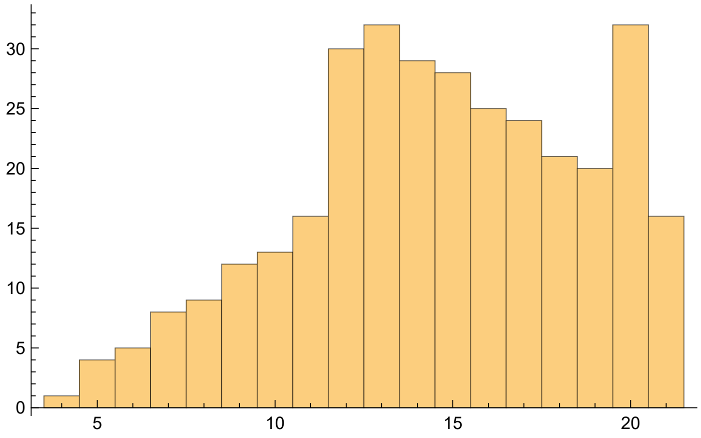
```
In[∘]:= suit = {"A", 2, 3, 4, 5, 6, 7, 8, 9, 10, "J", "Q", "K"};
        myDeck = Join[suit, suit, suit, suit];
        myDeck = {#} & /@ myDeck;
        myDeck = myDeck /. {"J" → 10, "Q" → 10, "K" → 10, {"A"} → {1, 11}};

In[∘]:= deck
Out[∘]=
        {{1, 11}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {10}, {10}, {10},
         {1, 11}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {10}, {10}, {10}}

In[∘]:= twoCards = Subsets[deck, {2}];

In[∘]:= twoCards⟦1⟧
Out[∘]=
        {{1, 11}, {2}}

In[∘]:= enumeratedAces = Tuples /@ twoCards;

In[∘]:= enumeratedTotal = Total[#, {2}] & /@ enumeratedAces;

In[∘]:= under21 = Select[#, # ≤ 21 &] & /@ enumeratedTotal;

In[∘]:= finalScore = Max /@ under21;

In[∘]:= Length[finalScore]
Out[∘]=
        325

In[∘]:= Count[finalScore, {}]
Out[∘]=
        0

In[∘]:= Histogram[finalScore]
Out[∘]=
```



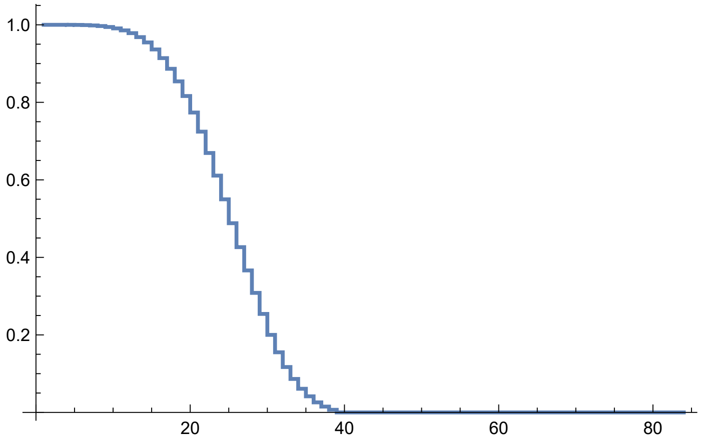
```
In[•]:= drawCards[deck_, n_] := Total[#, {2}] & /@ (Tuples /@ Subsets[deck, {n}])

        bustCards[cards_, limit_] := Max /@ (Select[#, # ≤ limit &] & /@ cards)

        regFail [cards_, minNeeded_, limit_] :=
         Count[bustCards[cards, limit], x_ /; -1 < x < minNeeded] / Length[cards]

        regSuccess [cards_, minNeeded_, limit_] :=
         Count[bustCards[cards, limit], x_ /; minNeeded ≤ x < limit] / Length[cards]

        critFail [cards_, minNeeded_, limit_] :=
         Count[bustCards[cards, limit], x_ /; x < 0] / Length[cards]

        critSuccess[cards_, minNeeded_, limit_] :=
         Count[bustCards[cards, limit], limit] / Length[cards]

        successVector[cards_, minNeeded_, limit_] :=
         N[#, 3] & /@ {minNeeded - 1,
           1 - ((minNeeded - 2) * 0.05),
           minNeeded,
           regSuccess[cards, minNeeded, limit] + critSuccess[cards, minNeeded, limit],
           regFail[cards, minNeeded, limit] + critFail[cards, minNeeded, limit],
           critSuccess[cards, minNeeded, limit],
           critFail[cards, minNeeded, limit],
           regSuccess[cards, minNeeded, limit],
           regFail[cards, minNeeded, limit]
          }

        bustOdds [list_] := N[Count[list, -∞] / Length[list], 3]

In[•]:= drawTwo = drawCards[myDeck, 2];
        drawThree = drawCards[myDeck, 3];
        drawFour = drawCards[myDeck, 4];
        drawFive = drawCards[myDeck, 5];
        drawSix = drawCards[myDeck, 6];

In[•]:= headings = {"Min. Roll", "Roll Success", "Min. Needed", "Tot. Success",
           "Tot. Fail", "Crit. Success", "Crit. Fail", "Reg. Success", "Reg. Fail"};
```

```
In[•]:= Grid[Prepend[Table[successVector[drawTwo, n, 21], {n, 2, 21}], headings],
         {Background → {None, {{White, Lighter[Blend[{Blue, Green}], 0.8]}}},
          Dividers → {{Gray, {LightGray}, Gray}}, Frame → True, Spacings → {2, {2, {0.7}, 2}}}]
```

Out[•]=

Min. Roll	Roll Success	Min. Needed	Tot. Success	Tot. Fail	Crit. Success	Crit. Fail	Reg. Success	Reg. Fail
1.00	1.	2.00	1.00	0	0.0483	0	0.952	0
2.00	0.95	3.00	1.00	0	0.0483	0	0.952	0
3.00	0.9	4.00	1.00	0	0.0483	0	0.952	0
4.00	0.85	5.00	0.995	0.00452	0.0483	0	0.947	0.00452
5.00	0.8	6.00	0.983	0.0166	0.0483	0	0.935	0.0166
6.00	0.75	7.00	0.967	0.0332	0.0483	0	0.919	0.0332
7.00	0.7	8.00	0.943	0.0573	0.0483	0	0.894	0.0573
8.00	0.65	9.00	0.914	0.0860	0.0483	0	0.866	0.0860
9.00	0.6	10.0	0.878	0.122	0.0483	0	0.830	0.122
10.0	0.55	11.0	0.837	0.163	0.0483	0	0.789	0.163
11.0	0.5	12.0	0.789	0.211	0.0483	0	0.741	0.211
12.0	0.45	13.0	0.695	0.305	0.0483	0	0.647	0.305
13.0	0.4	14.0	0.599	0.401	0.0483	0	0.551	0.401
14.0	0.35	15.0	0.510	0.490	0.0483	0	0.462	0.490
15.0	0.3	16.0	0.425	0.575	0.0483	0	0.377	0.575
16.0	0.25	17.0	0.348	0.652	0.0483	0	0.300	0.652
17.0	0.2	18.0	0.276	0.724	0.0483	0	0.228	0.724
18.0	0.15	19.0	0.211	0.789	0.0483	0	0.163	0.789
19.0	0.1	20.0	0.151	0.849	0.0483	0	0.103	0.849
20.0	0.05	21.0	0.0483	0.952	0.0483	0	0	0.952

```
In[∘]:= ListStepPlot[Table[bustOdds[bustCards[drawFour, n]], {n, 2, 84}]]
Out[∘]=
```



```
In[∘]:= Table[n, {n, 2, 22}]
Out[∘]=
        {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22}

In[∘]:= drawSeven = drawCards[myDeck, 7];

In[∘]:= bustOdds [list_] := N[Count[list, -∞] / Length[list]]

In[∘]:= bustOdds[drawSeven]
Out[∘]=
        0.999088

In[∘]:= N[399 / 400]
Out[∘]=
        0.9975
```

Odds of busting on 21 with 7 drawn cards is 0.999088, or about 1/1000. For comparison, odds of NOT getting 20&20 with a disadvantaged roll is 0.9975.

