TP5 — Gibbs, mines de charbon (corrige)

Source : TP5/gibbs_student_mines_corrige.ipynb — PDF compile (code execute).

In [1] :

```
import numpy as np
import matplotlib as mpl
import pandas as pd
import matplotlib.pylab as plt
from scipy import stats
import seaborn as sns
sns.set_theme(color_codes=True)
plt.close("all")
```

1. Coal mines

This model analyzes the data on mining accidents, represented as a sequence of count data $(X_i)_{i=1851}^{1967}$, covering the years from 1851 to 1967. Here, X_i indicates the number of mining accidents that occurred in year \(i \), resulting in a total of 112 observations.

Statistical Modeling

The data is assumed to follow a piecewise Poisson distribution, where the accident counts are modeled as independent and identically distributed (i.i.d.) random variables. Specifically:

$$ \begin{aligned} X_i &\sim \mathcal{P}(\lambda_1) &\text{ for } i = 1851, \ldots, \tau, \\ X_i &\sim \mathcal{P}(\lambda_2) &\text{ for } i = \tau+1, \ldots, 1967, \end{aligned} $$

where λ_1 and λ_2 are the Poisson rate parameters for the two time periods, separated by an unknown change-point τ. The goal is to estimate the rate of mining accidents before and after this change-point τ, where λ_1 represents the rate for the initial period, and λ_2 represents the rate following τ.

Prior Distributions

To complete the model, we specify prior distributions for the parameters. We assume:

• **λ_1** and **λ_2** follow Gamma priors with parameters $\Gamma(1, 10)$, which provides flexibility while still concentrating the rates around low values.

• **τ** follows a uniform prior over the set $\{1851, \dots, 1967\}$, reflecting no prior knowledge of the change-point location within this range.

The choice of priors and the structure of the Poisson model allows us to perform Bayesian inference on the rate parameters λ_1, λ_2, and the change-point τ, providing insights into any shifts in accident rates over time.

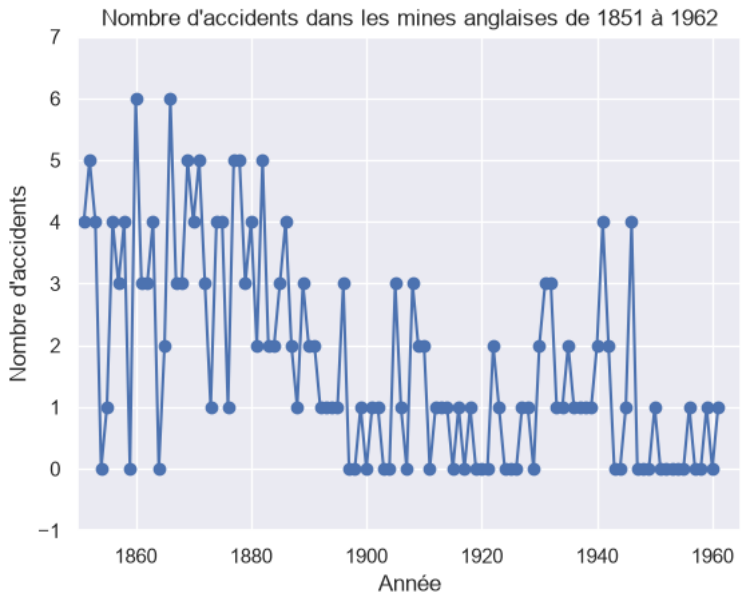
In [2] :

```
disasters_array = np.array([4, 5, 4, 0, 1, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6,
3, 3, 5, 4, 5, 3, 1, 4, 4, 1, 5, 5, 3, 4, 2, 5,
2, 2, 3, 4, 2, 1, 3, 2, 2, 1, 1, 1, 1, 3, 0, 0,
1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1,
0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2,
3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 4, 2, 0, 0, 1, 4,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1])

n_count_data = len(disasters_array)

years = np.array(range(1851,1962))

fig = plt.figure()
plt.plot(years, disasters_array,'-o')
plt.title('Nombre d\'accidents dans les mines anglaises de 1851 à 1962')
plt.xlabel('Année')
plt.ylabel('Nombre d\'accidents')
plt.axis([1850, 1965, -1, 7])
plt.show()
```




```
<Figure size 640x480 with 1 Axes>
```

