Backgammon – Analyzing Doubling

Professor Steven J Miller : sjm1@williams.edu

In[154]:=

```
double[numdo_, pdouble_, paccept_, pointswin_, print_] := Module[{},
   (* numdo is number of iterations *)
   (* pdouble/paccept is threshold for first to double 2nd to accept *)
   (* pointswin is how many points need to win. our
    model will be to toss a fair coin and each toss each person equally
    likely to get a point. if we take 100 points it makes it very easy. *)
   results = {};
   netresults = 0;
   onewin = 0;
   twowin = 0;
   doublesoffered = {};
   doublesaccepted = {};

   For[n = 1, n ≤ numdo, n++,
    {
     score = pointswin / 2;
     (* player one wins if reach pointswin points, player two if reach 0 *)
     pwin = score / pointswin; (* might as well start keeping track
      when probability of one player of winning is pdouble, and now double *)
     valuegame = 1; (* initially worth one point *)
     whocandouble = 1;
     (* set up so player 1 doubles first wlog *)
     score = Floor[pdouble * pointswin];
     doublesingameoffered = 0;
     doublesingameaccepted = 0;

     While[score > 0 && score < pointswin,
      {
       pwin = score / pointswin;
       If[print == 1, Print["Score = ", score,
         " and prob win = ", 1.0 pwin, " and value of game = ", valuegame]];
       (* see if player one can double *)
       If[pwin ≥ pdouble && whocandouble == 1,
        If[pwin > paccept,
          {
           (* decline *)
           results = AppendTo[results, valuegame];
           onewin = onewin + 1;
           netresults = netresults + valuegame;
           score = pointswin + 100; (* ends game! *)
           If[print == 1, Print["Two declines"]];
           doublesingameoffered++;
          },
          {
           (* accept *)
           valuegame = valuegame * 2;
```

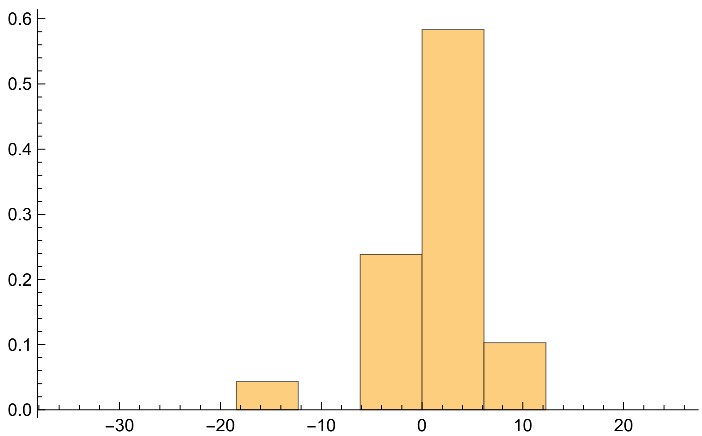
```
        whocandouble = 2;
        If[print == 1, Print["Two accepts"]];
        doublesingameoffered++;
        doublesingameaccepted++;
       }];
   ]; (* end of player one doubling *)
   (* see if player two can double *)
   If[1 - pwin ≥ pdouble && whocandouble == 2,
    If[1 - pwin > paccept,
      {
       (* decline *)
       results = AppendTo[results, -valuegame];
       twowin = twowin + 1;
       netresults = netresults - valuegame;
       score = -100; (* ends game! *)
       If[print == 1, Print["One declines"]];
       doublesingameoffered++;
      },
      {
       (* accept *)
       valuegame = valuegame * 2;
       whocandouble = 1;
       If[print == 1, Print["One accepts"]];
       doublesingameoffered++;
       doublesingameaccepted++;
      }];
   ]; (* end of player two doubling *)

   (* check to see if game should end *)
   (* do next turn *)
   If[Random[] < .5, score = score + 1, score = score - 1];

   If[score == pointswin,
    {
     onewin = onewin + 1;
     results = AppendTo[results, valuegame];
     netresults = netresults + valuegame;
     score = score + 100;
    }];
   If[score == 0,
    {
     twowin = twowin + 1;
     results = AppendTo[results, -valuegame];
     netresults = netresults - valuegame;
     score = - 100;
    }];

  }]; (* end of while loop *)
 doublesoffered = AppendTo[doublesoffered, doublesingameoffered];
 doublesaccepted = AppendTo[doublesaccepted, doublesingameaccepted];
}]; (* end of n loop *)
```

```
  Print["Prob double is ", 1.0 pdouble, " and prob accept is ", 1.0 paccept];
  Print["Player one's winning percentage = ", 100.0 onewin / numdo, "%."];
  Print["Player two's winning percentage = ", 100.0 twowin / numdo, "%."];
  Print["netresults / numgames = ", 1.0 netresults / numdo];
  Print["Average Abs[game value] = ",
   1.0 Sum[Abs[results[[k]]], {k, 1, Length[results]}] / numdo];
  Print["Ave number of doubles offered = ",
   1.0 Mean[doublesoffered], " and stdev = ", 1.0 StandardDeviation[doublesoffered]];
  Print["Ave number of doubles accepted = ",
   1.0 Mean[doublesaccepted], " and stdev = ", 1.0 StandardDeviation[doublesaccepted]];
  Print["Histogram of Results - how much the first to double wins."];
  Print[Histogram[results, Automatic, "Probability"]];
  Print["Histogram on number of doubles offered."];
  Print[Histogram[doublesoffered, Automatic, "Probability"]];
 ]; (* end of module *)
```

