CS 237 Fall 2019 Homework Three

Due date: PDF file due Thursday September 26 @ 11:59PM in GradeScope with 6-hour grace period

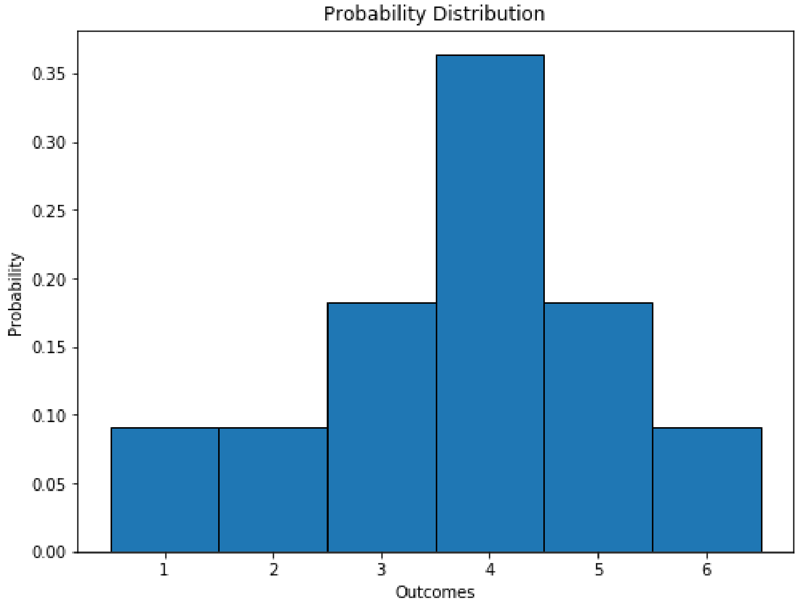
Late deadline: If submitted up to 24 hours late, you will receive a 10% penalty (with same 6 hours grace period)

General Instructions

Please complete this notebook by filling in solutions where indicated. Be sure to "Run All" from the Cell menu before submitting.

There are two sections to the homework: problems 1 - 8 are analytical problems about last week's material, and the remaining problems are coding problems which will be discuss in lab next week.

In [1]:
```
# Here are some imports which will be used in code that we write for CS 237

# Imports used for the code in CS 237

import numpy as np                  # arrays and functions which operate on array
import matplotlib.pyplot as plt     # normal plotting
import seaborn as sns               # Fancy plotting
import pandas as pd                 # Data input and manipulation

from math import log,pi                  # or just import math and use anything there, conflicts wit
from numpy.random import random, randint, uniform, choice, shuffle, seed
from collections import Counter

%matplotlib inline

# Use this next function to round to 4 digits.
# If the number is less than 0.00005 this will
# just produce 0.0000 so just print it normally.

def round4(x):
    return round(x+0.00000000001,4)

def round4_list(L):
    return [ round4(x) for x in L ]


# Useful code from HW 01

# This draws a useful bar chart for the distribution of the
# list of integers in outcomes

def show_distribution(outcomes, title='Probability Distribution', my_xticks = [], f_size = (8,6)):
    plt.figure(figsize=f_size)
    num_trials = len(outcomes)
    X = range( int(min(outcomes)), int(max(outcomes))+1 )
    freqs = Counter(outcomes)
    Y = [freqs[i]/num_trials for i in X]
    plt.bar(X,Y,width=1.0,edgecolor='black')
    if my_xticks != []:
        plt.xticks(X, my_xticks)
    elif (X[-1] - X[0] < 30):
        ticks = range(X[0],X[-1]+1)
        plt.xticks(ticks, ticks)
    plt.xlabel("Outcomes")
    plt.ylabel("Probability")
    plt.title(title)
    plt.show()

# Example of use

show_distribution([1,4,3,5,4,6,2,4,3,5,4])

# This function takes a list of outcomes and a list of probabilities and
# draws a chart of the probability distribution.
# It allows labels for x axis with numbers or strings; for the latter, you
# still need to give the numeric labels, but can overwrite them with your string labels.

def draw_distribution(Rx, fx, title='Probability Distribution', my_xticks = [], f_size = (8,6)):
    plt.figure(figsize=f_size)
    plt.bar(Rx,fx,width=1.0,edgecolor='black')
    plt.ylabel("Probability")
    plt.xlabel("Outcomes")
    if my_xticks != []:
        plt.xticks(Rx, my_xticks)
    elif (Rx[-1] - Rx[0] < 30):
        ticks = range(Rx[0],Rx[-1]+1)
        plt.xticks(ticks, ticks)
    plt.title(title)
    plt.show()
```



Analytical Problems Introduction

Some of these problems concern the familiar experiment of throwing two dice and counting the number of dots that show on both. It is always a good idea to draw the search space if possible, so here is a diagram such as I put on the board; this also shows you how to use HTML inside Markdown cells!