What about when you start to hit?

If you score less than a 12 on your first two cards, you should automatically hit, as it is impossible to bust. This needs to be taken into account, as what we really want is an odds table for the game state where the player has some chance of failing.

The way I go about doing this is a little different. I want to draw every *permutation* of three cards from the deck. I will sum the first two to get my starting number, and use the third number to calculate the value after hitting. I had previously used Mathematica's "Subsets" function to draw my cards, but although that generates all possible sets, it does so assuming order does not matter. In this case, order does matter, because the set {A, 4, K} for example denotes hitting on a 15 and drawing a king, where as {A,K,4} denotes hitting on a 21 and drawing a 4. I don't need all 6 permutations - {A,4,K} is equivalent to {4,A,K}, but it is less work to unnecessarily include them and make use of existing functions.

```
In[∘]:= threeCards = MapIndexed[myDeck⟦#⟧ &, Permutations[Range[52], {3}], {2}]
Out[∘]=
{{{1, 11}, {2}, {3}}, {{1, 11}, {2}, {4}}, {{1, 11}, {2}, {5}}, {{1, 11}, {2}, {6}}, {{1, 11}, {2}, {7}},
 {{1, 11}, {2}, {8}}, {{1, 11}, {2}, {9}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}},
 {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {1, 11}}, {{1, 11}, {2}, {2}}, {{1, 11}, {2}, {3}},
 {{1, 11}, {2}, {4}}, {{1, 11}, {2}, {5}}, {{1, 11}, {2}, {6}}, {{1, 11}, {2}, {7}}, {{1, 11}, {2}, {8}},
 {{1, 11}, {2}, {9}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}},
 {{1, 11}, {2}, {1, 11}}, {{1, 11}, {2}, {2}}, {{1, 11}, {2}, {3}}, {{1, 11}, {2}, {4}}, {{1, 11}, {2}, {5}},
 {{1, 11}, {2}, {6}}, {{1, 11}, {2}, {7}}, {{1, 11}, {2}, {8}}, … 132 537 …, {{10}, {10}, {7}},
 {{10}, {10}, {8}}, {{10}, {10}, {9}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {10}},
 {{10}, {10}, {10}}, {{10}, {10}, {1, 11}}, {{10}, {10}, {2}}, {{10}, {10}, {3}}, {{10}, {10}, {4}},
 {{10}, {10}, {5}}, {{10}, {10}, {6}}, {{10}, {10}, {7}}, {{10}, {10}, {8}}, {{10}, {10}, {9}},
 {{10}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {1, 11}},
 {{10}, {10}, {2}}, {{10}, {10}, {3}}, {{10}, {10}, {4}}, {{10}, {10}, {5}}, {{10}, {10}, {6}},
 {{10}, {10}, {7}}, {{10}, {10}, {8}}, {{10}, {10}, {9}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}}
Full expression not available ( original memory size: 35.7 MB )
```

These next two lines are rather long and complicated. I have included an itemized breakdown at the end of this notebook. “finalTotal” denotes the final value after hitting once. “firstTwoTotal” is the total of the first two cards drawn - what is being hit on.

```
finalTotal = bustCards[Total[#, {2}] & /@ (Tuples /@ Transpose[
       {Total[#, {2}] & /@ (Tuples /@ threeCards⟦All, ;; 2⟧), threeCards⟦All, 3⟧}]), 21];
firstTwoTotal = bustCards[Total[#, {2}] & /@ (Tuples /@ threeCards⟦All, ;; 2⟧), 21];
```

bustOddsStartAtN: What is the likelihood of busting (going over 21) if you hit an initial hand with sums to “N”?
afterHitAvgValue: Assuming you don’t bust, what average value will you be left with after hitting once?

```
bustOddsStartAtN[firstTotal_, finalTotal_, beforeHit_] :=
 bustOdds[Pick[finalTotal, Map[# == beforeHit &, firstTotal]]]
afterHitAvgValue[firstTotal_, finalTotal_, beforeHit_] :=
 N[
  Mean[
   Select[
    Pick[finalTotal, Map[# == beforeHit &, firstTotal]], # > 0 &]
  ], 3
 ]
```

```
In[∘]:= Integer[Commonest[Pick[finalTotal, Map[# == 7 &, firstTwoTotal]]]]
Out[∘]=
Integer[{17}]
```

```
In[∘]:= FromDigits[{17}]
Out[∘]=
17
```

```
In[•]:= Tally[Pick[finalTotal, Map[# == 7 &, firstTwoTotal]]]
Out[•]=
        {{18, 256}, {10, 224}, {11, 224}, {13, 256},
         {14, 256}, {15, 256}, {16, 256}, {17, 1024}, {9, 224}, {12, 224}}

        1024 /
Out[•]=
```

$$\frac{8}{25}$$

In[•]:= $N\left[\frac{8}{25}\right]$

```
Out[•]=
        0.32
```

In[•]:=

```
Grid[
 Prepend[
  Transpose[
   {Range[4, 21],
    Table[bustOddsStartAtN[firstTwoTotal, finalTotal, n], {n, 4, 21}],
    Table[afterHitAvgValue[firstTwoTotal, finalTotal, n], {n, 4, 21}]}
  ], {"Start Tot.", "Odds of Bust", "Avg. Non-Bust Tot."}
 ],
 {Background → {None, {{White, Lighter[Blend[{Blue, Green}], 0.8]}}},
  Dividers → {{Gray, {LightGray}, Gray}}, Frame → True, Spacings → {2, {2, {0.7}, 2}}}
]
```

Out[•]=

Start Tot.	Odds of Bust	Avg. Non-Bust Tot.
4	0	11.5
5	0	12.5
6	0	13.5
7	0	14.5
8	0	15.4
9	0	16.4
10	0	17.4
11	0	17.6
12	0.294	16.9
13	0.338	17.2
14	0.399	17.5
15	0.460	17.8
16	0.506	18.1
17	0.567	18.4
18	0.619	18.5
19	0.672	18.4
20	0.812	18.6
21	0	17.6

Some Strange Results

In[•]:=

```
Tally[Pick[threeCards[[All, ;; 2]], Map[# == 20 &, firstTwoTotal]]]
```

Out[•]=

```
{{{{1, 11}, {9}}, 800}, {{{9}, {1, 11}}, 800}, {{{10}, {10}}, 12 000}}
```

In[•]:=

$$\frac{12\,000 * (50 - 4)}{(12\,000 + 800 + 800) * 50}$$

Out[•]=

$$\frac{69}{85}$$

In[∘]:=

$$\mathrm{N}\left[\frac{69}{85}\right]$$

Out[∘]=

```
0.811765
```

In[∘]:=

```
N[46 / 50]
```

Out[∘]=

```
0.92
```

This is a very interesting result! If you have a total of 20, and you hit, you only have an 81% chance of busting. Online tables show 92% odds, which at a glance seems more reasonable. What gives? Well, if you have A/9 combo, you have 20 but will never bust on the next turn (There are 1600 permutations of this). If you have any 10/10 combo (12,000 permutations of this), you will bust if your next card isn't A. You drew two cards already, so 50 remain in the deck.

