Poker menteur

Au poker menteur, on utilise 5 dés avec des valeurs de 1 à 6, ou 9, 10, valet, dame roi et as.

- En lançant les 5 dés, on peut obtenir des combinaisons particulières classables dans un ordre conventionnel :
    1. rien
    2. une paire
    3. deux paires
    4. un brelan
    5. une petite suite
    6. une grande suite
    7. un full
    8. un carré
    9. une quinte ou poker
- Ces combinaisons sont-elles justifiables sur base de l'analyse combinatoire pour un seul lancer groupé des 5 dés ?
- Examiner en fonction du nombre de lancers simulés la convergence des statistiques vers les probabilités en fonction du type de main.

Solution

Voici un programme permettant de simuler et traiter des lancers

<sxh python; title : Poker_des_01.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

Poker menteur : lancers et mains possibles # 5 dés numérotés [1, 2, 3, 4, 5, 6]

import random

def lancer_de(nfaces = 6):

```
  # résultat lancer d'un seul dé
  try: return random.randint(1,nfaces)
  except: return 0
```

def lancer_des(ndes = 1, nfaces = 6):

```
  # résultat lancer de plusieurs dés. Sortie en liste triée.
  #try: return tuple(sorted([random.randint(1, nfaces) for i in
range(ndes)])) # retourne un tuple
  try: return sorted([random.randint(1, nfaces) for i in range(ndes)]) #
retourne une liste
  except: return ()
```

def main(lancer):

```
  # construit une liste triée par nombre d'occurrences croissant de 6 listes
```

```
  # de 2 nombres : nombre d'occurrences et valeur de la face
  comptage = sorted([[lancer.count(valeur), valeur] for valeur in
range(1,7)])
  #print comptage,  # décommenter pour visualiser comptage
  #
  # construit un tuple des nombres d'occurrences non nulles, par ordre
croissant
  signature = tuple([nombre for nombre, valeur in comptage if nombre != 0])
  #print signature,  # décommenter pour visualiser signature
  main = mains[signature] # qualification correspondante de la main
  if main == "rien, petite ou grande suite":
      # l'analyse de la face manquante permet de déterminer la main
      if lancer[0]==2:
          main = "grande suite"
      elif lancer[4]==5:
          main = "petite suite"
      else:
          main = "rien"
  return main
```

dictionnaire de correspondance pour les tuples des occurrences triées t5=tuple([5]) mains = { (1,1,1,1,1): “rien, petite ou grande suite”, # 3 possibilités dans ce cas (1,1,1,2): “une paire”, (1,2,2): “deux paires”, (1,1,3): “brelan”, (2,3): “full”, (1,4): “carre”, t5: “quinte (poker)” }

n_essais=100 # procéder en faisant varier le nombre for i in range(n_essais):

```
  lancer= lancer_des(5,6)
  print lancer, main(lancer)
```

</sxh>

Et voici un autre programme permettant d'analyser de manière exhaustive tous les lancers possibles. Les parties répétées ne sont pas reprises.

<sxh python; title : Poker_des_03.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

…

lancers_possibles={} histo_mains={} for f1 in range(1,7): # pour toutes les faces possibles du dé 1

```
  for f2 in range(1,7): # pour toutes les faces possibles du dé 2
      for f3 in range(1,7): # pour toutes les faces possibles du dé 3
          for f4 in range(1,7): # pour toutes les faces possibles du dé 4
              for f5 in range(1,7): # pour toutes les faces possibles du dé
5
                  lancer= tuple(sorted([f1,f2,f3,f4,f5]))
                  #print lancer, main(lancer)
                  if lancers_possibles.has_key(lancer):
                      lancers_possibles[lancer][1]+=1
                      histo_mains[lancers_possibles[lancer][0]]+=1
                  else:
```

```
                        main2=main(lancer)
                        lancers_possibles[lancer]=[main2,1]
                        if histo_mains.has_key(main2):
                            histo_mains[main2]+=1
                        else:
                            histo_mains[main2]=1
```

for c,v in lancers_possibles.iteritems():

```
  print c,v
```

for c,v in histo_mains.iteritems():

```
  print c,v
```

print 'somme = ',sum(histo_mains.values()) print '6**5 = ',6**5 </sxh>

Pour la comparaison entre les probabilités issues de l'analyse exhaustive et des statistiques issues de simulations, utiliser les dictionnaires exhaustifs pré-créés et simplement ajouter au programme des nombres correspondants aux occurrences simulées.

Pour 7776 lancers, on obtient par exemple ceci pour la comparaison de l'analyse combinatoire et de la simulation :

```
quinte (poker) [6, 3]
rien [480, 466]
full [300, 299]
carre [150, 175]
deux paires [1800, 1855]
grande suite [120, 142]
brelan [1200, 1192]
une paire [3600, 3518]
petite suite [120, 126]
```

Pour 7 776 000 lancers, on a par exemple :

```
quinte (poker) [6, 5929]
rien [480, 480707]
full [300, 300304]
carre [150, 149591]
deux paires [1800, 1800461]
grande suite [120, 120194]
brelan [1200, 1200807]
une paire [3600, 3597522]
petite suite [120, 120485]
```

From:
https://dvillers.umons.ac.be/wiki/ - **Didier Villers, UMONS - wiki**

Permanent link:
https://dvillers.umons.ac.be/wiki/teaching:exos:poker_menteur?rev=1383470535

Last update: **2013/11/03 10:22**