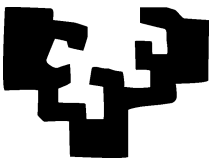
eman ta zabal zazu



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 your new skill to assess different alternatives in staffing a (highly simplified) hospital care service.

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

You have to size a medical urgent care facility to service a certain district. We will deal with an over-simplified situation. You may assume the following:

1. Hourly patients arrivals are randomly distributed as Poisson $P(\lambda = 10)$, each and every hour, day and night. All patients arrive at the start of each hour.
2. The number of patients a doctor can service per hour is also random, Poisson-distributed with mean $\mu = 1.2$; the work of each doctor proceeds independently of each other. After being taken care, patients are discharged.
3. If in a given hours more patients arrive than can be discharged, they have to wait for service.
4. You, as a manager, are required to set the number of doctors on duty so that the number of people waiting for attention does not exceed 5, except in very rare occasions that should occur less than 1% of the time.
5. For the purposes of this problem, we will only look at what happens at hourly boundaries. The number of people waiting at the end of hour t is defined as the number of people waiting at the end of hour $t - 1$ plus the number of arrivals minus the number of discharges.

3 Questions

1. What is the minimum number of doctors necessary to achieve the goal in 4 above?
2. What would happen in $\lambda > \mu \times d$ with d the number of doctors on duty?
3. What is the average queue length with 10 doctors? With 9?
4. If instead of being told that arrivals are Poisson-distributed with $\lambda = 10$ you were told that the population served is 1.000.000 persons and each has a probability $p = 1.3 \times 10^{-5}$ of requiring urgent care in a given hour, independently of each other, what would you choose as the distribution for the hourly arrivals?

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 the seed.

2. Clearly, the length of the queue at time t, Q_t, the number of arrivals, A_t, discharges, D_t, and the length of the queue at time $t-1$ are related by:

$$Q_t = Q_{t-1} + A_t - D_t \tag{1}$$

3. You can simulate arrivals and services per hour, for different numbers d of doctors on duty. Notice the independence assumption about the number of patients each doctor can serve hourly and use it to obtain the distribution of the total number of people they can serve per hour.

   At the beginning, you can set $Q_0 = 0$; this affects the length of the queue at $t = 1, 2, \ldots$, which may not be very realistic for small t. What is usually done is to discard the first observations (a so-called *transient*) and use the remaining.

4. You can simulate, for instance, 2448 hours and discard a transient made of the first 48. (You can simulate longer than that if you wish: the only downside is that it will take more time.)

5. For each hour, you have to

   - Generate a random Poisson with $\lambda = 10$, A_t.
   - Generate a random Poisson with $\lambda = 1.2 \times d$, D_t.
   - Compute the new length of the queue Q_t using relation (1).
   - Save the length of the queue, so yo can compute its average after 2400 hours (= 2448 - 48).
   - Notice that the queue can never be negative: there cannot be -2 persons waiting! When you compute Q_t using relation (1), you should convert any negative values to zero. The function `max` can help with that. Look:

     ```
     > max(3,4,5)
     [1] 5
     ```

```
> max(0,-2)
[1] 0
```

6. You can start by initializing a vector `Q` of length `N=2448` (or whatever you choose). Fill it with zeros, for instance:

   ```
   > N <- 2448
   > Q <- rep(0,N)
   ```

7. Once you have that vector , you have to write a loop which for `i` going from 2 to `N` computes relation (1).

8. Poisson random variables with any λ are easy to generate by means of function `rpois`; type `help(rpois)` to check the syntax.

9. Once you have simulated the length of the queue in `Q` for all times, you can easily drop the first 48 transient observations. Do something like:

   ```
   > Q <- Q[-c(1:48)]     # drops the first 48 observations
   ```

   or equivalently

   ```
   > Q <- tail(Q,n=-48)  # drops the first 48 observations
   ```

10. Counting how many times you have had a queue length greater than 5 is now quite easy.

    ```
    > L <- sum(Q > 5)
    ```

    Explanation: `Q > 5` evaluates to a vector of values `TRUE` of `FALSE`, according to whether the corresponding element of `Q` is or is not greater than 5. When we invoke the function `sum` on these vector, `TRUE` and `FALSE` are automatically converted to 1 and zero, so in `L` we get the total number of hours with queue lengths greater than 5.

11. If `L` is less than or equal 1% of the hours, you are all right. Otherwise you have to try with more doctors.

12. The computation of the average length of the queue is just as easy: use function `mean`.

13. The problem presented is just a (very) simple queuing model. It is simple enough that can be solved analytically. In practice, though, things are always more complicated: arrivals and service times vary over time, service times are not independent (for instance, two doctors may require the service of the X-ray unit at the same time in their diagnostic) and many other complexities appear: simulation is then the tool of choice.

14. 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.

15. 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.