**MONTE CARLO SIMULATION**

Monte Carlo methods are a form of stochastic modeling. In the context of mathematics, statistics, and various scientific disciplines, the term "stochastic" is often used to describe processes, systems, or variables that are inherently random or involve some degree of randomness. It is derived from the Greek word "stokhastikos," which means "able to guess" or "proceeding by guesswork," emphasizing the element of unpredictability or uncertainty in stochastic phenomena.

They rely on repeated random sampling to simulate and analyze complex systems or processes. Monte Carlo methods use randomness to generate a range of possible outcomes, which are then aggregated to estimate the properties of the system being modeled, such as probabilities, averages, or other statistical measures. Monte Carlo Simulation uses probability distribution for modeling a stochastic or a random variable.

Monte Carlo simulation is a computerized mathematical technique to generate random sample data based on some known distribution for numerical experiments. This method is applied to **risk quantitative analysis and decision-making problems**. This method is used by professionals of various profiles such as finance, project management, energy, manufacturing, engineering, research & development, insurance, oil & gas, transportation, etc.

**Monte Carlo Simulation ─ Important Characteristics**

Following are the three important characteristics of the Monte-Carlo method −

1. **Random Sample Generation:** Monte Carlo simulations rely on the generation of random samples from a probability distribution to simulate the behavior of a system. This randomness is crucial for exploring the various possible outcomes and their probabilities.
2. **Known Input Distribution:** The probability distribution of the input variables must be known or assumed based on available data or expert knowledge. This distribution is used to draw random samples for the simulation.
3. **Known Result Distribution:** While the exact result of each individual experiment may not be known beforehand, the distribution of the results or the statistical properties of interest (such as mean, variance, and confidence intervals) should be estimable through the simulation. The goal is to approximate these properties through repeated random sampling.

**Monte Carlo Simulation ─ Advantages**

* Easy to implement.
* Provides statistical sampling for numerical experiments using the computer.
* Provides approximate solutions to mathematical problems.
* Can be used for both stochastic and deterministic problems.

**Monte Carlo Simulation ─ Disadvantages**

* Time-consuming as there is a need to generate large number of samples to get the desired output.
* The results of this method are only the approximation of true values, not the exact.

**Montecarlo methods = example 1**

I play a game with p(win)=p

For each round, a game ends when I lose 2 games in a row.

what is the expected number of rounds?( Each round consists of a single game, and the game ends when you lose two rounds (or games) in a row.)

**Analytical solutions**

Le e= E(R)

e= (1) \*p + 2(1-p) (1-p) + ,,,,,,,

**monte Carlo solution**

we can simulate this using a set of random numbers to determine a win or a loss for each round with a specified probability. For example, 0.5 probability of winning and losing.

|  |  |
| --- | --- |
| Probability | outcome |
| >=0.5 | win |
| <0.5 | Loss |

**Random numbers**

0.9, 0.3,0.6, 0.5.0.4.0.3.0.9, 0.7

|  |  |  |
| --- | --- | --- |
| Game number | Rno | outcome |
| 1 | 0.9 | win |
| 2 | 0.3 | loss |
| 3 | 0.6 | win |
| 4 | 0.5 | win |
| 5 | 0.4 | Loss |
| 6 | 0.3 | Loss |

Round 1 -

Game 1 - random1 >=0.5 win

game 2 – random 2 <0.5 loss

game 3 – random 3<0.5 loss

The game ends and starts again

This can be solved by writing a computer program in MATLAB.

**ROUNDS = [ ]**

**For i in [1…..1m]**

**r=0. nloss=0 // initialization**

**While nloss <> 2**

**r=r+1**

**if Rand() >0.5**

**nloss=0**

**Else**

**nloss =nloss + 1**

**ROUNDS APPEND(r)**

**Return (ROUNDS]**

You can try with different probabilities. like 0.5, 0.75.9.0

Montecarlo has been applied in many areas such as finance, inventory, queuing, etc.

**Montecarlo simulation approach**

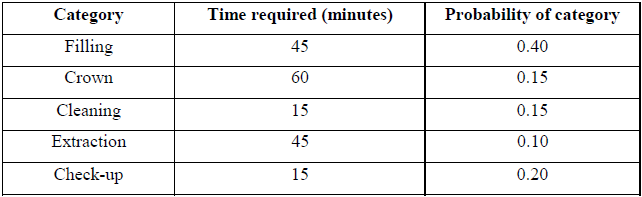
How to use Monte Carlo methods

Regardless of what tool you use, Monte Carlo techniques involves three basic steps:

1. Set up the predictive model, identifying both the dependent variable to be predicted and the independent variables (also known as the input, risk or predictor variables) that will drive the prediction.
2. Specify probability distributions of the independent variables. Use historical data and/or the analyst’s subjective judgment to define a range of likely values and assign probability weights for each.
3. Run simulations repeatedly, generating random values of the independent variables. Do this until enough results are gathered to make up a representative sample of the near infinite number of possible combinations.