Outcomes from Two Dice

	1	**2**	**3**	**4**	**5**	**6**
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Problem 1 (Conditional Probability)

Suppose you roll two fair dice and count the number of dots showing. What is the probability of a sum of 5 if

(A) the second roll is not 3?

(B) they land on different numbers?

Solution:

Let A = "The sum is 5", B = "The second roll is not 3", and C = "They land on different numbers."

A = {(1,4),(2,3),(3,2),(4,1)}, | A | = 4, P(A) = 4/36 = 1/9

B = {1,2,..., 6} x {1,2,..., 6} - {(1,3),(2,3),(3,3),(4,3),(5,3),(6,3)}, | B | = 36 - 6 = 30, P(B) = 30/36 = 5/6

C = {1,2,..., 6} x {1,2,..., 6} - {(1,1),(2,2),(3,3),(4,4),(5,5),(6,6)}, | B | = 36 - 6 = 30, P(C) = 30/36 = 5/6

(A)

$P(A \cap B) = P(\{(1,4),(3,2),(4,1)\}) = \frac{3}{36} = \frac{1}{12}$

but then:

$$\text{Answer} = P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{1}{12}}{\frac{5}{6}} = \boxed{0.1}$$

Or we may calculate using a diagram. B is shown by the shaded cells below and A is shown by boldface numbers:

Outcomes from Two Dice

	1	**2**	**3**	**4**	**5**	**6**
1	2	3	4	**5**	6	7
2	3	4	**5**	6	7	8
3	4	**5**	6	7	8	9
4	**5**	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

```
Thus
   P(A|B) = P(AB) / P(B)
          = (3/36) / (30/36)
          = 3/30
          = 0.1
```

(B) $A \cap C = A$, hence:

$$\text{Answer} = P(A \mid C) = \frac{P(A \cap C)}{P(C)} = \frac{P(A)}{P(C)} = \frac{\frac{1}{9}}{\frac{5}{6}} = \boxed{\frac{4}{30} = 0.1333}.$$

Again, using a diagram, we show C by the shaded cells below and A is shown by boldface numbers:

Outcomes from Two Dice

	1	**2**	**3**	**4**	**5**	**6**
1	2	3	4	**5**	6	7
2	3	4	**5**	6	7	8
3	4	**5**	6	7	8	9
4	**5**	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

```
Thus
   P(A|C) = P(AC) / P(C)
          = (4/36) / (30/36)
          = 4/30
          = 0.1333
```

Problem 2 (Conditional Probability)

Suppose we have two events A and B. For each of these three cases, give the numeric value or the simplest formula for $P(A \mid B)$:

(a) $A \cap B = \varnothing$ (A and B are disjoint)

(b) $A \subset B$

(c) $B \subset A$

Solution:

(a)

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(\varnothing)}{\Pr(B)} = \boxed{0}$$

(b)

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \boxed{\frac{\Pr(A)}{\Pr(B)}}$$

(b)

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(B)}{\Pr(B)} = \boxed{1}$$

Problem 3

(Based on a true story, more or less....)

Wayne falls asleep watching an old Clint Eastwood movie at 4am after preparing his lecture for CS 237...

He is lying on the sidewalk after robbing a bank, in pain and mulling over how to quantify the uncertainty of his survival, when Dirty Harry walks over. Dirty Harry pulls out his 44 Magnum and puts two bullets opposite each other in the six slots in the cylinder (e.g., if you number them 1 .. 6 clockwise, he puts them in 1 and 4), spins the cylinder randomly, and, saying "The question is, are you feeling lucky, probabalistically speaking, computer science punk?" points it at Wayne's head and pulls the trigger.... "CLICK!" goes the gun (no bullet) and Dirty Harry smiles... "How about that .... Let's see if this gun is memory-less!" Without spinning the cylinder again, he points the gun at Wayne's head and pulls the trigger again.

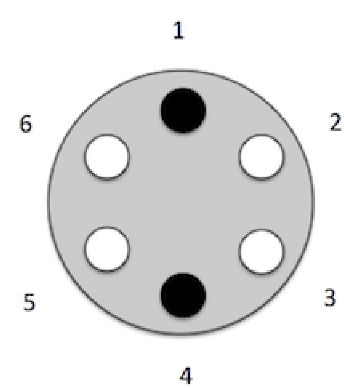
(a) What is the probability that (at least in my dream) you will have to have another instructor finish out CS 237 (because Wayne has a really BAD headache and can't continue)?

(b) Now, suppose that when Dirty Harry put the bullets in the gun, he put them right next to each other (e.g., in slots 1 and 2). He spins it as usual. What is the probability in this case that you will have another instructor finish teaching CS 237?

