Warmup questions

Prob. of rolling six =
$$\frac{1}{6} = 0.16$$

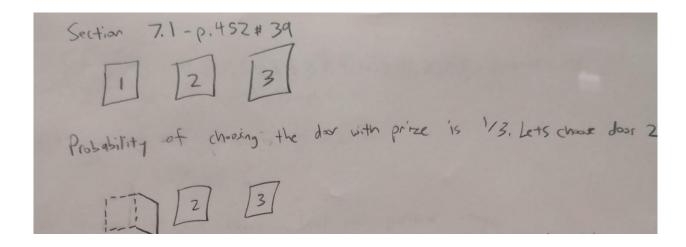
(2) Prob. of rolling six = $\frac{1}{6} = 0.16$

(3) Number of odd integers in first 100 positive integers

100/2 > 50 add

100/2 > 50 a

1. Probability of choosing door with the prize starts at 1/3. Let's choose the middle door, door (2).

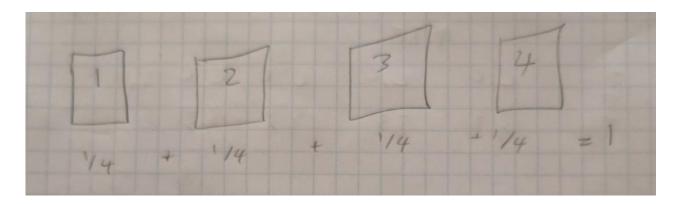


If (1) is opened (revealing no prize) our decision is to etiher stay with (2) or switch open (3). Upon first glance, this may appear to be a 50/50 choice if we don't take into consideration the other side of the equation, Monty's decisions. Monty, the host, knows which door holds the prize and it's assumed he will never reveal the door with the prize. By choosing (2), Monty's choices are either (1) or (3), but if (1) or (3) holds the prize then his only option is to open the remaining door without the prize since (2) was our original choice. From here, the scenario becomes clearer and let's assume (2) is the door with the prize. Assuming the prize was behind our original pick (2) and Monty opens (1), if we switch to (3) we lose. If we had originally chosen (3), Monty's only option is to open (1) and by switching from (3) to (2), we win the prize. If our original pick was (1), again Monty's only option is to open (3) and if we switch, we win the prize. This suggests by switching there's a chance your original choice was correct and if you decide to switch you would lose out on the prize, but you were basing your decision with a 1/3 probability. However, if your original choice was wrong (66% chance; and in this case doors (1) or (3)), Monty has only one option, to open the door without the prize. Thus, switching will give the contestant a greater chance to ultimately end up with the prize than deciding not to switch choices.

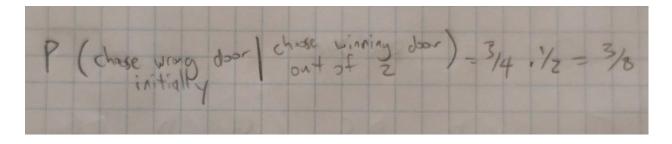
*As stated in Piazza, observing these type of questions as 100 doors helps put the probability into perspective. If you reframe the situation as a probability of 1/100 of initially choosing the winning door and Monty eliminates all but 2 doors from 100, should the contestant stay or switch, it becomes even more obvious switching is the right call.

2. (p.452 #40)

Similar to question 1 with only 3 doors, by not changing the selection after the host opens a losing door, the probability of the original choice being the winning door is still at ¼ or 25% since Monty only affected the probabilities of a potential switch (Monty always opens a door without the prize).



Prior to Monty opening the door, if the winning door was amongst the other 3 doors not chosen then we would say the probability was at 75% (3/4). With Monty opening one of the losing doors and by assuming the original door chose is wrong, it's a 50/50 chance you will choose the winning door between the last 2 remaining. The probability of choosing the wrong door initially (3/4) times the probability you choose the correct remaining door after Monty gets rid of one losing door (1/2) is equal to 3/8 or 37.5%.



(Section 7.6 – p. 476 #9)
 Similar to example 2 (p.471) and implementing Bayes' Theorem (calculating conditional probabilities)

4. (Section 7.3 – p.476 #10)

```
Infected with influency = P(I) = 4% = 0.04
Not infected with influence P(I)=1-P(I)=96%= 0.96
Patient infected with influence tests positive = P(TII) = 47% = 0.97
Parlient not infected with influence tests positive = P(TI I)=2%=0.02
Patient infected with influenza tests regutive=P(NII)=1-P(TII)=1-0.97=0.03
Patient not infected with influence tests negative = P(NII) = 1-P(TII) = 1-0.02 = 0.98
@ "probability patient intected given tests positive"
        P(IIT) = P(TIL) ·P(I)
                   P(T|I) P(I) + P(T|I) P(I)
                 - (0.97). (0.04)
                   (0.97) (0.04) + (0.02) (0.96)
                  - 0.0388

0.0388 + 0.0192 = 0.0388
                    = 0.6689655
  P(I|T)= P(TII).P(I)
                  P(T) = ) + P(T) P(I)
                  = (0.02).(0.96)
                   (0.02).(0.96)+(0.97).(0.04)
                  - 0.0192
                    0.0388+0.0192
                   = 0.0192 = 0.33103448
                     0.058
```

5.

5. (a)
$$\{1,2,2,2,3,4,5,5,6\}$$

Probability distributions $\{\rho(1) = \frac{1}{q} \quad \rho(2) = \frac{2}{q}, \frac{1}{2} \quad \rho(3) = \frac{1}{q} \quad \rho(4) = \frac{1}{q} \quad \rho(5) = \frac{1}{q}$

(b) $\rho(1) + \rho(3) + \rho(5) = \frac{1}{q} + \frac{1}{q} = \frac{2}{q} = \frac{1+\frac{q}{q}+2}{q} = \frac{4}{q}$

(c) $\rho(\text{sum} : c \neq from \neq rolls)$;

$$(1,6), (6,1), (2,5), (5,2), (3,4), (4,3)$$

$$\rho(1,6) = \frac{1}{q} \cdot \frac{1}{q} = \frac{1}{q} \quad \rho(6,1) = \frac{1}{q} \cdot \frac{1}{q} = \frac{1}{q} \quad \rho(2,5) = \frac{1}{3} \cdot \frac{2}{q} = \frac{6}{91}$$

$$\rho(5,2) = \frac{6}{91} \quad \rho(3,4) = \frac{1}{61} \quad \rho(4,3) = \frac{1}{61}$$

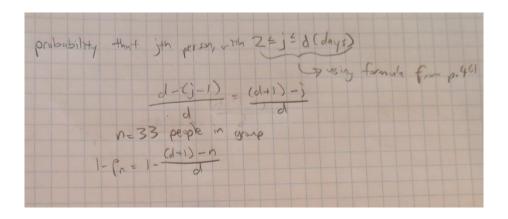
$$(\frac{1}{31}) + (\frac{1}{41}) + (\frac{6}{91}) + (\frac{6}{91}) + (\frac{1}{291}) + (\frac{1}{291}) = \frac{1}{691}$$

$$\rho(7 \mid enn) = \frac{1}{p(7 \mid enn)} \quad \rho(enn) = \rho(2) + (4) + \rho(6) = \frac{3}{q} + \frac{1}{q} + \frac{1}{q} = \frac{5}{q}$$

$$\rho(7 \mid enn) = \rho(2) + (4) + \rho(6) = \frac{3}{q} + \frac{1}{q} + \frac{1}{q} = \frac{5}{q}$$

$$\rho(7 \mid enn) = \frac{3}{81} \quad \rho(7 \mid enn$$

6.



With n=365 for standard birthday problems (n=366 for Chris' lectures due to his friend taking offense since she was born on a leap year), the python function calculated probabilities near 0.7742, so I knew the amount of days in a year on planet Probby had to be significantly more than 365 days. Using brute force, the algorithm determined the approximate number of days in the year (n) was closest to 772 while getting results close to 0.5 probability.

```
#Probability/Bayes HW week 13
#referenced http://interactivepython.org/runestone/static/everyday/2013/09/1_birthday.htmL
import random
def spock_saves_planet_probby(number_people):
    brute = [False] * 772  #used brute force to determine 772 days (with 365 days probability hovered close to 0.7742)
    for _ in range(number_people): #used n=33 for the number of inhabitants on the planet
        birthday = random.randint(0, 771)
                                           #had to establish random birthdays for the amount of days
        if brute[birthday]:
            return True
           brute[birthday] = True
    return False
def fifty_fifty_chance_shared_birthday(number_people, runs):
    same_birthday = 0
                         #establish variable for calculating probability
    for _ in range(runs):
        if spock_saves_planet_probby(number_people):
            same_birthday += 1
    return same_birthday / float(runs)
print(fifty_fifty_chance_shared_birthday(33,10000))
0.5017
```

(EXTRA CREDIT): COMIC SKETCH ON NEXT PAGE

✓ < \equiv \

Python 3

