Homework_05

October 2, 2019

1 Homework 5

1.0.1 Instructions

Your homeworks have two components: a written portion and a portion that also involves code. Written work should be completed on paper, and coding questions should be done in the notebook. You are welcome to LaTeX your answers to the written portions, but staff will not be able to assist you with LaTeX related issues. It is your responsibility to ensure that both components of the homework are submitted completely and properly to Gradescope. Refer to the bottom of the notebook for submission instructions.

```
In [6]: # Run this cell to set up your notebook

        import numpy as np
        from scipy import stats
        from datascience import *
        from prob140 import *

        # These lines do some fancy plotting magic
        import matplotlib
        %matplotlib inline
        import matplotlib.pyplot as plt
        plt.style.use('fivethirtyeight')

        # These lines make warnings look nicer
        import warnings
        warnings.simplefilter('ignore', FutureWarning)
```

1.0.2 1. Random Numbers of Trials

As in lecture and the textbook, we will use "p-coin" to mean a coin that lands heads with probability p.

a) In a randomized controlled experiment, it costs researchers t dollars to study each subject in the treatment group and c dollars to study each subject in the control group. Suppose each of n subjects is randomized into one of the two groups according to whether a p-coin lands heads (treatment) or tails (control). Find the expected total cost to study all n subjects.

b) I toss n p-coins. Those that land tails I toss again, and then I stop. Find the expected total number of heads.

c) I have one 0.25-coin, one fair coin, and three 0.75-coins. I pick one of the five coins at random and toss it till I get 10 heads. Find the numerical value of the expected number of tosses.

1.0.3 [Solution] Random Numbers of Trials

a) Let X be the number of heads, Y the number of tails, and T the total cost. Then $T = tX + cY$ so $E(T) = tnp + cn(1-p)$.

b) Let X be the total number of heads and H the number of heads in the initial set of of n tosses. Then $E(X \mid H) = H + (n-H)p$ and so $E(X) = np + (n-np)p = np(2-p)$.

c) Let T be the number of tosses and C the chance with which the coin lands heads. Then
$E(T) = 0.2E(T \mid C = 0.25) + 0.2E(T \mid C = 0.5) + 0.6E(T \mid C = 0.75$
$= 0.2 \cdot \frac{10}{0.25} \ + \ 0.2 \cdot \frac{10}{0.5} \ + \ 0.6 \cdot \frac{10}{0.75} \ = \ 20$
#newpage

1.0.4 2. Winning Color

In a population, the proportion of red elements is p_r, the proportion of blue elements is p_b, the proportion of green elements is p_g, and $p_r + p_b + p_g = 1$.

In each part below, set x equal to the quantity that you are trying to find, and develop an equation for x by conditioning on the first couple of draws. Try to write the simplest equation you can. Then solve for x.

Do not use any other method. The point of this exercise is for you to learn how to use the method outlined above. It's almost certainly going to be shorter than any other correct method.

a) Alan draws repeatedly at random with replacement from the population, betting that the color red will appear before the color blue. Find the chance that he wins his bet. Your final answer should be in terms of p_r and p_b only, and you should aim for the simplest possible form.

b) The answer to **a** can be applied in any situation where there are i.i.d. multinomial trials. For example, suppose a die is rolled repeatedly. Use your answer to Part **a** to find the chance that the face with one spot appears before any face with an even number of spots appears.

c) Now suppose Alan plays the following game with Katherine. They draw alternately at random with replacement from the population, with Alan drawing first. Alan wins if he draws red before Katherine has drawn blue. Katherine wins if she draws blue before Alan has drawn red. Find the chance that Alan wins the game. To keep your expressions simple, use the notation $q_r = 1 - p_r$ and $q_b = 1 - p_b$.

d) Find the expected duration of the game in Part **c**. That is, find the expected number of draws till there is a winner.

