

Combinatorics

Counting problems can be approached from the basic building blocks described in the first section: [Counting](#). However some counting problems are so ubiquitous in the world of probability that it is worth knowing a few higher level counting abstractions. When solving problems, if you can find the analogy from these canonical examples you can build off of the corresponding combinatorics formulas:

1. [Permutations of Distinct Objects](#)
2. [Permutations with Indistinct Objects](#)
3. [Combinations with Distinct Objects](#)
4. [Bucketing with Distinct Objects](#)
5. [Bucketing with Indistinct Objects](#)
6. [Bucketing into Fixed Sized Containers](#)

While these are by no means the only common counting paradigms, it is a helpful set.

Permutations of Distinct Objects

Definition: Permutation Rule

A permutation is an ordered arrangement of n distinct objects. Those n objects can be permuted in $n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 = n!$ ways.

This changes slightly if you are permuting a subset of distinct objects, or if some of your objects are indistinct. We will handle those cases shortly! Note that unique is a synonym for distinct.

Example: How many unique orderings of characters are possible for the string "BAYES"? **Solution:** Since the order of characters is important, we are considering all permutations of the 5 distinct characters B, A, Y, E, and S: $5! = 120$. Here is the full list:

BAYES, BAYSE, BAEYS, BAESY, BASYE, BASEY, BYAES, BYASE, BYEAS, BYESA, BYSAE, BYSEA, BEAYS, BEASY, BEYAS, BEYSA, BESAY, BESYA, BSAYE, BSAEY, BSYAE, BSYEA, BSEAY, BSEYA, ABYES, ABYSE, ABEYS, ABESY, ABSYE, ABSEY, AYBES, AYBSE, AYEB, ASYBE, ASYEB, ASEBY, ASEYB, YBAES, YBASE, YBEAS, YBESA, YBSAE, YBSEA, YABES, YABSE, YAEBS, YAESB, YASBE, YASEB, YEBAS, YEBSA, YEABS, YEASB, YESBA, YESAB, YSBAE, YSBEA, YSABE, YSAEB, YSEBA, YSEAB, EBAYS, EBASY, EBYAS, EBYSA, EBSAY, EBSYA, EABYS, EABSY, EAYBS, EAYSB, EASBY, EASYB, EYBAS, EYBSA, EYABS, EYASB, EYSBA, EYSAB, ESBAY, ESBYA, ESABY, ESAYB, ESYBA, ESYAB, SBAYE, SBAEY, SBYAE, SBYEA, SBEAY, SBEYA, SABYE, SABEY, SAYBE, SAYEB, SAEBY, SAEYB, SYBAE, SYBEA, SYABE, SYAEB, SYEBA, SYEAB, SEBAY, SEBYA, SEABY, SEAYB, SEYBA, SEYAB

Example: a smartphone has a 4-digit passcode. Suppose there are 4 smudges over 4 digits on the screen. How many distinct passcodes are possible?

Solution: Since the order of digits in the code is important, we should use permutations. And since there are exactly four smudges we know that each number in the passcode is distinct. Thus, we can plug in the permutation formula: $4! = 24$.

Permutations of Indistinct Objects

Definition: Permutations of Indistinct Objects

Generally when there are n objects and:

n_1 are the same (indistinguishable) and

n_2 are the same and

...

n_r are the same, then the number of distinct permutations is:

$$\text{Number of unique orderings} = \frac{n!}{n_1!n_2!\cdots n_r!}$$

Example: How many distinct bit strings can be formed from three 0's and two 1's?

Solution: 5 total digits would give $5!$ permutations. But that is assuming the 0's and 1's are distinguishable (to make that explicit, let's give each one a subscript). Here are the $3! \cdot 2! = 12$ different ways that we could have arrived at the identical string "01100" if we thought of each 0 and 1 as unique.