***Example:***Dr. Strong, a dentist schedules all his patients for 30-minute appointments. Some of the patients take more or less than 30 minutes depending on the type of dental work to be done. The following table shows the summary of the various categories of work, their probabilities and the time actually needed to complete the work.

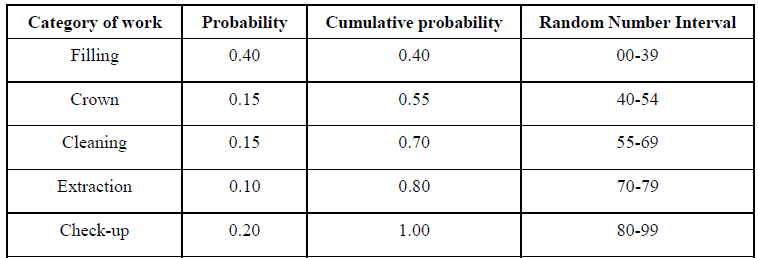
***Simulation Problem***



Simulate the dentist’s clinic for four hours and determine the **average waiting time** for the patients as well as the **idleness** of the doctor. Assume that all the patients show up at the clinic exactly at their scheduled arrival time, starting at 8.00 am. Use the following random numbers for handling the above problem: **40, 82, 11, 34, 25, 66, 17, 79**

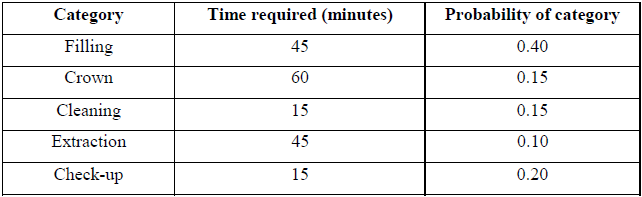
***Solution:***calculate cumulative probability andAssign the random number intervals to the various categories of work as shown in the table.

***Random Number Intervals Assigned to the Various Categories***



*Assuming the dentist clinic starts at 8.00 am, the arrival pattern and the service category are shown in the table.*

**40, 82, 11, 34, 25, 66, 17, 79**

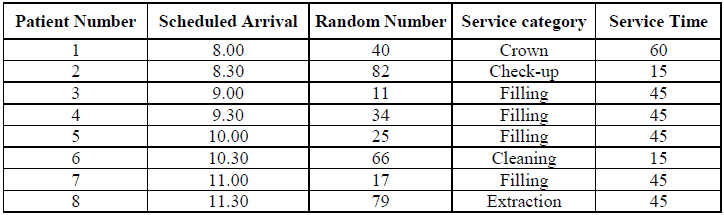


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Time** | **Rno** | **Category** | **Service begins at** | **Service times** | **End service** | **Waiting time** |
| **8.00** | **40** | **Crown** | **8.00** | **60** | **9.00** | **-** |
| **8.30** | **82** | **Check up** | **9.00** | **15** | **9.15** | **30** |
| **9.00** | **11** | **Filling** | **9.15** | **45** | **10.00** | **15** |
| **9.30** | **34** | **Filling** | **10.00** | **45** | **10.45** | **30** |
| **10.00** | **25** | **Filling** | **10.45** | **45** | **11.30** | **45** |
| **10.30** | **66** | **Cleaning** | **11.30** | **15** | **11.45** | **60** |
| **11.00** | **17** | **Filing** | **11.45** | **45** | **12.30** | **45** |
| **11.30** | **79** | **Extraction** | **12.30** | **45** | **1.15** | **60** |

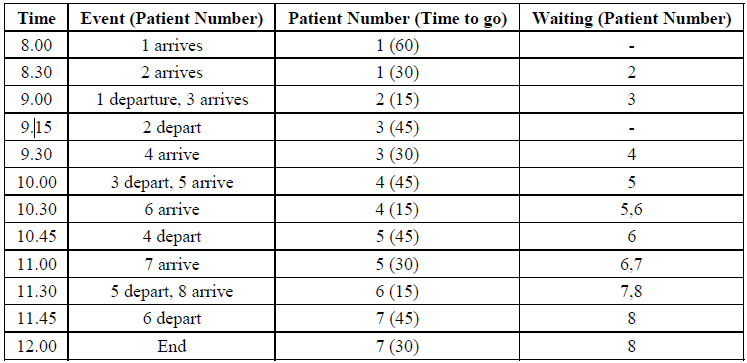
**Average waiting = 285/8=** 35.625 minutes.

