Introduction:

This project will analyze the situation presented to Dr. Weems, a physician with several years of experience in the field of pain management in chronic pain patients. We will use as a tool the discrete event simulation and more specifically the Jaamsim software, first to put into practice the operation of a private clinic where Dr. Weems worked which will allow us to better understand the situation and also allow us to evaluate the correct functioning of the simulation and corroborate how the practices of Dr. Weem allowed him to improve the operation of the private clinic and secondly to use the experience of Dr. Weem and through the simulation evaluate how it would impact a new policy in a new much larger clinic where Dr. Weems will begin to work.

Input Data Analysis.

We will now analyze the distributions of the times that will be the input variables in the simulation.

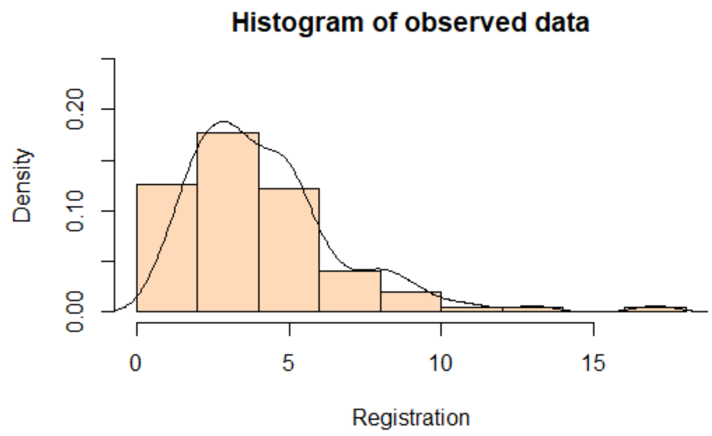
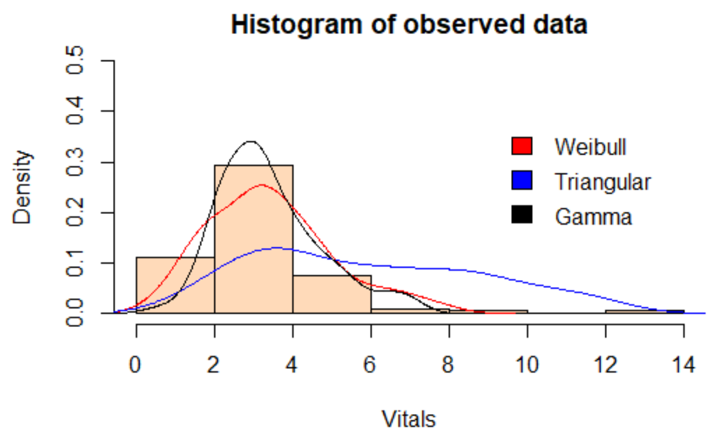
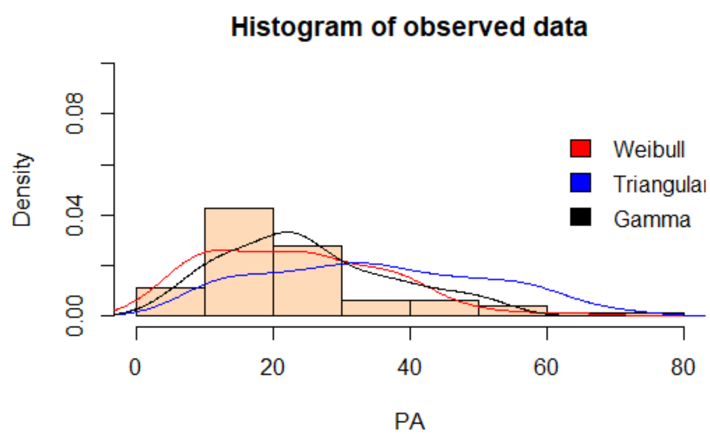
R was used to perform the analyses, mainly using the fitdistrplus and ExtDist libraries, among others. The bestdist function was used to choose the best distribution using the AIC (Akaike Information Criteria) criterion.

