Positions of Combinations in an Unordered Combinatoric Series

Abstract

Computing the number of possible combinations in an unordered non-repetitive set of all combinations generated by the (***n*** choose ***k***) problem is straightforward and related to the Binomial Theorem as well as the rows of Pascal's Triangle. Slightly less straightforward, however, is finding the lengths of the subsets, position of a combination or the combination at a position.

Introduction

Combinatorics provides the tools to answer many questions such as the number of unordered non-repetitive combinations of a set of given length and given subset length, the classic "***n*** choose ***k***" problem, the answer to which can be shown to bear relation to the values of rows of Pascal's Triangle and in turn to the Binomial Theorem.

A question asked less often though, is that given a combination and a list of all possible combinations, where would the given combination be located within the larger list? Conversely, given a list of all possible combinations and a position value, which combination would be located at that position? This might proceed from research into large combinatoric systems such as lottery drawings, where calculating the Pascal's Triangle would be cumbersome and listing all combinations in computer memory or file would be prohibitive. The efficient and algorithmically rigorous approach to directly calculate these answers is to apply combinatorics and set theory to determine a formula.

Subset Length

First the more common scenario, calculating the number of combinations in a set of length ***n***, choosing ***k*** set members per combination. For a set of length ***n***=6, choosing ***k***=3, the results are as follows:

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

$$\binom{6}{3}=\frac{6!}{3!(6-3)!}=\frac{720}{6(3)!}=\frac{720}{36}=20$$

Generating a Pascal's Triangle with ***n***=6 rows yields the following, and the value at the position (row = 6), (column = 3) is also found to be 20. This is true for all values of ***n*** and ***k***.

n=0 1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
n=6 1 6 15 20 15 6 1

k=0 k=3

This deals only with the total number of combinations for $\binom{n}{k}$ however, which while useful would not give a direct answer for other related questions. For instance, assuming the set ***n*** is the natural numbers (1,2,3,4,5,6) and choosing ***k***=3, how many of the $\binom{n}{k}$ combinations begin with the number 3? The answer to this question can still be found in a Pascal's Triangle, selecting different values. Since the answer sought is now the length of subsequent subsets of set ***n***, it will seem intuitive perhaps that the answer will involve ***n*** and ***k*** less than the values for the full set and the Pascal Triangle. The full list of all unordered combinations for ***n***=6 and ***k***=3 are:

(123, 124, 125, 126, 134, 135, 136, 145, 146, 156, 234, 235, 236, 245, 246, 256, 345, 346, 356, 456)

A simple count of elements reveals that for each numeric member of the set $\binom{n}{k}$, the length of the subset beginning with that member is:

1	=	10
2	=	6
3	**=**	**3**
4	=	1
5	=	0
6	=	0

Revisiting Pascal's Triangle, using the (row = ***n***-1), (column=***k***-1) values, the results correlate.

n=0 — 1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
n=5 — 1 5 10 10 5 1
k=0 k=2

Expressed as the Binomial Theorem equivalent:

$$\binom{5}{2}=\frac{5!}{2!(5-2)!}=\frac{120}{2(3)!}=\frac{120}{12}=10$$

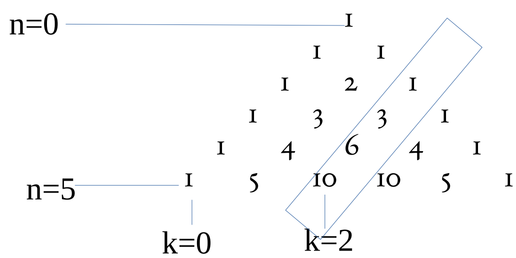
Using this method, the length of any and all subsets of the larger set may be determined. To generalize, the set of the length of subsets in set ***n***, choose ***k***, can be expressed as:

$$\left\{\left(\frac{(n-x)!}{((k-1)!(n-k-x)!)}\right):1\leqslant x\leqslant(n-k)\right\}$$

This provides the framework for finding the combination at a given position in the larger set ***n***, and for finding the position within the set given the combination.

Position of a Combination

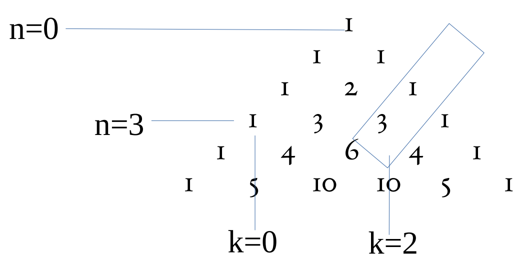
In this example, the position of the element (245) in the unordered combination set $\binom{6}{3}$ will be calculated. First the set of all **n** is correlated with the values of the Pascal's Triangle diagonal beginning at (row = ***n***-1), (column=***k***-1).



1	=	10
2	=	6
3	=	3
4	=	1
5	=	0
6	=	0

The highest order (leftmost) element of the desired combination is "2", so the position value is initialized at **x**=10, the sum of the lengths of all subsets prior to the index element "2". The iterative process is continued by removing the index elements of the target subset and all prior subsets, in this case (1, 2), from the original set of all elements and recalculating. This leaves a subset (3, 4, 5, 6) of the original set with length 4.

