Sujit Kanti Biswas
GT User ID: sbiswas67
GT ID: 903549376

Project 1: Martingale Report

1. In Experiment 1, based off the experiment results calculate the estimated probability of winning $80 within 1000 sequential bets. Explain your reasoning using the experiment thoroughly. (not based on plots).

   Answer:
   We shall check if the value of the last index of the numpy array is equal to 80 for each of the simulations (containing 1000 sequential spins/bets). If the last index has value 80, then it means that $80 was won in that simulation otherwise lost within those 1000 sequential spins/bets.

   For both the simulations of 10 and 1000 (containing 1000 sequential spins/bets) the frequency of winning simulation came as 10/10 and 1000/1000 respectively.
   Hence, the estimated probability of winning $80 within 1000 sequential spin/bets is 1 i.e., 100%.

   <Code Snippet – Commented out in the submitted code martingale.py >

```
######### Experiment-1a #########
simulation_times = 10
simulation_results = np.zeros((simulation_times, 1001))
win_count = 0
for loop in range(simulation_times):
    simulation_results[loop] = roulette_Exp1(win_prob)
    if simulation_results[loop][-1] == 80:
        win_count += 1                              #Winning simulation
print("Probability of winnings $80 with ", simulation_times, " simulations is ",
win_count/simulation_times)
….
….

######### Experiment-1b #########
simulation_times = 1000
simulation_results = np.zeros((simulation_times, 1001))
win_count = 0
for loop in range(simulation_times):
    simulation_results[loop] = roulette_Exp1(win_prob)
    if simulation_results[loop][-1] == 80:
        win_count += 1                              #Winning simulation
print("Probability of winnings $80 with ", simulation_times, " simulations is ", win_count /
simulation_times)
….
….
```

   <Output:>

```
(ml4t) Sujits-MacBook-Pro:martingale sujitkantibiswas$ python martingale.py
Probability of winnings $80 with  10  simulations is  1.0
Probability of winnings $80 with  1000  simulations is  1.0
```

2. In Experiment 1, what is the estimated expected value of our winnings after 1000 sequential bets? Explain your reasoning thoroughly.

Answer:
The estimate expected value (or mean) of winnings after 1000 sequential bets is $80.

a) As from question 1, we have calculated from the experiment data that probability of winning $80 within 1000 sequential bets is 1 or 100%. This shows the for each of the simulations we shall always win $80 at the end of the 1000 sequential spins/bets.

b) We can also show the same from the experiment data as follows:
Used the following code to calculate the mean for each of the bets across all the 10 and 1000 simulations respectively for experiment 1. The output shows that expected value (or mean value) reached $80 before the simulations completed the 1000th bet.

<Code Snippet – Commented out in the submitted code martingale.py>

```
######### Experiment-1a #########
…
result_mean = np.mean(simulation_results, axis=0)
print("Expected value for Experiment 1 for 10 simulations is ", result_mean)
…
######### Experiment-1b #########
…
result_mean=np.mean(simulation_results,axis=0)
print("Expected value for Experiment 1 with 1000 simulations is ", result_mean)
```

<Output>

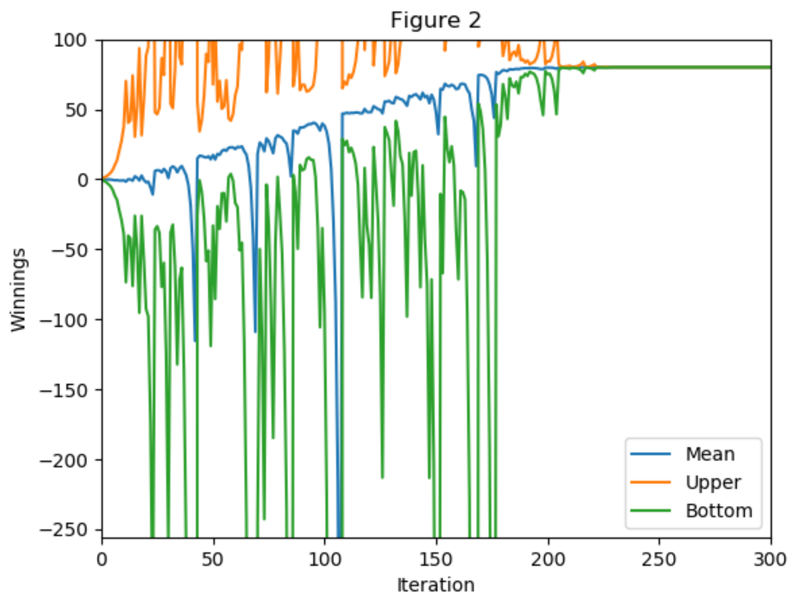
```
(ml4t) Sujits-MacBook-Pro:martingale sujitkantibiswas$ python martingale.py
Expected value for Experiment 1 for 10 simulations is  [ 0.   0.6  0.8 ... 80.  80.  80. ]
Expected value for Experiment 1 with 1000 simulations is  [ 0.0e+00 -3.0e-02 -5.1e-02 ...  8.0e+01  8.0e+01
8.0e+01]
```

3. In Experiment 1, does the standard deviation reach a maximum value then stabilize and/or converge as the number of sequential bets increases? Explain why it does (or does not) thoroughly.

Answer:
No, the standard deviation does not reach a maximum value and then gradually stabilize/converge. Rather the SD is fluctuating all over the place and then becomes zero as the mean of the winnings stabilizes at $80. Refer to Figure 2 below.

This is true because SD and Mean are taken for each of the 1000 spins/bets over multiple simulation runs. Since, the probability of winning $80 is 1 or 100% for 1000 sequential spins/bets (as seen in question 1), so each of the simulations are reaching $80 win after certain number of bets/spins and eventually they all stay at $80 which makes the SD to zero.

The reason for standard deviation to be completely fluctuating within a wide range of values before becoming zero (instead of a gradual convergence to zero) is because the result of a spin is random for each of the spins/bets across various simulations which create a completely unique sequence of Stochastic process for each of the simulations.



4. In Experiment 2, based off the experiment results calculate the estimated probability of winning $80 within 1000 sequential bets. Explain your reasoning using the experiment thoroughly. (not based on plots).

Answer:
We shall check if the value of the last index of the numpy array is equal to 80 for each of the 1000 simulations (containing 1000 sequential spins/bets). If the last index has value 80, then it means that $80 was won in that simulation otherwise lost within those 1000 sequential spins/bets.

For the 1000 simulation experiment (containing 1000 sequential spins/bets) the frequency of winning simulation came as 648/1000.
Hence, the estimated probability of winning $80 within 1000 sequential spin/bets is 0.648 i.e., 64.8%.

This makes sense, because for this experiment we are bounded by a spending limit of $256. So, for any of the simulations where the loss reached -$256, it shall be considered as a lost game.

<Code Snippet – Commented out in the submitted code martingale.py >

```
######### Experiment-2 #########
simulation_times = 1000
simulation_results = np.zeros((simulation_times, 1001))
win_count = 0
for loop in range(simulation_times):
```

```
    simulation_results[loop] = roulette_Exp2(win_prob)
    if simulation_results[loop][-1] == 80:
        win_count += 1  # Winning simulation
print("Exp2: Probability of winnings $80 with ", simulation_times, " simulations is ",
win_count / simulation_times)
```

<Output>

```
(ml4t) Sujits-MacBook-Pro:martingale sujitkantibiswas$ python martingale.py
Exp2: Probability of winnings $80 with  1000  simulations is  0.648
```

5. In Experiment 2, what is the estimated expected value of our winnings after 1000 sequential bets? Explain your reasoning thoroughly. (not based on plots)

Answer:
The estimate expected value (or mean) of winnings after 1000 sequential bets is approximately -$37.99.6

a) From the experiment 2 data with 1000 simulations, we can see that :
   - 351 simulations have lost $256 after 1000 sequential bets
   - 648 simulations have won $80 after 1000 sequential bets
   - 1 simulation has won $20 after 1000 sequential bets

   Now if we put these figures in a weighted average, we get…
   =-$256 * (351/1000) + $80 * (648/1000) + $20 * (1/1000)
   = -256 * 0.351 + 80 * 0.648 + 0.02
   = -89.856 + 51.84 + 0.02
   = - $37.996

b) We can also show the same from the experiment data as follows:
   Used the following code to calculate the mean for each of the bets across all the 1000 for experiment 2. The output shows that expected value (or mean value) reached -$37.996 before the simulations completed the 1000th bet.

   <Code Snippet – Commented out in the submitted code martingale.py>

```
######### Experiment-2 #########
…
…

result_mean=np.mean(simulation_results,axis=0)
print("Expected value for Experiment 2 is ", result_mean)
…
```

   <Output>

```
(ml4t) Sujits-MacBook-Pro:martingale sujitkantibiswas$ python martingale.py
Expected value for Experiment 2 is  [  0.     -0.05   -0.165 ... -37.99  -37.992 -37.996]
```

6. In Experiment 2, does the standard deviation reach a maximum value then stabilize and/or converge as the number of sequential bets increases? Explain why it does (or does not) thoroughly.

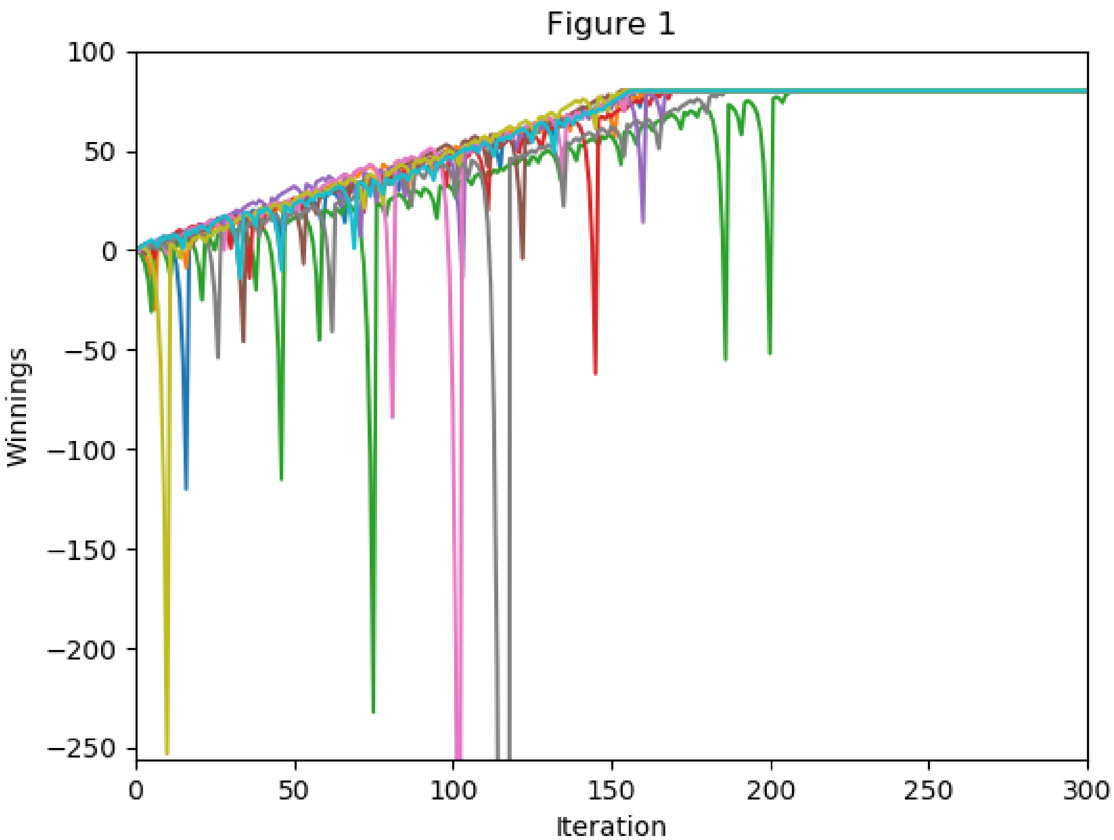
   Answer:

   Yes, the standard deviation reaches a maximum and then stabilizes after that as the number of sequential bets increases. Refer to Figure.

   This is because after a certain number of spins/bets most of the simulations also stabilize as being either $80 (win) or $-256(lost). As most of the simulation outcomes fall into any of these two categories ($80 or -$256), the standard deviation stabilizes as the variance between these two values.

