergleich zur Quadratur*

a) Schätzen Sie mittels Monte Carlo

$$I_1 = \int_0^1 e^{-x^2}\,dx.$$

Führen Sie Replikationen für $N \in \{10^3, 10^4, 10^5\}$ durch. Berichten Sie $\hat{\mu}$, SE, 95%-CI und vergleichen Sie Ihre Schätzung mit einer numerischen Quadratur (siehe Snippet unten).

b) Wiederholen Sie die Aufgabe für

$$I_2 = \int_0^1 \sqrt{x}\,dx.$$

Vergleichen Sie MC und Quadratur; diskutieren Sie die SE-Skalierung $\propto 1/\sqrt{N}$.

Hinweis: Sie können folgenden Code verwenden, um den Wert des gesuchten Integrals numerisch zu berechnen:

```python
import numpy as np
from scipy.integrate import quad

# Integranden
f1 = lambda x: np.exp(-(x**2))    # a)
f2 = lambda x: np.sqrt(x)         # b)

# Referenz via Quadratur:
val1, err1 = quad(f1, 0.0, 1.0)
val2, err2 = quad(f2, 0.0, 1.0)
print("I1 (quad) =", val1, "+/-", err1)
print("I2 (quad) =", val2, "+/-", err2)
```

Aufgabe 6 *Importance Sampling (Unterer Tail der Normalverteilung)*

Ziel ist es die (geringe) Wahrscheinlichkeit für eine Beobachtung im untere Tail zu berechnen:

$$p \ = \ \mathbb{P}(Z < -3), \quad Z \sim \mathcal{N}(0,1).$$

a) Implementieren Sie den *naiven* MC-Schätzer

$$\hat{p}_{\text{naiv}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{Z_i < -3\}$$

.

b) Verwenden Sie *Importance Sampling* mit verschobener Dichte $q = \mathcal{N}(\mu_q, 1)$ und *negativem* Shift

$$\mu_q \in \{-3.0,\ -3.5,\ -4.0\}.$$

Ziehen Sie Stichproben aus $X_i \sim q$ und gewichten Sie mit

$$w(x) = \frac{\phi(x)}{q(x)} = \exp\Big(- \mu_q \, x + \tfrac{1}{2}\mu_q^2 \Big),$$

sodass

$$\hat{p}_{\text{IS}} = \tfrac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{X_i < -3\} \, w(X_i)$$

.

c) Berichten Sie für jeden Wert von μ_q die folgenden Größen und vergleichen Sie mit der naiven Methode:

- $\hat{p}_{\text{IS}}$
- ESS
- 95%-CI (über Replikationen oder Batch-Means der Summanden)
- Laufzeit

Aufgabe 7 *Abhängige Sequenz (AR(1)) & Batch-Means-SE*

Wir betrachten eine AR(1)-Sequenz (Y_t) und eine Funktion g.

- **Bildungsgesetz (AR(1)):**

$$Y_t \ = \ \phi \, Y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),$$

mit Startwert $Y_0 = 0$ (oder aus der Stationärverteilung) und typischen Parametern z. B. $\phi = 0{,}8$, $\sigma = 1$.

- **Zu schätzen:** den Mittelwert $\mu = \mathbb{E}[g(Y)]$ für $g(y) = e^{-y^2}$

a) Generieren Sie N aufeinanderfolgende Realisationen $(Y_t)_{t=1}^N$ (zuzüglich Burn-in) mit folgendem Python-Helfer:

```
import numpy as np

def ar1(n, phi=0.8, sigma=1.0, burnin=1000, seed=0):
    rng = np.random.default_rng(seed)
    y = np.empty(n + burnin)
    y[0] = 0.0
    eps = rng.normal(0.0, sigma, size=n + burnin)
    for t in range(1, n + burnin):
        y[t] = phi * y[t-1] + eps[t]
    return y[burnin:]  # drop burn-in
```

b) Setzen Sie $X_t = g(Y_t)$ und schätzen Sie $\hat{\mu} = \frac{1}{N} \sum_{t=1}^{N} X_t$.

c) Vergleichen Sie den *naiven* SE $s/\sqrt{N}$ mit einem *Batch-Means*-SE (z. B. $B = 30$ Batches):

```
def batch_means_se(values, B=30):
    n = len(values)
    m = n // B
    use = m * B
    y = np.asarray(values[:use]).reshape(B, m).mean(axis=1)
    se = y.std(ddof=1) / np.sqrt(B)
    return se, B, m
```

d) Diskutieren Sie den Unterschied in Abhängigkeit von ϕ.

Aufgabe 8 *Stoppen bei erreichter Präzision (sequenziell)*

Stoppen Sie die MC-Schätzung bei erreichtem SE-Ziel (kein fixes N vorgegeben).

- **Zielpräzision:** $\varepsilon = 3 \times 10^{-4}$ für $g(x) = e^{-x^2}$ mit $X \sim \mathcal{U}(0, 1)$.
- **Vorgehen:** Ziehen Sie aus X in Batches der Größe $B = 1{,}000$, aktualisieren Sie nach jedem Batch $\widehat{\mathrm{SE}}$ und *stoppen* Sie, sobald $\widehat{\mathrm{SE}} \leq \varepsilon$ **oder** ein Budget $N_{\max} = 300{,}000$ ausgeschöpft ist.

Berichten Sie: finales N, $\hat{\mu}$, $\widehat{\mathrm{SE}}$, 95%-CI und ob das Budget eingehalten wurde.

Hinweis: Starten Sie mit einem kleinen Pilot (N_0), um s zu initialisieren.

Aufgabe 9 *Stichprobengröße aus Pilotdaten (Handrechnung)*

Sie planen ein MC-Experiment und möchten die *absolute Präzision* ε bei 95% Konfidenz erreichen. Aus einem Pilotlauf mit $N_0 = 5000$ erhalten Sie eine Schätzung $s = 0{,}45$ (für die Zielgröße $g(X)$).

a) Bestimmen Sie die nötige Gesamtstichprobe

$$N^* = \left\lceil \left(\frac{z_{0.975} \cdot s}{\varepsilon} \right)^2 \right\rceil \quad \text{mit} \quad z_{0.975} \approx 1{,}96,$$

für $\varepsilon = 0{,}01$.

b) Wie viele zusätzliche Ziehungen sind noch nötig (gegenüber N_0)?

c) Wiederholen Sie die Rechnung für $\delta = 10\%$ relativ zu $|\hat{\mu}| = 0{,}27$:

$$N^* = \left\lceil \left(\frac{z_{0.975} \cdot s}{\delta\,|\hat{\mu}|} \right)^2 \right\rceil.$$

Geben Sie Zwischenschritte und gerundete Endwerte an (auf ganze N aufrunden).