

## example\_basketball

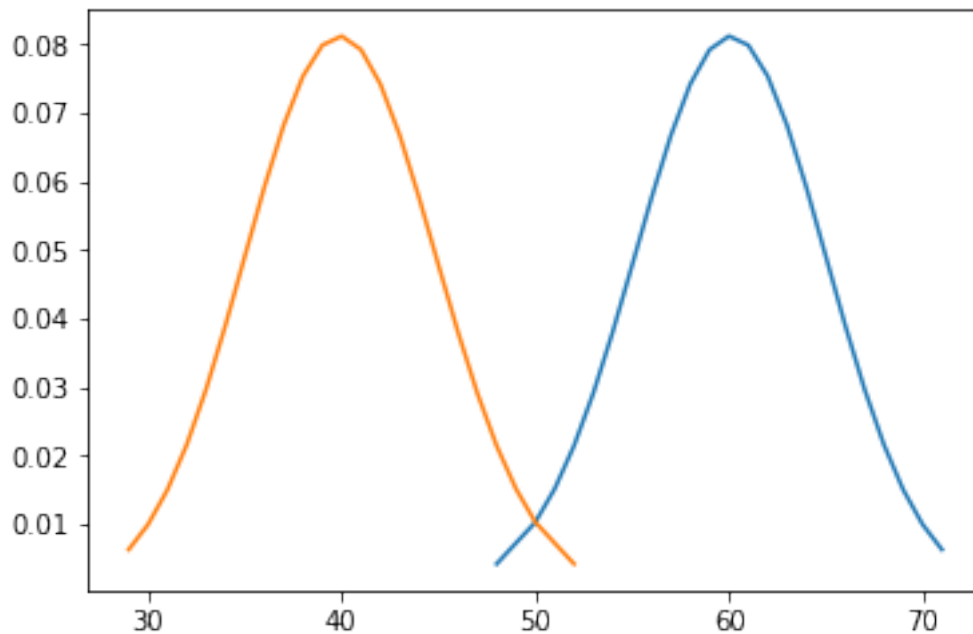
April 23, 2019

```
In [2]: import matplotlib.pyplot as plt
        from scipy.stats import binom
        import numpy as np
        import math

        trials = 100
        probb_two_point = 0.6
        probb_three_point = 0.4

        inputs_two_point = [binom.ppf(x, trials, probb_two_point) for x in np.arange(0, 1, 0.01)]
        inputs_three_point = [binom.ppf(x, trials, probb_three_point) for x in np.arange(0, 1, 0.01)]
        pdf_two_point = [binom.pmf(x, trials, probb_two_point) for x in inputs_two_point]
        pdf_three_point = [binom.pmf(x, trials, probb_three_point) for x in inputs_three_point]
        plt.plot(inputs_two_point[1:], pdf_two_point[1:])
        plt.plot(inputs_three_point[1:], pdf_three_point[1:])
```

```
Out[2]: [<matplotlib.lines.Line2D at 0x10ec23160>]
```



```

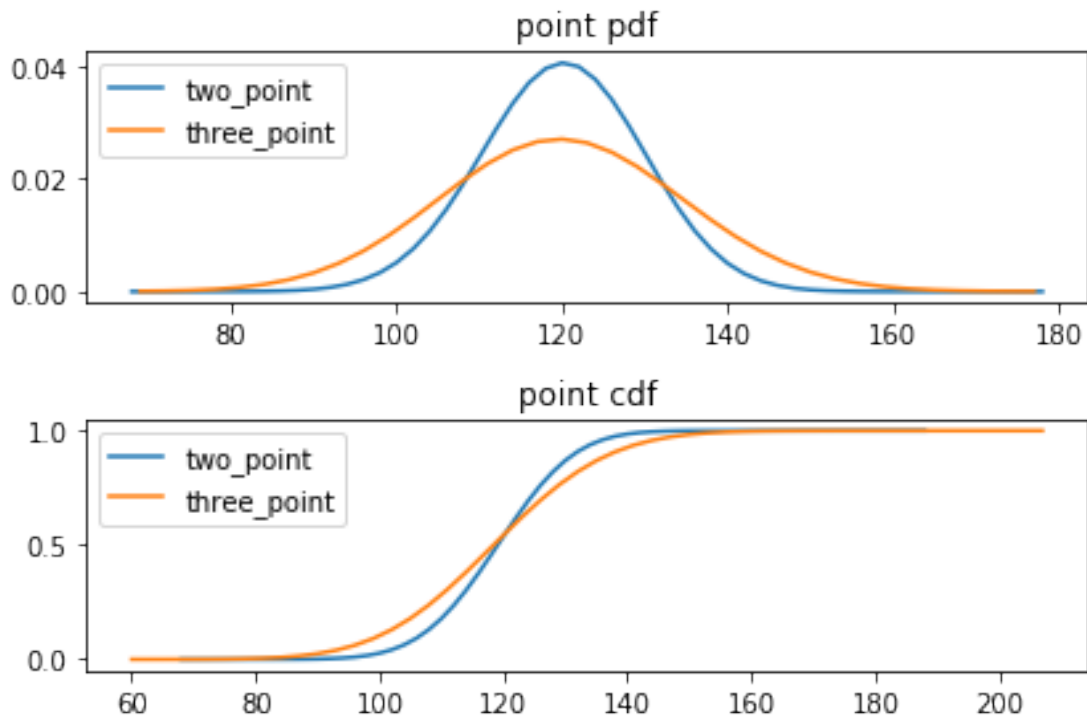
In [3]: trials = 100
        prob_two_point = 0.6
        prob_three_point = 0.4

        fig, axs = plt.subplots(2, 1)

        shot_attempts = list(range(101))
        points_twos = [2*s for s in shot_attempts]
        points_threes = [3*s for s in shot_attempts]
        pdf_two_point = [1 / 2 * binom.pmf(x, trials, prob_two_point) for x in shot_attempts]
        pdf_three_point = [1 / 3 * binom.pmf(x, trials, prob_three_point) for x in shot_attempts]
        axs[0].plot(points_twos[34:90], pdf_two_point[34:90], label='two_point')
        axs[0].plot(points_threes[23:60], pdf_three_point[23:60], label='three_point')
        axs[0].legend(loc='upper left')
        axs[0].set_title('point pdf')

        cdf_two_point = np.array(pdf_two_point).cumsum() * 2
        cdf_three_point = np.array(pdf_three_point).cumsum() * 3
        axs[1].plot(points_twos[34:95], cdf_two_point[34:95], label='two_point')
        axs[1].plot(points_threes[20:70], cdf_three_point[20:70], label='three_point')
        axs[1].legend(loc='upper left')
        axs[1].set_title('point cdf')
        plt.tight_layout()