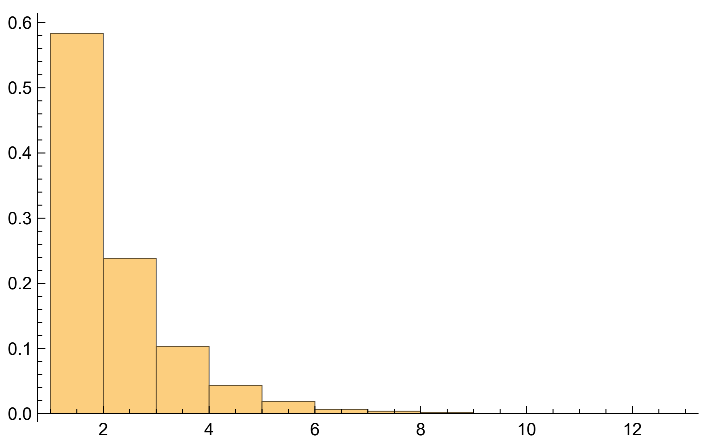
In[156]:=

```
Timing[double[10000, 70 / 100, 95 / 100, 100, 0]]
```

```
Prob double is 0.7 and prob accept is 0.95
Player one's winning percentage = 70.93%.
Player two's winning percentage = 29.07%.
netresults / numgames = 0.1304
Average Abs[game value] = 7.1576
Ave number of doubles offered = 1.7289 and stdev = 1.1398
Ave number of doubles accepted = 1.7289 and stdev = 1.1398
Histogram of Results - how much the first to double wins.
```



```
Histogram on number of doubles offered.
```



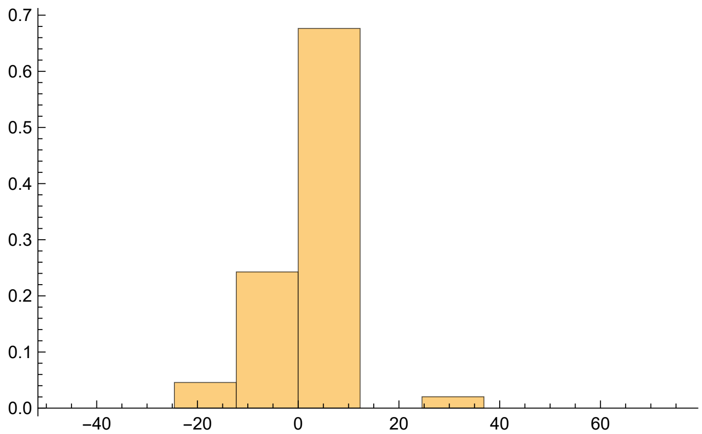
Out[156]=

```
{36.3438, Null}
```

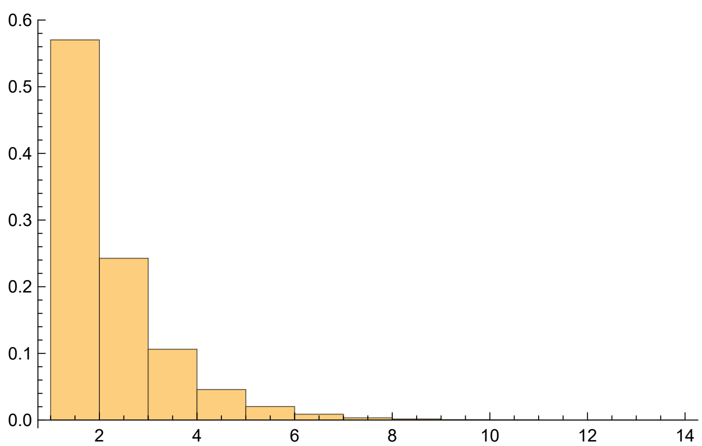
In[160]:=

```
Timing[double[40000, 70 / 100, 95 / 100, 100, 0]]
```

```
Prob double is 0.7 and prob accept is 0.95

Player one's winning percentage = 70.07%.

Player two's winning percentage = 29.93%.

netresults / numgames = -0.06205

Average Abs[game value] = 7.49655

Ave number of doubles offered = 1.7594 and stdev = 1.1595

Ave number of doubles accepted = 1.7594 and stdev = 1.1595

Histogram of Results - how much the first to double wins.
```




```
Histogram on number of doubles offered.
```



Out[160]=

```
{142.047, Null}
```

```
In[5]:= checkprob[numdo_, prob_, pointswin_] := Module[{},
    onewin = 0;
    twowin = 0;
    For[n = 1, n ≤ numdo, n++,
     {
      score = prob * pointswin;
      While[score > 0 && score < pointswin,
       {
        score = score + If[Random[] < .5, 1, -1];
       }]; (* end of while loop *)
      If[score == pointswin, onewin = onewin + 1, twowin = twowin + 1];
     }]; (* end of for loop *)
    Print["Prob p = ", prob];
    Print["Prob player 1 wins is ", onewin * 100.0 / numdo];
    Print["Prob player 2 wins is ", twowin * 100.0 / numdo];
   ];

In[8]:= checkprob[10000, .5, 100]

Prob p = 0.5

Prob player 1 wins is 49.9

Prob player 2 wins is 50.1

In[9]:= checkprob[10000, .72, 100]

Prob p = 0.72

Prob player 1 wins is 71.73

Prob player 2 wins is 28.27

In[10]:= checkprob[10000, .87, 100]

Prob p = 0.87

Prob player 1 wins is 87.5

Prob player 2 wins is 12.5

In[11]:= checkprob[10000, .87, 200]

Prob p = 0.87

Prob player 1 wins is 87.26

Prob player 2 wins is 12.74
```