1.0.5 [Solution] Winning Color

a) $x = p_r + p_g x$, so $x = p_r/(p_r + p_b)$

b) $1/4$

c) $x = p_r + q_r q_b x$, so $x = p_r/(1 - q_r q_b)$

d) $x = 1 + q_r(1 + q_b x)$, so $x = (1 + q_r)/(1 - q_r q_b)$

#newpage

1.0.6 3. Panda's Problem

Every day, Panda the black-and-white cat comes to our house for food. Assume that every day:

- We put the food out at the front door or at the back door, according to whether a p-coin lands heads or tails.
- Panda arrives at the door at which it found the food the previous day; if the food is not there, Panda is disappointed and trudges to the other door to eat.

a) Set up a four-state Markov Chain and find the long run expected proportion of days when Panda is disappointed.

b) Suppose that yesterday Panda arrived at the front door and was not disappointed. What is the chance of the same thing happening today? What is your best guess for the chance of the same thing happening (Panda arriving at the front door and not being disappointed) one year from now, assuming that the process continues as described?

c) Panda's strategy is to remember where the food was the previous day, and go to that door. Here are three other strategies that Panda might use:

- Always go to the front door.
- Always go to the back door.
- Remember where the food was the previous day, and go to the other door.

Compare each of these strategies to the strategy Panda uses: for what values of p do these result in a lower expected proportion of days of disappointment?

1.0.7 [Solution] Panda's Problem

a) State = yesterday's door, today's door

For example, the state FB represents the food was at the front door yesterday and at the back door today.

The transtion from FB to BF, for example, means yesterday, the food was at the front door, today, the food is at the back door, and tomorrow, the food will be at the front door.

The transition from FB to FB, for example, is not possible because the first state implies the food is at the back door today but the second state implies the food is at the front door today.

States	**FF**	**FB**	**BF**	**BB**
FF	p	q	0	0
FB	0	0	p	q
BF	p	q	0	0
BB	0	0	p	q

Panda is disappointed for the states FB and BF

The long run expected proportion of days when Panda is disappointed is $\pi(FB) + \pi(BF)$

Balance equations:

$\pi(FF) = p\pi(FF) + p\pi(BF)$

$\pi(FB) = q\pi(FF) + q\pi(BF)$

$\pi(BF) = p\pi(BF) + p\pi(BB)$

$\pi(BB) = q\pi(BF) + q\pi(BB)$

If we solve these, we get $\pi(BF) = \pi(FB) = pq$

Therefore, our final answer is:

$\pi(FB) + \pi(BF) = 2pq$

b) $P(\text{front door yesterday and not disappointed, front door today and not disappointed}) = P(FF, FF) = p$

$P(\text{Panda arriving at front door and not being disappointed one year from now}) = \pi(FF) = \frac{p}{q}\pi(BF) = p^2$ (from solving the system of equations from part a)

c) From part a, we know Panda's initial strategy results in disappointment 2pq of the time. We must find when each of these alternative methods result in a lower proportion of disappointment.

If Panda always goes to the front door, he is disappointed on days when we pick the back door. The expected proportion of the days we pick the back door is q. If $p > 1/2$, then this strategy is better than the one in (a) because it results in a lower expected proportion of days when Panda is disappointed ($q < 2pq$ if $p > 1/2$).

If Panda always goes to the back door, he is disappointed on days when we pick the front door. The expected proportion of days we pick the front door is p. If $p < 1/2$, then this strategy is better than the one in (a) because it results in a lower expected proportion of days when Panda is disappointed ($p < 2pq$ if $p < 1/2$).

If Panda always goes to the other door, then the expected proportion of days of disappointment is $\pi(FF) + \pi(BB) = p^2 + q^2$.

$p^2 + q^2 < 2pq$

$p^2 - 2pq + q^2 < 0$

$(p - q)^2 < 0$ This is not true for any p and q.

Therefore, since $p^2 + q^2 \geq 2pq$ for all p and q, this strategy is never better than the one described in (a).

#newpage ### 4. Jump Up, Fall Down ### Consider a Markov Chain with state space $0, 1, 2, \ldots, 12$ and transition behavior given by: - For $0 \leq i \leq 11$, the distribution of X_{n+1} given $X_n = i$ is uniform on $i + 1, i + 2, \ldots, 12$. - $P(12, 0) = 1$.

a) Complete the cell below to construct the transition matrix of this chain and assign it to the name `jump_fall`.

In [3]:
```
#Answer to 4a
s = np.arange(0, 13)
def transition_probs(i, j):
    if(i == 12):
        if(j == 0):
            return 1
        else:
            return 0
    elif(j > i):
        return 1/(12 - i)
    else:
        return 0


jump_fall_tbl = Table().states(s).transition_function(transition_probs)
jump_fall = jump_fall_tbl.to_markov_chain()
jump_fall
```

Out[3]:
```
     0         1         2         3         4         5         6         7  \
0  0.0  0.083333  0.083333  0.083333  0.083333  0.083333  0.083333  0.083333
```