Consider the 10/10. What are your odds of busting? Well, 46/50! That is our 92% from the internet. And we already know you have 0% odds from A/9.

In[∘]:=

$$\frac{1600}{12\,000+1600}*0+\frac{12\,000}{12\,000+1600}*0.92$$

Out[∘]=

```
0.811765
```

In[∘]:=

$$\frac{1600}{12\,000+1600}$$

Out[∘]=

$$\frac{2}{17}$$

In[∘]:=

$$\mathrm{N}\left[\frac{2}{17}\right]$$

Out[∘]=

```
0.117647
```

In[∘]:=

$$\mathrm{N}\left[\frac{2}{17}\right]$$

Out[∘]=

```
0.117647
```

In[∘]:=

$$\mathrm{N}\left[\frac{2}{15}\right]$$

Out[∘]=

```
0.133333
```

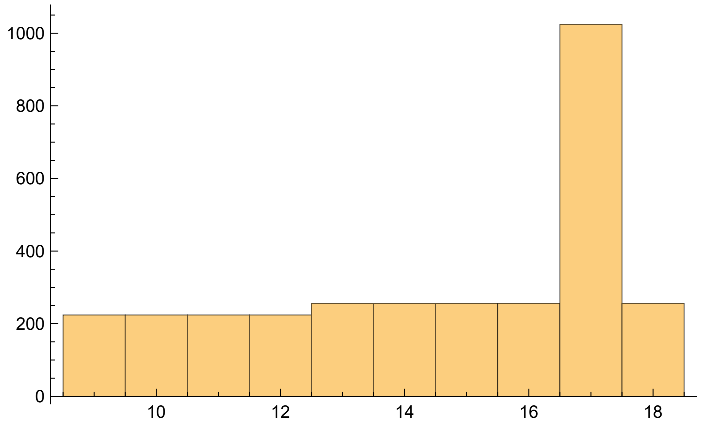
Rule of thumb

After you hit, you have a roughly 1/3 chance of going up by 10. Otherwise, your odds are evenly distributed.

For example, below is a histogram of what results from hitting on a 7. You have 1024 ways to end up at 17, versus 3200 total outcomes. Further, the histogram shows that your other outcomes are roughly evenly distributed.

```
In[∘]:= Histogram[
         Pick[finalTotal, Map[# == 7 &, firstTwoTotal]]]
        Length@Pick[finalTotal, Map[# == 7 &, firstTwoTotal]]
        N[1024 / 3200]
```

Out[∘]=



Out[∘]=

```
3200
```

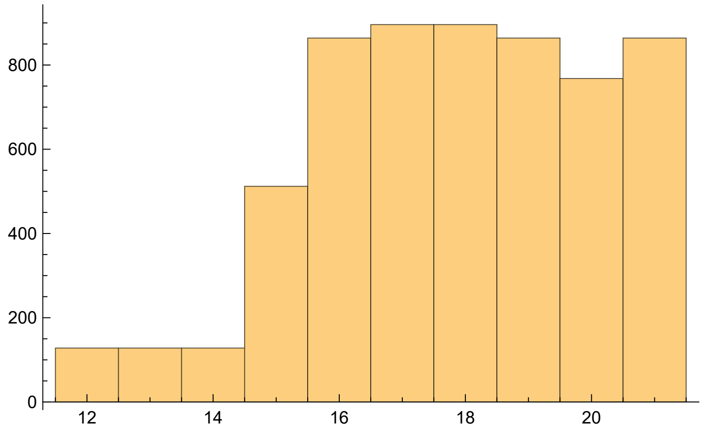
Out[∘]=

```
0.32
```

How to represent this for higher hitting at higher starting points can be a little ambiguous. For example, look at hitting on a 15 , shown below. What I’ve done throughout this notebook is “wrap” all values greater than 21 down to $-\infty$, which doesn’t show up on the histogram. Indeed, the most common outcome is a bust, which will occur 46% of the time.

```
In[∘]:= Histogram[
         Pick[finalTotal, Map[# == 15 &, firstTwoTotal]]]
```

Out[∘]=



Let’s extend this analysis to all starting values. The function below tells me what fraction of the the final outcomes are equal to the most common outcome.

```
In[•]:= modeFrac[number_] :=
         N[
          (Count[#, FromDigits[Commonest[#]]] &@
              Pick[finalTotal, Map[# == number &, firstTwoTotal]]) /
           (Length[Pick[finalTotal, Map[# == number &, firstTwoTotal]]]),
          3
         ]
```

In the table below...
Start Tot.: The total of our first two cards.
Aft. Hit Mode: The most common value occurring after drawing an additional card.
Odds of Aft. Hit Mode: What fraction of the final outcomes “Aft. Hit Mode” represents.

Up until 12, the point at which we are no longer guaranteed to be safe if we hit, we can expect to go up in value by 10, and we can expect that to occur about 1/3 of the time. After this point, we can expect to bust *at best* about 1/3 of the time, which gets worse as we go up in value (barring some funny business with valuing Aces at 1 or 11).

In[∘]:=
```
Grid[
 Prepend[
  Transpose@{Range[4, 21],
    Table[Commonest[Pick[finalTotal, Map[# == n &, firstTwoTotal]]], {n, 4, 21}],
    Table[modeFrac[n], {n, 4, 21}]},
  {"Start Tot.", "Aft. Hit Mode", "Odds. of Aft. Hit Mode"}
 ],
 {Background → {None, {{White, Lighter[Blend[{Blue, Green}], 0.8]}}},
  Dividers → {{Gray, {LightGray}, Gray}}, Frame → True, Spacings → {2, {2, {0.7}, 2}}}
]
```

Out[∘]=

Start Tot.	Aft. Hit Mode	Odds. of Aft. Hit Mode
4	{14}	0.320
5	{15}	0.320
6	{16}	0.320
7	{17}	0.320
8	{18}	0.320
9	{19}	0.320
10	{20}	0.320
11	{21}	0.320
12	{−∞}	0.294
13	{−∞}	0.338
14	{−∞}	0.399
15	{−∞}	0.460
16	{−∞}	0.506
17	{−∞}	0.567
18	{−∞}	0.619
19	{−∞}	0.672
20	{−∞}	0.812
21	{21}	0.300

It looks like we can expect about a 16, if we in

In[∘]:=
```
{N[Length@# / Length@firstTwoTotal], N[Mean@#]} &@
 Pick[finalTotal, Map[4 ≤ # ≤ 11 &, firstTwoTotal]]
```

Out[∘]=
```
{0.211161, 15.9571}
```

So, what’s the verdict here? Recall that our definition of success is “Get at least this number, but not over 21”. We have about a 70% chance being at or above 12 on our first two cards, putting the players in the “danger zone” of possibly busting. The 30% of the time we aren’t in the *Danger Zone*, expect the players to hit to about a 16. At this point, they will have a 50% chance of going bust on the next hit. If they don’t bust, expect an 18.

