

Combinations


<https://github.com/learn-co-curriculum/dsc-combinations>

<https://github.com/learn-co-curriculum/dsc-combinations/issues/new/choose>

Introduction

In the previous section, you learned about how to apply permutations. Permutations come in handy when we want to know how many ways we can order sets. Now, what if order is not important? That's where *combinations* come in.

Objectives

You will be able to:

- Describe how combinations are used when order is not important

Why combinations?

In some settings, the order of the selection is not important.

Let's go back to our example of a coverband creating a setlist. Imagine that the band is playing 3 songs out of their 8 song repertoire. How many ways can they select songs, assuming that the **order of the chosen songs is not important**? Here, we just want to know *which* three songs they play, and not which song goes first, second and last.

You can use a backward rationale here. You know that when order *did* matter, our answer was $8 * 7 * 6$. When having three elements, there are 6 possible orders (ABC, ACB, CAB, CBA, BAC, BCA), so the answer can be obtained by dividing our previous answer by 6.

This type of problem can be solved by using *combinations*. In general, combinations answer the question: "How many ways can we create a subset k out of n objects?". The subset is not ordered.

$$\binom{n}{k} = \frac{P_k^n}{k!} = \frac{\frac{n!}{(n-k)!}}{k!} = \frac{n!}{(n-k)!k!}$$

Applied to our example, this means that there are

$$\frac{8!}{(8-3)!3!} = \frac{8!}{(8-3)!3!} = \frac{8 * 7 * 6}{6} = 56$$



Chat

so there are 56 ways to choose 3 songs out of an 8 song repertoire.

Summary

In this section, you learned what combinations are and how to use them. Let's put this knowledge into practice!

How do you feel about this lesson?



Have specific feedback?

[Tell us here! \(https://github.com/learn-co-curriculum/dsc-combinations/issues/new/choose\)](https://github.com/learn-co-curriculum/dsc-combinations/issues/new/choose)