Travail pratique **Simulation de Monte-Carlo et paradoxe des anniversaires**

Ma date d'anniversaire ? Le 21 nivôse, vous aussi ?

Rappelons que le paradoxe des anniversaires vient de la réponse à la question suivante : Quel est le plus petit nombre de personnes à réunir pour avoir au moins une chance sur deux que deux personnes ou plus du groupe aient leur anniversaire le même jour ?

Si on considère 365 dates d'anniversaire possibles, toutes équiprobables, la réponse n'a rien de paradoxale mais contredit l'intuition de la plupart des gens car réunir 23 personnes suffit.

En effet, pour 365 dates équiprobables, la probabilité p_k qu'un groupe de k personnes en contiennent au moins deux ayant leur anniversaire le même jour est

$$p_k = 1 - \frac{A_k^{365}}{\overline{A_k^{365}}} = 1 - \frac{A_k^{365}}{365^k} = 1 - \frac{365!}{365^k(365-k)!}$$

et, à dix chiffres significatifs, on a

$$p_{22} = 0{,}4756953077\ldots \quad \text{et} \quad p_{23} = 0{,}5072972343\ldots$$

Et on simule, encore et encore

L'objectif premier de ce travail pratique est la mise en place d'une simulation de Monte Carlo permettant de calculer un estimateur $\hat{p}$ de la probabilité p que, dans un groupe de K personnes, au moins M d'entre elles aient leur anniversaire le même jour (dans un monde où chaque année compte Y jours et où les probabilités de naître tel ou tel jour sont équiprobables), ainsi qu'un intervalle de confiance pour cette probabilité.

Pour cela, vous devez compléter les sources fournies

- ▷ en ajoutant une classe implémentant l'interface `Experiment` et permettant de simuler l'expérience de Bernoulli suivante : choisir K dates au hasard, parmi Y dates équiprobables, et retourner 1 (succès) si au moins une date a été choisie M fois ou plus et 0 (échec) sinon ;
- ▷ en complétant la méthode `simulateTillGivenCIHalfWidth` de la classe `MonteCarloSimulation`, méthode qui doit permettre de simuler une expérience aléatoire donnée (implémentant l'interface `Experiment`) jusqu'à ce que la demi-largeur de l'intervalle de confiance au seuil $1-\alpha$, pour la mesure de performance retournée par l'expérience, passe en dessous d'une valeur maximale $\Delta_{\max}$.

La méthode `simulateTillGivenCIHalfWidth` prend sept paramètres :

- ▷ l'expérience `exp` à simuler ;
- ▷ le seuil `level` $(1-\alpha)$ de couverture de l'intervalle de confiance pour la performance asspociée à l'expérience ;

- ▷ la demi-largeur maximale `maxHalfWidth` ($\Delta_{\max}$) de l'intervalle de confiance produit ;
- ▷ le nombre `initialNumberOfRuns` (N_{init}) de réalisations à générer initialement ;
- ▷ le nombre `additionalNumberOfRuns` (N_{add}) de réalisations supplémentaires à générer tant que la borne $\Delta_{\max}$ pour la demi-largeur de l'intervalle de confiance n'est pas atteinte ;
- ▷ une source aléatoire `rnd` à passer à l'expérience à simuler ;
- ▷ un objet de type `StatCollector` permettant d'accumuler les résultats simulés et de calculer les estimateurs de base.

Afin d'obtenir un intervalle de confiance dont la demi-largeur ne dépasse pas la valeur $\Delta_{\max}$, la démarche suivante doit être utilisée :

1) On commence par réaliser N_{init} simulations de l'expérience (voir la méthode `simulateNRuns`).
2) À partir des données récoltées on calcule une estimation du nombre N de réalisations à générer afin d'obtenir un intervalle de confiance dont la demi-largeur ne dépasse pas $\Delta_{\max}$ (voir la page 48 du cours). Cette valeur de N est ensuite arrondie, vers le haut, au plus proche multiple de N_{add}.
3) La simulation est **poursuivie** jusqu'à atteindre N réalisations de l'expérience.
4) Si la demi-largeur de l'intervalle de confiance, calculée sur la base de ces N réalisations, est inférieure ou égale à $\Delta_{\max}$ le processus s'arrête. Sinon N_{add} simulations supplémentaires sont effectuées avant de recalculer un nouvel intervalle de confiance et de retester la condition d'arrêt. Ce processus est répété jusqu'à ce que la condition d'arrêt soit satisfaite.