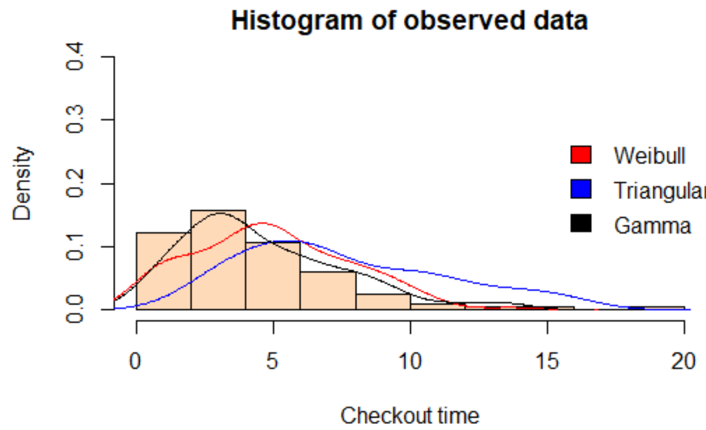
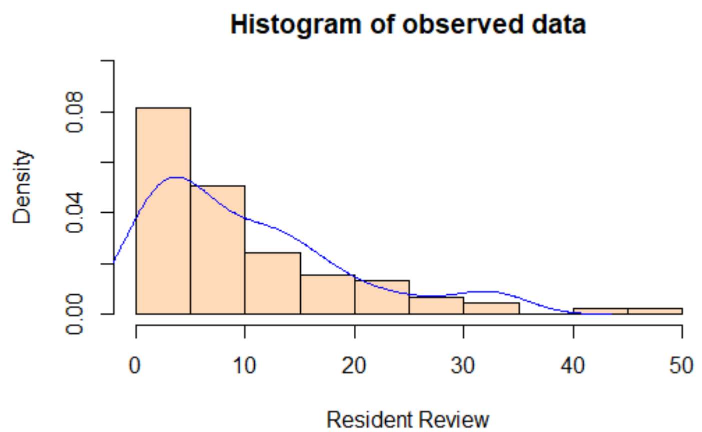
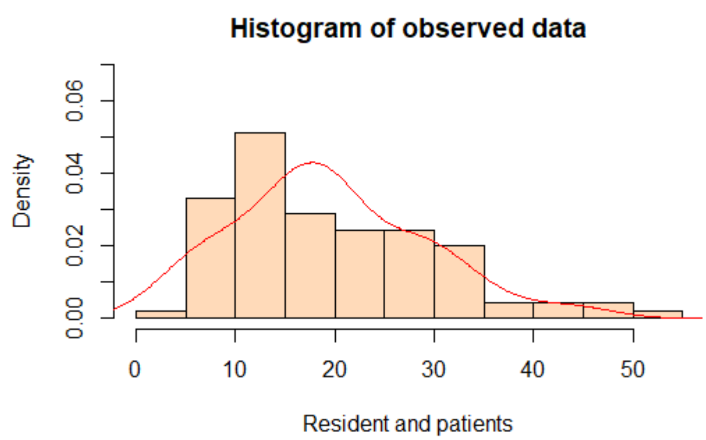
However, in some situations we added some distributions to the graph because the distribution chosen by the program as the best one failed the Smirnov-Kolmogorov test and in other situations because the program could not test them and therefore they were added manually and the decisions were made considering all that was available both the graphical inspection and the results of the bestdist() function and the KS test.

