hw2_S2023_pt2

February 2, 2023

1 Homework 2 Part 2

This is an individual assignment.

1.1 Description

Create or edit this Jupyter Notebook to answer the questions below. Use simulations to answer these questions. An analytical solution can be useful to check if your simulation is correct but analytical solutions alone will not be accepted as a solution to a problem.

1.2 Problem 1

Consider the experiment where you pick 3 cards at random from a deck of 52 playing cards (13 cards per suit) **without replacement**, i.e., at each card selection you will not put it back in the deck, and so the number of possible outcomes will change for each new draw.

Let D_i denote the event the card is diamonds in the ith draw.

Build a simulation to compute the following probabilities:

1. $P(D_1)$
2. $P(D_1 \cap \overline{D}_2)$
3. $P(D_1 \cap \overline{D}_2 \cap \overline{D_3})$
4. $P(D_3 | D_1 \cap \overline{D}_2)$

Note: to sample from a set ***without replacement****, consider use the function* `numpy.random.choice` *by controling the parameter* `replace`*.*

1. $P(D_1) = 0.25$

[54]:

```
import random
import numpy as np
import numpy.random as npr

#part 1 Probability of drawing a single diamond
#help(np.random.choice)



#Create a deck of cards
Deck = [0]*52
```

```python

#Diamonds will be 1 all other suits will be 0
for i in range(13):
    Deck[i] = 1;


#Run a simulation
num_sims = 100_000
event_counter = 0

for i in range(num_sims):
    if(np.random.choice(Deck)):
        event_counter += 1

print("Probability of drawing a diamond on the first draw is:", event_counter/
 ↪num_sims)
```

```
Probability of drawing a diamond on the first draw is: 0.24943
```

2. $P(D_1 \cap \overline{D}_2) = 0.191$

[55]:

```python
#Part 2 simulation
#reset the event counter
event_counter = 0


#now factor in card replacement
for i in range(num_sims):
    #Draw twice without replacement
    cards_drawn = np.random.choice(Deck,2,replace=False)

    #check to see if condition is met
    if(cards_drawn[0] == 1 and cards_drawn[1] == 0):
        event_counter += 1

print("The probability of drawing a diamond on the first\
 draw and not drawing a diamond on the second draw is:", event_counter/num_sims)
```

```
The probability of drawing a diamond on the first draw and not drawing a diamond
on the second draw is: 0.1914
```

3. $P(D_1 \cap \overline{D}_2 \cap \overline{D_3}) = 0.145$

[57]:

```python
#Part 3 simulation
#reset the event counter
event_counter = 0


for i in range(num_sims):
    #Draw three times without replacement
    cards_drawn = np.random.choice(Deck,3,replace=False)

    #check to see if condition is met
    if(cards_drawn[0] == 1 and cards_drawn[1] == 0 and cards_drawn[2] == 0):
        event_counter += 1

print("The probability of drawing a diamond on the first\
 draw and not drawing a diamond on the second and third draw is:",␣
 ↪event_counter/num_sims)
```

```
The probability of drawing a diamond on the first draw and not drawing a diamond
on the second draw is: 0.14584
```

4. $P(D_3|D_1 \cap \overline{D}_2) = 0.24$

[66]:
```
#Part 4 simulation

#reset the event counter
event_counter = 0

#A new deck is needed two cards were drawn so there are only 50 cards
new_deck = [0]*50



#A diamond was drawn so there are only 12 diamonds left
for i in range(12):
    new_deck[i] = 1;

#Begin the simulation
for i in range(num_sims):
    if(np.random.choice(new_deck)):
        event_counter += 1

print("The probability of drawing a diamond given the\
first draw was a diamond and the second was not is:", event_counter/num_sims)
```

```
The probability of drawing a diamond given thefirst draw was a diamond and the
second was not is: 0.23908
```

1.3 Problem 2

Suppose that you have 6 coins: 1 fair coins and 5 unfair coins. When flipped, the coins flip heads with probability 0.5, 0.6, 0.8, 0.2, 0.1 and 0.3, respectively. Consider the experiment where you choose a coin at random and flip it twice. Let E be the event that 2 heads were observed. Compute $P(\text{fair coin}|E)$.

Note: to encode specific probabilily of flipping heads, you may consider using the function `numpy.random.choice` by controlling the parameter `p`.

$P(\text{fair coin}|E) = 0.18$

[87]:

```
num_sims = 500_000
#Create the sample space of coins
coins = ["Fair", "Coin1", "Coin2", "Coin3", "Coin4", "Coin5"]
sides = ["H", "T"]

#reset the event_counter
event_counter = 0
fair_and_event = 0

for i in range(num_sims):
    #pick a coin
    coin_chosen = np.random.choice(coins)

    #Flip the coin twice based on what coin is chosen
    if coin_chosen == "Fair":
        #(1,2,p=[0.5,0.5]) 1 means 0 or 1, 2 means two flips
        #p = 0.5 is the probability of the fair coin
        flips = np.random.choice(sides,2,p=[0.5, 0.5])
    elif coin_chosen == "Coin1":
        flips = np.random.choice(sides,2,p=[0.6, 0.4])
    elif coin_chosen == "Coin2":
        flips = np.random.choice(sides,2,p=[0.8, 0.2])
    elif coin_chosen == "Coin3":
        flips = np.random.choice(sides,2,p=[0.2, 0.8])
    elif coin_chosen == "Coin4":
        flips = np.random.choice(sides,2,p=[0.1, 0.9])
    else:
        flips = np.random.choice(sides,2,p=[0.3, 0.7])

    #Check if the event occured
    if(flips[0] == 'H' and flips[1] == 'H'):
        event_counter += 1

    if(flips[0] == 'H' and flips[1] == 'H' and coin_chosen == "Fair"):
        fair_and_event += 1

#apply conditional probability equation
fair_given_event = fair_and_event/event_counter
print("The probability of a fair coin given two heads were flipped is:",␣
 ↪fair_given_event)
```

```
The probability of a fair coin given two heads were flipped is:
0.18104147608609353
```

2 Submit Your Solutions

Confirm that you've successfully completed the assignment.

Along with the Notebook, include a PDF of the notebook with your solutions.