No idle time

***Arrival Pattern of the Patients***

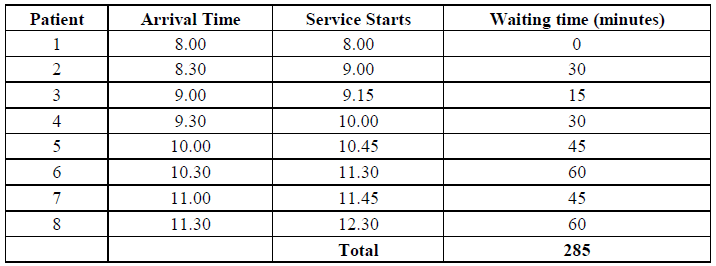
**

***The arrival, and departure patterns and patients’ waiting times are tabulated.***



The dentist was not idle during the simulation period. The waiting times for the patients are as given in table below.

***Patient's Waiting Time***



The average waiting time of patients = 285/8  
= 35.625 minutes.

**Discuss 5 common pitfalls to successful simulation**

#### **Distraction Pitfall**

Every simulation starts with developing a basic understanding of the question at hand. It is impossible to define the part relevant to modeling and simplifying reality without having achieved that understanding. The process of conceiving the question commonly tempts users to address seemingly related questions at the same time. This expansion is mostly due to external pressures (e.g., from clients or superiors). Intrinsic motivation exists as well and stems predominantly from a user's desire to take advantage of perceived synergies between answering the question at hand and preparing other existing or anticipated questions. While the intent is understandable, expanding the original question typically requires enlarging the part of reality to be modeled in the next step which makes the model structure more complex. We call this the distraction pitfall.

#### **Complexity Pitfall**

Upon formulating the research question and defining the target system, i.e. the corresponding part of reality to be modeled, the model structure is to be chosen. This task is often perceived as particularly challenging. It often requires (sometimes drastically) simplifying model entities and their relationships that exist in reality to enable modeling. At the same time, the model structure has to represent reality with sufficient precision for the simulation to yield applicable results. It is a balancing act between simplifying and exact representation. However, going to much in the direction of exact representation of the target system bears the risk of drowning in details and losing sight of the big picture. The resulting model structure becomes increasingly complex and comprehensive. Sometimes this complexity even causes a simulation project to fail. We call this the complexity pitfall.

#### Implementation Pitfall

Software support is often needed to generate the actual simulation model once the conceptual design is finalized. As the domain experts involved in modeling are often laymen with regards to IT implementation, they are at risk of choosing unsuitable software for the simulation. Often the implementation of a model structure is based on existing IT systems or readily available tools. As a result, the selected IT tool is often too weak if it is technically unable to support the functional scope of the simulation model. Conversely, an IT tool can be too powerful as well if only a fraction of its technical potential is used.[[10]](https://www.jasss.org/15/2/5.html#fn10) Such not well-managed interdependencies between the conceptual model and the IT implementation represent what we call the implementation pitfall.

#### Interpretation Pitfall

Upon completing and testing an implemented simulation model, one can finally work with it. However, one can often observe that users are prone to losing their critical distance to the results produced by a simulation. The effect appears to be the bigger the more involved the persons were in the model's development and/or application. Losing critical distance to the simulation results can be responsible for undiscovered model errors or reduced efforts in validation. It also has to be noted again that a model is a simplified representation of reality that can only yield valid results for the context it was originally created for. Going beyond that context means to lose the results' validity and can lead to improper or even false conclusions. It happens for example when the analyzed aspect is not part of the reality represented by the model or when the model is too simple and does not allow for valid interpretations. These wrong conclusions resulting from a loss of critical distance is what we call the interpretation pitfall.

#### Acceptance Pitfall

Even if one is convinced of the simulation results' validity and accuracy, this may not be true for third-party decision makers. In many settings, third-party decision makers have the final word and hardly know the model. The more distant they are to the modeling process and the more complex the simulation model is, the more skeptical they tend to be about the results. Ultimately, a simulation-based decision may be rejected because the actual simulation results are not accepted. This can mean that otherwise good and correct simulation results are ignored and discarded. A related observation is that doubts are to be raised particularly in situations in which the results do not meet the expectations of third-party decision makers.[[13]](https://www.jasss.org/15/2/5.html#fn13) Such expectations are also at the heart of a typical dilemma often faced in simulation modeling: If the results match the expectations, the simulation model and the results themselves are called trivial. If the results are surprising, they prompt third-party decision makers to question the correctness of the entire simulation. We call this cluster of related issues the acceptance pitfall

**MonteCarlo Flow Chart**