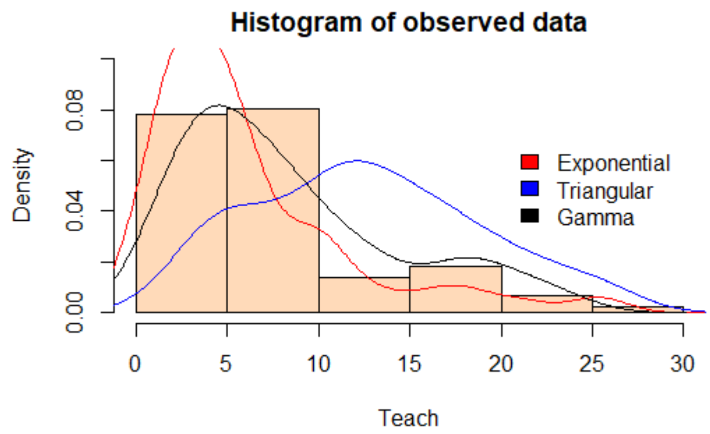
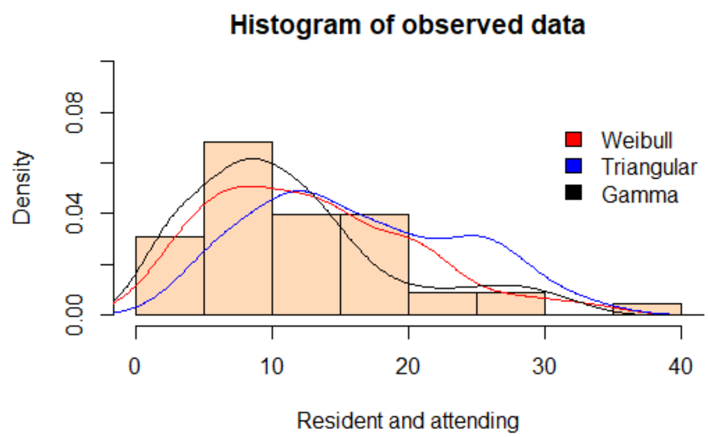
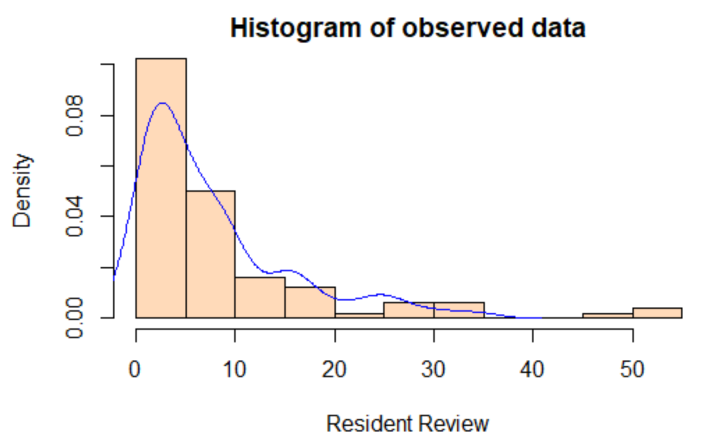
It should also be considered that in the KS test the result was obtained by simulation (as explained in the function itself) which means that if the same function is run several times the same result is not necessarily obtained (i.e. the p-value could be different from the one presented in the table).

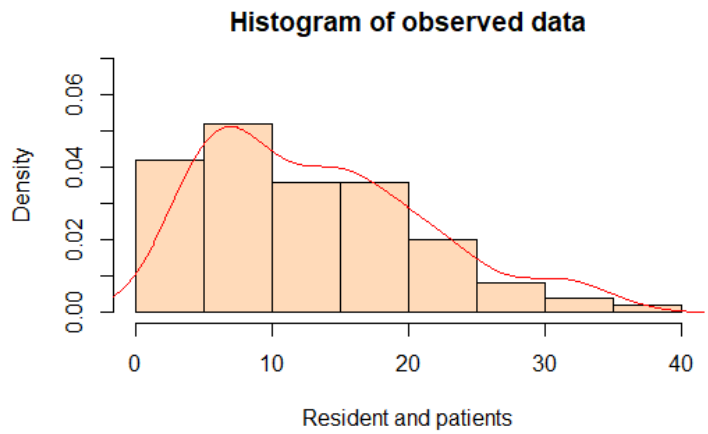
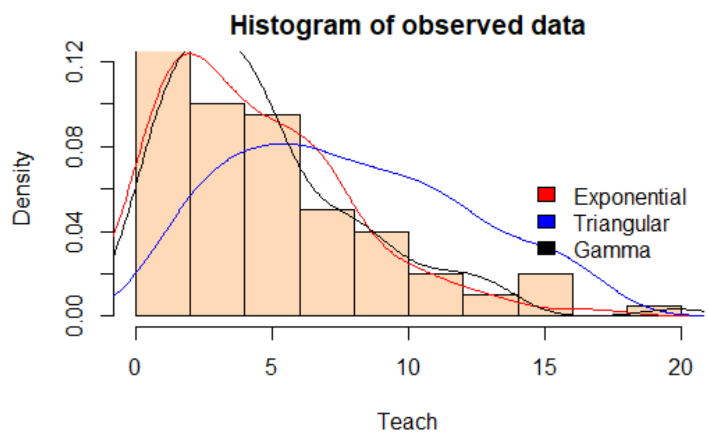
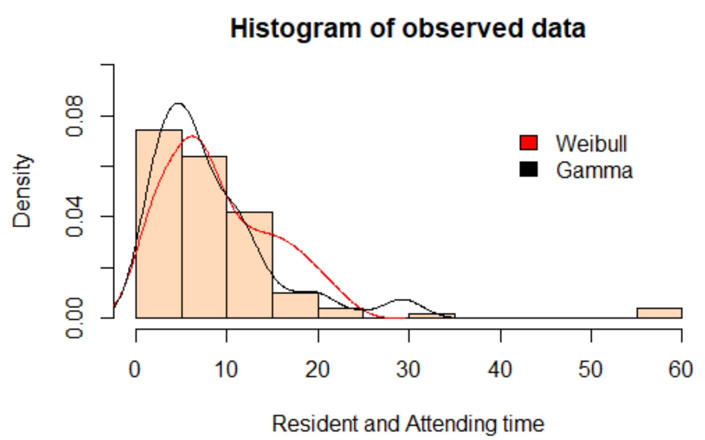
Input data for E-HOC clinic