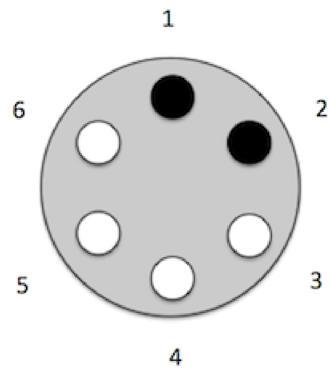
(c) Suppose Dirty Harry puts the bullets in two random positions in the cylinder and we don't have any idea where they are. He spins it as usual. Now what is the probability that I will not be able to finish teaching CS 237?

Hint: This has nothing to do with the memory-less property (Dirty Harry never took CS 237) .... and we could solve it by just considering what the probability is for each of the patterns (no bullet, no bullet) and (no bullet, bullet) as we go around the circle of slots in the cylinder.

Here is what the cylinder might look like in case (a) before Dirty Harry randomizes the positions by spinning it:



Here is what it might look like in case (b) before being randomized:



In case (c), of course, the bullets can be in any two slots.

You may solve this using conditional probabilities, or by any other method you wish.

Solution: To do this without expliciting invoking the conditional probability rule, we just consider the positions and the appropriate patterns.

(a) If the first slot had no bullet, then it must have been in position 2, 3, 5, or 6. Half of these, 3 and 6, are followed by a bullet, so the probability must be 1/2 = \boxed{0.5}.

(b) In this case, the first slot must have been in 3, 4, 5, or 6, and only one of these, 6, is followed by a bullet, so the probability must be 1/4 = 0.25.

(c) If the bullet positions are random, then after placing the first bullet, there are 5 positions left, 2 of which are next to the first bullet. Therefore the probability of case (b) is 2/5 and of case (a) is 3/5. Or, if you don't want to factor out the rotations by placing he first bullet in slot 1, you could think in terms of combinations: There are C(6,2) = 15 positions for two random bullets, 6 with the bullets next to each other, and 9 with at least one space between bullets, so again P((a))= 9/15 = 3/5 and P((b)) = 6/15 = 2/5

Thus the overall probability is 2/5 *1/4 + 3/5* 1/2 = = \boxed{0.4}.

You could, of course, do a decision tree....

Doing it using Conditional Probabilites

We could of course do this as a conditional probability problem, where A = "the second shot has a bullet in the slot" and B = "the first shot has no bullet in the slot" and we want to know P(A|B). Since Dirty Harry may be contemplating a third shot, let make no mistakes and be completely explicit about the sample space and the events involved. Using the numbering of the slots as in the diagram provided, we have

(a) Bullets in slots 1 and 4, rest empty:

```
 S = { (1,2), (2,3), (3,4), (4,5), (5,6), (6,1) }         // first number in pair is slot
 for first shot, second is slot for second shot
 A = { (3,4), (6,1) }                   P(A) = 2/6 = 1/3   // 2nd number in pair has bull
et
 B = { (2,3), (3,4), (5,6), (6,1) }     P(B) = 4/6 = 2/3   // 1st number in pair is empty
AB = A                                 P(AB) = 1/3

P(A|B) = P(AB)/P(B) = (1/3)/(2/3) = 1/2 = 0.5
```

(b) Bullets in slots 1 and 2, rest empty:

```
 A = { (1,2), (6,1) }                   P(A) = 2/6 = 1/3
 B = { (3,4), (4,5), (5,6), (6,1) }     P(B) = 4/6 = 2/3
AB = { (6,1) }                         P(AB) = 1/6

P(A|B) = P(AB)/P(B) = (1/6)/(2/3) = 1/4 = 0.25
```

Problem 4 (Independence)

Suppose you roll two dice and count the number of dots showing. Let A = "the sum of the dots showing on the two rolls is a prime number."

(a) Let B = "The first toss showed an even number of dots." Are A and B independent? Show your work.

(b) Let C = "the second toss was greater than the first." Are A and C independent? Show your work.

(c) Suppose we pick a value k for $1 \leq k \leq 6$ and let event D_k = "the first toss was greater than or equal to k." For which values of k are A and D_k independent? Show your work.