Q1. GIve the joint distribution of the model and the a posteriori distribution

Q1 — Corrigé. Notons $x=(x_0,\dots,x_{n-1})$, $n=112$, $\tau\in\{0,\dots,n-1\}$ l'indice de rupture (années $1851+\tau$). Vraisemblance : $L=\prod_{i\le\tau}e^{-\lambda_1}\lambda_1^{x_i}/x_i!\prod_{i>\tau}e^{-\lambda_2}\lambda_2^{x_i}/x_i!$. Priors : $\lambda_j\sim\Gamma(a_0=1,b_0=10)$ (shape/rate, $p(\lambda)\propto\lambda^{a_0-1}e^{-b_0\lambda}$), $\tau\sim\mathcal{U}\{0,\dots,n-1\}$. Loi jointe $p(x,\lambda_1,\lambda_2,\tau)=L\times p(\lambda_1)p(\lambda_2)p(\tau)$; posterior $p(\lambda_1,\lambda_2,\tau|x)\propto L\times p(\lambda_1)p(\lambda_2)$ (le prior de τ est constant).

Q2. Give the conditional distribution of λ_1 given λ_2,τ and the one of λ_2 given λ_1,τ.

Q3. Give the conditional distribution of τ given λ_1,λ_2

Q2 — Corrigé. Par conjugaison Poisson–Gamma : $\lambda_1\mid\tau,x\sim\Gamma(a_0+S_1(\tau),\,b_0+n_1(\tau))$ avec $n_1=\tau+1$, $S_1=\sum_{i\le\tau}x_i$; $\lambda_2\mid\tau,x\sim\Gamma(a_0+S_2(\tau),\,b_0+n_2(\tau))$, $n_2=n-\tau-1$, $S_2=\sum_{i>\tau}x_i$. (Ne dépend pas de l'autre λ.)

Q3 — Corrigé. $p(\tau=t\mid\lambda_1,\lambda_2,x)\propto \exp(-(t+1)\lambda_1-(n-t-1)\lambda_2)\,\lambda_1^{S_1(t)}\lambda_2^{S_2(t)}$, normalisée sur $t=0,\dots,n-1$ (calcul en log + log-sum-exp, tirage catégoriel).

Q4. Based on the previous questions, implement a Gibbs algorithm to sample from the a posteriori distribution

In [3] :

```
# Q4 : Gibbs Poisson-Gamma + rupture categorielle
# Convention : Gamma(shape, rate) ; np.random.gamma utilise shape & scale=1/rate.
a0, b0 = 1.0, 10.0

def gibbs_mines(x, n_iter=12000, burn=2000, seed=0):
rng = np.random.default_rng(seed)
n = len(x)
cumsum = np.concatenate(([0], np.cumsum(x)))
lam1 = np.empty(n_iter); lam2 = np.empty(n_iter); tau = np.empty(n_iter, dtype=int)
# init
t = n // 2
for it in range(n_iter):
n1, n2 = t + 1, n - t - 1
s1, s2 = cumsum[t + 1], cumsum[n] - cumsum[t + 1]
l1 = rng.gamma(a0 + s1, 1.0 / (b0 + n1))
l2 = rng.gamma(a0 + s2, 1.0 / (b0 + n2))
# log-poids de tau (vectorise)
tt = np.arange(n)
S1 = cumsum[tt + 1]
S2 = cumsum[n] - S1
logw = -(tt + 1) * l1 - (n - tt - 1) * l2 + S1 * np.log(max(l1, 1e-300)) + S2 *
np.log(max(l2, 1e-300))
logw -= logw.max()
w = np.exp(logw); w /= w.sum()
t = int(rng.choice(n, p=w))
lam1[it], lam2[it], tau[it] = l1, l2, t
return {"lambda1": lam1[burn:], "lambda2": lam2[burn:], "tau": tau[burn:]}

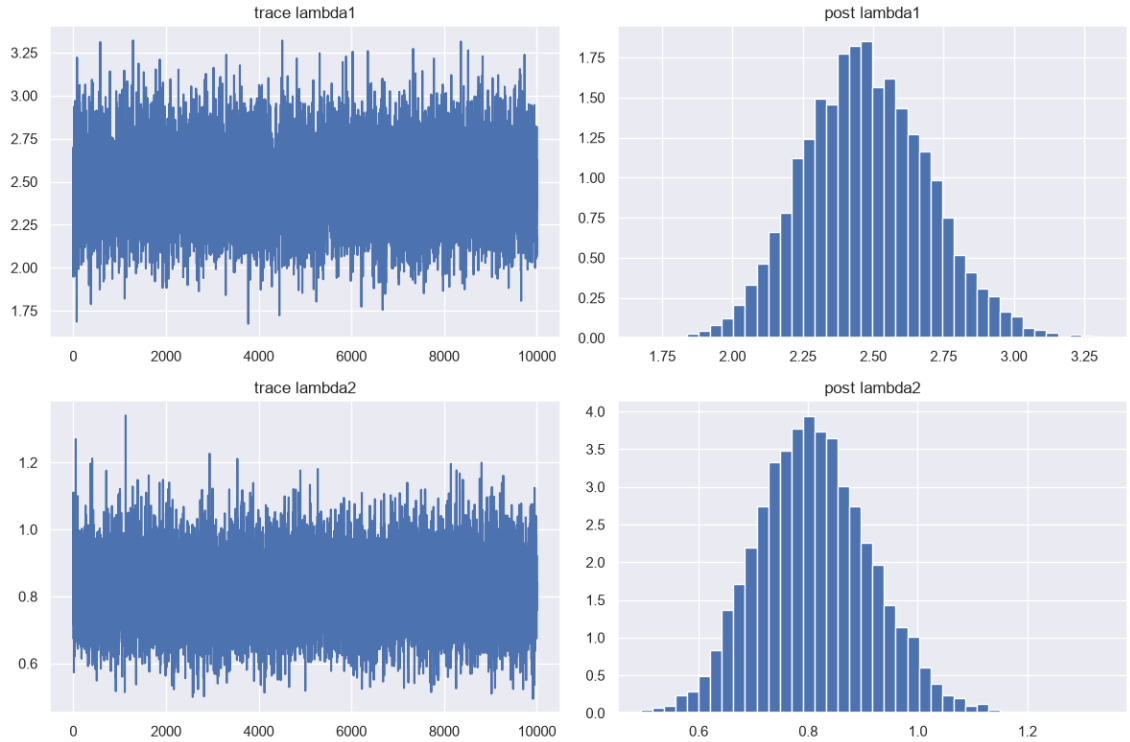
# fix annees (le notebook d'origine avait range(1851,1962) incoherent)
years = np.arange(1851, 1851 + len(disasters_array))
```

Q5. Use your Gibbs implementation, to estimate the a posteriori mean of the parameters

