Math 344: Lecture 13: Mathematica tutorial: Questions? Email sjm1@williams.edu

```
In[ ]:= Print[Hyperlink["https://www.youtube.com/watch?v=g1oj7CIqGM8"]]
        Print[Hyperlink[
          "http://web.williams.edu/Mathematics/sjmiller/public_html/math/handouts/latex.
            htm"]]
```

https://www.youtube.com/watch?v=g1oj7CIqGM8

http://web.williams.edu/Mathematics/sjmiller/public_html/math/handouts/latex.htm

Lecture on problem solving and coding

```
In[ ]:= Print[
         Hyperlink["https://web.williams.edu/Mathematics/sjmiller/public_html/341Fa21/
            mathematicaprograms/Math341_DiplomaProblems.nb"]]
        Print[
         Hyperlink["https://web.williams.edu/Mathematics/sjmiller/public_html/341Fa21/
            mathematicaprograms/Math341_DiplomaProblems.pdf"]]
```

https://web.williams.edu/Mathematics/sjmiller/public_html/341Fa21/mathematicaprograms/Math341_DiplomaProblems.nb

https://web.williams.edu/Mathematics/sjmiller/public_html/341Fa21/mathematicaprograms/Math341_DiplomaProblems.pdf

---

Long Suits in Bridge

In a hand of bridge, each of the four players is dealt 13 cards. It does not matter what order you get the cards, only which cards you get. NOTE: $nCr = \binom{n}{r}$

What is the probability you are dealt at least 7 cards in a suit?

```
In[1]:= (* Want the probability of at least 7
         in same suit in a hand of 13 cards*)
        (* create a simple deck
         good enough for this problem *)
        deck = {}; (* initializes deck to be empty *)
        For[i = 1, i ≤ 4, i++,
         For[j = 1, j ≤ 13,
           j++, deck = AppendTo[deck, i]];
        ] (* end of the i loop *)
        Print[deck]

        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
         2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4}

In[60]:= weightedatleastseven[numdo_] := Module[{},
           success = 0; (* counts how often happens *)
           For[n = 1, n ≤ numdo, n++,
            {
             hand = RandomSample[deck, 13];
             (* chooses a hand of 13 *)
             For[s = 0, s ≤ 5, s++, suit[s] = 0];
             (* initialize each suit count to be 0 *)
             For[c = 1, c ≤ 13, c++,
              {
               x = hand⟦c⟧;
               (*this calls in the c-th element*)
               suit[x] = suit[x] + 1;
               For[s = 1, s ≤ 4, s++,
```

```
        If[suit[s] ≥ 7, success = success + 1]];
       (* this gives a WEIGHTED count -
        get one point for 7 in a row,
       2 for 8 in a row, .... Even worse as can
        get multiple credit for 7 in a row! Exit
        the loop once you find 7 in a row or
        better put this outside the c loop!*)
      }];
     (* this is where
      should put the s For loop *)
    }]; (* end of the n loop *)
   Print["Observed success rate is ",
    100.0 success / numdo, "%."];
   theory =
    4.0 Sum[ Binomial[13, k] Binomial[39, 13 - k] ,
       {k, 7, 13}] / Binomial[52, 13];
   Print["Theoretical success rate is about ",
    100.0 theory, "%."];
  ] (* end of the module *)
```

In[61]:=
```
Timing[weightedatleastseven[10000]]
```

```
Observed success rate is 7.81%.

Theoretical success rate is about 4.03213%.
```

Out[61]=
```
{0.34375, Null}
```

In[62]:=
```
atleastseven[numdo_] := Module[{},
  success = 0; (* counts how often happens *)
  For[n = 1, n ≤ numdo, n++,
   {
    hand = RandomSample[deck, 13]; (* chooses a hand of 13 *)
    For[s = 0, s ≤ 5, s++, suit[s] = 0];
    (* initialize each suit count to be 0 *)
    For[c = 1, c ≤ 13, c++,
     {
      x = hand[[c]]; (*this calls in the c-th element*)
      suit[x] = suit[x] + 1;
     }];
    For[s = 1, s ≤ 4, s++,
     If[suit[s] ≥ 7, success = success + 1]];
   }
  ]; (* end of the n loop *)
  Print["Observed success rate is ",
   100.0 success / numdo, "%."];
  theory =
   4.0 Sum[ Binomial[13, k] Binomial[39, 13 - k] , {k, 7, 13}] /
     Binomial[52, 13];
  Print[
   "Theoretical success rate is about ", 100.0 theory, "%."];
 ] (* end of the module *)
```