Solution:

Outcomes from Two Dice

	1	**2**	**3**	**4**	**5**	**6**
1	**2**	**3**	4	**5**	6	**7**
2	**3**	4	**5**	6	**7**	8
3	4	**5**	6	**7**	8	9
4	**5**	6	**7**	8	9	10
5	6	**7**	8	9	10	**11**
6	**7**	8	9	10	**11**	12

(a) The outcomes which are prime are 2, 3, 5, 7, and 11. There are 15 occurrences of these numbers, thus

$$P(A) = \frac{15}{36} = \frac{5}{12}$$

and clearly

$$P(B) = \frac{18}{36} = \frac{1}{2}$$

and also:

$$P(A \cap B) = \frac{7}{36}$$

Clearly $P(A \cap B) \neq P(A) \cdot P(B)$,

$$\boxed{\text{so the two events are NOT independent.}}$$

(b) We have:

$$P(C) = \frac{21}{36} = \frac{7}{13}$$

and:

$$P(A \cap C) = \frac{7}{21} = 0.3334 \neq P(A) \cdot P(C) = \frac{7}{18} \cdot \frac{7}{13} = 0.2094$$

$$\boxed{\text{The two events are NOT independent.}}$$

(c) By inspection of the grid, we can see that it only works for $k = 1$ (the trivial case).

Problem 5 (Independence)

For three events A, B, and C, we know that

- A and C are independent,
- B and C are independent,
- A and B are disjoint,

Furthermore, suppose that $P(A \cup C) = \frac{2}{3}$, $P(B \cup C) = \frac{3}{4}$, $P(A \cup B \cup C) = \frac{11}{12}$.

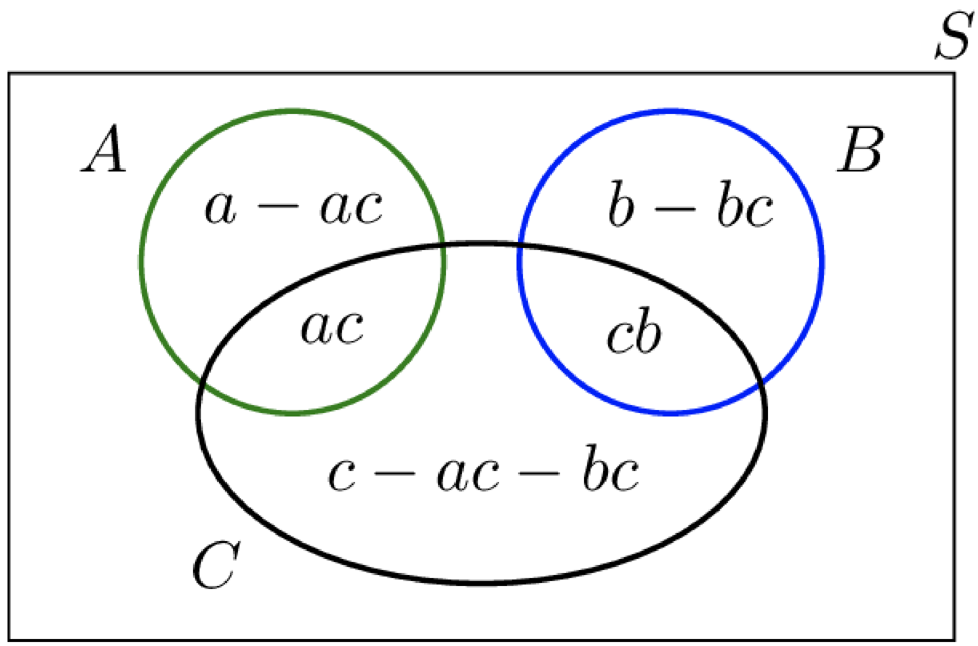
Find $P(A)$, $P(B)$, and $P(C)$.

Solution:

We have:

$$\Pr(A \cap C) = \Pr(A) \cdot \Pr(C) \quad \text{and} \quad \Pr(B \cap C) = \Pr(B) \cdot \Pr(C)$$

consider the following figure:



$$P(A) = a, P(B) = b, P(C) = c$$

we then get:

$$a + c - ac = \frac{2}{3}$$

and:

$$b + c - bc = \frac{3}{4}$$

and:

$$a + b + c - ac - bc = \frac{11}{12}$$

subtracting the third equation from the sum of the first two we get:

$$\Pr(C) = c = \boxed{\frac{1}{2}}$$

which gives:

$$\Pr(A) = a = \boxed{\frac{1}{3}}$$

and:

$$\Pr(B) = b = \boxed{\frac{1}{2}}$$

See Problem 3 on this page: https://www.probabilitycourse.com/chapter1/1_4_5_solved3.php (https://www.probabilitycourse.com/chapter1/1_4_5_solved3.php)

Problem 6

You randomly shuffle a 52-card deck of cards. We will consider what happens as we draw 3 cards from the deck to form a sequence of cards (for a - c) or a set of cards (d).

Answer each of the following questions, showing all relevant calculations. You should analyze these using tree diagrams, but no need to show the diagrams in your answer.

