Homework_01

September 5, 2019

```
In [2]: from prob140 import *
        from datascience import *
        import numpy as np
        from scipy import special

        import matplotlib.pyplot as plt
        %matplotlib inline
        import matplotlib
        matplotlib.style.use('fivethirtyeight')
```

1 Homework 1

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.

1.0.2 How to Do Your Homework

The point of homework is for you to try your hand at using what you've learned in class. The steps to follow:

- Go to lecture and sections, and also go over the relevant text sections before starting on the homework. This will remind you what was covered in class, and the text will typically contain examples not covered in lecture. The weekly Preparation Guide will list what you should read.
- Work on some of the practice problems before starting on the homework.
- Attempt the homework problems by yourself with the text, section work, and practice materials all at hand. Sometimes the week's lab will help as well. The two steps above will help this step go faster and be more fruitful.
- At this point, seek help if you need it. Don't ask how to do the problem — ask how to get started, or for a nudge to get you past where you are stuck.

- For a good measure of your understanding, keep track of the fraction of the homework you can do by yourself or with minimal help. It's a better measure than your homework score, and only you can measure it.

1.0.3 Rules for Homework

- Every answer should contain a calculation or reasoning. For example, a calculation such as $(1/3)(0.8) + (2/3)(0.7)$ or `sum([(1/3)*0.8, (2/3)*0.7])`is fine without further explanation or simplification. If we want you to simplify, we'll ask you to. But just $\binom{5}{2}$ by itself is not fine; write "we want any 2 out of the 5 frogs and they can appear in any order" or whatever reasoning you used. Reasoning can be brief and abbreviated, e.g. "product rule" or "not mutually excl."
- You may consult others (see "How to Do Your Homework" above) but you must write up your own answers using your own words, notation, and sequence of steps.
- We'll be using Gradescope. You must submit the homework according to the instructions in at the end of homework set.

1.0.4 1. First Repeat

Suppose you roll a die rolled repeatedly. Remember that in this class, dice are assumed fair unless the description says otherwise.

For \$k = 1, 2, 3, ..., 6 \$ let D_k be the event that the first k rolls all show different faces.

a) For \$k = 1, 2, 3, ..., 6 \$ let D_k be the event that the first k rolls all show different faces. Note that $P(D_1) = 0$ because you can't have different faces with just one roll.

Without doing any calculations, draw a Venn diagram that shows the events D_4 and D_5. Make it clear which is which, and justify your answer. Then enter one of the symbols $\leq$, $=$, and $\geq$ in the blank below.

$P(D_5)$ _____$P(D_4)$

b) For $k = 2, 3, \ldots, 7$ let F_k be the event that the kth roll is the first time you see a face that has already appeared. Write the event F_k in terms of the events $D_1, D_2, \ldots, D_6$.

c) In the Venn diagram that you drew in Part **a**, there's a region that corresponds to one of the events F_i for some i. Say which i it is, and shade that event F_i in your diagram. There's no need to draw a new diagram. Just shade the appropriate region in the diagram you already drew.

d) Thus far, you haven't used any fractions – just logic. Now for some calculation. In the code cell below, define a function `prob_D` that takes k as its argument and returns $P(D_k)$.

```
In [10]: def prob_D(k):
             rolls = np.arange(k)
             return np.prod( (6 - rolls) / 6 )
```

Run the cell below to make sure that it is consistent with your answer to Part **a**.

```
In [11]: prob_D(4), prob_D(5)

Out[11]: (0.2777777777777778, 0.09259259259259259)
```

d) In the code cell below, define a function `prob_F` that takes k as its argument and returns $P(F_k)$. Use your function `prob_D` in your definition.

Then use apply to complete the table first_repeat. Its first column contains k and its second column should contain $P(F_k)$. For a reminder of the use of apply, see the definition of the array different in Section 1.4.

```
In [12]: first_repeat = Table().with_column('k', np.arange(2, 8))

         def prob_F(k):
             return prob_D(k-1) - prob_D(k)

         all_probs_F_k = first_repeat.apply(prob_F, 'k')

         first_repeat = first_repeat.with_column('P(F_k)', all_probs_F_k)

         first_repeat
```

```
Out[12]: k    | P(F_k)
         2    | 0.166667
         3    | 0.277778
         4    | 0.277778
         5    | 0.185185
         6    | 0.0771605
         7    | 0.0154321
```

e) Run the cell below. If you created all_probs_F_k correctly, the sum should be a recognizable special value. Explain why the sum comes out that way.

```
In [13]: sum(all_probs_F_k)
```

```
Out[13]: 1.0
```

1.0.5 [Solution] First Repeat

a) $D_5 \subseteq D_4$ because if the first five rolls are all different then the first four had to be different. In the blank: $\leq$

b) $F_k = D_{k-1} \backslash D_k$

c) $i = 5$. It's the ring of points that are in D_4 but not in D_5.

e) The first repetition has to happen somewhere in the rolls 2 through 7. Also first repetition can't happen on two different rolls, so $F_2, F_2, \ldots, F_7$ are mutually exclusive. So the probabilities add up to 1.

You can also establish this algebraically using Part **b** and the difference rule, but the solution above is more illuminating.

1.0.6 2. A Different Approximation

At the end of Section 1.5 of the textbook, an approximate value of the chance of a collison in n trials involving N available codes is:

Approximation 1:

$$P(\text{collision}) \sim 1 - e^{-\frac{n^2}{2N}}$$

A simpler approximation that is often used is Approximation 2:

$$P(\text{collision}) \approx \frac{n^2}{2N}$$

See Wikipedia, for example, and keep in mind that their H is our N.

a) Derive Approximation 2 from Approximation 1. Refer to properties of the exponential function if you need to.

b) As you have seen, for $N = 365$ the chance of a collision is just over 0.5 when $n = 23$. Use Approximation 2 to find an approximate value of n by setting $P(\text{collision})$ to be 0.5 and N to be 365. Use the code cell below.

In [15]:
```
N = 365
p_collision = 0.5

"""Note: The value of n below is an approximation"""

n = (p_collision * 2*N) ** 0.5
n
```

Out[15]: 19.1049731745428

c) The answer to **b** is not great as an approximation to 23, but it's not terrible either. The simple approximation is a great way to get a rough sense of how many trials you need to for a specified collision probability when N is too large for exact calculations.

For example, suppose you use a 64-bit hash. Then there are $N = 2^{64} \approx 1.8 \times 10^{19}$ hash values. Use Approximation 2 to find an approximate number of trials n so that the probability of a collision is about 0.25. Use the code cell below to write an expression that evaluates to the numerical value of the n that you found.

In [17]:
```
(0.25 * 2 * (2 ** 64)) ** 0.5
```

Out[17]: 3037000499.97605

1.0.7 [Solution] A Different Approximation

a) Use the approximation

$$e^x \sim 1 + x$$

for small x.

Then $e^{-\frac{n^2}{2N}} \sim 1 - \frac{n^2}{2N}$, so $1 - e^{-\frac{n^2}{2N}} \sim 1 - (1 - \frac{n^2}{2N}) = \frac{n^2}{2N}$.

1.0.8 3. Heads in Coin Tossing

This is one of the fundamental models of probability theory. Note that unless otherwise specified, coins in this course are assumed to be fair.

This exercise is a series of quick observations. Before you start, look over some of the Combinatorics exercises in the Math Prerequisites set.

Suppose you toss a coin n times and note down the sequence of heads (H) and tails (T).

Fix an integer k such that $0 \leq k \leq n$.