(row = ***n***-1), (column=***k***-1).



3	=	3
4	=	1
5	=	0
6	=	0

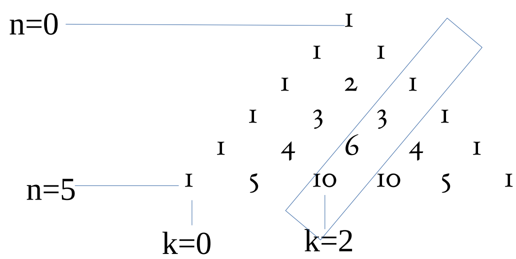
The next highest order element of the desired combination is "4", so the position value is incremented by the sum of the lengths of all prior subsets again, **x** = (10 + 3) = 13 . The iterative process is continued by removing the index elements of the target subset and all prior subsets, in this case (3, 4), from the working set and recalculating. This leaves a subset (5, 6) of the original set with length 2.

The element of the desired combination in this case is the lowest order (rightmost), the singles place if compared to decimal or other based numbers, so the sum of the lengths of all prior subsets is simply incremented by its position within the remaining subset (5, 6), resulting in **x** = (10 + 3 + 1) = 14. Checking this result by counting the positions of the fully enumerated list of all unordered combinations for ***n***=6 and ***k***=3 confirms the result:

(123, 124, 125, 126, 134, 135, 136, 145, 146, 156, 234, 235, 236, **245**, 246, 256, 345, 346, 356, 456)

Combination at a Position

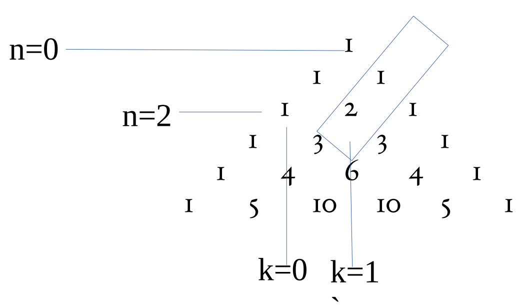
In this example, the 18th combination of the unordered combination set $\binom{6}{3}$ will be calculated. First the set of all **n** is correlated with the values of the Pascal's Triangle diagonal beginning at (row = ***n***-1), (column=***k***-1).



1	=	10
2	=	6
3	=	3
4	=	1
5	=	0
6	=	0

The position value [18] is greater than or equal to the sum of the lengths of prior subsets up to the index "3", but less than the sum of the lengths of prior subsets up to the index "4", indicating that the highest order element of the desired combination is "3". The position value is initialized with the sum of the lengths of all prior subsets, **x** = 16, and the index elements prior to and including "3", in this case (1,2,3), are removed from the original set, leaving a subset (4, 5 ,6) of length 3. The value for **k** is decremented, and values are recalculated.

(row = ***n***-1), (column=***k***-2).



4	=	2
5	=	1
6	=	0

The difference between the desired position value [18] and the position value **x** is greater than or equal to the sum of the lengths of prior subsets up to the index "4", but less than the sum of the lengths of prior subsets up to the index "5", indicating that the next highest order element of the desired combination is "4". The lengths of the prior subsets (there are no prior subsets, "4" was the lowest possible index) are added to the position value, **x** = (16 + 0) = 16 . The iterative process is continued by removing the index elements of the target subset and all prior subsets, in this case (4), from the working set and recalculating. This leaves a subset (5, 6) of the original set with length 2.

The element of the desired combination in this case is the lowest order (rightmost), the singles place if compared to decimal or other based numbers, so the index of the position of the final element within the remaining subset (5, 6) is simply the difference between the desired value (18) and the current position value **x**, or (18-16) = 2, resulting in the final element of "6", the full combination being "346". Checking this result by counting the positions of the fully enumerated list of all unordered combinations for ***n***=6 and ***k***=3 confirms the result:

(123, 124, 125, 126, 134, 135, 136, 145, 146, 156, 234, 235, 236, 245, 246, 256, 345, **346**, 356, 456)

Conclusion

The application of the Binomial Theorem, represented visually for demonstration purposes by Pascal's Triangle, may be employed to determine the position of a given combination within an unordered list of all possible combinations, without the need to enumerate and iterate the entire list. Conversely, a related algorithm may be employed to determine the combination at a given position within an unordered list of all possible combinations, again without the need to enumerate and iterate the entire list. Applications for this could be (and indeed have been) the study of the occurrence of combinations within all combinatoric possibilities, to analyze whether the occurrences of them are patternistic or bounded within random or pseudorandom systems. This could aid in the measurement of bias within random number or combination generators, preliminary detection of patternicity in presumed random systems (radio static, etc), and other research.

Addendum

Included in a GitHub repository are Python 3 functions capable of performing the operations described in this paper, returning a combination given a position, and returning a position given a combination. Analogous functions for permutations, ordered and unordered, repeats allowed and repeats disallowed conditions are also covered. This code is released under an open MIT license and permission is granted for use, modification, and distribution without attribution or remuneration. No warranties are expressed or implied.

https://github.com/MeatAcorn/combinatorics