(a) Suppose you draw three cards from the deck with replacement; what is the probability that the first card is red, the second is a spade, and the third is a facecard (Jack, Queen, or King)? [Hint: these are ordered!]

(b) Suppose you draw three cards from the deck with replacement; what is the probability that the first and third cards have the same color, but the second is a different color?

(c) Repeat (b), but without replacement (cards are not put back). [Hint: a tree diagram might be useful.]

Solution:

(a) Since we replace the cards, the three draws are independent, and we can just multiply the probabilities:

$$P(red, spade, facecard) = \frac{26}{52} \cdot \frac{13}{52} \cdot \frac{12}{52} = \boxed{0.0288}$$

(b) Again, we can just multiply the probabilities, but there are various cases to consider, since there are 8 possible equiprobable outcomes {BBB, BBR, ..., RRR} and the event is {BRB,RBR}. Clearly the probability is = $\boxed{0.25}$.

(c) Now the probabilities are different, and you can do it with a tree diagram or by considering all the possible combinations of red and black that come out:

```
all same color:     prob = (26/52)*(25/51)*(24/50) = 0.1109
2 of one, 1 of the other:     prob = (26/52)*(25/52)*(26/52) = 0.1202

The sequences we are interested in are BRB and RBR, so the
answer is 2*0.1202 = = 0.2404.
```

Problem 7

You have two bags of colored balls. Bag A contains 3 red and 2 black balls; and Bag B contains 1 red ball and 5 black balls. Both bags are shaken so that they are in random order but the bags are kept separate.

Answer each of the following questions, showing all relevant calculations. If necessary, analyze these using a tree diagram, but no need to show the diagram in your answer.

(a) Suppose you draw a ball from Bag A and find it is red; then you draw a ball from Bag B and find that it is black; you throw all the balls from both bags into a third bag, shake it, and draw a ball. What is the probability that it is black?

(b) Suppose you flip a fair coin to choose one of the bags, and then draw one ball. What is the probability that it is red?

(c) Suppose you tell a friend to flip a fair coin to choose one of the bags, and then draw a ball without telling you which bag it came from; you see that it is red. What is the probability that it came from Bag A?

Solution:

(a) $\frac{2+4}{4+5} = \boxed{0.6667}$

(b) $\frac{1}{2} \cdot \frac{3}{5} + \frac{1}{2} \cdot \frac{1}{6} = \boxed{0.3833}$

(c)

P(red and from A) = $\frac{1}{2} \cdot \frac{3}{5} = \frac{3}{10}$;

P(red and from B) = $\frac{1}{2} \cdot \frac{1}{6} = \frac{1}{12}$

P(red) = $\frac{3}{10} + \frac{1}{12} = 0.3833$

P(red | from A) = $\frac{0.3}{0.3833} = \boxed{0.7826}$.

Problem 8 (Circular Permutations)

In how many ways can 7 people { A, B, C, D, E, F, G } be seated at a round table if

(a) A and B must not sit next to each other;

(b) C, D, and E must sit together (i.e., no other person can sit between any of these three)?

(c) A and B must sit together, but neither can be seated next to C or D.

Consider each of these separately. For (c) you may NOT simply list all possibilities, but must use the basic principles we have developed (you may check your work with a list if you wish).

Hint: Conceptually, think of the groups of two or three people as one "multi-person" entity in the overall circular arrangement. However, a "multiperson" is an unordered entity, and you will have to think about how many ways a "multiperson" could be ordered. It may help to draw a diagram, fixing a particular person at the top of the circle (thereby eliminating the duplicates due to rotations).

Solution:

(a) We must eliminate all those where the two are seated together (effectively, being one combined "multi-person" among 6), i.e., $\frac{6!}{6}$, and then account for the duplication (since the two can be in any order, i.e., are a permutation of 2 people). (Another example of the unordering principle.) So we have $\frac{2!*6!}{6} = 2! * 5! = 240$ ways that these two can be seated together. So 720 - 240 = 480 ways.

(b) The "multi-person" has three people, and so we have $\frac{5!}{5} = 4! = 24$ arrangements including the multi-person, and 3! permutations of the three in the group. Thus 24*6 = 144 ways.

(c) Choose two people to sit next to (AB) or (BA), C(3, 2) = 3

These two people can be arranged in 2! = 2 ways

The other three can sit in 3! = 6 ways

A, B can sit in 2! = 2 ways

3 * 2 * 6 * 2 = 72 ways

Another way of thinking about it:

The multi-person has 2 people, so we have 5 people (including one multi-person). Therefore there are 6!6=5!=120 ways, and 2! permutations of the multi-person. Therefore 120 * 2 = 240 without the constraint on C and D.

Now we must eliminate all those where C and D sit next to the multi-person.