This is already complicated, and unfortunately only get’s worse... The longest possible hand you can have, and still be safe, is 8 cards long - {A, A, A, A, 2, 2, 2, 2} for a total of 12, just entering the *Danger Zone*. But your odds of busting noticeably higher than in the above table, 36% (44 cards remain, 12 of which have the value “10” which will bust you). So, you could continue through all these other hands, logging where they spit you out and compiling that into a final table with odds of 12-21 after perfect play.

```
In[◦]:= 1 / Binomial[52, 8]
Out[◦]=
```

$$\frac{1}{752\,538\,150}$$

```
In[◦]:=
```

$$N\left[\frac{1}{752\,538\,150}\right]$$

```
Out[◦]=
```

$$1.32884\times 10^{-9}$$

Monte Carlo

So, rather than directly calculating all of this, let’s do a monte carlo simulation.

Draw 8 cards at random, as 8 is the maximum number we can draw and just hit a value of 12.

```
In[◦]:= draw = RandomChoice[myDeck, 8]
Out[◦]=
{{10}, {4}, {5}, {1, 11}, {10}, {9}, {9}, {5}}
```

Let’s use a fun edge case instead. A player wouldn’t hit a 21, but technically they could, and in this case that wouldn’t bust them.

```
In[◦]:= draw = {{1, 11}, {10}, {2}, {5}}
Out[◦]=
{{1, 11}, {10}, {2}, {5}}
```

This is the pattern of numbers we will sum over. The first list assumes we count the Ace as a 1, and the second assumes we count it as an 11.

```
In[◦]:= Tuples@draw
Out[◦]=
{{1, 10, 2, 5}, {11, 10, 2, 5}}
```

Calculate the cumulative sum - the result of drawing one more card.

```
In[◦]:= Accumulate /@ Tuples@draw
Out[◦]=
{{1, 11, 13, 18}, {11, 21, 23, 28}}
```

We need to find the *soonest possible location* where we meet or surpass 12, as we have entered the

Danger Zone where we may start busting. The below function gives all locations of numbers between (inclusive) 12 and 21, in {list, location in list} ordered pairs (Mathematica starts counting at one).

```
In[∘]:= Position[Accumulate /@ Tuples@draw, _? (12 ≤ # ≤ 21 &)]
Out[∘]=
        {{1, 3}, {1, 4}, {2, 2}}
```

Choose ordered pair with the lowest second value, which corresponds to the fewest number of cards needed to enter the *Danger Zone*.

```
In[∘]:= MinimalBy[Position[Accumulate /@ Tuples@draw, _? (12 ≤ # ≤ 21 &)], Last]
Out[∘]=
        {{2, 2}}
```

Get the value where we have entered the *Danger Zone*. This is the final result of a single simulated run!

```
In[∘]:= Extract[Accumulate /@ Tuples@draw,
         MinimalBy[Position[Accumulate /@ Tuples@draw, _? (12 ≤ # ≤ 21 &)], Last]⟦1⟧]
Out[∘]=
        21
```

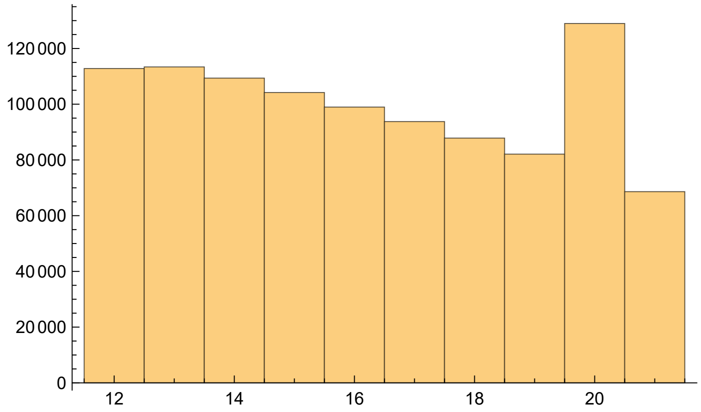
```
In[∘]:= monteFunc[draw_] := Extract[Accumulate /@ Tuples@draw,
          MinimalBy[Position[Accumulate /@ Tuples@draw, _? (12 ≤ # ≤ 21 &)], Last]⟦1⟧]
```

Here is how we would do a single run with a random draw of 8 cards.

```
In[∘]:= monteFunc[RandomChoice[myDeck, 8]]
Out[∘]=
        20
```

And here is how we generate our final data set.

```
In[∘]:= monteData = Table[monteFunc[RandomChoice[myDeck, 8]], {n, 1, 1000000}];
In[∘]:= Histogram[monteData]
Out[∘]=
```



```
In[∘]:= 13 010 / 100 000
Out[∘]=
         1301
        ──────
        10 000
```

```
In[∘]:= N[ 1301  ]
          ──────
          10 000
Out[∘]=
        0.1301
```

```
In[∘]:= fracMatch[value_] := N[Count[monteData, _?(# ≥ value &)] / Length[monteData], 3]
```

```
In[∘]:= Grid[
         Prepend[
          Transpose@{Range[12, 21],
            Table[fracMatch[n], {n, 12, 21}]},
          {"Get at least", "Odds"}
         ],
         {Background → {None, {{White, Lighter[Blend[{Blue, Green}], 0.8]}}},
          Dividers → {{Gray, {LightGray}, Gray}}, Frame → True, Spacings → {2, {2, {0.7}, 2}}}
        ]
```

Out[∘]=

Get at least	Odds
12	1.00
13	0.887
14	0.774
15	0.664
16	0.560
17	0.461
18	0.368
19	0.280
20	0.198
21	0.0686

And there we have it! Our (approximate) riskless play. 100% chance of getting at least a 12, about a 50% chance of getting at least a 16 or 17, and a 7% chance of getting a perfect 21. Odds of blackjack, about 5%.

If a DM asks for a 16, you have a 56% chance of getting it right off the bat. But the other 44% of the time, you're gonna have to hit at least once, busting you about 50% of the time after that first hit.

