Math205_Week5.Mon.sagews

Author          Daniel Dugger
Date            2018-10-23T00:26:01
Project         140169db-6121-47d4-bbb8-38353d2304e4
Location        Math205_Week5.Mon.sagews

Original file  Math205_Week5.Mon.sagews

(when downloading, rename file to Math205_Week5.Mon.sagews)

```python
list=[1,2,3,4,5,6]
for a in list:
    print a**2
```

```python
i=0
while i<=5:
    print list[i]**2
    i=i+1
```

```python
suits=['S','C','H','D']
ranks=['A',2,3,4,5,6,7,8,9,10,'J','Q','K']
deck=[]
    #two nested loops                     #This code makes the deck of cards, using two nested loops.
for a in suits:
    for b in ranks:
        deck.append([b,a])

print deck
```

```python
def random_choices(L,k):
    #randomly choose k things from list L, taking them out as we go along. Return the k things in a list.
    output=[]
    i=1
    while i<=k:
        #loop for the k different picks
        a=randint(0,len(L)-1)
        #pick random spot in the current list
        output.append(L[a])
        #put the ath entry of L into our output list
        L=L[:a]+L[a+1:]
        #update the list L by removing the ath object
        i=i+1
    return output

random_choices(deck,5)

def experiment():
    #Randomly choose a hand of 5 cards, and return True if it has at least two of the same rank.
    hand=random_choices(deck,5)
    #First randomly choose the hand of 5 cards
    #   print hand                    #used for debugging
    ranks=[]
    for a in hand:
        #create a new list called "ranks" with only the ranks of the 5 cards
        ranks.append(a[0])
    for b in ranks:
        #for each item in "ranks", see if it appears at least 2 times.
        if ranks.count(b)>=2:
            return True
    return False

experiment()

False

experiment()

[[8, 'H'], ['J', 'S'], [9, 'S'], [8, 'D'], [5, 'C']]
True

experiment()

False

def trials(n):
    #This is the usual "trials" function that we seen many times.
    successes=0
    while i<=n:
        if experiment():
            successes=successes+1
        i=i+1
    return successes*0.0/n

trials(10)
[[8, 'H'], ['A', 'C'], ['Q', 'C'], [7, 'S'], ['J', 'S']]
[['J', 'D'], ['Q', 'D'], ['Q', 'H'], ['Q', 'S'], [4, 'D']]n
[[10, 'C'], ['J', 'H'], [9, 'H'], [2, 'C'], ['Q', 'C']]
[[4, 'H'], ['Q', 'S'], [6, 'S'], [2, 'D'], ['A', 'S']]
[5, 'S']n
[['J', 'C'], [9, 'C'], [7, 'H'], [7, 'H']]
[['J', 'D'], ['K', 'C'], [9, 'H'], [8, 'S']]
[[10, 'H'], ['J', 'S'], ['Q', 'H'], [3, 'H'], [7, 'H']]
[['J', 'D'], ['K', 'C'], [9, 'S'], ['Q', 'H'], [9, 'D']]
0.400000000000000

trials(100)
0.520000000000000
trials(10000)
0.497300000000000