Luis Ignacio Pulido

Python for Color Arrangement: A Case Study in Permutations and Combinations

Permuting and Combining

In the context of analyzing occurrences within a sample space, various methods are typically employed to quantify the frequency and relationships among different elements. These methods often include constructing a frequency table, utilizing diagrams such as Venn or tree diagrams, or formulating probability functions. While these techniques are effective for discrete, finite, and relatively small sample spaces, they may become insufficient when dealing with scenarios involving multiple interacting elements.

For instance, consider a situation where a person decides whether to bring an umbrella on a given day, contingent upon the probability of rain. If this decision process is repeated over a week, calculating the likelihood of successful outcomes (e.g., not getting wet) on a specific number of days becomes more complex. Assuming a fixed probability for the latter, we could compute the associated probabilities of carrying an umbrella and experiencing rain or not. Now, if we were to repeat this experiment over a week and calculate the chances of being successful on three days out of the whole week, we realize that such an event could occur on Monday, Tuesday, and Wednesday, or Monday, Tuesday, and Friday, or any other combination of three days within the week. This complexity arises from the numerous possible combinations of outcomes across different days within the week. Such observation underscores the need to adopt a combinatorial approach to statistical problems, which systematically considers all potential arrangements of scenarios. In this report, we aim to explore these concepts both theoretically and practically, leveraging mathematical principles and applying computational tools such as Python to deploy an automated solution to our problem.

When making a selection from a set of elements where the order of decision is not important, we are dealing with combinations of them. For instance, when considering the days of the week, selecting Monday, Tuesday, and Wednesday is different from choosing Monday, Tuesday, and Thursday, even though both selections consist of the same days. However, is the arrangement Monday, Tuesday, and Wednesday any different from Tuesday, Wednesday, and Monday? The answer depends on the problem we are analyzing and the specific outcome we are interested in. In many cases, such as ours, these options are considered the same. Therefore, when applying a binomial distribution to a fixed probability of success, we utilize the combinatorial operator.

Now, let's consider a scenario where we are arranging people in a line. In this case, the order of placement matters; placing one person second or third results in different arrangements, even though the individuals (elements) remain the same. This concept is known as permutation. The different arrangements of a set of distinct objects are referred to as permutations.

The relationship between permutations and combinations is defined by the factorial. For example, if we want to determine how many ways we can choose 3 days of the week, we apply the combination operator to 7 with 3, resulting in a total of 35 possible combinations. If we are interested in the order of these dates, we then multiply the result by the factorial of 3 (3!), yielding 210. Interestingly, this is equivalent to the product of 7 times 6 times 5, representing the number of elements we can choose from our series in each iteration.

Problem Proposal

Let's consider a scenario where there are seven distinct individuals, each uniquely identified, and they are divided into three groups. These groups are structured in the following manner: one consists of three individuals, another group consists of three individuals, and the remaining individual forms the third team. It's important to note that these sets are assigned permanent seating arrangements, and any alteration to their positions is prohibited.

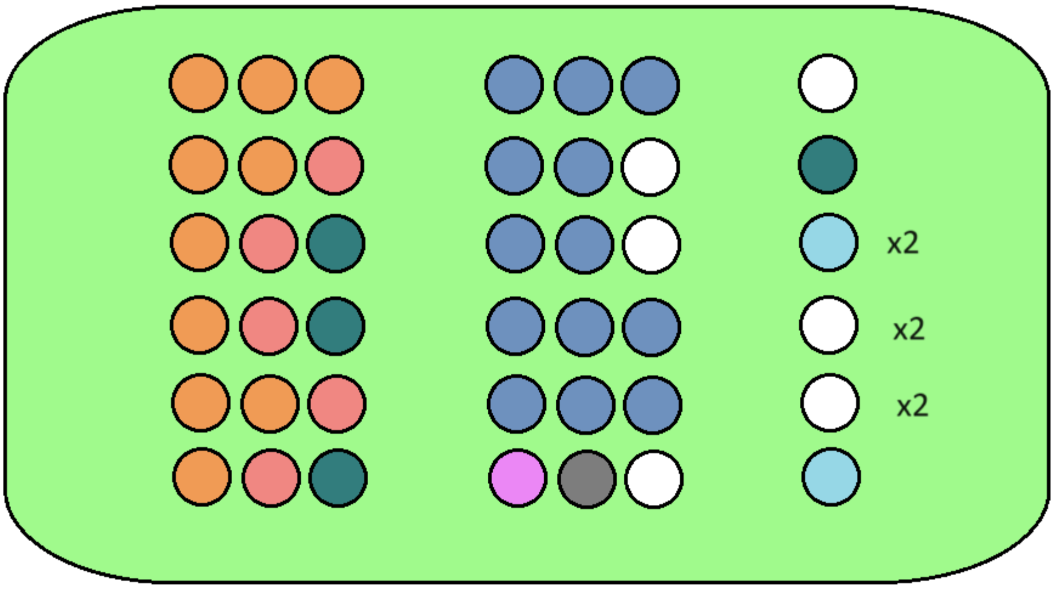
Additionally, there are seven distinct colors available to be distributed among these individuals. The arrangement of colors among the individuals within each group is significant, as the order of colors matters. For instance, the arrangement "Blue - Blue - Red" is distinct from "Blue - Red - Blue".

A key constraint in this problem is that the same color cannot be used in multiple groups. If a particular color is assigned to any individual within one group, it can only be utilized within that specific group; no other group may employ the same color. However, colors are allowed to be repeated in intra-groups. This is, color similarity between clusters is banned but same team members can match.

The exercise: Find how many color arrangements can be derived conditional to the previous information.

Problem Logic

To approach our problem systematically, we will first enumerate all possible scenarios and then formalize them using mathematical and coding structures. Consider our case in which we need to select from 7 distinct colors and allocate them among three groups, ensuring that no color is repeated across groups. One extreme scenario involves each group receiving a unique color: the first group is assigned 1 color, the second group receives another one, and the last group (comprising a single individual) is given another one among the remainder. Conversely, an alternative scenario entails all members within a single group having different color coding, calling for the selection of all colors from the available 7 options and distributing them accordingly. Between these extremes lie various possibilities. For instance, the first group may consist of 2 colors while the second group comprises 1 color, or vice versa. Additionally, both the first and second groups could each contain 2 colors. Visualizing these scenarios provides insight into the range of possibilities:



According to the latter, in scenarios where we have “n” colors in one group and “m” in another, we observe that permutations within each cluster do not alter the overall distribution outcome. Therefore, when both groups contain the same number of colors, we multiply the total number of scenarios by two to account for the possibility of interchanging the positions of the groups, resulting in equivalent distributions.

However, if both groups have the same number of colors but differ in color compositions, permutations within each group are considered equivalent. Therefore, in these cases, we do not multiply by two. In particular, in the last case where all 7 colors are utilized and distributed across all 3 groups, we are essentially arranging the colors to fit the groups while ensuring no color repetition between groups and this represents the maximum utilization of available colors.

Mathematical Depicting

Our initial case involves using three colors for the first group, followed by one color for the second group (with an extra color for the last person). This requires selecting three colors from seven for the first group and distributing them accordingly, followed by multiplying by a factor of four for the second group and repeating the process for the last group. Note: this scenario is then doubled as it's equivalent to using three colors for the second group and one color for the first.

$$\binom{7}{3}(3!)(4 \cdot 3 \cdot 2) = \frac{7!}{3!4!}3!\,(4 \cdot 3 \cdot 2) = 7!$$

Now, we'll examine the scenario where the first group comprises three different colors while the second group consists of only two. We begin by selecting three items from our initial pool of seven colors. Then, from the remaining four colors, we choose two and arrange them in any order, incorporating an additional three factorial permutations. Finally, we select from the two remaining colors for the last member. Notably, the process of selecting two elements out of three and later distributing them is equivalent to distributing two elements in three spaces, particularly when one is repeated. Following the **first selecting and then arranging principle**, we first choose two colors from the four left for the second group (4 choose 2), then arrange these colors in the three spaces (3 choose 2), and multiply by two factorial for the permutations. This result is then doubled to account for the same scenario with the second group swapped for the first.

$$\binom{7}{3}(3!)\binom{4}{2}(3!)(2\cdot 2) = \frac{7!}{3!4!}3!\frac{4!}{2!2!}(3!)(2\cdot 2) = 7!\,3! = (6)7!$$

Recalling another scenario, we'll take advantage of all the colors in our palette. Thus, we'll make three selections out of seven for the first group and arrange them in any order, incorporating an additional three factorial permutations. Then, we'll choose from the remaining colors to distribute accordingly among the other members. Notably, this process is essentially the same as choosing from the seven colors how to distribute them in any way, resulting in precisely seven factorial permutations!

$$\binom{7}{3}(3!)\binom{4}{3}(3!)(1) = \frac{7!}{3!4!}3!\frac{4!}{3!1!}(3!) = 7!$$