$0_1 \ 1_0 \ 1_1 \ 0_2 \ 0_3$

$0_1 \ 1_0 \ 1_1 \ 0_3 \ 0_2$

$0_2 \ 1_0 \ 1_1 \ 0_1 \ 0_3$

$0_2 \ 1_0 \ 1_1 \ 0_3 \ 0_1$

$0_3 \ 1_0 \ 1_1 \ 0_1 \ 0_2$

$0_3 \ 1_0 \ 1_1 \ 0_2 \ 0_1$

$0_1 \ 1_1 \ 1_0 \ 0_2 \ 0_3$

$0_1 \ 1_1 \ 1_0 \ 0_3 \ 0_2$

$0_2 \ 1_1 \ 1_0 \ 0_1 \ 0_3$

$0_2 \ 1_1 \ 1_0 \ 0_3 \ 0_1$

$0_3 \ 1_1 \ 1_0 \ 0_1 \ 0_2$

$0_3 \ 1_1 \ 1_0 \ 0_2 \ 0_1$

Since identical digits are indistinguishable, all the listed permutations are the same. For any given permutation, there are $3!$ ways of rearranging the 0's and $2!$ ways of rearranging the 1's (resulting in indistinguishable strings). We have over-counted. Using the formula for permutations of indistinct objects, we can correct for the over-counting:

$$\text{Total} = \frac{5!}{3! \cdot 2!} = \frac{120}{6 \cdot 2} = 10$$

Example: How many *distinct* orderings of characters are possible for the string "MISSISSIPPI"?

Solution: In the case of the string "MISSISSIPPI", we should separate the characters into four distinct groups of indistinct characters: one "M", four "I"s, four "S"s, and two "P"s. The number of distinct orderings are:

$$\frac{11!}{1!4!4!2!} = 34,650$$

Example: Consider the 4-digit passcode smart-phone from before. How many distinct passcodes are possible if there are 3 smudges over 3 digits on the screen?

Solution: One of 3 digits is repeated, but we don't know which one. We can solve this by making three cases, one for each digit that could be repeated (each with the same number of permutations). Let A, B, C represent the 3 digits, with C repeated twice. We can initially pretend the two C 's are distinct

$[A, B, C_1, C_2]$. Then each case will have $4!$ permutations: However, then we need to eliminate the double-counting of the permutations of the identical digits (one A , one B , and two C 's):

$$\frac{4!}{2! \cdot 1! \cdot 1!}$$

Adding up the three cases for the different repeated digits gives

$$3 \cdot \frac{4!}{2! \cdot 1! \cdot 1!} = 3 \cdot 12 = 36$$

Part B: What if there are 2 smudges over 2 digits on the screen?

Solution: There are two possibilities: 2 digits used twice each, or 1 digit used 3 times, and other digit used once.

$$\frac{4!}{2! \cdot 2!} + 2 \cdot \frac{4!}{3! \cdot 1!} = 6 + (2 \cdot 4) = 6 + 8 = 14$$

You can use the power of computers to enumerate all permutations. Here is sample python code which uses the built in `itertools` library:

```
>>> import itertools

# get all 4! = 24 permutations of 1,2,3,4 as a list:
>>> list(itertools.permutations([1,2,3,4]))
[(1, 2, 3, 4), (1, 2, 4, 3), (1, 3, 2, 4), (1, 3, 4, 2), (1, 4, 2, 3), (1, 4, 3, 2),
 (2, 1, 3, 4), (2, 1, 4, 3), (2, 3, 1, 4), (2, 3, 4, 1), (2, 4, 1, 3), (2, 4, 3, 1),
 (3, 1, 2, 4), (3, 1, 4, 2), (3, 2, 1, 4), (3, 2, 4, 1), (3, 4, 1, 2), (3, 4, 2, 1),
 (4, 1, 2, 3), (4, 1, 3, 2), (4, 2, 1, 3), (4, 2, 3, 1), (4, 3, 1, 2), (4, 3, 2, 1)]

# get all 3!/2! = 3 unique permutations of 1,1,2 as a set:
>>> set(itertools.permutations([1,1,2]))
{(1, 2, 1), (2, 1, 1), (1, 1, 2)}
```

Combinations of Distinct Objects

Definition: Combinations

A combination is an unordered selection of r objects from a set of n objects. If all objects are distinct, and objects are not "replaced" once selected, then the number of ways of making the selection is:

$$\text{Number of unique selections} = \frac{n!}{r!(n-r)!} = \binom{n}{r}$$

Here are all the $10 = \binom{5}{3}$ ways of choosing three items from a list of 5 unique numbers:

```
# Get all ways of choosing three numbers from [1,2,3,4,5]
>>> list(itertools.combinations([1,2,3,4,5], 3))
[(1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 3, 4), (1, 3, 5), (1, 4, 5), (2, 3, 4), (2, 3, 5),
 (2, 4, 5), (3, 4, 5)]
```

Notice how order doesn't matter. Since $(1, 2, 3)$ is in the set of combinations, we don't also include $(3, 2, 1)$ as this is considered to be the same selection. Note that this formula does not work if some of the objects are indistinct from one another.

How did we get the formula $\frac{n!}{r!(n-r)!}$? Consider this general way to select r unordered objects from a set of n objects, e.g., "7 choose 3":

1. First consider permutations of all n objects. There are $n!$ ways to do that.
2. Then select the first r in the permutation. There is one way to do that.
3. Note that the order of r selected objects is irrelevant. There are $r!$ ways to permute them. The selection remains unchanged.
4. Note that the order of $(n - r)$ unselected objects is irrelevant. There are $(n - r)!$ ways to permute them. The selection remains unchanged.

$$\text{Total} = \frac{n!}{r! \cdot (n - r)!} = \binom{n}{r}$$

Example: In the Hunger Games, how many ways are there of choosing 2 villagers from district 12, which has a population of 8,000?

Solution: This is a straightforward combinations problem. $\binom{8000}{2} = 31,996,000$.

Part A: How many ways are there to select 3 books from a set of 6?

Solution: If each of the books are distinct, then this is another straightforward combination problem.

There are $\binom{6}{3} = \frac{6!}{3!3!} = 20$ ways.

Part B: How many ways are there to select 3 books if there are two books that should not both be chosen together? For example, if you are choosing 3 out of 6 probability books, don't choose both the 8th and 9th edition of the Ross textbook.

Solution: This problem is easier to solve if we split it up into cases. Consider the following three different cases:

Case 1: Select the 8th Ed. and 2 other non-9th Ed. books: There are $\binom{4}{2}$ ways of doing so.

Case 2: Select the 9th Ed. and 2 other non-8th Ed. books: There are $\binom{4}{2}$ ways of doing so.

Case 3: Select 3 books from the 4 remaining books that are neither the 8th nor the 9th edition: There are $\binom{4}{3}$ ways of doing so.

Using our old friend the Sum Rule of Counting, we can add the cases:

$$\text{Total} = 2 \cdot \binom{4}{2} + \binom{4}{3} = 16$$

Alternatively, we could have calculated all the ways of selecting 3 books from 6, and then subtract the "forbidden" ones (i.e., the selections that break the constraint).

Forbidden Case: Select 8th edition and 9th edition and 1 other book. There are $\binom{4}{1}$ ways of doing so (which equals 4). Total = All possibilities - forbidden = $20 - 4 = 16$ Two different ways to get the same right answer!

Bucketing with Distinct Objects

In this section we are going to be counting the many different ways that we can think of stuffing elements into containers. (It turns out that Jacob Bernoulli was into voting and ancient Rome. And in ancient Rome they used urns for ballot boxes. For this reason many books introduce this through counting ways to put balls in urns.) This "bucketing" or "group assignment" process is a useful metaphor for many counting problems.

The most common case that we will want to consider is when all of the items you are putting into buckets are distinct. In that case you can think of bucketing as a series of steps, and employ the step rule of counting. The first step? You put the first distinct item into a bucket (there are number-of-buckets ways to do this). Second step? You put the second distinct item into a bucket (again, there are number-of-buckets ways to do this).

Bucketing Distinct Items:

Suppose you want to place n distinguishable items into r containers. The number of ways of doing so is:

$$r^n$$

You have n steps (place each item) and for each item you have r choices

Problem: Say you want to put 10 distinguishable balls into 5 urns (No! Wait! Don't say that! Not urns!). Okay, fine. No urns. Say we are going to put 10 different strings into 5 buckets of a hash table. How many possible ways are there of doing this?

Solution: You can think of this as 10 independent experiments each with 5 outcomes. Using our rule for bucketing with distinct items, this comes out to 5^{10} .

Bucketing with Indistinct Objects

While the previous example allowed us to put n distinguishable objects into r distinct groups, the more interesting problem is to work with n indistinguishable objects.

Divider Method:

Suppose you want to place n indistinguishable items into r containers. The divider method works by imagining that you are going to solve this problem by sorting two types of objects, your n original elements and $(r - 1)$ dividers. Thus, you are permuting $n + r - 1$ objects, n of which are the same (your elements) and $r - 1$ of which are the same (the dividers). Thus the total number of outcomes is:

$$\frac{(n + r - 1)!}{n!(r - 1)!} = \binom{n + r - 1}{n} = \binom{n + r - 1}{r - 1}$$

The divider method can be derived via the "Stars and Bars" method. This is a creative construction where we consider permutations of indistinguishable items, represented by stars $*$, and dividers between our containers, represented by bars $|$. Any distinct permutation of these stars and bars represents a unique assignment of our items to containers.