So... Where does that leave us? I'd use this above-table to (without telling the players), set baseline assuming perfectly safe play. This is basically your dice roll. On a soft failure (not busting), they don't get what they want, but it isn't awful. If they do push their luck, hit, and bust, something pretty bad happens.

```
Out[∘]=
{13, 17, 13, 13, 13, 15, 15, 19, 20, 16, 17, 14, 19, 16, 13, 16, 14, 20, 19, 12, 19, 13, 13, 17, 16, 20, 19, 16,
 18, 15, 19, 13, 12, 18, 13, 18, 13, 12, 15, 16, 17, 16, 12, 17, 20, 15, 20, 13, 20, 20, 20, 16, 17, 16, 13, 19,
 12, 18, 18, 18, 13, 15, 21, 14, 20, 14, 12, 21, 14, 13, 20, 18, 12, 12, 16, 20, 17, 16, 18, 14, 13, 17, 14, 13,
 18, 17, 20, 18, 13, 15, 17, 16, 13, 12, 13, 17, 18, 14, 13, 15, 20, 12, 14, 14, 15, 18, 15, 18, 12, 13, 12, 21,
 12, 17, 13, 14, 13, 16, 16, 16, 15, 13, 20, 21, 12, 15, 12, 15, 13, 18, 14, 15, 15, 18, 16, 16, 21, 17, 20, 20,
 20, 18, 20, 13, 20, 14, 18, 17, 12, 18, 15, 13, 12, 14, 21, 14, 13, 19, 14, 14, 20, 21, 21, 21, 13, 20, 13, 13,
 14, 16, 13, 16, 17, 16, 17, 20, 13, 16, 14, ··· 99 642 ···, 13, 15, 19, 15, 16, 19, 21, 12, 12, 13, 16, 14, 12,
 13, 16, 16, 18, 13, 13, 19, 14, 14, 16, 15, 17, 16, 12, 17, 21, 13, 18, 14, 16, 15, 12, 21, 19, 19, 19, 19, 17,
 20, 17, 12, 15, 15, 18, 20, 20, 14, 20, 21, 17, 14, 16, 19, 15, 17, 13, 16, 20, 18, 20, 20, 14, 20, 18, 18, 18,
 20, 20, 15, 14, 13, 17, 12, 14, 15, 20, 16, 17, 14, 14, 14, 20, 16, 20, 20, 12, 18, 15, 12, 14, 13, 14, 15, 20,
 20, 15, 15, 18, 16, 16, 20, 13, 17, 13, 19, 14, 19, 21, 17, 12, 12, 14, 15, 14, 12, 15, 17, 17, 19, 18, 20, 15,
 20, 20, 18, 18, 21, 17, 17, 18, 17, 13, 19, 13, 17, 12, 20, 20, 16, 19, 18, 19, 13, 20, 15, 21, 14, 15, 15,
 14, 15, 15, 16, 16, 13, 16, 17, 12, 14, 15, 12, 13, 14, 12, 12, 15, 15, 18, 16, 21, 19, 14, 14, 21, 12, 16}
Full expression not available ( original memory size: 2.4 MB )
```

```
In[∘]:= Flatten[Accumulate /@ Tuples@draw]
Out[∘]=
{3, 6, 7, 17, 27, 37, 47, 51, 3, 6, 17, 27, 37, 47, 57, 61}

In[∘]:= Position[Accumulate /@ Tuples@draw, _?(12 ≤ # ≤ 21 &)]
Out[∘]=
{{1, 3}, {1, 4}, {2, 2}}

In[∘]:= First@MinimalBy[Position[Accumulate /@ Tuples@draw, _?(12 ≤ # ≤ 21 &)], Last]
Out[∘]=
{2, 2}

In[∘]:= Extract[Accumulate /@ Tuples@draw,
         MinimalBy[Position[Accumulate /@ Tuples@draw, _?(12 ≤ # ≤ 21 &)], Last]〚1〛]
Out[∘]=
21

In[∘]:= (Accumulate /@ Tuples@draw)〚2, 2〛
Out[∘]=
21

In[∘]:= {{2, 2}}〚0〛
Out[∘]=
List
```

```
In[∘]:= Subsets[myDeck, {3}]
Out[∘]=
{{{1, 11}, {2}, {3}}, {{1, 11}, {2}, {4}}, {{1, 11}, {2}, {5}}, {{1, 11}, {2}, {6}}, {{1, 11}, {2}, {7}},
 {{1, 11}, {2}, {8}}, {{1, 11}, {2}, {9}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}},
 {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {1, 11}}, {{1, 11}, {2}, {2}}, {{1, 11}, {2}, {3}}, {{1, 11}, {2}, {4}},
 {{1, 11}, {2}, {5}}, {{1, 11}, {2}, {6}}, {{1, 11}, {2}, {7}}, {{1, 11}, {2}, {8}}, {{1, 11}, {2}, {9}},
 {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {10}}, {{1, 11}, {2}, {1, 11}},
 {{1, 11}, {2}, {2}}, {{1, 11}, {2}, {3}}, {{1, 11}, {2}, {4}}, {{1, 11}, {2}, {5}}, {{1, 11}, {2}, {6}},
 {{1, 11}, {2}, {7}}, {{1, 11}, {2}, {8}}, … 22037 …, {{7}, {8}, {10}}, {{7}, {9}, {10}}, {{7}, {9}, {10}},
 {{7}, {9}, {10}}, {{7}, {9}, {10}}, {{7}, {10}, {10}}, {{7}, {10}, {10}}, {{7}, {10}, {10}}, {{7}, {10}, {10}},
 {{7}, {10}, {10}}, {{7}, {10}, {10}}, {{8}, {9}, {10}}, {{8}, {9}, {10}}, {{8}, {9}, {10}}, {{8}, {9}, {10}},
 {{8}, {10}, {10}}, {{8}, {10}, {10}}, {{8}, {10}, {10}}, {{8}, {10}, {10}}, {{8}, {10}, {10}}, {{8}, {10}, {10}},
 {{9}, {10}, {10}}, {{9}, {10}, {10}}, {{9}, {10}, {10}}, {{9}, {10}, {10}}, {{9}, {10}, {10}},
 {{9}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}, {{10}, {10}, {10}}}
Full expression not available (original memory size: 6 MB)

In[∘]:= Accumulate[%139]
••• Accumulate: Objects of unequal length in {{2}, {7}, {8}, {1, 11}, {5}, {9}, {4}, {1, 11}} cannot be added.
Out[∘]=
Accumulate[{{2}, {7}, {8}, {1, 11}, {5}, {9}, {4}, {1, 11}}]

In[∘]:= Tuples /@ %
••• Tuples: Nonatomic expression expected at position {1, 1} in Tuples[{2}]. ⓘ
••• Tuples: Nonatomic expression expected at position {1, 1} in Tuples[{7}]. ⓘ
••• Tuples: Nonatomic expression expected at position {1, 1} in Tuples[{8}]. ⓘ
••• General: Further output of Tuples::normal will be suppressed during this calculation. ⓘ
Out[∘]=
{Tuples[{2}], Tuples[{7}], Tuples[{8}], Tuples[{1, 11}],
 Tuples[{5}], Tuples[{9}], Tuples[{4}], Tuples[{1, 11}]}

In[∘]:= 16 / 44
Out[∘]=
```

