**Combinatorics**

**Definitions**

Experiment – something which can have multiple outcomes.

Sample Space – all outcomes, and assume equally likely so far. W{S} = number of outcomes

Event Space – set of all desired outcomes. W{E} = number desired outcomes.

Have to find the sample space multiplicity W{S} and the event space multiplicity W{E}. Then the probability of the event is:



*It’s imperative, however, that all events in S be equiprobable.*

**Example**

Throw dice twice. List sample space. List event space. Probability the sum will be 8. What is the most likely sum?



What are sums?



Probabilities?



Most likely option is 7, with probability 1/6. Maybe draw probability distribution of sum to see Bell-shaped curve. P(sum of identically distributed random variables) approaches a Bell shaped curve.

**General Counting Rules**

Here are the basics:

* If a process can be broken down into a set of steps: x1∩x2∩…xn, then the number of ways to do the entire process is W{x1,x2,…,xn} = n1n2⋯nn.
* If a process can be broken down into a set of steps: x1∪x2∪…xn, then the number of ways to do the entire process is W{x1,x2,…,xn} = n1 + n2 + ⋯ + nn.

**Example**

How many outcomes when throw dice twice? That’s 6×6 = 36. How many outcomes where sum is 8? That’s 2+6, 3+5, 4+4, 5+3, 6+2 = 5. So P(8) = 5/36.

**Example**

How many 3-character passwords can one create if it must be in the form NLP where N is a single digit number, L is a letter (upper or lower case), and P is a punctuation symbol (.!?,:;)? This would be W = 10\*52\*6 = 3120.

**Example**

How many 3-character passwords can one create if it must be in the form NLP where N is a single digit number, L is a letter (upper or lower case), and P is a punctuation symbol (.!?,:;) but order doesn’t matter? This would be W = 10\*52\*6 = 3120 for a particular order, but then we can have in order NLP, NPL, PLN, PNL, LNP, LPN. So that 3120 + 3120 + 3120 + 3120 + 3120 + 3120 = 6∙3120 = 18720.

**Special Counting Rules**

Gonna do a few examples…for reference, I’m going to call a tuple, Tnp , a group of numbers of dimensionality p, like (2,1,4,3), each drawn from a set of n. If replacement is allowed, then we can also have something like (1,1,2,2). If order matters, then (1,2,3,4) is distinct from (2,1,4,3). If order doesn’t matter, then these would be identical. Note p can be larger than n when we have replacement.

**W{Tnp} w/replacement from n distinct elements (order matters)**

First I’ll look at situations where we extract p-tuples (t1, t2, t3, …, tp) from a drawer of n things, where each time we have the same n distinct elements to choose from. Note p could be larger than n, in principle. This is:



since we have n choices for the first element, n choices for the second element, …, n choices for the last element. Maybe call it:



**Example**

Say you have four different characters c1, c2, c3, c4 to choose from to create a two-character long password. How many passwords can you create? Well, sample space is of form T = (t1, t2) where t1 can equal t2 and order matters. This would be:



which is W = np = 42 = 16 socks.

**Example**

How many 6 digit passwords can you create? This is W = 610 = 1 million. How many passwords can you create out of 6 alpha-numeric characters? This 636.

**Example**

Say we flip a coin r times and record the sequence. How many times, n, would we have to repeat r coin flips to have a p = 40% chance of getting that sequence again? Let’s look, rather to the odds of not getting that sequence ever, in n sequences. This will be:



Now we solve for n,



**W{Tnp} w/o replacement from n distinct elements (order matters)**

Next I’ll do some examples of extracting tuple w/o replacement from n distinct elements…This is:



This is because we have n choices for the first element, n-1 choices for the next, n-2 choices for the ensuing, etc., until we have n – p + 1 choices for the last one. This works out to n(n-1)(n-2)⋯(n-p+1) = n!/(n-p)!. Typical notation:



**Example**

Say you have four different characters c1, c2, c3, c4 to choose from to create a two-character long password. How many passwords can you create if you can’t choose same character twice? Sample space is tuples of the form T = (t1, t2) where t1 and t2 cannot be the same and order matters. This would be:



And multiplicity is W = n!/(n-p)! = 4!/2! = 4∙3 = 12.