First consider where both C and D sit next to the multi-person. There are 2 ways for C and D to sit next to the multi-person, 2! ways for A and B to sit next to each other, and 3! ways for the remaining people to sit, so 2*2*3! = 24 ways

Then, consider where only C or only D sit next to the multi-person. Choose 1 from C or D: C(2, 1) = 2 ways. There are 2! ways for A and B to sit next to each other. There are 2 ways for the the chosen one and the multi-person to sit next to each other. The other one in C, D who doesn't sit next to the multi-person can choose C(3, 1) = 3 positions to sit. There are 3! for the rest of the 3 people to sit. So 2*2!*2*3*3! = 144

So 240-24-144 = 72

Lab Introduction: Poker Probability

In this lab we will explore Poker Probability, which is calculating the probability of various hands in the game of poker. This is, again, exploring how to confirm our theoretical understanding with experiments. If our experiments, as we increase the number of trials, converge to our theoretical calculation, then we have almost certainly analyzed it correctly.

There are many versions of poker (see here (http://www.wikihow.com/Play-Poker)) but the game we will study is called "five-card draw." It is described (https://www.pokernews.com/strategy/5-card-draw-rules-how-to-play-five-card-draw-poker-23741.htm) as follows:

> Once everyone has paid the ante, each player receives five cards face down. A round of betting then occurs. If more than one player remains after that first round of betting, there follows a first round of drawing. Each active player specifies how many cards he or she wishes to discard and replace with new cards from the deck. If you are happy with your holding and do not want to draw any cards, you “stand pat.” Once the drawing round is completed, there is another round of betting. After that if there is more than one player remaining, a showdown occurs in which the player with the best five-card poker hand wins.

Here is an excellent short YT video on the basics of Poker: YT (https://www.youtube.com/watch?v=xfqMC3G37VE)

The only part we will care about is the final calculation of which hand wins: basically, the least probable hand wins. When you learn poker, then, one of the first things you have to learn is the ordering of the hands from most to least likely. *Poker probability* refers to calculating the exact probabilities of hands. The Wikipedia article (https://en.wikipedia.org/wiki/Poker_probability) contains the exact results and the formulae used to calculate them.

In this lab we will develop a framework for dealing 5-card hands and empirically estimating the probabilites of various hands. In fact, we will be able to do nearly all the hands commonly encountered. Our only constraint is that for the rarest hand, a Royal Flush, since there are only 4 such hands, the probability is so small it would take too long to get a reasonable estimate, and so we shall ignore this case.

This lab should help your understanding of the counting techniques covered in lecture.

Preface: Card Games and Probability

First we will first explore how to encode a standard deck of 52 playing cards, how to perform various tests on cards, and how to deal hands. To remind you, here is the illustration showing all the cards: cards (http://www.cs.bu.edu/fac/snyder/cs237/images/PlayingCards.png).

In [2]:
```
# We will represent cards as a string, e.g., 'AC' will be Ace of Clubs

# Denominations: 2, ..., 10,  'J' = Jack, 'Q' = Queen, 'K' = King, 'A' = Ace
Denominations = ['2','3','4','5','6','7','8','9','10','J','Q','K','A']

# Suits 'S' = Spades, 'H' = Hearts, 'D' = Diamonds, 'C' = Clubs
Suits = ['C', 'H', 'S', 'D']

# Note that colors are determined by the suits (hearts and diamonds are red, others black,
# so, AC is Black

# List comprehensions are a great way to avoid explicit for loops when creating lists

Deck =  [(d+s) for d in Denominations for s in Suits]   # Note the double for loop

print( Deck )
```

```
['2C', '2H', '2S', '2D', '3C', '3H', '3S', '3D', '4C', '4H', '4S', '4D', '5C', '5H', '5S', '5
D', '6C', '6H', '6S', '6D', '7C', '7H', '7S', '7D', '8C', '8H', '8S', '8D', '9C', '9H', '9S',
'9D', '10C', '10H', '10S', '10D', 'JC', 'JH', 'JS', 'JD', 'QC', 'QH', 'QS', 'QD', 'KC', 'KH',
'KS', 'KD', 'AC', 'AH', 'AS', 'AD']
```

In [3]:
```
# Now we can "deal" cards by choosing randomly from the deck

seed(0)                             # seed makes sure that all your computations start with the same
                                    # this not really important, and only necessary for debugging an

def dealCard():
    return choice(Deck)             # choice randomly chooses an element of a list

print( dealCard() )
```

```
KC
```

In [4]:
```
choice(Deck,10)
```

Out[4]:
```
array(['KD', '2C', '2D', '2D', 'JD', '4H', '6D', '7H', 'AS', 'JC'],
      dtype='<U3')
```