$$\frac{4}{11}$$

```
In[∘]:= N[4/11]
Out[∘]=
0.363636
```

```
Length@final
```

Some of that got a little complicated, so I copy the parts where I go step-by-step here

```
In[∘]:= threeCards⟦All, ;; 2⟧
Out[∘]=
{{{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}},
 {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}},
 {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}},
 {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}},
 {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}},
 {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}},
 {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, {{1, 11}, {2}}, … 132 515 …,
 {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}},
 {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}},
 {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}},
 {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}},
 {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}},
 {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}}
Full expression not available (original memory size: 25.9 MB)
```

```
In[∘]:= Total[#, {2}] & /@ (Tuples /@ threeCards⟦All, ;; 2⟧)
Out[∘]=
{{3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13},
 {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13},
 {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13},
 {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13}, {3, 13},
 {3, 13}, {3, 13}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14},
 {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14},
 {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14},
 {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, {4, 14}, … 132 422 …, {20}, {20}, {20}, {20},
 {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20},
 {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20},
 {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20},
 {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20},
 {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}, {20}}
Full expression not available (original memory size: 23.3 MB)
```

```
In[∘]:= Transpose[{Total[#, {2}] & /@ (Tuples /@ threeCards⟦All, ;; 2⟧), threeCards⟦All, 3⟧}]
Out[∘]=
{{{3, 13}, {3}}, {{3, 13}, {4}}, {{3, 13}, {5}}, {{3, 13}, {6}}, {{3, 13}, {7}}, {{3, 13}, {8}},
 {{3, 13}, {9}}, {{3, 13}, {10}}, {{3, 13}, {10}}, {{3, 13}, {10}}, {{3, 13}, {10}}, {{3, 13}, {1, 11}},
 {{3, 13}, {2}}, {{3, 13}, {3}}, {{3, 13}, {4}}, {{3, 13}, {5}}, {{3, 13}, {6}}, {{3, 13}, {7}},
 {{3, 13}, {8}}, {{3, 13}, {9}}, {{3, 13}, {10}}, {{3, 13}, {10}}, {{3, 13}, {10}}, {{3, 13}, {10}},
 {{3, 13}, {1, 11}}, {{3, 13}, {2}}, {{3, 13}, {3}}, {{3, 13}, {4}}, {{3, 13}, {5}}, {{3, 13}, {6}},
 {{3, 13}, {7}}, {{3, 13}, {8}}, {{3, 13}, {9}}, {{3, 13}, {10}}, {{3, 13}, {10}}, {{3, 13}, {10}},
 {{3, 13}, {10}}, {{3, 13}, {1, 11}}, {{3, 13}, {2}}, {{3, 13}, {3}}, {{3, 13}, {4}}, {{3, 13}, {5}},
 {{3, 13}, {6}}, … 132 515 …, {{20}, {9}}, {{20}, {10}}, {{20}, {10}}, {{20}, {10}}, {{20}, {10}},
 {{20}, {1, 11}}, {{20}, {2}}, {{20}, {3}}, {{20}, {4}}, {{20}, {5}}, {{20}, {6}}, {{20}, {7}},
 {{20}, {8}}, {{20}, {9}}, {{20}, {10}}, {{20}, {10}}, {{20}, {10}}, {{20}, {10}}, {{20}, {1, 11}},
 {{20}, {2}}, {{20}, {3}}, {{20}, {4}}, {{20}, {5}}, {{20}, {6}}, {{20}, {7}}, {{20}, {8}}, {{20}, {9}},
 {{20}, {10}}, {{20}, {10}}, {{20}, {10}}, {{20}, {10}}, {{20}, {1, 11}}, {{20}, {2}}, {{20}, {3}},
 {{20}, {4}}, {{20}, {5}}, {{20}, {6}}, {{20}, {7}}, {{20}, {8}}, {{20}, {9}}, {{20}, {10}}, {{20}, {10}}}
Full expression not available (original memory size: 39.5 MB)
```

In[∘]:=
```
Tuples /@
 Transpose[{Total[#, {2}] & /@ (Tuples /@ threeCards⟦All, ;; 2⟧), threeCards⟦All, 3⟧}]
```

Out[∘]=
```
{{{3, 3}, {13, 3}}, {{3, 4}, {13, 4}}, {{3, 5}, {13, 5}}, {{3, 6}, {13, 6}}, {{3, 7}, {13, 7}}, {{3, 8}, {13, 8}},
 {{3, 9}, {13, 9}}, {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}},
 {{3, 1}, {3, 11}, {13, 1}, {13, 11}}, {{3, 2}, {13, 2}}, {{3, 3}, {13, 3}}, {{3, 4}, {13, 4}}, {{3, 5}, {13, 5}},
 {{3, 6}, {13, 6}}, {{3, 7}, {13, 7}}, {{3, 8}, {13, 8}}, {{3, 9}, {13, 9}}, {{3, 10}, {13, 10}},
 {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}}, {{3, 1}, {3, 11}, {13, 1}, {13, 11}},
 {{3, 2}, {13, 2}}, {{3, 3}, {13, 3}}, {{3, 4}, {13, 4}}, {{3, 5}, {13, 5}}, {{3, 6}, {13, 6}},
 {{3, 7}, {13, 7}}, {{3, 8}, {13, 8}}, {{3, 9}, {13, 9}}, {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}},
 {{3, 10}, {13, 10}}, {{3, 10}, {13, 10}}, {{3, 1}, {3, 11}, {13, 1}, {13, 11}}, {{3, 2}, {13, 2}},
 {{3, 3}, {13, 3}}, ··· 132 520 ···, {{20, 10}}, {{20, 10}}, {{20, 10}}, {{20, 1}, {20, 11}}, {{20, 2}},
 {{20, 3}}, {{20, 4}}, {{20, 5}}, {{20, 6}}, {{20, 7}}, {{20, 8}}, {{20, 9}}, {{20, 10}}, {{20, 10}},
 {{20, 10}}, {{20, 10}}, {{20, 1}, {20, 11}}, {{20, 2}}, {{20, 3}}, {{20, 4}}, {{20, 5}}, {{20, 6}}, {{20, 7}},
 {{20, 8}}, {{20, 9}}, {{20, 10}}, {{20, 10}}, {{20, 10}}, {{20, 10}}, {{20, 1}, {20, 11}}, {{20, 2}},
 {{20, 3}}, {{20, 4}}, {{20, 5}}, {{20, 6}}, {{20, 7}}, {{20, 8}}, {{20, 9}}, {{20, 10}}, {{20, 10}}}
Full expression not available (original memory size: 24.4 MB)
```

In[∘]:=
```
Total[#, {2}] & /@ (Tuples /@
   Transpose[{Total[#, {2}] & /@ (Tuples /@ threeCards⟦All, ;; 2⟧), threeCards⟦All, 3⟧}])
```

Out[∘]=
```
{{6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {12, 22}, {13, 23}, {13, 23}, {13, 23}, {13, 23},
 {4, 14, 14, 24}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {12, 22}, {13, 23},
 {13, 23}, {13, 23}, {13, 23}, {4, 14, 14, 24}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20},
 {11, 21}, {12, 22}, {13, 23}, {13, 23}, {13, 23}, {13, 23}, {4, 14, 14, 24}, {5, 15}, {6, 16}, {7, 17},
 {8, 18}, {9, 19}, {10, 20}, {11, 21}, {12, 22}, {13, 23}, {13, 23}, {13, 23}, {13, 23}, {6, 16}, {8, 18},
 {9, 19}, {10, 20}, {11, 21}, {12, 22}, {13, 23}, {14, 24}, {14, 24}, {14, 24}, {14, 24}, {5, 15, 15, 25},
 {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {12, 22}, {13, 23}, {14, 24}, {14, 24}, {14, 24},
 {14, 24}, {5, 15, 15, 25}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {12, 22}, ··· 132 436 ···,
 {26}, {27}, {28}, {29}, {30}, {30}, {30}, {30}, {21, 31}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29},
 {30}, {30}, {30}, {30}, {21, 31}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {30}, {21, 31},
 {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {30}, {30}, {30}, {21, 31}, {22}, {23}, {24}, {25},
 {26}, {27}, {28}, {29}, {30}, {30}, {30}, {30}, {21, 31}, {22}, {23}, {24}, {25}, {26}, {27}, {28},
 {29}, {30}, {30}, {30}, {30}, {21, 31}, {22}, {23}, {24}, {25}, {26}, {27}, {28}, {29}, {30}, {30}}
Full expression not available (original memory size: 23.3 MB)
```

How many Ace & 9 Combinations?

