For counting problems we have 3 basic formulas

- The counting principle which states if there are n_i choices at stage i, then the total number of choices for all stages is
$$n_1 \cdot n_2 \cdots n_j.$$
- The number of *permutations* of size r from n *distinct* objects is
$${}_nP_r = n \cdot (n-1) \cdot \cdots (n-r+1) = \frac{n!}{(n-r)!}.$$
  A permutation is a selection where the order of selection is important.

  A special case is a permutation of size n or simply a permutation or reordering. There are $n!$ of these.
- The number of *combinations* of size r from n distinct objects is "n choose r" or
$${}_nC_r = \frac{{}_nP_r}{r!} = \frac{n!}{(n-r)!r!} = \binom{n}{r}$$
  A combination is a selection where the order is not important. For example, when dealing cards or the lottery numbers.

To implement these on the computer, you need to learn two functions: `prod()` to multiply terms together and `choose()` to implement the formula for combinations.

1 Using `prod()` to find factorials

There is no factorial function built in, but we can use the `prod` function to find factorials.

For example, to find 3! we can enter in the numbers 1,2,3 and then multiply with `prod()`

```
> x = c(1,2,3)
> prod(x)
[1] 6
```

This is a little silly as we entered the numbers 1,2,3 by hand. Its better to let the computer find sequences for us. Expressions like `a:b` will give a list of numbers from `a` to `b`.

```
> 1:3
[1] 1 2 3
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> 10:1
 [1] 10  9  8  7  6  5  4  3  2  1
> 10:(10-5+1)
[1] 10  9  8  7  6
```

So to find a factorial is easy. We can find 7! by

```
> prod(1:7)
[1] 5040
```

An for any n, we can assign it first, then find the factorial

```
> n = 10
> prod(1:n)
[1] 3628800
```

1.1 Find the following factorials

1. Find 69!
2. Find 70!. Why do you think 69! is the largest you can find on most calculators?
3. In a class of 30 students, the number of different arrangements for the top 5 students is ${}_{30}P_5$. What is this number? (Divide factorials)

2 Using `choose()` for binomial coefficients

The `choose()` function will compute $\binom{n}{k}$. For example, there are "52 choose 5" different poker hands. This number is found with

```
> choose(52,5)                  # n first, k second
[1] 2598960
```

That's over 2.5 million.

2.1 In a lottery with 52 balls, it is typical to choose 6 of them. How many different combinations are possible?

2.2 A group of 10 basketball players splits into two teams of 5. How many different ways can this be done?

3 Lotteries

Is a state lottery deserving of the nickname "stupid tax"? Let's compute the probabilities and expectations of the lottery and see.

First, a generalized lottery consists of the following

- There are N lottery balls.
- We select a of them to be drawn.
- A drawing is held where b balls are drawn at random without replacement.

What we want to do is count is to compare the balls we selected with the ones actually selected and count the matches.

Mathematically, this is easy to do if we think of our selection are marking a of the N balls with a color, say red, and leaving the other $N-a$ a different color, say blue.

Then when they select b balls, we want to know how many red ones are in their selection.

How can we count this? For example, how many ways are there to match exactly 1 ball? In there sample of b they must choose exactly 1 from the red ones and $b-1$ from the blue ones. Break this into two stages. There are "a choose 1" ways to select the red one, and "$N-a$ choose $n-1$" ways to pick the blue ones. That is

$$\text{number of ways } = \binom{a}{1}\binom{N-a}{n-1}$$

How does this change if we want to match k balls where k is unspecified? Well from the a red balls we need to choose k, and from the $N-a$ blue ones, we need to choose $n-k$. Thus

$$\text{number of ways to match exactly } k \; = \binom{a}{k}\binom{N-a}{n-k}$$

Now, to get *probabilities*, we need to divide by the total number of ways to choose n balls from N. This is simply $\binom{n}{N}$. So we get

$$\text{Probability of matching exactly } k \text{ balls } = \frac{\binom{a}{k}\binom{N-a}{n-k}}{\binom{N}{n}}$$

For example, in a simple lottery with 10 total balls where we mark 5 and then 6 are selected the probability of 3 matches and then 1 match is

```
> choose(a,k)*choose(N-a,n-k)/choose(N,n)
[1] 0.4762
> choose(a,k)*choose(N-a,n-k)/choose(N,n)
[1] 0.02381
```

So we see in this silly example, it is much more likely to get 3 matches than 1.

3.1 For a lottery of 80 balls, where we mark 10 and then 20 are chosen. What are the probabilities of matching exactly 0,1 and then 2 balls? Which is more likely?

3.2 Can you explain the output of these commands

```
> N = 80; n = 20; a = 10; k = 0:a
> choose(a,k)*choose(N-a,n-k)/choose(N,n)
 [1] 4.579e-02 1.796e-01 2.953e-01 2.674e-01 1.473e-01 5.143e-02
 [7] 1.148e-02 1.611e-03 1.354e-04 6.121e-06 1.122e-07
```

4 expectation

Now to decide if playing the lottery is a "stupid tax". Suppose you are reimbursed the following amounts in a simple lottery where there are 10 balls, we mark 4 and 6 are selected

- If you match 0,1 or 5 you get paid $1.
- If you match 2 or 3 you must pay $1.

What is the expected amount you make per game?
The answer is found by multiplying the payout times the probability of the given value.
The probabilities are

```
> N = 10; n = 6; a = 4; k = 0:a
> choose(a,k)*choose(N-a,n-k)/choose(N,n)
[1] 0.004762 0.114286 0.428571 0.380952 0.071429
```

So we get the expected value is

$$1 \cdot 0.004762 + 1 \cdot 0.114286 - 1 \cdot 0.428571 - 1 \cdot 0.380952 + 1 \cdot 0.071429.$$

Don't rush to the calculator yet. We can do this easily as follows. Make a string of numbers 1,1,-1,-1,1 and then multiply and add

```
> pay = c(1,1,-1,-1,1)
> probs =  choose(a,k)*choose(N-a,n-k)/choose(N,n)
> sum(pay * probs)
[1] -0.619
```

So we **lose** about 62 cents, on average, each time we play.

4.1 Now a more realistic example. Suppose there are 52 lottery balls, we mark 6 and 6 are selected. We get paid as follows

- If we match 0,1,2 or 3 we must pay $1.
- If we match 4 we get paid $10
- If we match 5 we get paid $1,000
- If we match 6 we get paid $1,000,000

What is our expected return? (Use

```
> pay = c(-1,-1,-1,-1,10,1000,1000000)
```

4.2 Suppose there are only 35 balls instead of 52, but otherwise the game is the same. What is the expected return now?

4.3 What is different if there are only 30 balls, but all other things are equal.