

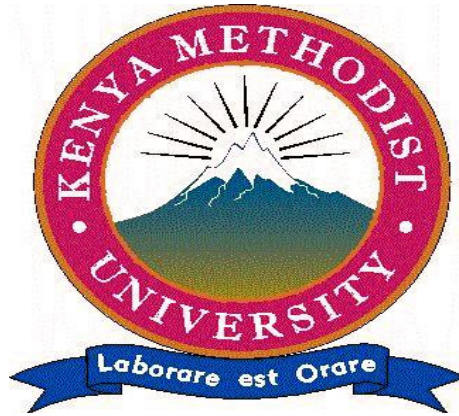


Introduction Simulation and modeling Notes

Computer Science (Kenya Methodist University)



Scan to open on Studocu



KENYA METHODIST UNIVERSITY

FACULTY OF SCIENCES

DEPARTMENT OF COMPUTER INFORMATION SYSTEMS

Course Code: CISY403

Course Title: SIMULATION AND MODELING

1st Trimester, January 2009

List of Contents

Course Outline

Module 1: Introduction

Module 2: Simple Queuing

Module 3: Monte Carlo Simulation in Excel

Module 4: Simulation Software

Module 5: Confidence Intervals

Module 6: Probability Distributions

Module 7: Random Number Generators

Module 8: Output Data Analysis for a Single System

Course Outline: CISY403 Simulation and Modelling (3)

Course Purpose:

The course introduces the students to simulation and modelling as the technique of solving problems by the observation of the performance, over the time, of a dynamic model of the system.

Course Objectives:

At the end of the course students will be able:

- To analyze computer and communication systems through case studies.
- To demonstrate understanding of system modeling through the competent use of computer simulation methods in mathematical modeling techniques.

Course Content: Introduction and basic simulation procedures. Model classification: Monte Carlo simulation, discrete-event simulation, continuous system simulation, mixed continuous/discrete-event simulation. Queuing networks: analytical and simulation modeling of queuing systems. Input and Output Analysis: random numbers, generating and analyzing random numbers, sample generation, place and execution driven simulation point and internal estimation. Probability distributions.

Teaching Methods:

- Class lectures that involve proper explanation various simulation models.
- Tutorials that entail solving of problems by both students and lecturer.
- Practical sessions in the lab and practical demonstrations.
- Group discussions among the students and active participation in class.
- Regular CATs and assignments that are discussed after grading.

Reference Textbooks:

- Modeling and Simulation: The Computer Science of Illusion
- Raczynski , S.: Modelling and Simulation: The Computer Science of Illusion; John Wiley, 2006.
- Birta, Louis G., Arbez, Gilbert : *Modelling and Simulation Exploring Dynamic System Behaviour*; Springer, 2007.
- Law and Kelton: *Simulation Modeling and Analysis*; 3rd Edition, McGraw Hill, 1991.

Teaching Tools:

- Computer installed with a simulator program such as MS Office Excel.

Assessment

Assignments and/or Cats – 40%

Exam – 60%

Module 1: INTRODUCTION

Simulation refers to a broad collection of methods and applications to mimic the behaviour of real systems, usually on a computer with appropriate software.

Modeling

Computer simulation deals with models of systems. A system is a facility or process either actual or planned, such as:

- A bank with different kinds of customers, servers, facilities.
- A central insurance claims office where a lot of paperwork is received, reviewed, copied, filed and mailed by people and machines.
- A supermarket with inventory control, checkout, and customer service.
- Etc

Systems are often studied to measure performance, improve its operation, or design it if it doesn't exist.

How about just playing with the real system?

It might be possible to experiment with the actual physical system. For instance:

- Some cities have installed entrance-ramp traffic lights on the freeway system to experiment with different sequencing to find settings that make rush hour as smooth and safe as possible.
- A supermarket manager might try different policies for inventory control and checkout personnel assignment to see what combinations seem to be most profitable and provide best service.
- A computer facility can experiment with different network layouts and job priorities to see how they affect machine utilization turnaround.

However, sometimes you can't (or shouldn't) play with the system:

- You can't experiment with alternative factory layouts if it's not yet built.
- Even in an existing factory, it might be costly to change an experimental layout that might not work anyway.
- It would be hard to run twice as many customers through a bank to see what will happen when a nearby branch closes.
- Trying a new check-in procedure at an airport might initially cause a lot of people to miss their flights if there are unforeseen problems with the new procedure.
- It's not possible to try out a new procedure in an emergency room of a hospital.

In these situations, you might build a model to serve as a stand-in for studying the system and ask pertinent questions about what would happen in the system if you did this or that.

Physical Models

This is a physical replica or scale model of the system, sometimes called *iconic model*. For example,

- Physical flight simulators are widely used to train pilots.
- Simulated control rooms have been developed to train operators for nuclear power plants.

Logical Models

Such a model is a set of approximations and assumptions, both structural and quantitative; about the way the system does or will work. A logical model is represented in a computer program that exercised to address the model's behaviour; if your model is a valid representation of your system, you hope to learn about the system's behaviour.

After making the assumptions and stating the assumptions for a valid logical model, you need to find a way to deal with the model and analyze its behaviour. If the model is simple enough, you may be able to use traditional mathematical tools like queuing theory, differential equation or linear programming. However, most systems are pretty complicated and for such models, there may not be exact mathematical solutions worked out, which is where simulation comes in.

COMPUTER SIMULATION

Computer simulation refers to methods for studying a wide variety of models of real world systems by numerical evaluation using software designed to imitate the systems' operations and characteristics, often over time.

Why Simulate?

- Predict behaviour before building
 - Prototypes are often cheaper than building
 - Proof-of-concept
 - Evaluate design trade-offs
 - Sell concepts to others
- Predict for future expectations
 - Weather forecasts, hurricane paths
 - Stock market
 - Satellite and asteroid orbits and changes
 - Earthquakes
- System characterization testing
 - Sensitivity analysis
 - Accuracy determination
 - Behaviour familiarity
- Pretend (virtual environments)
 - Training tools
 - Games
 - Interactive controller
 - Realism experience for system use

Advantages of Simulation

- The main reason for simulation's popularity is its ability to deal with very complicated models of correspondingly complicated systems. This makes it a versatile and powerful tool.

- Simulation offers an improvement in performance/price ratios of computer hardware, making it ever more cost effective to do what was prohibitively expensive.
- Advances in simulation software power, flexibility, and ease of use have moved the approach from the realm of tedious and error-prone low-level programming to the arena of quick and valid decision making.

Disadvantages of simulation

- Because many real systems are affected by uncontrollable and random inputs, many simulation models involve random (stochastic) input components, causing their output to be random too.
- To deal with the uncertainty, you might be able make a lot of over-simplification assumptions about the system. Unfortunately though, such an over-simplified model will probably not be a valid representation of the system.

Different Kinds of Simulations

- **Static Vs Dynamic:** Time does not play a role in static models as it does in dynamic models.
- **Continuous Vs Discrete:** In a continuous model, the state of the system can change continuously over time; for example, the level of a reservoir as water flows in and out, and as precipitation and evaporation occur. In a discrete model, change can occur only at specific times, machines going down and coming back up at specified times, and breaks for workers. You can have elements of both continuous and discrete change in the same model, which are called *mixed continuous-discrete models*. An example might be a refinery with continuously changing pressure inside vessels and discretely occurring shutdowns.
- **Deterministic Vs Stochastic:** models that have no random input are deterministic; a strict appointment-book operation with fixed service times would be an example. Stochastic models operate with random input – like a bank with randomly arriving customers requiring varying service times.

How simulation gets done

- a. **By hand:** in the beginning people did simulation by hand. Though currently, hand simulation may seem impractical, there are some aspects of it that are common to most simulations:
 - The purpose is to estimate something whose value would be hard to compute exactly.
 - The estimate we get at the end is not going to be exactly right, i.e., it has some errors associated with it.
 - It seems intuitive that the more “experiments” you make, the smaller the error is likely to be and thus the better the estimate is likely to be.

In 1920's and 1930s, statisticians began using random-number machines and tables in numerical experiments to help them develop and understand statistical theory. For example, Guinness Brewery employee W.S.Gossett did numerical sampling experiments to help him gain insight into what was going on in mathematical statistics. (To protect his job at Guinness, he published his research under the pseudonym “Student” and also developed the t distribution used widely in statistical inference.)

- b. **Programming in General-Purpose Languages:** As digital computers appeared in the 1950s and 1960s, people began writing computer programs in general-purpose languages like FORTRAN to do simulations of more complicated systems

- c. **Simulation Languages:** Special purpose simulation languages like GPSS, SIMSCRIPTS, SLAM and SIMAN became very popular and are in wide use. However, the user has to learn about their features to use them effectively.
- d. **High level simulators:** thus, several high-level simulator products emerged that are easier to use. However, the domain of many simulators can also be rather restrictive and are generally not as flexible as a user might like.

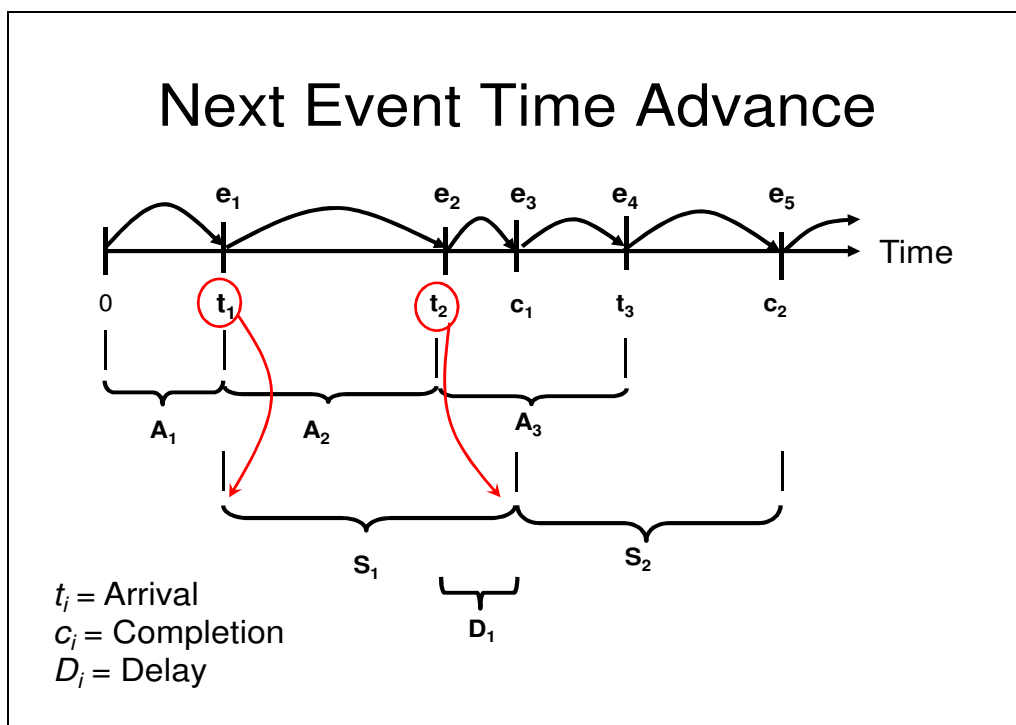
Module 2: SIMPLE QUEUEING

In the previous chapter, I mentioned that we shall be using Arena for our simulation. However, we shall not limit ourselves to Arena, actually we shall simply illustrate its use but mostly see how Monte Carlo simulation is used in EXCEL, but that's in the next chapter.

The simulation models we shall be considering are discrete, dynamic, and stochastic and will be called discrete-event simulations models (DE models). DE modelling concerns the modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time.

Because of the dynamic nature of DE models, we must keep track of the current value of simulated time as the simulation proceeds. Two principal approaches have been suggested:

- Next-event time advance: the simulation clock is initialized to zero and the times of occurrence of future events are determined.
- Fixed-increment time advance does not skip over inactive times.



An example

The system

Consider a single service facility with a single server – e.g., a one teller bank – for which we would like to estimate the expected average delay in queue of arriving customers, where the delay in queue of a customer is the length of time interval from the instant of his arrival at the facility to the instant he begins being served. For the objective of estimating the average

delay of a customer, the state variables for a DE model of the facility would be the status of the server, i.e. idle or busy, the number of customers waiting in the queue to be served (if any), and the time of arrival of each person waiting in the queue. The status of the server is needed to determine, upon a customer arrival, whether the customer can be served immediately or must join the end of the queue. When the server completed serving a customer,, the number of customers in the queue is used to determine whether the server will become idle or begin serving the first customer in the queue. The time of arrival of a customer is needed to compute his delay in the queue.

Below is a detailed illustration of the Next-event time advance approach for a single-server queuing system, which needs the following notation:

Next Event Time Advance

t_i = time of arrival of i th item (customer), $t_0 = 0$

$A_i = t_i - t_{i-1}$ = interval time between $(i-1)$ st and i th arrivals of items (customers)

S_i = time that a server actually spends serving i th event (customer)

D_i = delay in queue of i th event (customer)

$c_i = t_i + D_i + S_i$ = time that i th event (customer) completes service and departs

e_i = time of occurrence of i th event of any type (i th value the simulation clock takes on, excluding the value $e_0 = 0$)

Each of the above quantities will generally be a random variable.

Components and Organization of a DE simulation Model

- System State – Collection of state variables necessary to describe the system at a particular time.
- Simulation Clock – Current value of simulated time.
- Event list – list of future time event values.
- Statistical Counters – accumulators of simulation results
- Initialization Routine – start up state definition for time zero (t_0)
- Timing Routine – A subprogram determining next event list and event times
- Event Routine – Updates system state for each event type

- Library Routines – random event generators (dist. functions)
- Report Generator – final performance report
- Main Program – overall glue

SIMULATION OF A SINGLE –SERVER QUEING SYSTEM

Consider the simulation of the single server banking system that we describe above. The interarrival times, A_1, A_2, \dots are independent, identically distributed (IID) random variables. A customer who arrives and finds the server idle enters service immediately, and the service times S_1, S_2 of the successive customers...are IID random variables that are independent of the interarrival times.

The simulation will begin in the “empty and idle” state, i.e no customers are present and the server is idle. At time 0 we will begin waiting for the arrival of the first customer, which will occur after the first interarrival time A_1 . We wish to simulate this system until a fixed number (n) of customers have completed their delays in queue; i.e., the simulation will stop when the n th customer enters service.

To measure the performance of the system we will look at three things as below:

1. Average delay
2. Expected average queue length
3. Expected utilization

Average delay

On a given run of the simulation, we do not exclude the possibility of a customer having a delay of zero in the case of an arrival finding the system empty and idle. Delays with a value of zero are counted in the average \bar{a} , since if many delays were zero this would represent a system providing very good service and our output measure should reflect this. $d(n)$ gives information about system performance from the customers' point of view.

Performance Measures

1. Average customer delay (customer concern)

$$\hat{d}(n) = \frac{\sum_{i=1}^n D_i}{n} = \bar{D}(n)$$

Please note that under the sum sign it is **i=1**.

Expected average queue length

- Let q_n = average number of customers not being served over the duration n
- Let $Q(t)$ = number in queue at time t
- Let $T(n)$ = time required for n delays
- Let p_i = expected proportion of time that $Q(t) = i$
-

Performance Measures

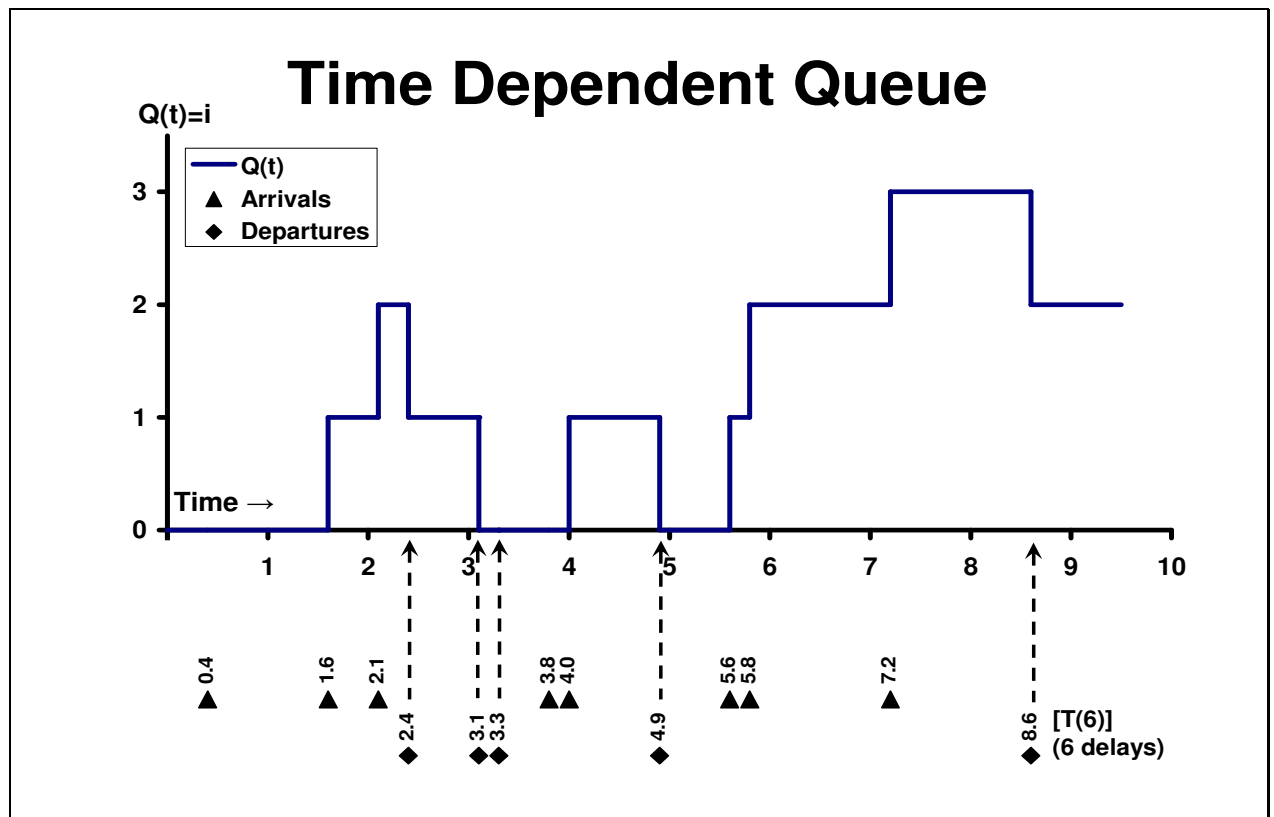
- Observed average queue length
- Expected (estimated) average queue length

$$q(n) = \frac{\sum_{i=0}^{\infty} i p_i}{n}$$

$$\hat{q}(n) = \frac{\sum_{i=0}^{\infty} i \hat{p}_i}{n}$$

p_i = proportion of time for each i

In the time dependent queue graph below, we have 6 delays, so $n=6$. Arrivals occur at times 0.4, 1.6, 2.1, 3.8, 4.0, 5.6, 5.8 and 7.2. departures occur at times 2.4, 3.1, 3.3, 4.9 and 8.6 and the simulation ends at time **T(6)=8.6**.



- Let T_i = total time that length is i
- Then: $T(n) = T_0 + T_1 + T_2 + \dots$
- And: $\hat{p} = T_i / T(n)$

$$\hat{q}(n) = \frac{\sum_{i=0}^{\infty} iT_i}{T(n)}$$

To compute the expected value of $q(n)$, we must first compute the T_i 's which can be read off the figure above. For example T_0 , is the intervals where the queue is of length 0, which is between 1.6 and 0.0, and 4.0 and 3.1 and 5.6 and 4.9.

iT_i is then the multiplication of i (queue length) times the corresponding value of T_i . For examples $T_0 = 3.2$, hence iT_i for this will be 0×3.2 .

Computing $\hat{q}(n)$

- From the example time graph:

- First compute the T_i 's (pick from graph):

$$\begin{aligned} T_0 &= (1.6 - 0.0) + (4.0 - 3.1) + (5.6 - 4.9) = 3.2 \\ T_1 &= (2.1 - 1.6) + (3.1 - 2.4) + 4.9 - 4.0 + 5.8 - 5.6 = 2.3 \\ T_2 &= (2.4 - 2.1) + (7.2 - 5.8) = 1.7 \\ T_3 &= (8.6 - 7.2) = 1.4 \end{aligned}$$

- Then the sum:

$$\sum_{i=0}^{\infty} iT_i = (0 \times 3.2) + (1 \times 2.3) + (2 \times 1.7) + (3 \times 1.4) = 9.9$$

- Finally: $\hat{q}_6 = 9.9 / 8.6 = 1.15$

Expected utilization

The measure of how busy the server is. This is the expected utilization of the server which is the expected proportion of time during the simulation that the server is not idle.

Computing Utilization Metric

- Define a 'busy' function:

$$B(t) = \begin{cases} 1 & \text{if the server is busy at time } t \\ 0 & \text{if the server is idle at time } t \end{cases}$$

- $\hat{u}(n)$ can be expressed as a proportion of time $B(t)=1$

$$\hat{u}(n) = \frac{(3.3 - 0.4) + (8.6 - 3.8)}{8.6} = \frac{7.7}{8.6} = 0.90$$

The server above from the graph was busy between times 0.4 and 3.3 and times 3.8 and 8.6.

0.90 indicates that the server was busy 90% of the time during the simulation. For many simulations involving servers of some sort, utilization statistics are quite informative in identifying bottlenecks, (utilizations near 100%, coupled with heavy congestions measured for the queues leading in), or excess capacity (low utilization).

(Refer to the Simple Queuing snapshots in the main notes for a detailed description of what happens in the simulation of the above model.)

Components of a queuing system

A queuing system is characterized by three components:

- Arrival process – describing how entities arrive to the system. If A_i be the inter-arrival time, we shall denote the mean inter-arrival time by $E(A)$ and $\lambda=1/E(A)$ as the arrival rate of the entities.
- Service mechanism – specifying the number of servers, whether each server has its own queue or there is one queue feeding all servers, and the probability distribution of the entities' service times. If S_i be the service time, we shall denote the mean service time of a customer $E(S)$ and call $\omega=1/E(S)$ the service rate of the entities.
- Queue discipline – the rule that a server uses to choose the next entity from the queue (if any) when the server completes the service of the current entity. Commonly used queue disciplines are:
 - FIFO(First in first out)
 - LIFO(Last in first out)
 - Priority – entities are served in order of their importance.

Some Definitions New and Review

- IID - Independent and Identically Distributed
 - Exponential random distribution – arrivals, departures
- GI – General Independent – arrivals
- G - General – service
- M - Markovian, memoryless of previous events, exponential distribution
- E_k – k-Erlang – summation of exponential distributions
- D – Deterministic – fixed times

System Notation

- General model notation
- $\langle \text{Arrival type} \rangle / \langle \text{Service type} \rangle / \langle \# \text{servers} \rangle$
 - GI/G/s – general queue
 - M/M/1 – single server queue with exponential arrival and service times.
- For more than one server, $s > 1$
 - Therefore, M/M/s

Steps in a simulation study

1. Formulate the problem and plan the study.
2. Collect data and define a model. For example, in modelling a bank, one might collect inter-arrival time and service times and use these data to specify inter-arrival and service-time distribution for use in the model.
3. Establish whether the model is valid.
4. Construct a computer program and verify. The simulation modeller must decide whether to program the model in a general-purpose language such as C or in a specially designed simulation language such as SIMAN.
5. Make pilot runs.
6. Validate the pilot runs.
7. Design experiments. It must be decided what they system designs to simulate, if there are more alternatives that one can reasonably simulate.
8. Make production runs.
9. Analyze output data. Statistical techniques are used to analyze the ourput data from the production runs.
10. Document, present and implement the results.

Other types of simulation

1. **Continuous simulation** concerns the modelling over time of a system by a representation in which the state variable change continuously with respect to time. Such models typically involve differential equations that give relationships to the rates of change of the state variables with time.
2. **Combined discrete-continuous simulation:** Since some sytems are neither completely discrete nor completely continuous, the need many arise to construct a model with aspects of both.
3. **Monte carlo simulation** is a scheme for employing random numbers and will be discussed in the next module.

Advantages, disadvantages and pitfalls of simulation

Advantages

- Most complex, real world systems with stochastic elements cannot be accurately described by a mathematical model that can be evaluated analytically.
- Simulation allows one to estimate the performance of an existing system under some projected set of operating conditions.

- Alternative proposed system designs can be compared via simulation to see which best meets a specified requirement.
- In a simulation, we can maintain much better control over experimental conditions than would generally be possible with experimenting with the system itself.
- Simulation allows us to study a system with a long time frame e.g. an economic system.

Disadvantages

- Each run of a stochastic model produces only estimates of a model's true characteristics for a particular set of input parameters.
- Simulation models are often expensive and time-consuming to develop.
- The large volume of numbers produced by a simulation study or the persuasive impact of a realistic animation often create a tendency to place greater confidence in a study's results than is justified.

MODULE 3: MONTE CARLO SIMULATION IN EXCEL

A **Monte Carlo method** is a technique that involves using random numbers and probability to solve problems. When we would like to accurately estimate the probabilities of uncertain events; for example, what is the probability that a new product's cash flows will have a positive net present value (NPV)? What is the risk factor of our investment portfolio? Monte Carlo simulation enables us to model situations that present uncertainty and then play them out on a computer thousands of times.

The term **Monte Carlo Method** was coined by S. Ulam and Nicholas Metropolis in reference to games of chance, a popular attraction in Monte Carlo, Monaco. Another theory is that, the name *Monte Carlo simulation* comes from the computer simulations performed during the 1930s and 1940s to estimate the probability that the chain reaction needed for an atom bomb to detonate would work successfully. The physicists involved in this work were big fans of gambling, so they gave the simulations the code name *Monte Carlo*.

Computer simulation has to do with using computer models to imitate real life or *make predictions*. When you create a model with a spreadsheet like Excel, you have a certain number of *input parameters* and a few equations that use those inputs to give you a set of *outputs* (or *response variables*). This type of model is usually **deterministic**, meaning that you get the same results no matter how many times you re-calculate.

Many companies use Monte Carlo simulation as an important part of their decision-making process. Here are some examples.

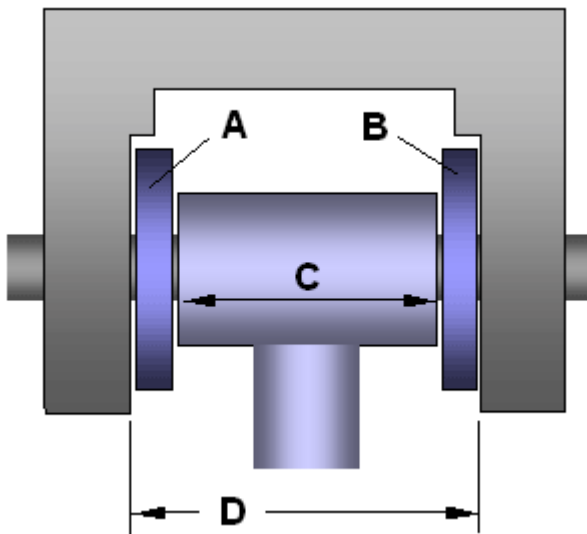
- General Motors, Proctor and Gamble, Pfizer, Bristol-Myers Squibb, and Eli Lilly use simulation to estimate both the average return and the risk factor of new products. At GM, this information is used by the CEO to determine which products come to market.
- GM uses simulation for activities such as forecasting net income for the corporation, predicting structural and purchasing costs, and determining its susceptibility to different kinds of risk (such as interest rate changes and exchange rate fluctuations).
- Lilly uses simulation to determine the optimal plant capacity for each drug.
- Proctor and Gamble uses simulation to model and optimally hedge foreign exchange risk.
- Sears uses simulation to determine how many units of each product line should be ordered from suppliers—for example, the number of pairs of Dockers trousers that should be ordered this year.

- Oil and drug companies use simulation to value "real options," such as the value of an option to expand, contract, or postpone a project.
- Financial planners use Monte Carlo simulation to determine optimal investment strategies for their clients' retirement.

Monte Carlo simulation is a method for *iteratively* evaluating a deterministic model using sets of random numbers as inputs. This method is often used when the model is complex, nonlinear, or involves more than just a couple uncertain parameters. A simulation can typically involve *over 10,000 evaluations* of the model, a task which in the past was only practical using super computers. By using **random inputs**, you are essentially turning the deterministic model into a stochastic model. The example below demonstrates this concept with a very simple problem.

Stochastic Model Example

A stochastic model is one that involves probability or randomness. In this example, we have an assembly of 4 parts that make up a hinge, with a pin or bolt through the centers of the parts. Looking at the figure below, **if $A + B + C$ is greater than D** , we're going to have a hard time putting this thing together.



Let's say we have a million of each of the different parts, and we randomly select the parts we need in order to assemble the hinge. **No two parts are going to be exactly the same size!** But, if we have an idea of the range of sizes for each part, then we can **simulate** the selection and assembly of the parts mathematically.

The table below demonstrates this. Each time you press "**Calculate**", you are simulating the creation of an assembly from a random set of parts. If you ever get a **negative clearance**, then that means the combination of the parts you have selected will be too large to fit within dimension **D**. Do you ever get a negative clearance?

Tolerance Stack-Up Model			
Part	Min	Max	Random
A	<input type="text" value="1.95"/>	<input type="text" value="2.05"/>	<input type="text"/>
B	<input type="text" value="1.95"/>	<input type="text" value="2.05"/>	<input type="text"/>
C	<input type="text" value="29.5"/>	<input type="text" value="30.5"/>	<input type="text"/>
D	<input type="text" value="34"/>	<input type="text" value="35"/>	<input type="text"/>
Clearance, D-(A+B+C):			<input type="text"/>

This example demonstrates almost all of the steps in a Monte Carlo simulation. The **deterministic model** is simply $D-(A+B+C)$. We are using **uniform distributions** to generate the values for each input. All we need to do now is press the "calculate" button a few thousand times, record all the results, create a histogram to visualize the data, and calculate the probability that the parts cannot be assembled.

Of course, you don't want to do this manually. That is why there is so much software (including Excel) for automating Monte Carlo simulation. All we need to do is follow the **five simple steps** listed below:

Step 1: Create a parametric model, $y = f(x_1, x_2, \dots, x_q)$.

Step 2: Generate a set of random inputs, $x_{i1}, x_{i2}, \dots, x_{iq}$.

Step 3: Evaluate the model and store the results as y_i .

Step 4: Repeat steps 2 and 3 for $i = 1$ to n .

Step 5: Analyze the results using histograms, summary statistics, confidence intervals, etc.

ON TO MS EXCEL

Open your MS EXCEL...

RAND()

When you type the formula `=RAND()` in cell A1 then copy and paste it to cell A2 and below for as far as you want, you get a number that is equally likely to assume any value between 0 and 1. Thus, around 25 percent of the time, you should get a number less than or equal to 0.25; around 10 percent of the time you should get a number that is at least 0.90, and so on. To demonstrate how the RAND function works, take a look at the figure below.

You run a simulation by pressing F9, for each press, it's a different run of the simulation as you will notice the numbers randomly changing.

Clipboard		Font		Alignment						
D5		fx		=(COUNTIF(A1:A400,">=0.25")-COUNTIF(A1:A400,">0.5"))/400						
	A	B	C	D	E	F	G	H	I	J
1	0.529798		mean	0.48552						
2	0.349041									
3	0.525689		fraction							
4	0.229605		0.25	0.2625						
5	0.015577		0.25-0.50	0.265						
6	0.719466		0.5-0.75	0.23						
7	0.471377		0.75-1.00	0.2425						
8	0.028899									
9	0.171928									
10	0.510255									
11	0.465012									
12	0.910587									
13	0.902525									
14	0.580092									
15	0.929353									
16	0.031877									
17	0.727234									
18	0.200089									
19	0.296131									
20	0.648559									
21	0.827948									
22	0.509995									
23	0.675647									
24	0.354719									
25	0.52661									
26	0.424578									
27	0.857148									

I copied from cell A2 to A400 the formula *RAND()* from A1. I named the range A1:A400 *Data*. Then, in column D, I tracked the average of the 400 random numbers (cell F1) and used the COUNTIF function in D4 to determine the fractions that are between 0 and 0.25, 0.25 and 0.50, 0.50 and 0.75, and 0.75 and 1. I did this as follows:

In D4 type

=(COUNTIF(A1:A400,"<= 0.25"))/400

This finds the % of the 400 values that are between 0 and 0.25.

In D5 type

=(COUNTIF(A1:A400,">=0.25")-COUNTIF(A1:A400,">0.5"))/400

In D6 type

=(COUNTIF(A1:A400,">=0.5")-COUNTIF(A1:A400,">0.75"))/400

In D7 type

=(COUNTIF(A1:A400,">=0.75"))/400

When you press the F9 key, the random numbers are recalculated. Notice that the average of the 400 numbers is always approximately 0.5, and that around 25 percent of the results are in intervals of 0.25. These results are consistent with the definition of a random number. Also note that the values generated by RAND in different cells are independent. For example, if the random number generated in cell A1 is a large number (for example, 0.99), it tells us nothing about the values of the other random numbers generated.

Discrete random variables in Monte Carlo

Suppose the demand for a calendar is governed by the following discrete random variable:

Demand	Probability
10,000	0.10
20,000	0.35
40,000	0.3
60,000	0.25

How can we have Excel play out, or simulate, this demand for calendars many times? The trick is to associate each possible value of the RAND function with a possible demand for calendars. The following assignment ensures that a demand of 10,000 will occur 10 percent of the time, and so on.

Demand	Random number assigned
10,000	Less than 0.10
20,000	Greater than or equal to 0.10, and less than 0.45
40,000	Greater than or equal to 0.45, and less than 0.75
60,000	Greater than or equal to 0.75

F5		fx		0.75				
	A	B	C	D	E	F	G	H
1						CUTOFFS	DEMAND	
2						0	10000	
3						0.1	20000	
4						0.45	40000	
5	TRIALS		RAND			0.75	60000	
6	1	40000	0.521014					
7	2	20000	0.317385		FRACTIONS OF TIME			
8	3	20000	0.190934		10000	0.114286		
9	4	20000	0.36534		20000	0.428571		
10	5	20000	0.167799		40000	0.228571		
11	6	20000	0.221464		60000	0.2		
12	7	40000	0.683582					
13	8	60000	0.905689					
14	9	40000	0.691382					
15	10	60000	0.981692					
16	11	10000	0.0851					
17	12	20000	0.232562					
18	13	40000	0.48424					
19	14	60000	0.771267					
20	15	20000	0.380977					
21	16	10000	0.078052					
22	17	20000	0.144886					
23	18	20000	0.191316					
24	19	40000	0.732151					
25	20	20000	0.431913					
26	21	60000	0.938257					
27	22	60000	0.769774					

The key to our simulation is to use a random number to initiate a lookup from the table range F2:G5 (named *lookup*). **You should know how to name ranges of cells!**

Random numbers greater than or equal to 0 and less than 0.10 will yield a demand of 10,000; random numbers greater than or equal to 0.10 and less than 0.45 will yield a demand of 20,000; random numbers greater than or equal to 0.45 and less than 0.75 will yield a demand of 40,000; and random numbers greater than or equal to 0.75 will yield a demand of 60,000. I generated 100 random numbers by copying from C6 to C7:C105 the formula *RAND()*. I then generated 100 trials, or iterations, of calendar demand by copying from B6 to B7:B105 the formula *VLOOKUP(C6,lookup,2)*. This formula ensures that any random number less than 0.10 generates a demand of 10,000, any random number between 0.10 and 0.45 generates a demand of 20,000, and so on. In the cell range F8:F11, I used the *COUNTIF* function to determine the fraction of our 100 iterations yielding each demand. When we press F9 to recalculate the random numbers, the simulated probabilities are close to our assumed demand probabilities.

How can a greeting card company determine how many cards to produce?

In this section, I'll demonstrate how Monte Carlo simulation can be used as a decision-making tool. Suppose that the demand for a Valentine's Day card is governed by the following discrete random variable:

Demand	Probability
10,000	0.10
20,000	0.35
40,000	0.3
60,000	0.25

The greeting card sells for \$4.00, and the variable cost of producing each card is \$1.50. Leftover cards must be disposed of at a cost of \$0.20 per card. How many cards should be printed?

REFER THE NOTES BELOW WITH MY EXCEL TABLE ON THE NEXT PAGE.

Basically, we simulate each possible production quantity (10,000, 20,000, 40,000, or 60,000) many times (for example, 1000 iterations). Then we determine which order quantity yields the maximum average profit over the 1000 iterations. You can find the data for this section in the table below. I've assigned the cell range G3:H6 the name *lookup*. Our sales price and cost parameters are entered in cells C4:C6.

I then enter a trial production quantity (40,000 in this example) in cell C1. Next I create a random number in cell C2 with the formula `=RAND()`. As previously described, I simulate demand for the card in cell C3 with the formula `VLOOKUP(C2,lookup,2)`.

The number of units sold is the smaller of our production quantity and demand. In cell C8, I compute our revenue with the formula `MIN(produced,demand)*unit_price`. In cell C9, I compute total production cost with the formula `produced*unit_prod_cost`.

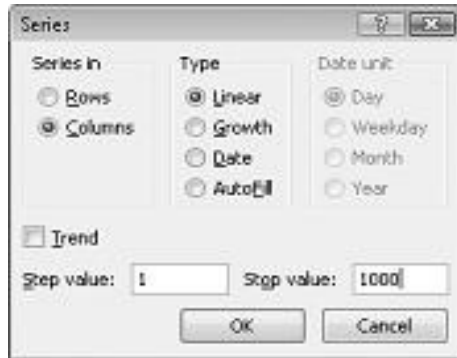
If we produce more cards than are in demand, the number of units left over equals production minus demand; otherwise no units are left over. We compute our disposal cost in cell C10

with the formula $\text{unit_disp_cost} * \text{IF}(\text{produced} > \text{demand}, \text{produced} - \text{demand}, 0)$. Finally, in cell C11, we compute our profit as $\text{revenue} - \text{total_var_cost} - \text{total_disposing_cost}$.

We would like an efficient way to press F9 many times (for example, 1000) for each production quantity and tally our expected profit for each quantity. This situation is one in which a two-way data table comes to our rescue.

C3 fx =VLOOKUP(C2,lookup2,2)									
	A	B	C	D	E	F	G	H	I
1		Produced	40000						
2		rand#	0.953838				CUTOFFS	DEMAND	
3		demand	60000				0	10000	
4		unit production cost	1.5				0.1	20000	
5		unit price	4				0.45	40000	
6		unit disp cost	0.2				0.75	60000	
7									
8		revenue	160000						
9		total var cost	60000						
10		total disposing cost	0						
11		profit	100000						
12									
13		25000	46174.17	59261.26	47711.71				
14		0	12091.07	47622.71	74208.54				
15	100000	10,000	20,000	40,000	60,000	production quantity			
16	1	25000	50000	100000	66000				
17	2	25000	50000	100000	66000				
18	3	25000	50000	-26000	-60000				
19	4	25000	50000	100000	150000				
20	5	25000	50000	16000	66000				
21	6	25000	50000	100000	150000				
22	7	25000	50000	-26000	-60000				
23	8	25000	50000	16000	-18000				
24	9	25000	50000	16000	-18000				
25	10	25000	50000	100000	-18000				
26	11	25000	50000	16000	-60000				
27	12	25000	50000	-26000	-18000				

In the cell range A16:A1015, I entered the numbers 1–1000 (corresponding to our 1000 trials). One easy way to create these values is to start by entering 1 in cell A16. Select the cell, and then on the Home tab in the Editing group, click Fill, and select Series to display the Series dialog box. In the Series dialog box, shown in Figure 60-6, enter a Step Value of 1 and a Stop Value of 1000. In the Series In area, select the Columns option, and then click OK. The numbers 1–1000 will be entered in column A starting in cell A16.