Description	Histogram	Kolmogorov-Smirnov test p-value	Probabilistic distribution to be used

Registration Time		0.1079	Weibull Parameters Shape = 2.1070178 Scale= 3.9813604
Vitals time		Based on bestdist() function and by graphical inspection	Weibull Parameters Shape = 2.1070178 Scale= 3.9813604
PA		0.07331	Weibull Parameters Shape = 1.8026405 Scale= 25.6339776

Checkout		0.6873	Gamma Parameters Shape = 2.30719868 Rate= 0.48835041
Resident Review time for New patients		0.12	Exponential Parameters rate= 0.09578947 Mean=1/rate
Resident and patient time for new patients		0.4005	Weibull Parameters Shape = 2.002653 Scale= 22.855105

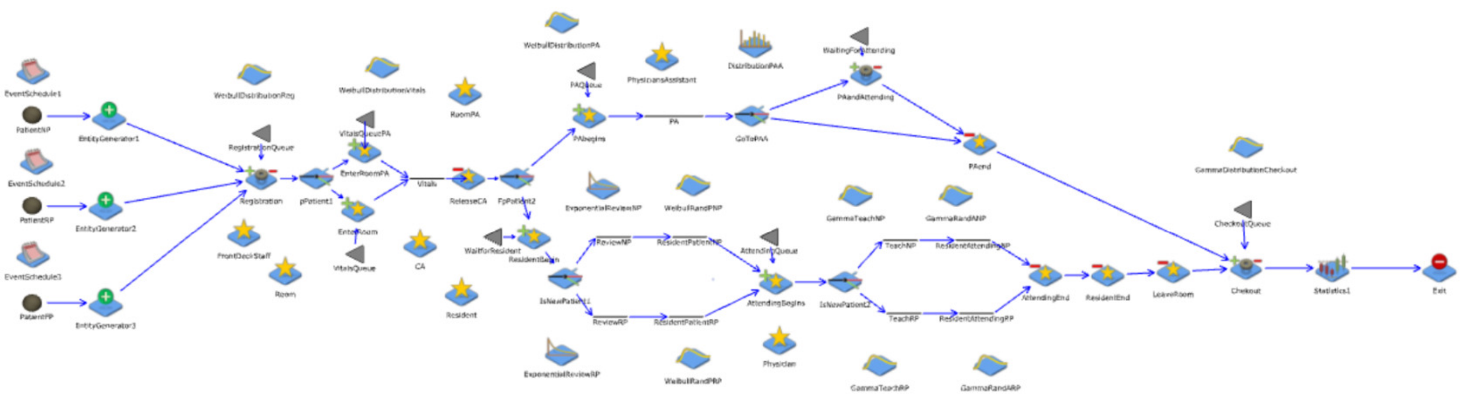
Teach time for new patients		0.4901	Gamma Parameters Shape = 1.9478865 Rate= 0.2481202
Resident and attending for new patients		0.4074	Gamma Parameters Shape = 2.82483794 Rate= 0.22688457
Resident Review time for Return patients		0.281	Exponential Parameter: Rate= 0.10752688

Resident and patients time for return patients		0.4676	Weibull Parameters Shape= 1.7386142 Scale= 14.6522292
Teach time for return patients		0.3667	Gamma Parameters Shape = 1.6173491 Rate= 0.3128335
Resident and attending for return patients		0.281	Gamma Parameters Shape = 1.91459012 Rate= 0.20722056 Mean = shape/rate Source: Wikipedia

Simulation Models

E-HOC clinic model

Let us now describe how the simulation works.



Return patients and follow-up patients) in the model using a calendar, i.e. they are generated using the supplied template. When patients arrive, they go to the registration area, which is modeled by an **Entityprocessor** module that allows me to take the resource, simulate the service and finally release the resource in a single step. Subsequently, we use a **Branch** to separate patients according to their type since Follow up patients use one room and other patients use different rooms, i.e. they are not the same resource. When entering the room, the CA resource is occupied and the Vitals service is simulated with a **Delay** module. The next module is a **Release** module which allows me to release the CA resource and then the patient remains in the room since the room is not yet released. Then a Branch is used again to separate the patients by type, it is important to mention that in order to recognize the type of patient, some attributes were assigned to the units, these attributes were assigned at the beginning when the unit was generated. After this we have 2 branches, the upper one that processes follow-up patients and the lower one that processes the rest of the patients. Let us first describe the upper branch, i.e. Follow-up patients. The next module in this upper branch is a **Seize** module that allows me to occupy the physician's assistant (PA) and then use a delay that will allow me to simulate the time the assistant spends with the patient. Then there is a branch since 50% of the time the main physician is required for this we use an Entityprocessor module that will allow me to require the physician to hold him while he is with the assistant and then release him. Then it goes to another **Release** module that allows me to release the room and the assistant. Now let's look at the bottom. Here after releasing the CA resource we use a Seize module to occupy a Resident and then use another **branch** module to separate the new patients from the return patients since the processing times are different we have then **2 Conveyors** modules that were used this time to simulate the service time for each type of patients, the first one for the **Resident Review** and the second one for the **Resident and Attending**. Then we go to a Seize module where we require the main physician that will be used in the next 2 modules that represent the processes **Teach** and **Resident and attending** and then we have 3 release modules, one to release the physician, one to release the resident and one to release the room, it should be noted that this could be done in a single module but to improve the understanding of the model was done separately. Now we move on to another entityprocessor module to represent the CheckOut and finally we move on to the system output.

OUTPUT ANALYSIS

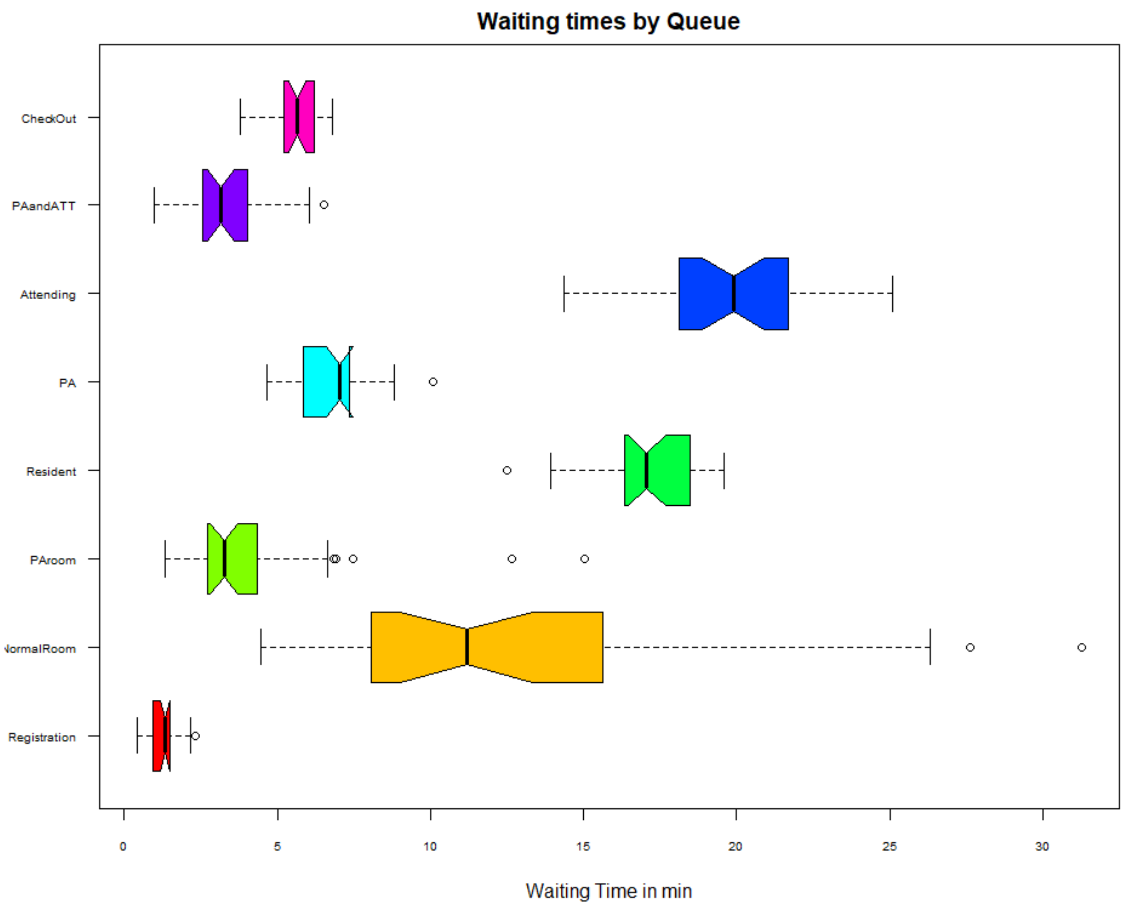
First of all, we must clarify that these analyses are based on 30 replications, which will allow, if necessary, to resort to 95% confidence intervals based on the assumption of normality, since our n would be sufficiently large to make this assumption.

In addition, each replication is for 1 working day so that the 30 replications correspond to 30 days of operation.

The entries to the system were made through the appointment template.

As Dr. Weem described in his approach, the most important measure of customer satisfaction is waiting times, so we will focus our analysis on waiting times.

After doing our simulation run and getting the results we transfer them to an excel sheet and then we read them from R and we can make a graphical comparison of the waiting times.



Here we can see, for example, that the longest waiting times are waiting for the doctor and waiting for the residents. Which means that Dr. Weems' analysis of trying to reduce these times makes a lot of sense.

In our second scenario we will simulate the same situation considering the doctor's ideas and we will reduce the average times by half.

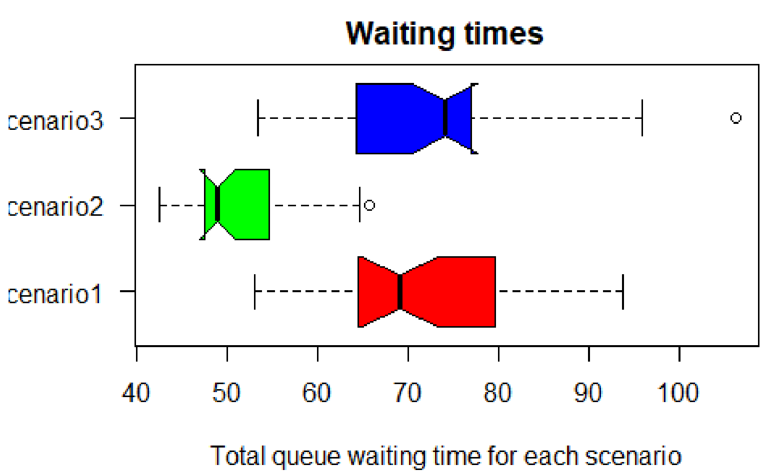
In our third scenario we are going to make the simulation by making a revision in the scheduling template, the idea is to suggest to the staff in charge of making appointments that they should do their best to ensure that in each day the largest number of patients are Follow-up patients, i.e. in this template there are 5 New patients 5 Follow-up patients and 7 Return patients if we can reverse this situation and achieve for example that there are 5 New patients 5 Return patients and 7 Follow-up patients this would also allow us to never quote 2 New patients or 2 Return patients at the same time. Our new schedule would then be as follows:

App Time	Patient Type
0800	New
0800	Follow-up
0815	Return
0830	Follow-up
0845	New
0845	Follow-up
0900	Return
0915	Follow-up
0930	New
0930	Follow-up
1000	Return
1015	Follow-up
1030	Return
1045	New
1100	Follow-up
1100	Return
1115	New

As we can see, this is an approach that is not very complicated to implement.

The files containing the scenarios would be: EHOCclinic.cfg , EHOCclinicScenario2.cfg y EHOCclinicScenario3.cfg, this could have been done in the same file but for the sake of ease of reviewing the scenarios, I have decided to implement them separately.

And now comparing the total time clients spend in queue while in the clinic we have:



Now, the total average queue time would be:

Scenario1(Original) = 71.34 , Scenario2 = 51.27 , Scenario3 = 72.86

Obviously the scenarios that are visually perceived to be different are 1 and 2, but can these differences be considered statistically significant?

So we created an 95% confidence interval for the difference between the average waiting times for scenarios 1 and 2.

(15.29635; 24.84056) as we can see the 2 extremes of the interval are positives Therefore, the average time for scenario 1 is approximately 15 to 24 minutes longer than for scenario 2.

Conclusion

With all of the above, it is reasonable to try to implement the changes since reducing total waiting times by at least 15 minutes is quite significant in terms of customer satisfaction and consequently would have a positive impact on the clinic's reputation.

In today's competitive times, offering a better quality of service means staying in the competitive marketplace and may even help to attract new customers.