In [5]:
```
# When dealing a hand in cards, the selection of cards is without replacement, that is, cards are
# the deck one by one and not put back. This can be simulated in the choice function by setting th
# parameter to False.

seed(0)

def dealHand(withReplacement = False,size = 5):
    return choice(Deck,size,withReplacement)                   # chooses a list of size elements

print( dealHand() )
```

```
['9C' 'JH' '4D' '10S' '2S']
```

In [6]:
```
# extract the denomination and the suit from a card

def denom(c):
    return c[0:-1]

def suit(c):
    return c[-1]

# The function rank(c) will simply return the position of the card c PLUS 2 in the list 2, 3, ....
# way in our code below. Although in the diagram given lecture, Ace is below 2, the Ace is actuall
# above the King, for example in determining a straight, under "Ace high rules."

#  rank(2) = 2, ...., rank(10) = 10, rank(Jack) = 11, rank(Queen) = 12, rank(King) = 13, rank(Ace)

def rank(c):
    return Denominations.index(denom(c))+2

# Now we want to identify various kinds of cards

def isHeart(c):
    return ( suit(c) == 'H')

def isDiamond(c):
    return ( suit(c) == 'D')

def isClub(c):
    return ( suit(c) == 'C')

def isSpade(c):
    return ( suit(c) == 'S')

def isRed(c):
    return ( isHeart(c) or isDiamond(c) )

def isBlack(c):
    return (not isRed(c))

def isFaceCard(c):
    return rank(c) >= 11 and rank(c) <= 13
```

Example Problem: What is probability that a 5-card hand has exactly 3 red cards?

Remember that in finite probability, for any event A,

$$P(A) = \frac{|A|}{|S|}.$$

Therefore, what we need to do in problems involving the probability of various kinds of hands in card games is to count the number of possible such hands, and divide by the total number of all possible hands. We developed analytical tools in lecture to do this, but here we are going to estimate it with repeated trials of dealing hands and testing for a given kind of hand.

In general for all but the last problem, we will use 100,000 trials to get a reasonable estimate of the probability. Since 1/100000 = 0.00001 this means our resolution for experimental probabilities is 5 decimal places.

In [7]:

```
# Return True iff the number of red cards in the hand h is 3
def threeRed(h):
    redCards = [c for c in h if isRed(c)]
    return (len(redCards) == 3)

seed(0)

# Run the experiment for 10,000 trials
# Print out probability that a 5-card hand has exactly 3 red cards

num_trials = 10**5
trials = [dealHand() for k in range(num_trials)]       # create list of 10000 hands randomly dealt

if(num_trials <= 10):       # Just for this example, you don't need to do this unless you are debu
    print(trials)

hands =  [threeRed(h) for h in trials]                 # convert this to list of true and false va

if(num_trials <= 10):
    print(hands)

prob = hands.count(True) / num_trials                  # count the number of True values and divic

# probability for 100,000 trials should be close to analytical value of 0.3251

print('\nProbability of exactly 3 red cards in a 5-card hand is ' + str(prob))
```

```
Probability of exactly 3 red cards in a 5-card hand is 0.32225
```

In [8]:

```
# here is another way to do it, but it is a bit cryptic!

# trials was calculated above

seed(0)

prob2 = sum( [1 for h in trials if threeRed(h)] ) / num_trials

print('\nProbability of exactly 3 red cards in a 5-card hand is ' + str(prob2))
```

```
Probability of exactly 3 red cards in a 5-card hand is 0.32225
```

In [9]:

```
# If you like cryptic, then you can put it all, including the calculation of trials in one line!
# Here are two examples. Note: these run the experiment again,  so they won't match the previous 2

seed(0)

prob3 = sum( [1 for h in [dealHand() for k in range(num_trials)] if threeRed(h)] ) / num_trials

print('\nProbability of exactly 3 red cards in a 5-card hand is ' + str(prob3))

seed(0)

prob4 = sum( [1 for k in range(num_trials) if threeRed(dealHand())] ) / num_trials

print('\nProbability of exactly 3 red cards in a 5-card hand is ' + str(prob4))
```

```
Probability of exactly 3 red cards in a 5-card hand is 0.32225

Probability of exactly 3 red cards in a 5-card hand is 0.32225
```

In [10]:
```
## Finally, constructing the lists each time is ok if they do not get too large, or
## you are running too many trials, in which case, you should code this using counters
## and with NO list comprehensions.

seed(0)

num_trials = 10**5
num_3D = 0

for k in range(num_trials):
    if (threeRed(dealHand())):
            num_3D += 1

print('\nProbability of exactly 3 red cards in a 5-card hand is',round4(num_3D / num_trials))
```

```
Probability of exactly 3 red cards in a 5-card hand is 0.3223
```

Example: What is probability that a 5-card hand has at least 3 Diamonds?

