

Générateurs (semi-)automatiques

de plein de trucs

Mercredi 23 avril 2014

stephane@gonnord.org - <http://blog.mp933.fr/>

EXERCICE 1 Lancer Spyder/whatever, créer un nouveau document python, immédiatement sauvé au bon endroit. Écrire une commande absurde dans le fichier ; exécuter.

1 TP de probabilités expérimentales

Objectif : observer le comportement d'une variable aléatoire via ses moyennes sur un certain nombre de tirages.

1.1 Algorithme

- Choisir un machin qui respecte un vague loi. Bref, une variable aléatoire. Inutile de connaître sa loi, son espérance, sa variance... Cela peut être un tirage à pile ou face, un dé non biaisé... ou biaisé, une loi continue... un jeu asymétrique, une marche...
- Faire tester le comportement de la moyenne de n « tirages » lorsque n tend vers l'infini (loi des grands nombres).
- Faire 100 ($\simeq +\infty$) tirages et noter la moyenne. Faire ça N fois avec N grand. L'histogramme doit ressembler à une gaussienne à peu près connue (théorème central limite)...
- Si c'est faisable, calculer la moyenne/l'écart-type (exo de maths); comparer alors l'histogramme à la gaussienne fournie par le théorème central limite.

1.2 Des exemples

En pièces jointes : un DM et un TP.

- Dans le DM, on observe des lancers de pile/face, puis on simule un jeu de dé absurde (improvisé) paramétré... pour essayer de déterminer expérimentalement l'espérance de gain d'un des joueurs. Fournit un exercice facile et de bon goût au prof de maths!
- Dans le TP, on observe des lancers de dés : loi des grands nombres¹ et théorème central limite².

1.3 À vous de jouer

1. Un petit project Euler (numéro 205). Vous n'allez pas réussir à le résoudre par simulation, mais vous pouvez quand même obtenir les premières décimales!

Peter has nine four-sided (pyramidal) dice, each with faces numbered 1, 2, 3, 4. Colin has six six-sided (cubic) dice, each with faces numbered 1, 2, 3, 4, 5, 6.

Peter and Colin roll their dice and compare totals : the highest total wins. The result is a draw if the totals are equal.

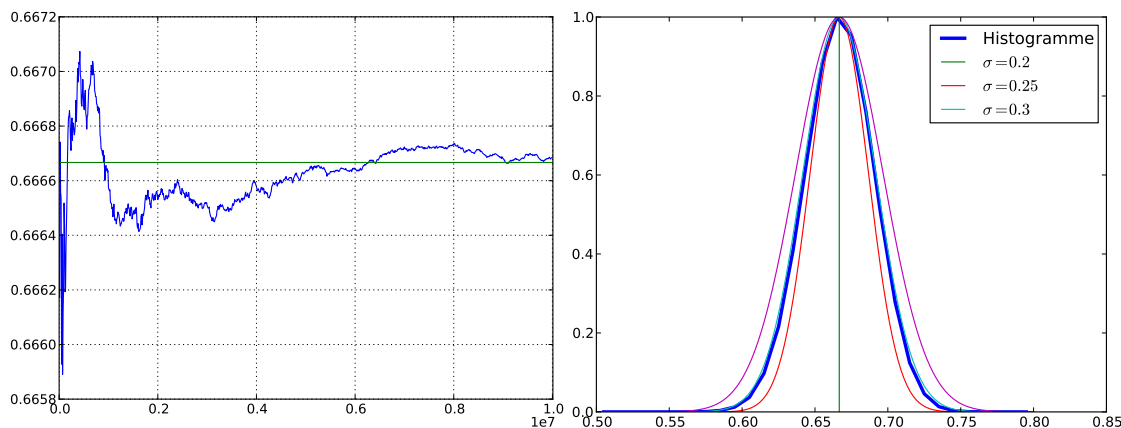
What is the probability that Pyramidal Pete beats Cubic Colin? Give your answer rounded to seven decimal places in the form 0.abcdefg

2. En reprenant le fichier solution du tp pré-cité, étudier expérimentalement l'espérance et la variance du maximum de deux variables vérifiant une loi de répartition uniforme sur $[0, 1]$.

Ce n'est qu'après coup, en notant que les espérances et écart-type ressemblaient à des choses simples que j'ai fait le calcul (trivial), qui a confirmé les observations.

1. Mais mais mais... la moyenne des lancers semble tendre vers l'espérance!

2. la distribution après 100 lancers ressemble à une gaussienne



Moyennes sur 10^7 termes, puis distribution sur 10^6 runs de moyennes de 100 tirages.

1.4 Idées en vrac

- Jouer avec des chaînes de Markov.
- Estimer des écarts-types ; y compris à l'œil (en comparant à la Gaussienne) ; ça peut être amusant.
- Par dichotomie, trouver un paramètre rendant un jeu paramétré équitable.
- Sur projecteuler.net, des tas de choses sympas...
- Tester le paradoxe des anniversaires (simulations vs. théorie).
- Dans le très excellent « Probabilités pour les non probabilistes » (W. Appel), vous trouverez des tas d'exercices joliment scénarisés ! Ça marche aussi avec d'autres livres de probabilités, mais c'est moins sympa...

2 TP/DM/DS sur machine : une chasse au trésor

Objectif : mettre en œuvre les algorithmes et/ou idées de la partie « ingénierie numérique », avec en particulier les questions d'approximations numériques.

2.1 Mise en place

- Faire le casting, en choisissant les éléments du programme à tester (calculs d'intégrale, résolution de systèmes, d'équations sur les réels, d'équations différentielles...).
- Écrire le scénario, en commençant par les étapes posant le moins de problèmes de précision...
- Réaliser la carte au trésor, en partant d'un paramètre ajustable, et en enchaînant les étapes. On écrit en parallèle la résolution, pour vérifier que les résultats sont raisonnables (précis et distinguables)
- Lancer un script qui génère tout ça !
- Constater qu'on a mal réglé les paramètres : tous les trésors sont au même endroit.
- Recommencer...

On peut aussi faire une course à point, où les items sont indépendants, et rapportent un nombre de points variables à cumuler.

2.2 Un exemple

En annexes, un devoir à la maison (pour ceux que ça amusait). On détermine un paramètre à partir de sa date de naissance ; ce paramètre nous donne un nom de fichier, dans lequel il y a des données numériques à extraire, fournissant un système linéaire à résoudre... donnant un paramètre à utiliser dans une équation polynomiale à résoudre, dont la solution est un paramètre intervenant dans une intégrale... dont le résultat est un paramètre pour une équation différentielle qu'il convient de résoudre numériquement, et qui donne accès au trésor !

2.3 À vous de jouer

En utilisant `numpy/scipy`, trouvez votre trésor !

2.4 Idées en vrac

- On peut assez facilement inclure des bases de données (pour la fin de l'année...)
- Les élèves peuvent créer eux-même leur chasse au trésor.
- On doit pouvoir inclure des choses non numériques et plus algorithmiques : le résultat d'un travail sur une permutation, etc.

3 TP/DS : bases de données

Objectif pour les élèves : se familiariser avec SQL, et la pratique effective de bases de données. Pour le professeur : trouver/générer rapidement des bases de données et des requêtes associées.

3.1 Création des tables

Il s'agit de travailler avec de grosses tables, pour lesquelles une requête SQL sera indispensable (à 5 entrées, les yeux vont en général plus vite que les doigts...)

- Il existe des tables toutes faites sur les sites des collègues (prépa ou université : les bases de données sont beaucoup enseignées, et depuis longtemps, en licence)...
- On peut importer une table depuis un fichier csv (directement depuis votre clicodrome de bases de données). Vous en trouvez par exemple sur opendata.paris.fr. Attention, site un peu addictif!
- Un petit script python permet d'alimenter facilement une base : que ce soit une base SQLite via `sqlite3` ou une base MySQL via `MySQLdb`...

3.2 Des exemples de bases (de données...)

Dans les documents, vous trouverez les bases suivantes, ainsi que quelques scripts (et/ou csv) ayant permis de les générer.

- Une base absurde de 100000 triangles (trois longueurs). Histoire de faire des requêtes simples ; a plutôt bien marché comme premier exemple au premier TP).
- Une base de communes/départements/régions française : permet de faire des jointures, pour de vraies requêtes intéressantes (plaît bien aux élèves ; testé en TP avant l'introduction formelle des jointures, et c'était une bonne idée!).
- Une base de colleurs/collés/colles : écrite il y a longtemps... et ça se voit (elle est très mal conçue). Je la laisse néanmoins (le mode de création pourra servir). Il y a un TP « prof » pour la création de la base, et un TP « élève » pour les requêtes.
- Deux csv récupérés sur opendata.paris.fr : les parcs et jardins de Paris, ainsi que les prénoms enregistrés par l'état civil de Paris de 2004 à 2012 (une mine de requêtes!).

3.3 Choix des requêtes

C'est un peu toujours les mêmes :

- projections, sélections ;
- jointures ;
- fonctions d'agrégation.

Dans chaque cas, on peut (selon contexte) demander de concevoir une requête répondant à tel problème.. ou au contraire demander la signification de telle requête.

Quelques exemples ?

- Combien de fois votre prénom a-t-il été donné à Paris de 2004 à 2012 ?
- Quel ont été les dix prénoms les plus donnés ? Les moins ?
- Quels sont les prénoms donnés plus de cent fois chaque année et au moins 1200 fois sur les 9 années ?
- Quels sont les communes ayant plus de lettres dans leur nom que d'habitants ?
- Quelle est la population totale vivant dans des communes de moins de 1000 habitants ?
- Quel collé a la meilleure moyenne ?
- Quels sont les colleurs ayant donné au moins 5 fois une note sous la moyenne ?

3.4 À vous de jouer

- Importer la base des prénoms : créer une base sqlite depuis le fichier csv.
 - Poser (sur cette base de prénoms) dix questions associées à des requêtes SQL... dont plusieurs que vous savez traiter !
 - Créer une base des parcs et jardins depuis Python :
 - choisir un schéma relationnel raisonnable (quelle information voulez-vous retenir?) ;
 - créer la table dans un fichier .db créé depuis python (utiliser les modèles fournis) ;
 - ouvrir le csv, et charger la base ;
 - penser à commiter !
- Ensuite, vous pouvez concevoir quelques requêtes associées à cette bases...

4 DS : couvrir le programme

« Cette année encore, l'épreuve de *<insert whatever here>* a permis de bien évaluer les candidats sur l'ensemble du programme. »

Bon, entre le manque de temps et le fait que ce générateur ne marche pas si bien que ça... je me contenterai de vous fournir trois DM (en plus des deux déjà présentés) et un DS dans l'archive jointe.