Ayez confiance !

Trois séries de simulations vous permettront d'observer le comportement de la méthode.

1) Premièrement vous calculerez une estimation et un intervalle de confiance pour la probabilité p_{23} d'avoir au moins deux personnes avec la même date d'anniversaire dans un groupe de 23 personnes. Vous commencerez par calculer un intervalle de confiance dont la demi-largeur ne dépasse pas 10^{-4} puis vous recommencez le processus à deux reprises en divisant, à chaque fois, par deux la demi-largeur maximale.

   Votre programme affichera, au minimum, l'estimation $\hat{p}$ de la probabilité d'observer au moins 2 dates identiques, le nombre de réalisations générées en tout ainsi que la demi-largeur de l'intervalle de confiance (et l'intervalle de confiance lui-même si vous voulez).

   Les valeurs des différents paramètres à utiliser sont les suivantes :

   - ▷ seuil de confiance `level` $= 1 - \alpha = 95\%$;
   - ▷ demi-largeur maximale initiale `maxHalfWidth` $= \Delta_{\max} = 10^{-4}$;
   - ▷ nombre initial de réalisations `initialNumberOfRuns` $= N_{\text{init}} = 10^6$;
   - ▷ nombre de réalisations supplémentaires `additionalNumberOfRuns` $= N_{\text{add}} = 10^5$;
   - ▷ taille des groupes de personnes : $K = 23$;

- ▷ nombre de jours par année : $Y = 365$;
- ▷ nombre minimum de personnes ayant leur anniversaire le même jour : $M = 2$;
- ▷ graine du générateur de nombres pseudo-aléatoires : `0x134D6EE` (à utiliser au début de chacune des trois simulations).

2) Dans un deuxième temps vous étudierez le seuil de couverture des intervalles de confiance calculés à partir du théorème central limite.

   Plus précisément, après avoir réinitialisé votre générateur pseudo-aléatoire avec la graine donnée, vous utiliserez la méthode `simulateNRuns` pour générer, à chaque fois, un million ($N = 10^6$) de réalisations de l'expérience, avec les valeurs $K = 23$, $Y = 365$ et $M = 2$ comme ci-dessus. Vous répéterez le processus 1000 fois afin de calculer 1000 intervalles de confiance au seuil de 95% tout en déterminant ceux qui contiennent la vraie valeur de p (c.-à-d. la valeur p_{23} donnée en début d'énoncé).

   Vous utiliserez les résultats obtenus pour calculer un seuil empirique de couverture des intervalles calculés (le pourcentage des intervalles contenant p) et un intervalle de confiance (toujours à 95%) associé à cette estimation.

3) Finalement vous déterminerez le nombre minimal K de personnes à réunir pour avoir plus d'une chance sur deux que $M = 3$ personnes au moins aient leur anniversaire le même jour (pour des années comptant $Y = 365$ jours). Comme précédemment vous réinitialiserez votre générateur pseudo-aléatoire avant le début des calculs. Vous effectuerez votre recherche pour des groupes dont la taille est comprise entre $K = 80$ et $K = 100$.

À vos marques, prêts, . . . Stop ! C'est fini

- ▷ Le travail de programmation est à effectuer par groupe de deux, en Java, version 21 (ou 23).
- ▷ L'archive contenant les sources à compléter, est disponible sur le site Cyberlearn du cours.
- ▷ Vous devez rendre une archive (au format `zip`) contenant toutes les sources de votre projet complété. Vous prêterez une attention toute particulière aux commentaires de votre implémentation de la méthode `simulateTillGivenCIHalfWidth`. Votre archive contiendra également un fichier texte (ou markdown) contenant les résultats pour les trois séries de simulations demandées.
- ▷ Vous devez rendre votre travail sur Cyberlearn au plus tard le **dimanche 26 janvier 2025** (avant minuit).