In [11]:
```
# Print out probability that a 5-card hand has 3, 4, or 5 diamonds.

seed(0)

def atLeast3Diamonds(h):
    return (len([c for c in h if isDiamond(c)]) >= 3)

num_trials = 10**5

trials = [dealHand() for k in range(num_trials)]
hands =  [atLeast3Diamonds(h) for h in trials]                 # convert this to list of true and
prob = hands.count(True) / num_trials

# Should be close to analytical value of 0.0928

print('Probability of at least 3 diamonds in a 5-card hand is ' + str( prob ))
```

```
Probability of at least 3 diamonds in a 5-card hand is 0.09278
```

Problem 9: What is probability of a flush in Poker?

In Poker, a *flush* is 5 cards of the same suit, but excludes straight flushes and royal flushes; these, however, are so rare (there are only 40 of them in all), that they are around or below our resolution (0.00001), so we just will determine if all suits are the same.

In [12]:
```
```

```
Probability of a flush in 5-card poker is 0.00176
```

Problem 10: What is probability of a straight in Poker?

In poker, a *straight* a hand in which the ranks form a contiguous sequence, e.g., 2,3,4,5,6. The suits do not matter. Also, for simplicity, we will use the "deuce-to-seven low rules (https://en.wikipedia.org/wiki/List_of_poker_hands#Straight)" whereby the Ace must count as a high card (above the King). This simplifies the calculation a little bit, since we can just sort the cards by rank and check if they are contiguous.

In [13]:
```
```

```
Probability of a straight in poker 0.002
```

Problem 11: Rank Signature of a poker hand

Let us define the *rank signature* of a hand as an ordered histogram of the ranks occurring in the hand; that is, we count the frequency of the ranks occurring in the hand, and *order* this sequence. Here are some examples:

- Five cards all of different ranks (e.g., Ace, 4, 2, King, 8): `[1,1,1,1,1]`
- One pair, 2 cards of the same rank, and 3 more all of different ranks (e.g., 2,2,6,3,Ace): `[1,1,1,2]`
- Two pair, 2 pairs (of different ranks) and one card of a different rank (e.g., 2,2,Ace,3,Ace): `[1,2,2]`
- Full house, 2 cards of the same rank, and 3 cards of the same rank (e.g., 8,Jack,8,8,Jack): `[2,3]`

Many poker hand can be defined solely in terms of the ranks involved. The importance of this concept is that once we write a function to estimate the probability of a given signature, we can then immediately calculate the probability of many different poker hands.

For this problem you must write a function which calculate the probability that a 5-card hand has a given signature and verify it by calculating the probability of no two ranks being the same (first choose 5 different ranks, then consider all possible enumerations of suits):

$$\frac{\binom{13}{5} * 4^5}{\binom{52}{5}} = 0.5071$$

Note: This is NOT the same as "no pair/high card" since in Poker, this hand means you don't have a flush or a straight (both of which have 5 cards of different rank). To find out the correct probability, you would need to subtract the possibility of a flush or a straight (or both, a straight-flush), which we will do in the next problem.

In [23]:

```
['4D' '5C' 'QH' 'QC' '2C']
[1, 1, 1, 1, 1]
Counter({12: 2, 4: 1, 5: 1, 2: 1})
dict_keys([4, 5, 12, 2])
[1, 1, 2, 1]
[1, 1, 1, 2]
```

Out[23]: False

Problem 12: Using rank signature to calculate six different poker hands

Problem 12 (A): What is probability of No Pair/High Card in Poker?

For this, you must use the rank signature of 5 different ranks (as in the previous problem). For this first one, you must calculate the cumulative probability (i.e., exclude straights and flushes).

In [15]:

```
Probability of No pair/high card in poker is 0.50317
```

Problem 12 (B): What is probability of One Pair in Poker?

In [16]:

```
Probability of one pair in poker is 0.42096
```

Problem 12 (C): What is probability of Two Pairs in Poker?

In [17]:

```
Probability of two pair in poker is 0.04783
```

Problem 12 (D): What is probability of Three of a Kind in Poker?

In [18]:

```
Probability of three of a kind in poker is 0.02073
```

Problem 12 (E): What is probability of a Full House in Poker?

In [19]:

```
Probability of a full house in poker is 0.0015
```

Problem 12 (F): What is probability of Four of a Kind in Poker?

In [20]:

```
Probability of four of a kind in poker is 0.00022
```

Optional: What is probability of a Straight Flush in Poker?

We will ignore the difference between this and a Royal Flush, so this is essentially calculating the cumulative probability (probability of this hand or better).

You will have to do this at least 10**6 times to get an accurate result.

In [ ]: