python example: functions and recursion

author: Michael O'Donnell, 10/15/19

In the early 1600s, Galileo was asked to explain the fact that, although the number of triples of integers from 1 to 6 with sum 9 is the same as the number of such triples with sum 10, when three dice are rolled, a 9 seemed to come up less often than a 10— supposedly in the experience of gamblers.

In probability it's clear that rolling a 9 or 10 are equally likely. But, attempt to recreate the gamblers experience.

In [1]:

```
# imports
import random
import pandas as pd
import matplotlib.pyplot as plt
```

A function to simulates rolling three dice:

In [2]:

```
def roll_three_dice():
    die_one = random.randrange(1,6,1)
    die_two = random.randrange(1,6,1)
    die_three = random.randrange(1,6,1)

    # return the sum of the three dice
    return(die_one + die_two + die_three)
```

In [3]:

```
# testing the above function, roll the dice!
print("sum of rolling three dice:", roll_three_dice())
```

```
sum of rolling three dice: 5
```

A function to simulate rolling three dice x amount of times:

In [4]:

```
def roll_em_again(num_of_rolls):
    nines = 0
    tens = 0

    # roll the dice a selected number of times
    for i in range(0, num_of_rolls):
        sum_of_roll = roll_three_dice()

        if sum_of_roll == 9:
            nines = nines + 1
        elif sum_of_roll == 10:
            tens = tens + 1

    # create dataframe to easily display results
    data = [['nine', nines], ['ten', tens]]
    df = pd.DataFrame(data, columns = ['Sum of Roll', 'Frequency'])
    df['Probability'] = round(df['Frequency']/num_of_rolls, 4)

    return df

    # bonus, display results as a bar chart
    #plt.bar(df['Sum of Roll'], df['Frequency'])
    #plt.show()
```

In [5]:

```
# test the above function, roll three dice five thousand times!
print(roll_em_again(5000))
```

```
  Sum of Roll  Frequency  Probability
0        nine        763       0.1526
1         ten        698       0.1396
```

Now, let's recreate the gamblers experience.

A function to recreate the gamblers experience. How many times did they roll the dice?:

In [6]:

```
def gamblers_experience(number_of_rolls_list):
    # create dataframe for results
    gamblers_df = pd.DataFrame(columns = ['Number of Rolls', 'Probability of Nin

    # calculate the prob of nines, tens at each number of total rolls
    for j in number_of_rolls_list:
        df = roll_em_again(j)
        data = pd.DataFrame([[j, df.iloc[0][2], df.iloc[1][2]]], columns = ['Num

        gamblers_df = gamblers_df.append(data)

    gamblers_df = gamblers_df.set_index('Number of Rolls')
    print(gamblers_df)
```

```
In [10]: gamblers_experience([25,100,500,1000,5000,10000])
```

```
                  Probability of Nine  Probability of Ten
Number of Rolls
25                             0.1200              0.0800
100                            0.1100              0.2100
500                            0.1420              0.1640
1000                           0.1620              0.1580
5000                           0.1536              0.1532
10000                          0.1510              0.1434
```

From the simulation above, the gamblers would've experienced different results at different number of rolls.