

EPFL

# Microsimulation



We have seen that aggregation consists in filling a table where each row corresponds to an individual, each column corresponds to an alternative, and within each cell, we apply the choice model for the individual and the alternative. The problem is that this table can be very big. In particular, when the number of alternatives is very large, it may be difficult to maintain in a big database all possible values of the probability. In this case, we'll rely on microsimulation. And this is the topic of this video, is to understand what microsimulation is all about.

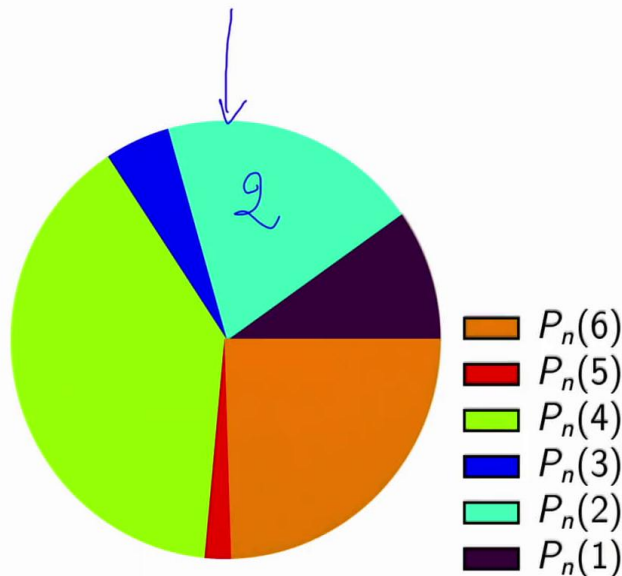
Notes

Summary



0m 04s

# Microsimulation



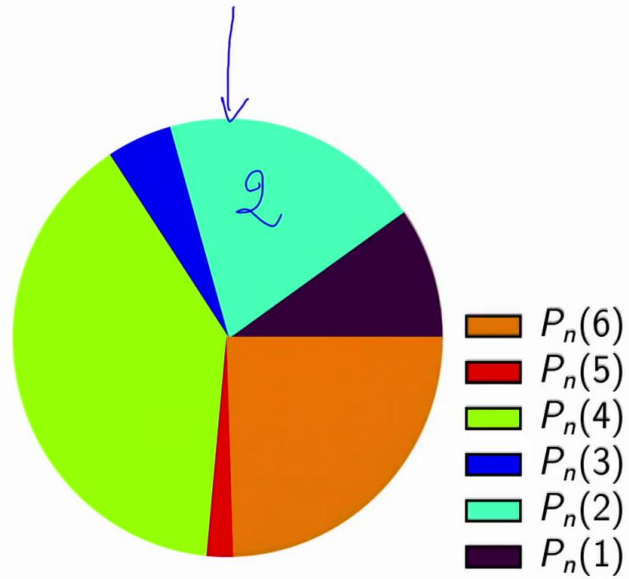
The idea of microsimulation is that, instead of looking at the choice probability for each of the available alternatives in the choice set, we will decide somehow what is the actual choice that a given individual is supposed to make. To do that, let's assume that we organize the choice probabilities as a pie chart. In this example, you have an example with six alternatives, and you can represent them in a circle. And the area in the circle is proportional to the choice probability. Now, suppose that you play roulette wheel. Then, you throw randomly a number, which is equally distributed around the circumference of this circle. So, clearly, the roulette wheel will select one of these six sections. But the probability to lie in a section will be exactly the choice probability. So, therefore, for a given individual  $n$ , we'll throw the wheel, the roulette wheel, and depending on where the wheel is stopping, we will decide that this is the alternative that this individual has chosen. So in this special case, the individual is supposed to select alternative 2, which is the one corresponding to this. This is the concept of microsimulation. Of course, we know that individual  $n$  will not always choose alternative two.

Notes

Summary



# Microsimulation



The model tells us that the individual will select alternative two with some probability. And actually, the next time we throw the same wheel, with the same probability distribution, we may end up in another area of the circle and, therefore, end up with a different chosen alternative. It means that this exercise will have to be repeated several times, because one realization only is just due to chance, to pure random events. So the microsimulation will have to be repeated several times.

Notes

Summary



2m 12s

# Microsimulation

## Simulated choice

- ▶ For each observation, draw  $R$  times from the choice model.
- ▶ Define  $\hat{y}_{inr} = 1$  if alternative  $i$  has been generated by draw  $r$ , 0 otherwise.
- ▶ Approximation:

$$P_n(i|x_n; \theta) \approx \frac{1}{R} \sum_{r=1}^R \hat{y}_{inr}.$$

## Warning

It is **invalid** to select the alternative with the highest probability.

So for each individual, for each observed values of the explanatory variables, we will generate  $R$  times a choice. So we will draw (what we call "draw" is throwing the roulette wheel as we saw in the previous slides) we will draw  $R$  times from the choice model. And for each draw  $r$ , we call  $\hat{y}_{inr}$  the indicator that says "has individual  $n$  selected alternative  $i$  during draw  $r$ , or not?" It will be one if yes, it will be zero otherwise. And now, using these generated choices, these simulated choices, we can actually approximate the value of the probability. The probability will be simply the average number of times that this alternative has been selected. And it can be shown, actually, that when capital  $R$  goes to infinity, this will become an equality. So this is called "microsimulation". And I want to emphasize here the fact that, whenever you need a realization of the choice, this is how you should do it. I have seen several people doing something different. They would associate with a choice model, the alternative with the highest probability. This is wrong. Because such an application would always predict the same alternative, one hundred percent of the time.

Notes

Summary



# Microsimulation

## Simulated choice

- ▶ For each observation, draw  $R$  times from the choice model.
- ▶ Define  $\hat{y}_{inr} = 1$  if alternative  $i$  has been generated by draw  $r$ , 0 otherwise.
- ▶ Approximation:

$$P_n(i|x_n; \theta) \approx \frac{1}{R} \sum_{r=1}^R \hat{y}_{inr}.$$

## Warning

It is **invalid** to select the alternative with the highest probability.

Which is not correct, because we know that each of the alternatives has a non zero probability to be selected. So, it is really invalid to select the alternative with the highest probability. You need to apply this roulette wheel technique, this microsimulation technique, in order to generate one realization of the choice.

Notes

Summary



## Aggregate market shares

Number of individuals choosing alternative  $i$

$$N(i) = \frac{1}{R} \sum_{n=1}^N \sum_{i=1}^R \hat{y}_{inr}.$$

Share of the population choosing alternative  $i$

$$w(i) = \frac{N(i)}{N} = \frac{1}{N} \frac{1}{R} \sum_{n=1}^N \sum_{i=1}^R \hat{y}_{inr}.$$

So now, once you have generated this, but then it is very simple. What is the number of individuals choosing alternative  $i$ ? Well, you just count. For each individual, you count the average time that this alternative has been chosen, by simply summing all the  $y$ 's, and dividing by  $R$ . And if you want the share of the population, you simply divide  $N(i)$  by capital  $N$ , which is the total size of the population. And this is equal to  $W(i)$ , the market share in the population for alternative  $i$ .

Notes

Summary



4m 32s

# Microsimulation

For each  $r$

Population	Alternatives				Total
	1	2	...	$J$	
1	$\hat{y}_{11r}^0$	$\hat{y}_{21r}^1$	$\dots^0$	$\hat{y}_{J1r}^0$	1
2	$\hat{y}_{12r}$	$\hat{y}_{22r}$	$\dots$	$\hat{y}_{J2r}$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$N$	$\hat{y}_{1Nr}$	$\hat{y}_{2Nr}$	$\dots$	$\hat{y}_{JNr}$	1
Total	$N(1)$	$N(2)$	$\dots$	$N(J)$	$N$

So let's go back to the table that we have seen before. So remember, this table has one row per individual, and one column per alternative. So each time we do a microsimulation, so each time we generate a choice, basically we calculate the values of these  $\hat{y}$  hat. But now, these values can be zero or one, for a given  $r$ . For example, if individual one has selected alternative two, we will have a one here, zero, zero, and zero everywhere else. Again, as before, the sums will have the same interpretations as before. So for each row, it will sum up to one, because only one alternative has been chosen, and then, for each column, it will calculate the number of people who have selected that alternative for the given draw  $r$ . But the advantage now is that this matrix, this big table, contains a lot of zeroes. Actually, for each row, it contains  $J-1$  zeroes. So it's definitely inefficient to store it in that way, and we can make it more compact. And this is the whole point of doing microsimulation.

Notes

Summary



5m 09s



# Microsimulation

## In practice

Population	Draw			
	1	2	...	$R$
1	$i_{11}$	$i_{12}$	...	$i_{1R}$
2	$i_{21}$	$i_{22}$	...	$i_{2R}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$N$	$i_{N1}$	$i_{N2}$	...	$i_{NR}$

Typically, what we do, is that all these tables are gathered together in one table like this, where now we have one row for each person in the population as before. But now, we have one column per draw. And this column only contains the label of the alternative which has been chosen for this draw. And, therefore, the complexity of this table, is controlled by the analyst. Because the capital  $R$  is your decision. Even if the number of alternatives is high, you can keep the size of this matrix (in terms of number of columns) relatively low. And this is the whole point of microsimulation.

Notes

Summary



6m 24s



So this is how we deal with the problem that occurs when this table has too many columns. The most important problem is when the table has too many rows. And this happens almost all the time, when the number of people in the population is really large. And to do this, we need to rely on something called "sample enumeration". And this is the topic of the next subject.

Notes

Summary



7m 04s