Let's consider the scenario where we utilize two colors for the first group and a single one for the second group. We begin by selecting 2 colors out of the initial 7, followed by choosing 1 from the remaining 5 for the second group (multiplying the first selection by 5). Then, we multiply this by 4 to account for the last group. Lastly, we double this operation (as in previous cases) to accommodate the possibility of applying the same scenario with groups 1 and 2 reversed.

$$\binom{7}{2}(3!)(5\cdot 4\cdot 2) = \frac{7!}{2!5!}(3!)(5\cdot 4\cdot 2) = 7!$$

We will also account for the case in which we make use of 2 colors for the first group and another 2 for the second one. For this task, we apply the combinations operator by choosing 2 out of seven for the first cluster, and choosing again 2 out of 5 colors left for the second set of people. Remember to multiply by the factorials to account for all the possible arrangements of people once their colors are assigned. We finish this case by multiplying by 3 to measure the possibility of selecting a color for the last person out of the 3 left.

$$\binom{7}{2}(3!)\binom{5}{2}(3!)(3) = \frac{7!}{2!5!}(3!)\frac{5!}{2!3!}(3!)(3) = \frac{7!}{2!}\frac{3!}{2!}(3) = (9/2)7!$$

Of course, our last and simplest case is the one in which all group members inside the same cluster are assigned the same color. This means we have seven choices for the first group, 6 for the second one and 5 left for the last person.

$(7 \cdot 6 \cdot 5) = 210$

Adding these results together, we get to a total of: $\frac{325}{24}7! = 68250$. One notices how number seven plays a crucial role in the answer, which matches the number of colors available in our problem.

Deployment in Python

We will deploy the information concerning this problem into Python to establish a method for verifying the correctness of our results. Utilizing a programming language—though this process is equally feasible in R, Julia, and others—we will construct our scenario, adhering to the defined restrictions, and then enumerate the observations within our final sample.

1) We will generate a dataframe which contains all possible combinations of 7 digit numbers using our 7 color possibilities. To do this, we first input this arrangements into a list and then pipe this into an structured data frame to ease further operations

```
import pandas as pd
import itertools

all_combinations = [''.join(map(str, comb)) for comb in
itertools.product(range(1, 8), repeat=7)]

df = pd.DataFrame({'Combinations': all_combinations})
```

2) We will split our combinations into different groups according to the problem description. Let's take advantage of regular expressions to extract the first 3

digits, second 3 digits and last digit (accounting for the last person). Recall “^” will indicate our expression to read from the beginning of the line whereas “$” will split the data from the end of the string. By making use of lambda functions we can easily construct a pipeline which feeds the next operation and makes all these operations in our data at once.

```
df = df.assign(Group_1 =
df.Combinations.str.extract("(^\d\d\d)")).assign(Group_2 = lambda x:
x.Combinations.str.extract("(\d\d\d\d$)")).assign(Group_2 = lambda x:
x.Group_2.str.extract("(^\d\d\d)")).assign(Group_3 = lambda x:
x.Combinations.str.extract("(\d$)"))
```

3) Our last step will be to define a function that checks if there are any duplicated digits (colors) between groups and assign a “True” attribute if this is the case. We will have to apply this function for all group combinations, this is, Group 1 vs Group 2, Group 1 vs Group 3 and Group 2 vs Group 3. Following this technique, we will filter for instances of the dataframe where we find such an attribute equal to “False” in order to keep the data rows that match the problem constraints.

```
def check_duplicates(row):

    group_1 = row['Group_1']
    group_2 = row['Group_2']

    for i in group_1:

        if i in group_2:

            return True

    return False  # Return False if no duplicates are found

# Apply the function row-wise to the DataFrame

df["first_second"] = df.apply(check_duplicates, axis=1)

df = df.loc[df.first_second == False]
```

The conclusion we've drawn from our analysis is that the solution to our problem, as verified through both mathematical-statistical approaches and programming in Python, is 68,250. While there are numerous methods to verify this result using programming, the approach provided was chosen for its intuitiveness and ease of understanding. Additionally, there are various analytical methods to reach the same conclusion using different combinatory operations. It's important to remember that calculating $\binom{7}{3}(3!)$ (which represents choosing 3 items from 7 and arranging them in order) yields the same result as performing P(7,3) directly, which calculates the number of permutations of 7 items taken 3 at a time. Both operations fundamentally compute the same quantity, demonstrating the flexibility and interconnectedness of combinatory principles.