```



```

In [15]: def game_outcome_probabilities(shot_attempts, prob_three, prob_two):
    """probability of a three point only shooting team beating a two point only
    shooting team  $x_{N\_pt}$  = number of baskets made for the  $N$  point shooting team
    Prob(three_point_team_wins) =
    Prob(two_point_team_points < three_point_team_points) =
     $P(2x_{two\_pt} < 3x_{three\_pt}) =$ 
     $P(x_{two\_pt} < 1.5x_{three\_pt}) =$ 
    Sum[n=0..trials]P( $x_{two\_pt} < 1.5x_{three\_pt} / x_{three\_pt} = n$ )P( $x_{three\_pt} = n$ )

    Args:
        shot_attempts (int): number of shots taken by each team
        prob_three (float): probability of score for the three point taking team
        prob_two (float): probability of score for the two point taking team

    Returns:
        dict: {'three wins': float, 'tie': float, 'two wins': float}
    """
    prob_three_wins = 0
    prob_tie = 0
    for made_threes in np.arange(shot_attempts + 1):
        prob_made_threes = binom.pmf(made_threes, shot_attempts, prob_three)
        max_two_made_still_lose = math.floor(1.5*made_threes)
        if 2 * max_two_made_still_lose == 3 * made_threes:
            prob_tie = (prob_tie
                        + binom.pmf(max_two_made_still_lose, shot_attempts, prob_two)
                        * prob_made_threes)
            max_two_made_still_lose = max_two_made_still_lose - 1
        if max_two_made_still_lose < 0:
            continue
        two_make_cdf = binom.cdf(max_two_made_still_lose, shot_attempts, prob_two)
        prob_three_wins = prob_three_wins + two_make_cdf * prob_made_threes

    res = {'three wins': prob_three_wins,
          'tie': prob_tie,
          'two wins': 1 - prob_three_wins - prob_tie}
    return res

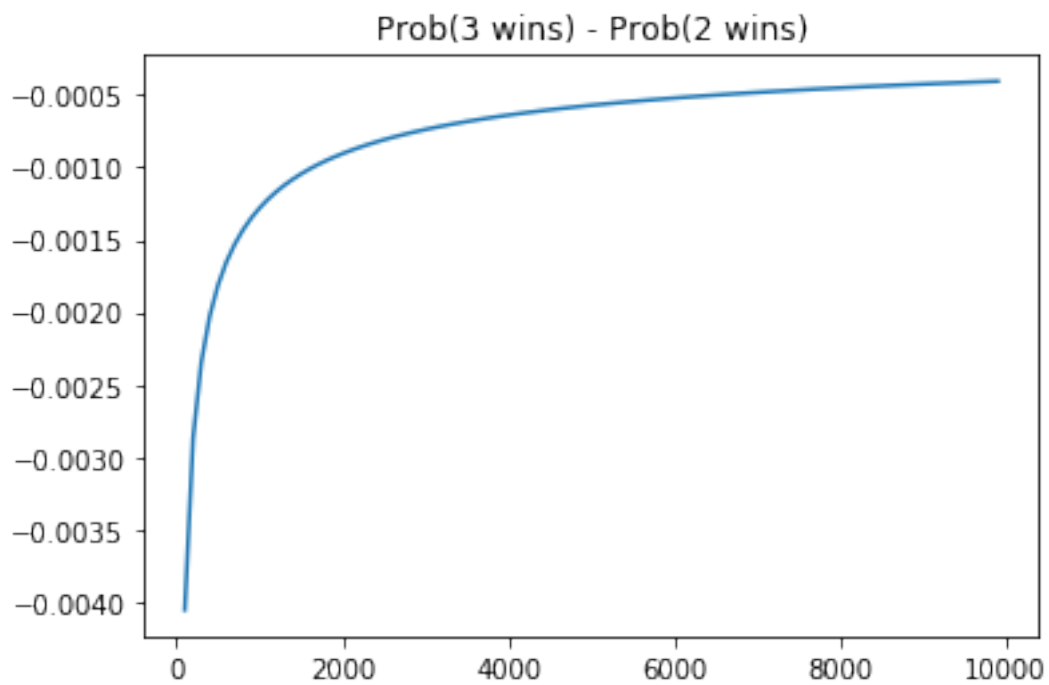
In [26]: shot_attempts = np.arange(0, 10000, 100)
        probs = [game_outcome_probabilities(shots, 0.4, 0.6) for shots in shot_attempts]
        prob_differential = [prob['three wins'] - prob['two wins'] for prob in probs]
        plt.plot(shot_attempts[1:], prob_differential[1:])
        plt.title('Prob(3 wins) - Prob(2 wins)')

```

```

Out[26]: Text(0.5, 1.0, 'Prob(3 wins) - Prob(2 wins)')

```



In [ ]:

In [ ]: