Poker hand simulation - two pairs probability

Global parameters

```
In [1]:  sample_size = 100
         simulation_size = 200
         poker_hand_size = 5
         confidence = 0.95
```

Create class for deck cards

```
In [2]:  from enum import Enum

         class Suit(Enum):
             __order__ = 'CLUBS DIAMONDS HEARTS SPADES'
             CLUBS = 1
             DIAMONDS = 2
             HEARTS = 3
             SPADES = 4

         class Card:
             def __init__(self, value, suit):
                 self.value = value
                 self.suit = suit
```

Create deck

```
In [3]:  deck = []
         for v in range(13):
             for s in Suit:
                 deck.append(Card(v, s))
```

Declare simulation function

```
In [4]: import random

        def runSimulation():

            two_pairs = 0

            for n in range(simulation_size):

                #shuffle deck
                random.shuffle(deck)

                #counts
                pairs_count = 0

                #init dic
                count = {}
                for i in range(13):
                    count[i] = 0

                #count cards
                for i in range(poker_hand_size):
                    card = deck[i].value
                    count[card] += 1

                #count pairs
                for i in range(13):
                    if count[i] == 2:
                        pairs_count += 1

                #count two pairs
                if pairs_count == 2:
                    two_pairs += 1

            probability = two_pairs / simulation_size

            return probability
```

Run simulations

```
In [5]: data = []
        for i in range(sample_size):
            probability = runSimulation()
            data.append(probability)
```

Print statistics

```
In [6]:  import pandas as pd

         df = pd.DataFrame(data)
         df.describe()
```

Out[6]:

	0
count	100.000000
mean	0.046800
std	0.015595
min	0.015000
25%	0.038750
50%	0.045000
75%	0.055000
max	0.095000

```
In [7]:  import numpy as np
         import scipy.stats

         a = 1.0 * np.array(data)
         m, se = np.mean(a), scipy.stats.sem(a)
         h = se * scipy.stats.t.ppf((1 + confidence) / 2., sample_size - 1)
         print("confidence interval is: [" + str(m - h) + ", " + str(m + h) + '
```

```
confidence interval is: [0.0437056906787055, 0.04989430932129452] at
95%
```

Print global parameters

```
In [8]:  print("confidence: " + str(confidence))
         print("sample_size: " + str(sample_size))
         print("simulation_size: " + str(simulation_size))
         print("poker_hand_size: " + str(poker_hand_size))
```

```
confidence: 0.95
sample_size: 100
simulation_size: 200
poker_hand_size: 5
```