7. Include figures 1 through 5.

   Answer:



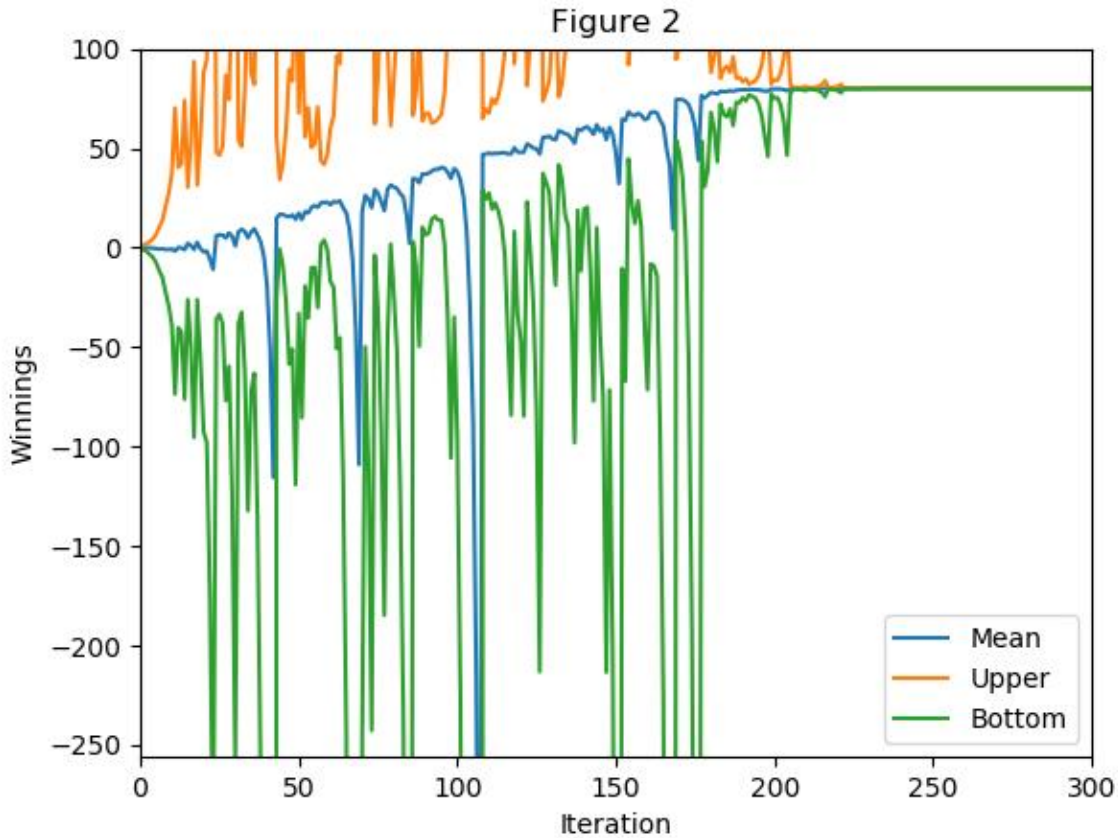


Figure 2
Winnings
Iteration
100
50
0
−50
−100
−150
−200
−250
0
50
100
150
200
250
300
Mean
Upper
Bottom

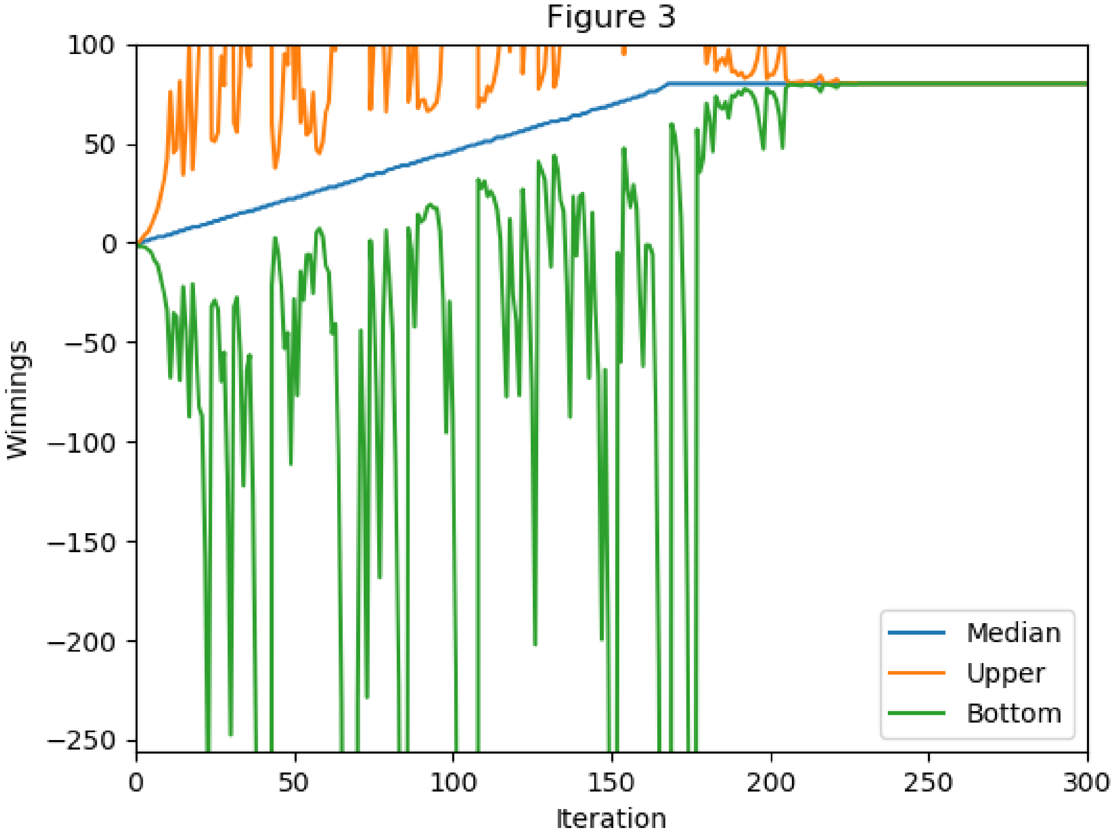


Figure 3
Winnings
Iteration
Median
Upper
Bottom
100
50
0
−50
−100
−150
−200
−250
0
50
100
150
200
250
300

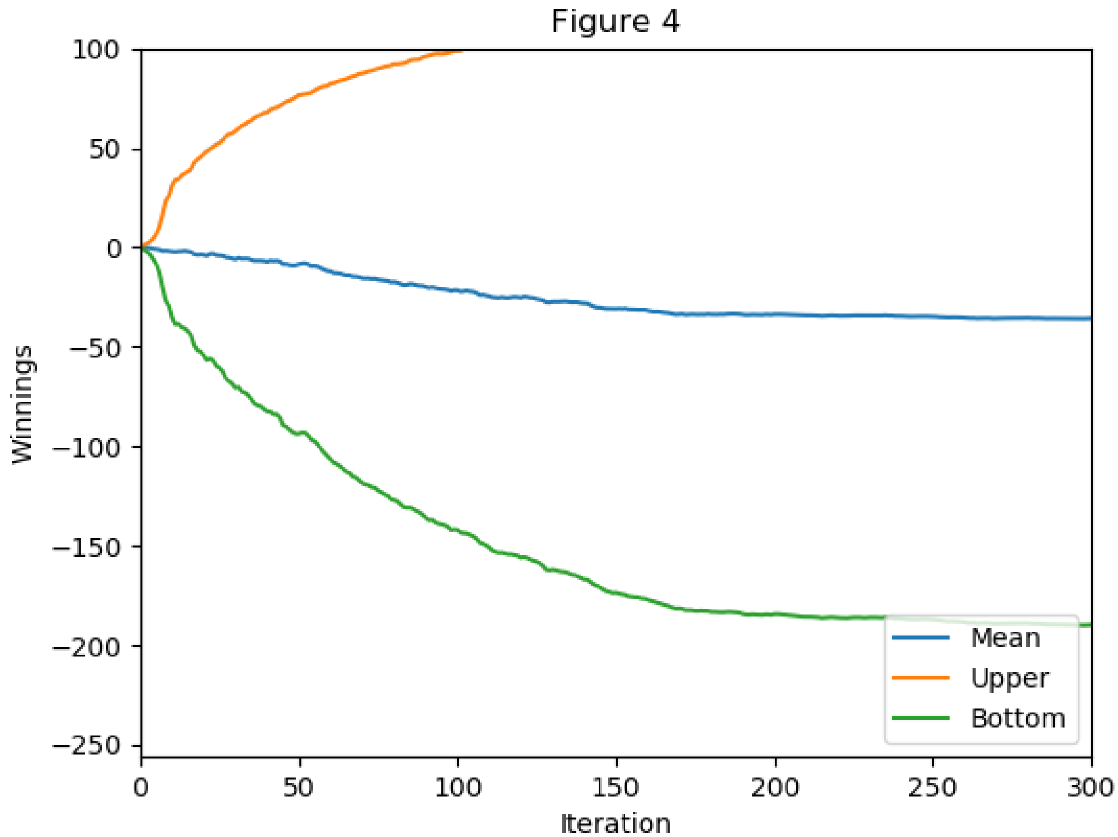


Figure 4
Winnings
100
50
0
−50
−100
−150
−200
−250
0
50
100
150
200
250
300
Iteration
Mean
Upper
Bottom

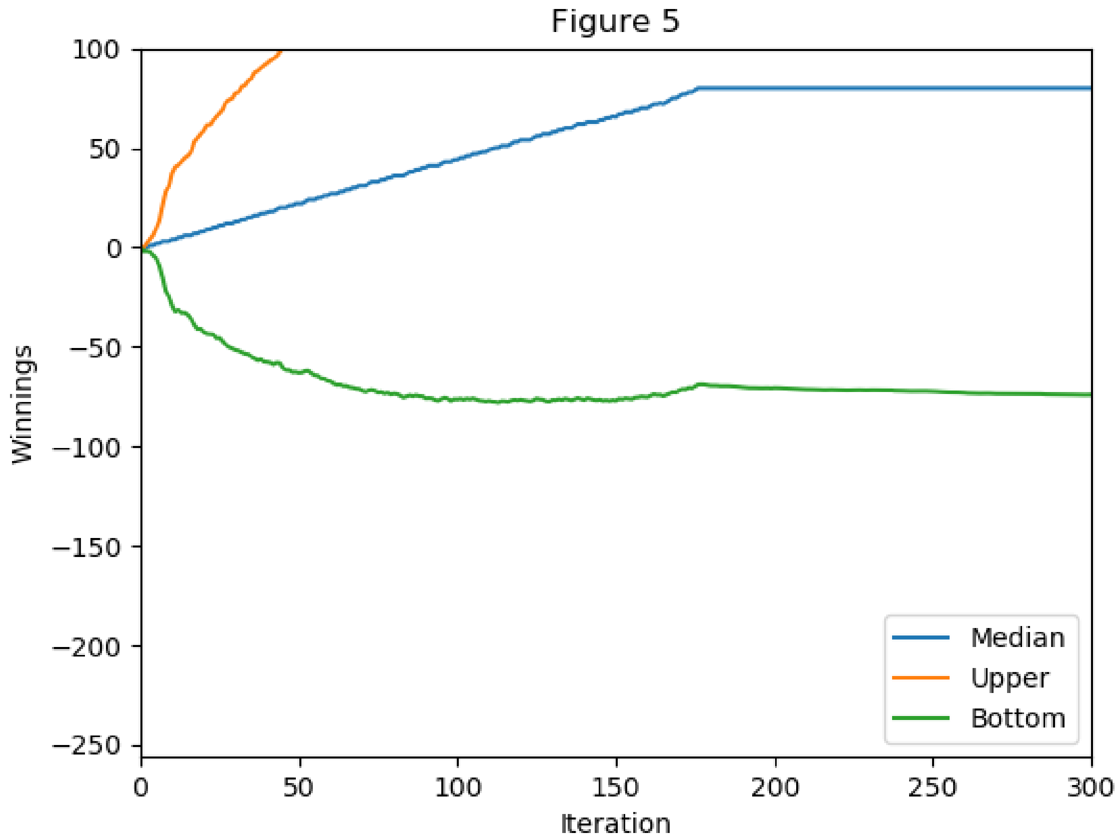


Figure 5
Winnings
Iteration
100
50
0
−50
−100
−150
−200
−250
0
50
100
150
200
250
300
Median
Upper
Bottom