Imagine we want to separate 5 indistinguishable objects into 3 containers. We can think of the problem as finding the number of ways to order 5 stars and 2 bars $*****| |$. Any permutation of these symbols represents a unique assignment. Here are a few examples:

$**|*|**$ represents 2 items in the first bucket, 1 item in the second and 2 items in the third.

$*****| |$ represents 4 items in the first bucket, 0 item in the second and 1 items in the third.

$| |*****$ represents 0 items in the first bucket, 0 item in the second and 5 items in the third.

Why are there only 2 dividers when there are 3 buckets? This is an example of a [fence-post-problem](#). With 2 dividers you have created three containers. We already have a method for counting permutations with some indistinct items. For the example above where we have seven elements in our permutation ($n = 5$ stars and $r - 1 = 2$ bars):

$$\text{Number of unique orderings} = \frac{n!}{n_1!n_2!} = \frac{(n + r - 1)!}{n!(r - 1)!} = \frac{7!}{5!2!} = 21$$

Part A: Say you are a startup incubator and you have \$10 million to invest in 4 companies (in \$1 million increments). How many ways can you allocate this money?

Solution: This is just like putting 10 balls into 4 urns. Using the Divider Method we get:

$$\text{Total ways} = \binom{10+4-1}{10} = \binom{13}{10} = 286$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 = 10$, where x_i represents the investment in company i such that $x_i \geq 0$ for all $i = 1, 2, 3, 4$.

Part B: What if you know you want to invest at least \$3 million in Company 1?

Solution: There is one way to give \$3 million to Company 1. The number of ways of investing the remaining money is the same as putting 7 balls into 4 urns.

$$\text{Total Ways} = \binom{7+4-1}{7} = \binom{10}{7} = 120$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 = 10$, where $x_1 \geq 3$ and $x_2, x_3, x_4 \geq 0$. To translate this problem into the integer solution equation that we can solve via the divider method, we need to adjust the bounds on x_1 such that the problem becomes $x_1 + x_2 + x_3 + x_4 = 7$, where x_i is defined as in Part A.

Part C: What if you don't have to invest all \$10 M? (The economy is tight, say, and you might want to save your money.)

Solution: Imagine that you have an extra company: yourself. Now you are investing \$10 million in 5 companies. Thus, the answer is the same as putting 10 balls into 5 urns.

$$\text{Total} = \binom{10+5-1}{10} = \binom{14}{10} = 1001$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 + x_5 = 10$, such that $x_i \geq 0$ for all $i = 1, 2, 3, 4, 5$.

Bucketing into Fixed-Sized Containers

Bucketing into Fixed-Sized Containers:

If n objects are distinct, then the number of ways of putting them into r groups of objects, such that group i has size n_i , and $\sum_{i=1}^r n_i = n$, is:

$$\frac{n!}{n_1!n_2!\cdots n_r!} = \binom{n}{n_1, n_2, \dots, n_r}$$

where $\binom{n}{n_1, n_2, \dots, n_r}$ is special notation called the multinomial coefficient.

You may have noticed that this is the exact same formula as "Permutations With Indistinct Objects". There is a deep parallel. One way to imagine assigning objects into their groups would be to imagine the groups themselves as objects. You have one object per "slot" in a group. So if there were two slots in group 1, three slots in group 2, and one slot in group 3 you could have six objects (1, 1, 2, 2, 2, 3). Each unique permutation can be used to make a unique assignment.

Problem:

Company Camazon has 13 distinct new servers that they would like to assign to 3 datacenters, where Datacenter A, B, and C have 6, 4, and 3 empty server racks, respectively. How many different divisions of the servers are possible?

Solution: This is a straightforward application of our multinomial coefficient representation. Setting $n_1 = 6, n_2 = 4, n_3 = 3$, $\binom{13}{6, 4, 3} = 60,060$.

Another way to do this problem would be from first principles of combinations as a multipart experiment. We first select the 6 servers to be assigned to Datacenter A, in $\binom{13}{6}$ ways. Now out of the 7 servers remaining, we select the 4 servers to be assigned to Datacenter B, in $\binom{7}{4}$ ways. Finally, we select the 3 servers out of the remaining 3 servers, in $\binom{3}{3}$ ways. By the Product Rule of Counting, the total number of ways to assign all servers would be $\binom{13}{6}\binom{7}{4}\binom{3}{3} = \frac{13!}{6!4!3!} = 60,060$.