In[63]:=
```
atleastseven[100 000]
```

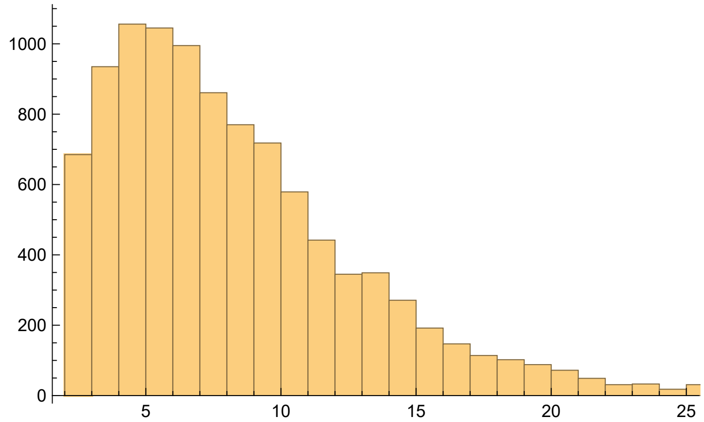
Observed success rate is 4.018%.

Theoretical success rate is about 4.03213%.

In[40]:=
```
(* darth vader problem - fail with prob p,
die after first fail *)
darthvader[numdo_, p_, numfailsallowed_] :=
 Module[{},
```

```
  (* p is the prob of failing,
  numdo is number of iterations *)
  servicetime = {}; (* how long live *)
  For[n = 1, n ≤ numdo, n++,
   {
    havenotfailed = numfailsallowed;
    tasksdone = 0;
    While[havenotfailed > 0,
     {
      x = Random[];
      (* if x ≤ p, failed, else live *)
      tasksdone = tasksdone + 1;
      If[x ≤ p,
       havenotfailed = havenotfailed - 1];
     }]; (* end of while loop *)
    servicetime =
     AppendTo[servicetime, tasksdone];
   }]; (* end of the n loop *)
  Print[Histogram[servicetime]];
  Print[1.0 Mean[servicetime]];
  Print[1.0 StandardDeviation[servicetime]];
 ] (* end of module *)

In[43]:= darthvader[10000, .25, 2]
```



7.9232

4.86153

The Double Sixes Game

You have two fair die.

On each turn you can roll one or both of the die.

The goal is to have both show a 6.

Thus once one of the die lands on a 6 you can stop rolling it.

Questions:

- How many rolls do you expect before you have double sixes?
- What is the probability you win on your first turn? On your second? On your n^{th}?

```
In[68]:= f[n_] := 2 (5 / 6) ^n - (25 / 36) ^n
         g[n_] := 1 - f[n] (* probability succeed by n *)
         success[n_] := g[n] - g[n - 1];
         (* probability succeed at n *)

In[71]:= doublesixes[numdo_] := Module[{},
           count = {};
           For[m = 1, m ≤ numdo, m++,
            {
             firstdie = 0; seconddie = 0; rolls = 0;
             While[firstdie + seconddie < 12,
              {
               rolls = rolls + 1;
               die1 = RandomInteger[{1, 6}];
               die2 = RandomInteger[{1, 6}];
               If[die1 == 6, firstdie = 6];
               If[die2 == 6, seconddie = 6];
              }];
             count = AppendTo[count, rolls];
            }];
           theory = {};
           For[k = 1, k ≤ 30, k++, theory = AppendTo[theory, {k + .5, success[k]}]];
           Print[Show[Histogram[count, Automatic, "Probability"], ListPlot[theory]]];
          ]
```