In[∘]:=
```
Binomial[13, 1] * Binomial[12, 1] * Binomial[4, 1] * Binomial[4, 1]
```

Out[∘]=

2496

How many 10 & 10 combinations?

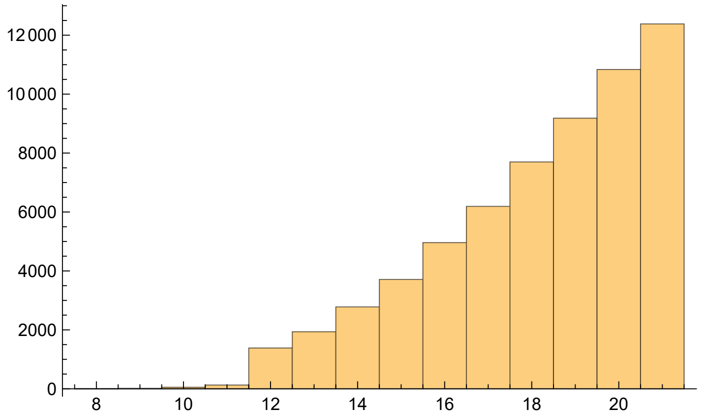
```
Binomial[13, 4] Binomial[]
```

In[∘]:=
```
Binomial[13, 1]
```

Out[∘]=

13

```
In[•]:= Binomial[4, 4]
Out[•]=
        1

In[•]:= 13 * 13
Out[•]=
        169

In[•]:= N[Mean[thirdAtN]]
Out[•]=
        -∞

          25
In[•]:= N[──]
          2
Out[•]=
        12.5

In[•]:= N[Count[thirdAtN, x_ /; x < 0] / Length[thirdAtN]]
Out[•]=
        0.811765

          27
In[•]:= N[──]
          80
Out[•]=
        0.3375

          228
In[•]:= N[───]
          775
Out[•]=
        0.294194

In[•]:= critFail[thirdAtN, 0, 21]

        ••• Select: Nonatomic expression expected at position 1 in Select[14, #1 ≤ 21 &]. ⓘ

        ••• Select: Nonatomic expression expected at position 1 in Select[15, #1 ≤ 21 &]. ⓘ

        ••• Select: Nonatomic expression expected at position 1 in Select[16, #1 ≤ 21 &]. ⓘ

        ••• General: Further output of Select::normal will be suppressed during this calculation. ⓘ
Out[•]=
        228
        ───
        775

In[•]:= Length[Pick[finalTotal, Map[# == 12 &, firstTwoTotal]]]
Out[•]=
        12 400

In[•]:= Tuples[{{"A"}, {1, 2}}]
Out[•]=
        {{A, 1}, {A, 2}}
        Join[Subsets[myDeck, {1}]
Out[∘]=
        {{{1, 11}}, {{2}}, {{3}}, {{4}}, {{5}}, {{6}}, {{7}}, {{8}}, {{9}}, {{10}}, {{10}},
         {{10}}, {{10}}, {{1, 11}}, {{2}}, {{3}}, {{4}}, {{5}}, {{6}}, {{7}}, {{8}},
         {{9}}, {{10}}, {{10}}, {{10}}, {{10}}, {{1, 11}}, {{2}}, {{3}}, {{4}}, {{5}},
         {{6}}, {{7}}, {{8}}, {{9}}, {{10}}, {{10}}, {{10}}, {{10}}, {{1, 11}}, {{2}},
         {{3}}, {{4}}, {{5}}, {{6}}, {{7}}, {{8}}, {{9}}, {{10}}, {{10}}, {{10}}, {{10}}}

In[∘]:= Length[drawFive]
Out[∘]=
        2 598 960

In[∘]:= Count[drawFive, -∞]
Out[∘]=
        2 458 988

        Subsets

In[∘]:= Histogram[drawCards[myDeck, 4]]
Out[∘]=
```



```
In[∘]:= DeleteElements[myDeck, 1 → {{1, 11}, {1, 11}}]
Out[∘]=
        {{2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {10}, {10},
         {10}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {10}, {10}, {10},
         {1, 11}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {10}, {10}, {10},
         {1, 11}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {10}, {10}, {10}}

In[∘]:= drawTwo
Out[∘]=
        {{3, 13}, {4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {11, 21},
         {11, 21}, {11, 21}, {2, 12, 12, 22}, {3, 13}, {4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18},
         {9, 19}, {10, 20}, {11, 21}, {11, 21}, {11, 21}, {11, 21}, {2, 12, 12, 22}, {3, 13},
         {4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {11, 21},
         {11, 21}, {11, 21}, {2, 12, 12, 22}, {3, 13}, {4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18},
         {9, 19}, {10, 20}, {11, 21}, {11, 21}, {11, 21}, {11, 21}, {5}, {6}, {7}, {8}, {9},
{10}, {11}, {12}, {12}, {12}, {12}, {3, 13}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {11},
{12}, {12}, {12}, {12}, {3, 13}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12},
{12}, {12}, {12}, {3, 13}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {12},
{12}, {12}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {13}, {13}, {13}, {4, 14}, {5},
{6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {13}, {13}, {13}, {4, 14}, {5}, {6}, {7},
{8}, {9}, {10}, {11}, {12}, {13}, {13}, {13}, {13}, {4, 14}, {5}, {6}, {7}, {8}, {9},
{10}, {11}, {12}, {13}, {13}, {13}, {13}, {9}, {10}, {11}, {12}, {13}, {14}, {14},
{14}, {14}, {5, 15}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {14}, {14},
{14}, {5, 15}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14},
{5, 15}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14}, {11},
{12}, {13}, {14}, {15}, {15}, {15}, {15}, {6, 16}, {7}, {8}, {9}, {10}, {11}, {12},
{13}, {14}, {15}, {15}, {15}, {15}, {6, 16}, {7}, {8}, {9}, {10}, {11}, {12}, {13},
{14}, {15}, {15}, {15}, {15}, {6, 16}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14},
{15}, {15}, {15}, {15}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {7, 17}, {8}, {9},
{10}, {11}, {12}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {7, 17}, {8}, {9}, {10},
{11}, {12}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {7, 17}, {8}, {9}, {10}, {11},
{12}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {15}, {16}, {17}, {17}, {17}, {17},
{8, 18}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {17}, {17}, {17}, {8, 18},
{9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {17}, {17}, {17}, {8, 18}, {9},
{10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {17}, {17}, {17}, {17}, {18}, {18},
{18}, {18}, {9, 19}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {18}, {18},
{18}, {9, 19}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {18}, {18},
{18}, {9, 19}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {18}, {18},
{18}, {19}, {19}, {19}, {19}, {10, 20}, {11}, {12}, {13}, {14}, {15}, {16}, {17},
{18}, {19}, {19}, {19}, {19}, {10, 20}, {11}, {12}, {13}, {14}, {15}, {16}, {17},
{18}, {19}, {19}, {19}, {19}, {10, 20}, {11}, {12}, {13}, {14}, {15}, {16}, {17},
{18}, {19}, {19}, {19}, {19}, {20}, {20}, {20}, {11, 21}, {12}, {13}, {14}, {15},
{16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12}, {13}, {14}, {15},
{16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12}, {13}, {14}, {15},
{16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {20}, {20}, {11, 21}, {12}, {13},
{14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12}, {13},
{14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12}, {13},
{14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {3, 13}, {4, 14},