**Example**

Ten students line up to enter a classroom. Two specific students, Alex and Jordan, must **not** stand next to each other. How many valid line-ups are possible? Well if there are no restrictions, then there are 10! ways. And we can subtract from this the number of ways to put Alex and Jordan together. This is 2∙9! for the two ways to order Alex and Jordan, the 9 places to put the first one, and the 8! ways to place the others. So result is 10! – 2∙9! = 8∙9!

**Example**

Ten students line up to enter a classroom. What is the probability that the two most misbehaved students, Alex and Jordan, are right next to each other? This would be:



**Example**

What are the odds that in a room full of 30 people, no two people share the same birthday? Can say that sample space is of form T = (t1, t2, …., t30) where tj’s can be equal and order matters (switch t’s and that switches peoples’ birthdays). Sample space multiplicity is:



The event space is of form T = (t1, t2, …, t30) where order matters and the t’s cannot be equal.

Its multiplicity is:



So our answer is:



Why don’t we consider the sample space S to be equal to (t1, t2, …, t30) where t’s are dates, but the order doesn’t matter, i.e., just a set of dates? I’d say because the different tuples aren’t equally likely. For instance, (September, September, …, September) isn’t as likely as (September, September, …, October) since there’s way more of the latter (30 possiblities) than the former (just 1). I think the key issue with the inequivalence of the two ways of counting is that we have sampling w/ replacement.

**Example**

Out of a class of 30, 3 people are chosen at random to forego recess and weed the garden, sweep the sidewalks, clean the classroom. Say there are 13 boys and 17 girls in the class. What is the probability that all three students chosen are girls? Well we’d have:



And since we have 17 girls, then:



So the probability is:



**Example**

Twenty people throw their names in a hat, and three people are chosen. What is the probability a given person is chosen? Well the sample space is T = (P1, P2, P3), where Pj denotes a person and their place in the tuple denotes the order in which they were chosen. Pj cannot be equal and their order does matter. So:



Say it is person 4 that wins. Then the event points look like (4, P2, P3) or (P1, 4, P3) or (P1, P2, 4)? This has multiplicity:



So the probability is:



Another way is to say S = number of ways to choose 3 winners out of 20. And E is number of ways to choose 2 alternative winners out of 19. Then,



Could also say it’s P = 1 – P(that person not getting selected). Then,



What’s the probability a given *pair* of people being chosen (maybe they both know the raffle-drawer-outer)? Well, this could be found by taking the two given people out of the pool and finding the number of ways to choose 1 out of the remaining 18. So,



Could also say it’s P = 1 – P(not both those people getting selected) = 1 – P(choose neither) – P(choose 1 of them) – P(choose other one of them). Then,



Or could alternately frame it as:



**W{Tnp} w/ replacement from n distinct elements (order doesn’t matter)**

Now I’ll do some examples of extracting sets from a drawer, where we have the same n distinct elements to choose from each time. This is:



Note p can be larger than n. Can think of this as lining up your p = 10 sock choices, and separating them by n – 1 bars. The bars delineate which of the elements the choice belongs to. So for instance say we have n = 3 different socks, and that we’re making a set of 10 choices. Then we put n – 1 = 2 bars somewhere within the set of p = 10 to delineate how many of each type of sock we have. Something like this:



This would say 3 of sock A, 4 of sock B, and 3 of sock C in the first line. And in the second line we’d have 7 of sock A and 3 of sock B and 0 of sock C. And in the third line we’d have 0 of sock A, 0 of sock B, and 10 of sock C. The number of distinct sets is the number of distinct ways we can arrange the letters and bars. Can think of this as the number of ways to rearrange n – 1 + p elements, which is (n – 1 + p)!, but then have to divide the number of ways we can interchange the n – 1 bars which is (n – 1)!, and also divide out the number of ways we can interchange the identities of the p elements which is p!.

