**Idée de projet unique et complet :**

**Simulation et prédiction des flux dans un hôpital / service d’urgence**

Ce projet est **très riche** et permet d’utiliser quasiment **toutes les lois** du cours :

**1️⃣ Arrivées des patients → Loi de Poisson**

* Hypothèse : les arrivées aux urgences suivent un **processus de Poisson** (nombre d’arrivées par heure/jour).
* Vérification : test d’ajustement sur données simulées ou réelles.
* Utilité : prédire combien de patients vont arriver demain (distribution de Poisson).

**2️⃣ Temps entre deux arrivées → Loi exponentielle**

* Lien : si les arrivées suivent un processus de Poisson, les intervalles de temps suivent une **loi exponentielle**.
* Application : simuler les temps d’attente entre deux patients.
* Estimation du paramètre θ\theta par MoM ou MLE.

**3️⃣ Gravité du cas → Loi de Bernoulli / Binomiale**

* Exemple : probabilité qu’un patient soit "cas grave" (succès/échec → Bernoulli).
* Si on observe nn patients, le nombre de cas graves suit une **loi Binomiale**.
* Application : prédire le nombre de cas graves dans une journée.

**4️⃣ Nombre d’examens nécessaires → Loi géométrique**

* Exemple : chaque patient doit subir des examens (ex. prise de sang, radios).
* On peut modéliser le nombre d’examens jusqu’à obtention d’un résultat concluant par une **loi géométrique**.
* Intérêt : temps moyen de diagnostic.

**5️⃣ Durée d’hospitalisation → Loi exponentielle ou Gamma**

* Exponentielle si on suppose une mémoire sans effet,
* Gamma si plusieurs "étapes" indépendantes avant sortie.
* Application : estimer le temps moyen d’occupation d’un lit.

**6️⃣ Taux de succès d’un traitement → Loi Beta**

* On peut modéliser la probabilité de succès d’un traitement comme une **loi Beta** (cas typique en bayésien).
* Exemple : après observation de xx succès sur nn patients, la loi a posteriori est Beta(x+1,n−x+1)\text{Beta}(x+1, n-x+1).
* Application : donner un intervalle de confiance bayésien pour l’efficacité d’un traitement.

**7️⃣ Capacité hospitalière → Loi Uniforme / distributions empiriques**

* Exemple : modéliser un délai "aléatoire" (ex. temps administratif, transport, imprévisible) par une **loi uniforme**.
* Cela complète le modèle en ajoutant de la variabilité.

**🎯 Résultats attendus du projet**

1. **Modélisation probabiliste complète** d’un service hospitalier (arrivées, attentes, diagnostics, hospitalisations, sorties).
2. **Estimation des paramètres** par méthode des moments et maximum de vraisemblance.
3. **Comparaison de plusieurs lois** pour chaque phénomène (tests d’adéquation, QQ-plot).
4. **Simulation Monte Carlo** pour prévoir un planning d’occupation des lits / surcharge.
5. **Interface utilisateur (Streamlit/Dash)** :
   * Choix d’un scénario (faible affluence vs épidémie),
   * Visualisation des distributions,
   * Estimation du risque de saturation des lits,
   * Intervalles de confiance et prédictions affichés clairement.

**💡 Impact pour ton portfolio**

* **Problème réel et concret (gestion d’un hôpital)** → très parlant dans le domaine **santé** mais aussi en **data science appliquée**.
* **Rigueur mathématique** → tu utilises presque toutes les lois étudiées.
* **Technicité** → estimation des paramètres, IC, tests d’adéquation, simulation.
* **Praticité** → interface utilisable par un gestionnaire d’hôpital pour prendre des décisions (combien de lits prévoir ? combien de médecins de garde ?).
* Tu montres que tu sais **transformer des maths abstraites en outil opérationnel**.

/////////////////////////////////

Excellente question 👌.  
Si je t’ai proposé **Python + Numpy/Scipy/Statsmodels + Streamlit**, ce n’est pas un choix au hasard, c’est stratégique. Voilà le raisonnement :

**1️⃣ Pourquoi Python ?**

* **Langage dominant** en **data science/statistiques** (90% des projets académiques et industriels).
* Énorme écosystème : presque toutes les lois de probabilité que tu vois en cours sont **déjà implémentées**.
* Lisible et simple → tu peux te concentrer sur la **méthodologie statistique**, pas sur la syntaxe.
* Universel : facile de passer de la recherche académique au monde pro.

**2️⃣ Pourquoi Numpy / Scipy / Statsmodels ?**

Ce sont les **blocs fondamentaux** pour appliquer la théorie vue en cours :

* **Numpy** : génération aléatoire et manipulation de vecteurs/matrices (indispensable).
* **Scipy.stats** : toutes les lois étudiées en stats (Poisson, exponentielle, binomiale, beta, Student, etc.) sont déjà codées → tu peux générer, calculer densité, quantiles, tests, etc.
* **Statsmodels** : spécialisé en **inférence statistique** → intervalles de confiance, maximum de vraisemblance, régressions statistiques.

👉 Ça correspond directement à tes cours (méthode des moments, MLE, IC, tests d’hypothèse).

**3️⃣ Pourquoi Pandas ?**

* Pour gérer tes données (simulation ou données réelles).
* Te permet de manipuler des tableaux comme en Excel, mais avec la puissance de Python.

👉 Très utile si tu passes à des datasets hospitaliers ou financiers réels.

**4️⃣ Pourquoi Matplotlib / Seaborn ?**

* Tu dois **montrer visuellement** que les lois collent aux données (histogrammes, densité, QQ-plot).
* Sans la visualisation, c’est abstrait → avec, tu prouves la validité de tes estimateurs.

**5️⃣ Pourquoi Streamlit (interface) ?**

* Facile à utiliser → quelques lignes de code suffisent pour créer une **appli web interactive**.
* Gratuitement hébergeable sur **Streamlit Cloud** → parfait pour ton portfolio (un recruteur pourra tester ton projet en ligne).