{5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {11, 21}, {11, 21},
{11, 21}, {2, 12, 12, 22}, {3, 13}, {4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19},
{10, 20}, {11, 21}, {11, 21}, {11, 21}, {11, 21}, {2, 12, 12, 22}, {3, 13}, {4, 14},
{5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {11, 21}, {11, 21},
{11, 21}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {12}, {12}, {12}, {3, 13}, {4},
{5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {12}, {12}, {12}, {3, 13}, {4}, {5}, {6},
{7}, {8}, {9}, {10}, {11}, {12}, {12}, {12}, {12}, {7}, {8}, {9}, {10}, {11}, {12},
{13}, {13}, {13}, {13}, {4, 14}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13},
{13}, {13}, {13}, {4, 14}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {13},
{13}, {13}, {9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14}, {5, 15}, {6}, {7},
{8}, {9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14}, {5, 15}, {6}, {7}, {8},
{9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14}, {11}, {12}, {13}, {14}, {15},
{15}, {15}, {15}, {6, 16}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {15},
{15}, {15}, {6, 16}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {15}, {15},
{15}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {7, 17}, {8}, {9}, {10}, {11}, {12},
{13}, {14}, {15}, {16}, {16}, {16}, {16}, {7, 17}, {8}, {9}, {10}, {11}, {12}, {13},
{14}, {15}, {16}, {16}, {16}, {16}, {15}, {16}, {17}, {17}, {17}, {17}, {8, 18},
{9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {17}, {17}, {17}, {8, 18}, {9},
{10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {17}, {17}, {17}, {17}, {18}, {18},
{18}, {18}, {9, 19}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {18},
{18}, {18}, {9, 19}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {18},
{18}, {18}, {19}, {19}, {19}, {19}, {10, 20}, {11}, {12}, {13}, {14}, {15}, {16},
{17}, {18}, {19}, {19}, {19}, {19}, {10, 20}, {11}, {12}, {13}, {14}, {15}, {16},
{17}, {18}, {19}, {19}, {19}, {19}, {20}, {20}, {20}, {11, 21}, {12}, {13}, {14},
{15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12}, {13}, {14},
{15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21}, {12},
{13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {20}, {11, 21},
{12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21},
{12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21},
{12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {11, 21},
{12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {3, 13},
{4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21}, {11, 21},
{11, 21}, {11, 21}, {2, 12, 12, 22}, {3, 13}, {4, 14}, {5, 15}, {6, 16}, {7, 17},
{8, 18}, {9, 19}, {10, 20}, {11, 21}, {11, 21}, {11, 21}, {11, 21}, {5}, {6}, {7},
{8}, {9}, {10}, {11}, {12}, {12}, {12}, {12}, {3, 13}, {4}, {5}, {6}, {7}, {8}, {9},
{10}, {11}, {12}, {12}, {12}, {12}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {13},
{13}, {13}, {4, 14}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {13}, {13},
{13}, {9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14}, {5, 15}, {6}, {7}, {8},
{9}, {10}, {11}, {12}, {13}, {14}, {14}, {14}, {14}, {11}, {12}, {13}, {14}, {15},
{15}, {15}, {15}, {6, 16}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {15},
{15}, {15}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {7, 17}, {8}, {9}, {10}, {11},
{12}, {13}, {14}, {15}, {16}, {16}, {16}, {16}, {15}, {16}, {17}, {17}, {17}, {17},
{8, 18}, {9}, {10}, {11}, {12}, {13}, {14}, {15}, {16}, {17}, {17}, {17}, {17},
{17}, {18}, {18}, {18}, {18}, {9, 19}, {10}, {11}, {12}, {13}, {14}, {15}, {16},
{17}, {18}, {18}, {18}, {18}, {19}, {19}, {19}, {19}, {10, 20}, {11}, {12}, {13},
{14}, {15}, {16}, {17}, {18}, {19}, {19}, {19}, {19}, {20}, {20}, {20}, {11, 21},
{12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20}, {20}, {20},
{11, 21}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20}, {20},
{20}, {11, 21}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20},
{20}, {11, 21}, {12}, {13}, {14}, {15}, {16}, {17}, {18}, {19}, {20}, {20}, {20},
{20}, {3, 13}, {4, 14}, {5, 15}, {6, 16}, {7, 17}, {8, 18}, {9, 19}, {10, 20}, {11, 21},
{11, 21}, {11, 21}, {11, 21}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {12}, {12},
{12}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {13}, {13}, {13}, {9}, {10}, {11},
{12}, {13}, {14}, {14}, {14}, {14}, {11}, {12}, {13}, {14}, {15}, {15}, {15}, {15},
{13}, {14}, {15}, {16}, {16}, {16}, {16}, {15}, {16}, {17}, {17}, {17}, {17}, {17},
{18}, {18}, {18}, {18}, {19}, {19}, {19}, {19}, {20}, {20}, {20}, {20}, {20}, {20}}
```

```
In[•]:= Tuples[suit]
```

```
Tuples: Nonatomic expression expected at position {1, 1} in Tuples[{A, 2, 3, 4, 5, 6, 7, 8, 9, 10, ≪3≫}].
```

```
In[•]:= Tuples[Tuples[deck, 2]]
```

```
Tuples: The length of the output of
    Tuples[{{{1, 11}, {1, 11}}, {{1, 11}, {2}}, {{1, 11}, {3}}, {{1, 11}, {4}}, {{1, 11}, {5}}, {{1, 11}, {6}}, {{1, 11}, {7}}, {{1, 11}, {8}}, {{1, 11}, {9}}, {{1
        , 11}, {10}}, ≪2694≫}] should be a machine integer.
```

```
Out[•]=
Tuples[{{{1, 11}, {1, 11}}, {{1, 11}, {2}}, {{1, 11}, {3}}, {{1, 11}, {4}}, {{1, 11}, {5}}, {{1, 11}, {6}},
  {{1, 11}, {7}}, {{1, 11}, {8}}, {{1, 11}, {9}}, {{1, 11}, {10}}, {{1, 11}, {10}}, {{1, 11}, {10}},
  {{1, 11}, {10}}, … 2678 …, {{10}, {1, 11}}, {{10}, {2}}, {{10}, {3}}, {{10}, {4}}, {{10}, {5}},
  {{10}, {6}}, {{10}, {7}}, {{10}, {8}}, {{10}, {9}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}, {{10}, {10}}}]
Full expression not available (original memory size: 0.5 MB)
```

```
In[•]:= Total[{{1, 11}, {1, 11}, 10}]
Out[•]=
{12, 32}
```

```
In[•]:= Tuples[{{1, 11}, {10}}]
Out[•]=
{{1, 10}, {11, 10}}
```

```
In[•]:= List[10]
Out[•]=
{10}
```