**Example**

Say you have four different characters c1, c2, c3, c4 to choose from to create a two-character long password. How many passwords can you create if no two passwords can have the same characters, regardless of order? Sample space is of the form T = (t1, t2) with replacement, but order doesn’t matter. This would be:



And the multiplicity is W = (n+p-1)!/(n-1)!p! = (4-1+2)!/(4-1)!2! = 5!/3!2! = 5∙4/2 = 10.

**W{Tnp} w/o replacement from n distinct elements (order doesn’t matter)**

And now I’ll do some examples of extracting a set w/o replacement from n distinct elements…this is:



It’s easiest to work backwards. Let W be the number of distinct ways to pull out p elements from n. Then for each of these distinct ways, there would p! ways to turn it into an indistinct way. So Wp! ought to equal the number of indistinct ways to pull out p of them. So then Wp! = n!/(n-p)!. And therefore:



A backwards way to think of it is, we can get the number of indistinct ways as n!/(n-p)!. And then to get the number of distinct ways, we just have to divide by all the ways we can permute the tuple’s elements, which is p!. Typical notation:



**Example**

Say you have four different characters c1, c2, c3, c4 to choose from to create a two-character long password. How many passwords can you create if no two passwords can have the same elements, regardless of order? And you can’t choose the same character twice? Sample space is of the form T = (t1, t2) where t1 and t2 cannot be the same and order doesn’t matter. This would look like:



And its multiplicity is W = n!/(n-p)!p! = 4!/2!2! = 6. So answer is: 4!/2!2! = 4∙3/2 = 6.

**Example**

If you remember that the numbers in your pin are 5,1,2,5, how many possible combinations are there? What are your odds of guessing the right one? There should be 4!/2! = 12 possilbe unique numbers we can construct out of these guys. Note we’re dividing by 2! because we have to erase distinctions between the order in which the two fives appear. Another way to reason about it is that we can choose C42 = 6 distinct places for the two 5’s to be placed. And then we can put the 1 and 2 in the other two slots in 2 different ways, so that gives us 6×2 = 12 possibilities. Explicitly, we have:



**Example**

If you know that numbers in your pin are 5, 1, 2, 5, and you also know that the 5’s are not right next to each other, how many options are there? What are your odds of guessing the right one? Well I guess we have: 4!/2! – 2∙3!/2! = 2∙3!/2! = 6 possibilities. Note we’re dividing by 2! because we have to erase distinctions between the order in which the two fives appear. Explicitly, the options are:



So odds are 1/6.

**Example**

To play the lottery, you pick five numbers and one power ball number. Typically, the five range from 1 to 70, and the power ball from 1 to 25. Order doesn’t matter for the five. So how many possibilities are there? Sample space is of the form T = (t1, t2, t3, t4, t5)⊗(t6) where t1,2,3,4,5 cannot equal each other and order doesn’t matter. But t6 is independent. Its multiplicity is:



and



So odds of winning is:



So basically the number of people living in the United States. If you match 4 numbers out of 5, *and* you get the powerball, then you win $50,000. What are odds of this? Well? You need to choose 4 out of the correct 5 and 1 out of the incorrect 65. And then the correct powerball. So we need:



So probability is:



If you match 4 numbers out of the 5, you get $100. What are the odds of doing this? Well? You need to choose 4 out of the correct 5 and 1 out of the incorrect 65. And you can choose any powerball number so this is:



So probability is:



If you match 3 and get the powerball, you also win $100. What are odds of this?



And so,



**Example**

What are the odds that in a room full of 30 people, exactly 3 people share the same birthday? Sample space multiplicity is:



The event space is comprised of choosing three people to have the same birthday, and 27 people to not have the same birthdays. So this would be:



So our answer is:



**Example**

Say 30 people apply for a job. How many candidates must one interview so that there is at least a 90% probability to interview 3 of the top 5? Well say we interview n candidates. The sample space would look like T = (t1, t2, t3, …, tn) where order doesn’t matter and they can’t be equal. So sample space multiplicity is:



And the event space is is the number of ways for n to comprise at least 3 of the top five. The event space would look like: Tp = (t1, t2, t3)⨂(t4, t5, …, tn) ∪ (t1, t2, t3, t4)⊗(t5, t6, …, tn) ∪ (t1, t2, t3, t4, t5)⊗(t6, t7, …, tn). Where the first tuple is drawn from the top 5 and the second tuple is drawn from the rest. And the multiplicity of this is:



So probability is:



We’d just have to try different n’s and see which one works. And looks like n = 23 is the smallest one that will do.

**Example**

An employer has a pool of 30 applicants. What are the odds that if she chooses 10 to interview she will get at least one of the top 3 candidates? This should be:



and,



So odds are:



Can also formulate as p = 1 – odds of getting none of the top 3. This would be:



**Example**

An employer has a pool of 30 applicants. What is the fewest number that she needs to interview so that she has at least a 95% chance of interviewing one of the top 3 candidates? Let’s do one by one. If interview 1, then have only



probability of getting one of top 3. This makes sense since there are 3 out of 30. If do two, then



If do three, then,



If do 4, 5, 6, etc., then,



Let’s jump to 10,



Looks like its 19,



In general, we’re looking for the solution to:



Could also frame it this way. What is the probability of not getting any of the three top candidates. Then the probability of getting at least one of the top three candidates is.



This reproduces the results above. Does this simplify?



So we have a cubic equation. No thanks. But if numerically solve, we find k = 18.3 about, congruent with our answer k = 19 above.

**Example**

Suppose the homecoming queen has come down to two people: Ann and Marie. To decide, a judge has to draw names out of hat. Because Ann had the most votes, she gets to put 15 slips of paper with her name on it in the hat. Marie had the second most votes, and gets to put 5 slips of paper with her name on it in the hat. A judge draws out 3 slips of paper. What is the likelihood that Marie has her name on most of them? Let’s look at the sample space. This is T = (t1, t2, t3) where tj’s cannot be equal and order doesn’t matter. Its multiplicity is:



and the event space will look like (t1)⊗(t2, t3) ∪ (,)⨂(t1, t2, t3), where the first tuple is drawn from Ann’s pile, and the second tuple from Marie’s pile.



So the probability is:



Note that if we had said that order *did* matter for the tuples, then we’d just have to multiply both W{E} and W{S} by 3!. This wouldn’t change the probability though. I think the key reason here is because we have sampling w/o replacement.

**Example**

John chooses 6 letters from the alphabet and Robbie chooses 8. What is the probability at least 4 letters are chosen in common? Well, sample space looks like Tp = (j1, j2, …, j6)⨂(r1, r2, …., r8). For each, order doesn’t matter and they can’t be equal. So its multiplicity is:



One way to think of the event space is as follows. Our event space tuples can be factored into (c1, c2, c3, c4)⨂(j5, j6)⨂(r5, r6, r7, r8) ∪ (c1, c2, c3, c4, c5)⨂(j6)⨂(r6, r7, r8) ∪ (c1, c2, c3, c4, c5, c6)⨂(,)⨂(r7, r8), where (c’s) denote the common elements, while (j’s) and (r’s) denote the distinct elements of John and Robbie. So W{E} = number of ways to choose 4 letters in common from 26, and then 2 and 4 not in common + number of ways to choose 5 letters in common from 26, and then 1 and 3 not in common + number of ways to choose 6 letters in common from 26, and then 0 and 2 not in common. This is:



Note this is the same as choosing the 2 and 4, 1 and 3, 0 and 2 in reverse order:



And our result is:



Another way to think of the event space is this. So *per* choice of Robbie’s letters, the number of ways John can choose 4, 5, 6 letters that match Robbie’s is the number of ways he can choose 4, 5, 6 letters from the 8 Robbie has, and then 2, 1, 0 letters from the rest:



If we multiply by all of Robbie’s possible choices, then we get the total number of sample points where we get at least four matches:



So the probability is:



**Example**

An advisory board ruled 5-3 against a plaintiff (all 5 women against and all 3 men for). If we may suppose that any 5 of the board members were as likely to vote against as any other, what are the odds the vote would’ve split along gender lines?

Our sample space is the number of ways to pick 5 people out of the 8. This is:



and the event space is the one sample where all five women are in the group of five. This is:



So the probability is:



**Example**

Out of a class of 30, 5 people are chosen at random to work and forego recess. Say there are 13 boys and 17 girls in the class. What is the probability that 3 students chosen are girls and 2 are boys? Well we’d have:



and,



So the probability is:



What’s the most likely scenario? Well we have to try:



So looks like choosing 3 girls and 1 boy is the most likely. And can verify that these numbers add up to C(30,5) = 142506.

**Example**

Say you have n socks, nb of which are blue, and nr of which are red. How many distinct ways can you arrange them on the floor? For instance, we would count the first two of the arrangements below as identical, and so not distinct.



Helps to work backwards. Consider all the distinct arrangements, W. Then for each arrangement, we can recover the total number of arrangements regardless (n!), by taking each distinct arrangment and multiplying by the number of ways we could permute the red socks with themselves (nr!) and the blue socks with themselves (nb)!. For instance, in the first arrangment, we could generate the same thing by choosing any of the 4 red socks to go in the 1st position, then any of the 3 remaining to go in the 2nd position, then any of the 2 remaining to go in the 5th position, and last any of the 1 remaining to go in the 10th position. So Wnr!nb! = n!. And therefore,



Another way to think of it is this. You have nr red socks, and you have to assign each of them a coordinate. This is Tn\_r tuple basically. There are n!/(n-nr)! ways to extract a Tn\_r tuple coordinate, per our discussion above. But then it doesn’t matter if our Tuple is (to use the first two arrangements in diagram above as illustration) (1,2,5,10) or (5,10,2,1). In other words we have to divide out all the nr! permutations of the numbers in our Tuple. So our answer is n!/(n-nr)!nr! = n!/nb!nr!. And it doesn’t matter the order of the remaining blue socks, so there is nothing further to account for.

Or another more direct way is to simply say we’ve got nr red socks and n coordinates. And we want to find the number of unique ways to create an nr long tuple.

**W{Tnp1,p2…,pm} w/o replacement from n distinct elements (order doesn’t matter)**

In this case we’re looking at a tuple of tuples I guess Tnp1,p2,p3 = ((t1, t2), (t3, t4, t5), (t6, t7, t8, t9, t10)). The sizes of the subtuples is fixed to p1, p2, and p3 (2, 3, 5 in this case). The number of ways we can assign n elements to the n slots (and it’s kind of presumed p1 + p2 + p3 = n), regardless of sub-tuple is n! And then we account for indistinct arrangements by dividing by the number of ways we can permute the elements of the three subtuples within themselves, i.e., dividing by p1! and p2! and p3!. So we have:



Generalizing to more than three subgroups is straightforward. Another way to look at is this.



which is to say, we multiply together the number of ways to extract a unique set of p1 guys from the n elements, with the number of ways to extract a unique set of p2 guys from the remaining n-p1 elements, with the number of ways to extract a unique p3 guys from the last remaining n-p1-p2 = p3 elements (this is just 1, in fact, as can see from the last term in that product). Maybe call it:



**Example**

Say you have n socks, nb of which are blue, nr of which are red, and ng of which are green. How many distinct ways can you arrange them on the floor? For instance, we would count the first two of the arrangements below as identical, and so not distinct.



Well this is:



**Example**

Construction jobs were assigned to 20 day-laborers, including 4 white people. Job A required 6 people, job B required 4 people, job C required 5 people, and job D required 5 people. What is the probability that all 4 white people get job A? Well, sample space is:



And the event space is the number of ways we can assign, out of a total of 16 people (all 4 white people have already been assigned) 2 to job A, 4 to job B, 5 to job C, and 5 to job D. This is:



So the probability is:



**Example**

On school beautification day, out of a class of 30, 15 people are chosen to help plant bushes, 10 people are chosen to weed, and 5 people are chosen to pick up trash. Say there are 13 boys and 17 girls in the class. What is the probability that all of the people assigned to to plant bushes are boys?



And since we have 13 boys, all assigned to plant bushes, we have to work out next the number of ways we can assign, out of 17 remaining people, 2 to plant bushes, 10 to weed, and 5 to pick up trash. This is:



So the probability is:



**Example**

Do resistor problem?