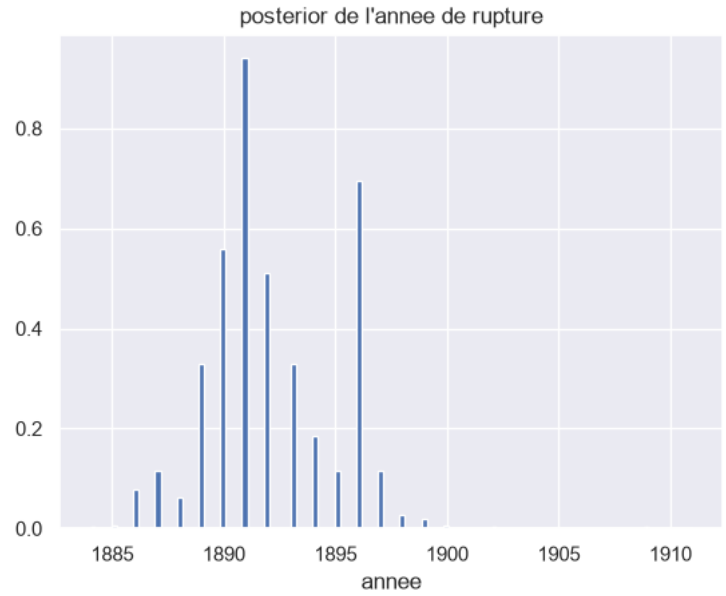
In [4] :

```
# Q5 : moyennes a posteriori + traces + histogrammes
res = gibbs_mines(disasters_array, n_iter=15000, burn=5000, seed=1)
for k in ["lambda1", "lambda2"]:
print(f"{k} : moyenne post = {res[k].mean():.3f}, std = {res[k].std():.3f}")
print(f"tau (indice) : moyenne = {res['tau'].mean():.1f} -> annee ~
{1851+res['tau'].mean():.1f}")
print(f"annee mediane de rupture = {1851+int(np.median(res['tau']))}")
fig, ax = plt.subplots(2, 2, figsize=(12, 8))
ax[0,0].plot(res["lambda1"]); ax[0,0].set_title("trace lambda1")
ax[0,1].hist(res["lambda1"], bins=40, density=True); ax[0,1].set_title("post lambda1")
ax[1,0].plot(res["lambda2"]); ax[1,0].set_title("trace lambda2")
ax[1,1].hist(res["lambda2"], bins=40, density=True); ax[1,1].set_title("post lambda2")
plt.tight_layout(); plt.show()
plt.hist(1851 + res["tau"], bins=len(disasters_array), density=True)
plt.title("posterior de l'annee de rupture"); plt.xlabel("annee"); plt.grid(True);
plt.show()
# overlay moyennes sur les donnees
plt.plot(years, disasters_array, "-o", ms=3, label="donnees")
plt.axhline(res["lambda1"].mean(), color="r", label="lambda1 post-mean")
plt.axhline(res["lambda2"].mean(), color="g", label="lambda2 post-mean")
plt.axvline(1851 + np.median(res["tau"]), color="k", linestyle="--", label="rupture
mediane")
plt.legend(); plt.grid(True); plt.show()
```

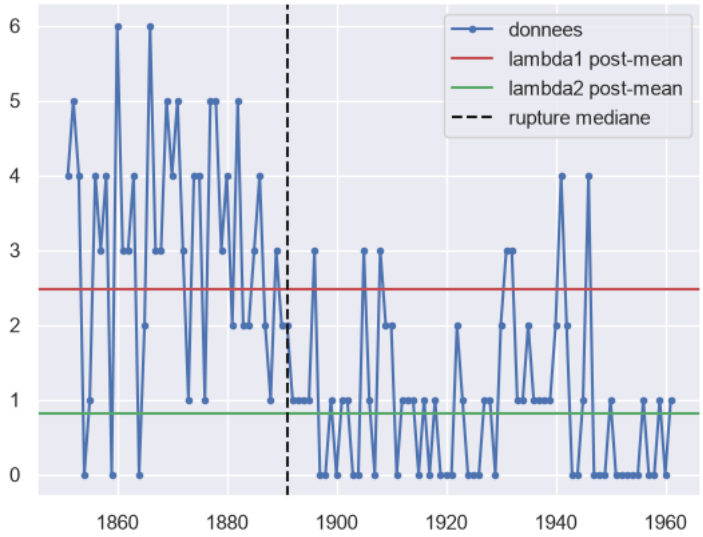
```
lambda1 : moyenne post = 2.475, std = 0.223
lambda2 : moyenne post = 0.811, std = 0.104
tau (indice) : moyenne = 41.1 -> annee ~ 1892.1
annee mediane de rupture = 1891
```




```
<Figure size 1200x800 with 4 Axes>
```




```
<Figure size 640x480 with 1 Axes>
```




```
<Figure size 640x480 with 1 Axes>
```

Q6. (optional) Use you Gibbs implementation to estimate α-credible regions for the parameters

In [5] :

```
        # Q6 (optionnel) : regions credibles alpha (intervalles HPB approximes par quantiles)
        alpha = 0.05
        for k in ["lambda1", "lambda2"]:
        lo, hi = np.quantile(res[k], [alpha/2, 1-alpha/2])
        print(f"{k} IC 95% = [{lo:.3f}, {hi:.3f}]")
        t_sorted = np.sort(1851 + res["tau"])
        lo, hi = np.quantile(t_sorted, [alpha/2, 1-alpha/2])
        print(f"rupture IC 95% = [{lo:.0f}, {hi:.0f}]")
```

```
lambda1 IC 95% = [2.061, 2.930]
lambda2 IC 95% = [0.620, 1.022]
rupture IC 95% = [1887, 1897]
```