Here goes... you actually don't need to know any fancy math for this. You just need some basic probability. But it gets tedious.

Suppose we want to know the probability of having a queen next to a king. (This is what I'll refer to as a "success".)

Define the following functions: $P(q, k, n) =$ probability of having a queen next to a king given that you have n cards left in the deck, q of which are queens, k of which are kings, and that the previous card you dealt was neither a queen nor a king.

In other words, P is the probability of success given a certain mix of cards remaining. Note that n is the number of remaining cards, not the total number of cards.

Let $Q(q, k, n) =$ probability of success given that the previous card dealt was a queen. The variables are the same as for the function P. Notice that capital Q is a function and little q is a variable.

Let $K(q, k, n) =$ probability of success given that the previous card dealt was a king. The variables are the same as for the function P.

Now the easy part: Figure out the base cases.

- $P(0, k, n) = 0$, i.e. if you have no queens left, you have a zero probability of success (since the previous card was neither a Q nor K.)
- $P(q, 0, n) = 0$, same reason as above.
- $P(q, k, 0) = 0$, (there are no cards left!)
- $P(q, k, 1) = 0$, i.e. you can't succeed if there's only one card left.
- $Q(q, 0, n) = 0$, i.e. you need at least one king remaining (but not a queen – the previous card dealt was a queen after all)
- $Q(q, k, 0) = 0$, i.e. you need at least one card in order to succeed from here.
- $K(0, k, n) = 0$, i.e. you need at least one queen remaining.
- $K(q, k, 0) = 0$, i.e. you need at least one remaining card.

Now the moderately tricky part. Define the functions recursively.

First let's do $Q(q, k, n)$. If the card you just dealt was a queen and you have n cards left, what are the ways you can succeed from here?

- Possibility 1: The next card you deal is a king. Probability of that is $\frac{k}{n}$.
- Possibility 2: The next card you deal is a queen, and then you succeed with the remaining $n-1$ cards. Probability of the next card being a queen is $\frac{q}{n}$, and the probability of success after that is $Q(q-1, k, n-1)$. Overall probability: $\frac{q}{n} \cdot Q(q-1, k, n-1)$
- Possibility 3: The next card is neither a queen nor a king. You then have $n-1$ cards left, and you succeed with the remaining $n-1$ cards. The probability of an indifferent card being next is $\frac{n-q-k}{n}$ and the probability of success after that is $P(q, k, n-1)$ (since you'd then be starting "fresh"). Overall probability: $\frac{n-q-k}{n} \cdot P(q, k, n-1)$.
- Total probability of success given that you just dealt a queen $= Q(q, k, n) = \frac{k}{n} + \frac{q}{n} \cdot Q(q-1, k, n-1) + \frac{n-q-k}{n} \cdot P(q, k, n-1)$

Yes, this is a recursive function, but all the recursive evaluations involve smaller parameter values, and the whole computation bottoms out eventually (because of the base cases).

If you've gotten this far, then you're past the worst of it. You now define $K(q, k, n)$ similarly and you get:

$K(q, k, n) = \frac{q}{n} + \frac{k}{n} \cdot K(q, k-1, n-1) + \frac{n-q-k}{n} \cdot P(q, k, n-1)$

This also follows by symmetry. (The role of the kings and queens are interchangeable.)

Finally, we define $P(q, s, n)$. Recall that this is the probability of success from a new deck of n cards, or equivalently from a deck with n cards remaining but where the previous card was neither a king nor a queen. Here are the three cases to consider:

- Possibility 1: The next card you deal is a king and then you succeed from there. Probability of that is $\frac{k}{n} \cdot K(q, k-1, n-1)$
- Possibility 2: The next card you deal is a queen and then you succeed from there. Probability of that is $\frac{q}{n} * Q(q-1, k, n-1)$
- Possibility 3: Next card is indifferent. Just like possibility 3 from earlier: $\frac{n-q-k}{n} \cdot P(q, k, n-1)$

Therefore:

$$P(q,k,n) = \tfrac{k}{n} \cdot K(q,k-1,n-1) + \tfrac{q}{n} \cdot Q(q-1,k,n-1) + \tfrac{n-q-k}{n} \cdot P(q,k,n-1)$$

Now you just write a computer program that codes up these three functions and you evaluate $P(4,4,52)$.

The one complication that comes up is that the recursion branches into a **huge** tree. Each function call results in two other function calls, and n only drops by 1 each time! I think it's basically intractable unless you shorten the computation somehow.

Here's how I fixed that:

1. I cached the values of K, Q, and P so that I never had to evaluate these for the same parameters twice. Huge, huge savings. (I suppose this is equivalent to doing the whole thing with dynamic programming, for those of you who are familiar with that.)
2. Look at the first term of P: $\frac{k}{n} \cdot K(q,k-1,n-1)$. Clearly if k is zero, there's no need to evaluate $K(q,k-1,n-1)$. You can make this optimization in every instance of a recursive call, i.e. before you recurse, check the coefficient to see if it's zero. This results in a small speedup.
3. Finally, the formulas for Q and K are symmetric. So if you fill in the cache for the value of (for instance) $Q(3,2,10)$ then you can also fill in the value for $K(2,3,10)$. This also gives you a small speedup.

Once you make these optimizations, the whole things runs really fast, and it spits out 0.486.

I realize this looks long and complicated, but it's really just a long sequence of simple ideas. Let me know if anything here is unclear.