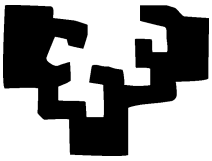
eman ta zabal zazu



Universidad del País Vasco

Euskal Herriko Unibertsitatea

Activity 1

1 Synopsis.

What we are set about to do. In a previous seminar you saw a simple example on how to estimate things by simulation, using the Monte Carlo method. In this activity you will put to good use to find the distribution of a statistic that would be analytically intractable.

What you need. You need to be fully acquainted with the content of previous seminars and practice assignments. You will also need access to a computer equipped with R.

2 Problem description

Tadpoles are amphibians that usually live in the water; frogs in the early stages of their lives are tadpoles which undergo a metamorphosis to reach their adult form.

Zoologists have been interested in knowing whether tadpoles display some sort of social instinct and like the vicinity of one another. If that were the case, the average distance between tadpoles set free to swim in an open space would be smaller than it would be if they swam at random, not caring for their likes.

A simple experiment has been carried out as follows: at a random time, a picture is taken of a group of n tadpoles in a given area, and the average distance between them computed. (Refer to Figure 1 for details.) Care is taken that the area is homogeneous, with no differences of temperature or concentration of nutrients than might explain some clustering of the tadpoles unrelated to their hypothesized social instinct.

The problem, though, is that although some theoretical results exist about the average distance D between n points thrown at random on a regular area of R^2 (such as rectangle), we don't know much about what values of D to expect. The following guidelines suggest a way of finding out by means of simulation.

3 Finding the distribution of D by simulation

1. Assume that our picture contains 20 tadpoles in a rectangle of 15×10 cm.

2. Define a matrix `coord` of 20 rows and 2 columns, and fill it with zeros:

```
> coord <- matrix(0,nrow=20,ncol=2)
```

Figure 1: Measuring distances between tadpoles. From a still photograph such as the one below, distances from each tadpole t_i to all others can be measured, and the average $D^* = \sum_{i \neq j} d(t_i, t_j)/\binom{n}{2}$ computed.

3. Define a vector `D` of length $N = 500$ where you will save some values.

```
> N <- 500
> D <- rep(0,N)
```

4. Repeatedly do the the following for i from 1 to N.

   (a) Generate 20 random points in a rectangle 15×10 cm. You only have to generate the first coordinates as $U(0, 15)$ and the second coordinates as $U(0, 10)$. Keep the coordinates of each point in a row of `coord`.

   (b) Compute all the (ordinary, euclidean) distances between rows of `coord` and their average. You could do it by looping over the rows of `coord`, but it is much easier to use the R function `dist`; see the hints below.

   (c) Save the results in element i of `D`.

5. At the end of the N iterations, plot the histogram of the values you have generated in `D`. That gives you an idea of what to expect from the average distance between 20 tadpoles if they swim "at random". If the true average distance D^* computed form a picture such as that in Figure 1 is much smaller that expected, you would have reason to reject the hypothesis of random positions of the tadpoles.

4 Hints and comments

1. When using random numbers, always start your program with a sentence such as

   ```
   > set.seed(12345)
   ```

   This will ensure that your code gives always the same answers when invoked with the same inputs. It doesn't matter which value you give to the seed.

2. When you need to repeat something a large number of times, you use a `for` loop. Most of the work you need to do here would be enclosed in a loop such as:

   ```
   >   for (i in 1:N) {
          ...
       }
   ```

3. Generating the random coordinates with uniform distribution inside the rectangle can be done with function `runif`. The easiest (and fastest) is to generate all first coordinates and all second coordinates at once:

   ```
   > coord[,1] <- runif(n=20, min=0, max=15)
   > coord[,2] <- runif(n=20, min=0, max=10)
   ```

4. As mentioned above, R has a function `dist` that will compute at once all distances among the rows of a matrix. (You can obtain details typing `help(dist)`.) You could use:

   ```
   >   distances <- dist(coord)
   ```

   This returns you an object which is not a full matrix: it would be wasteful to return twice the same thing (distances from point j to k and from k to j are equal). You can obtain all distances in vector form using the following recipe:

   ```
   > dis.matrix  <- as.matrix(distances)
   > dis.vector  <- dis.matrix[ lower.tri(dis.matrix) ]
   ```

5. Each time you do this, `dis.vector` will contain all distances among pairs of simulated points. If you are at iteration i, save the mean of this distances in element `i` of `D`.

6. As you exit the loop over `i` you have in `D` $N = 500$ "observations" artificially generated of the average random distance between 20 tadpoles in a 15×10 cm rectangle. If the true observation D^* computed as described in Figure 1 looks smaller than expected, you would have reason to reject the hypothesis of random swim. Plotting the histogram only requires code as:

   ```
   > hist(D)
   ```

7. The problem presented is just a (very) simple case in which simulation is used to obtain some insight on what would be expected under certain conditions. Loosely speaking, the reasoning is: "If A (= tadpoles are not social) is true, what experimental evidence (= inter-tadpole distances when they swim freely) would I expect ?" If the evidence you find is substantially different, you would have good reason to doubt A is true.

   The good news is that in many practical situations we can determine theoretically what can be expected (and learning just how to do that will occupy us for much of the course). Only in situations that become analytically intractable do we have to resort to simulation.

8. These situations are common in practice. It was mentioned in the introduction that theoretical results exist for average distances in a regularly shaped region. But as soon as the problem becomes more involved (for instance, if there is a stone in the middle of the region, emerging from the water) we have to use simulation.

9. We will not be simulating real life models (there is a subject on Operations Research where you can learn about this sort of problems), but will make further use of simulation to study properties of estimators in Activity 2.

10. You can present your work in any form you wish, as long as you show your results right next to the code used to generate them. For this, you can hardly do better that use R Markdown from inside Rstudio, in the manner demonstrated in class.