Next we enter our possible production quantities (10,000, 20,000, 40,000, 60,000) in cells B15:E15. We want to calculate profit for each trial number (1 through 1000) and each production quantity. We refer to the formula for profit (calculated in cell C11) in the upper-left cell of our data table (A15) by entering `=C11`.

We are now ready to trick Excel into simulating 1000 iterations of demand for each production quantity. Select the table range (A15:E1014), and then in the Data Tools group on the Data tab, click What If Analysis, and then select Data Table. To set up a two-way data table, choose our production quantity (cell C1) as the Row Input Cell and select any blank cell (we chose cell I14) as the Column Input Cell. After clicking OK, Excel simulates 1000 demand values for each order quantity.

To understand why this works, consider the values placed by the data table in the cell range C16:C1015. For each of these cells, Excel will use a value of 20,000 in cell C1. In C16, the column input cell value of 1 is placed in a blank cell and the random number in cell C2 recalculates. The corresponding profit is then recorded in cell C16. Then the column cell input value of 2 is placed in a blank cell, and the random number in C2 again recalculates. The corresponding profit is entered in cell C17.

By copying from cell B13 to C13:E13 the formula `AVERAGE(B16:B1015)`, we compute average simulated profit for each production quantity. By copying from cell B14 to C14:E14 the formula `STDEV(B16:B1015)`, we compute the standard deviation of our simulated profits for each order quantity. Each time we press F9, 1000 iterations of demand are simulated for each order quantity. Producing 40,000 cards always yields the largest expected profit. Therefore, it appears that producing 40,000 cards is the proper decision.

The Impact of Risk on Our Decision If we produced 20,000 instead of 40,000 cards, our expected profit drops approximately 22 percent, but our risk (as measured by the standard deviation of profit) drops almost 73 percent. Therefore, if we are extremely averse to risk,

producing 20,000 cards might be the right decision. Incidentally, producing 10,000 cards always has a standard deviation of 0 cards because if we produce 10,000 cards, we will always sell all of them without any leftovers.

Confidence Interval for Mean Profit A natural question to ask in this situation is, into what interval are we 95 percent sure the true mean profit will fall? This interval is called the *95 percent confidence interval for mean profit*. A 95 percent confidence interval for the mean of any simulation output is computed by the following formula:

$$\text{Mean Profit} \pm \frac{1.96 * \text{profit std. dev.}}{\sqrt{\text{number iterations}}}$$

You will learn more about confidence intervals in one of the modules and you will be able to understand this more.

In cell J11, I computed the lower limit for the 95 percent confidence interval on mean profit when 40,000 calendars are produced with the formula $D13 - 1.96 * D14 / \text{SQRT}(1000)$. In cell J12, I computed the upper limit for our 95 percent confidence interval with the formula $D13 + 1.96 * D14 / \text{SQRT}(1000)$. These calculations are shown in Figure below

		LCI	57960.2
		UCI	63757.5

In this simulation run, we are 95 percent sure that our mean profit when 40,000 calendars are ordered is between \$57,960 and \$63,757.

That's it! You should now have an idea of how monte carlo simulation is used for forecasting and simulations. Try the example in the website below and see how you understand it.

<http://www.vertex42.com/ExcelArticles/mc/SalesForecast.html>

Exercise

1. A GMC dealer believes that demand for 2005 Envoys will be normally distributed with a mean of 200 and standard deviation of 30.

His cost of receiving an Envoy is \$25,000, and he sells an Envoy for \$40,000. Half of all the Envoys not sold at full price can be sold for \$30,000. He is considering ordering 200, 220, 240, 260, 280, or 300 Envoys. How many should he order?

2. A small supermarket is trying to determine how many copies of *People* magazine they should order each week. They believe their demand for *People* is governed by the following discrete random variable:

Demand	Probability
15	0.10
20	0.20
25	0.30
30	0.25
35	0.15

3. The supermarket pays \$1.00 for each copy of *People* and sells it for \$1.95. Each unsold copy can be returned for \$0.50. How many copies of *People* should the store order?

Module 4: SIMULATION SOFTWARE

Comparison of simulation languages with general-purpose languages

The following are some advantages of programming a simulation model in a simulation language rather than a general-purpose language e.g. FOTRAN, C, PASCAL.

- Simulation languages automatically provide most of the features needed in programming a simulation model, resulting in a significant decrease in programming time.
- They provide a natural framework for simulation modelling. Their basic building blocks are more closely akin to simulation than are those in a language like FOTRAN.
- Simulation models are generally easier to change when written in a simulation language.
- Most simulation languages provide dynamic storage allocation during execution.
- They provide better error detection because many potential types of errors have been identified and checked for automatically.

On the other hand, many simulation models are still written in general-purpose languages. Some advantages of this choice are:

- Most modelers already know a general-purpose language, but this is often not the case with a simulation language.
- FOTRAN or BASIC is available on virtually every computer, but a particular simulation language may not be accessible on the computer that the analyst wants to use.
- The program may require less execution time than the corresponding program written in a simulation language.
- They may allow greater programming flexibility than certain simulation languages.
- Software cost may be lower.

CLASSIFICATION OF SIMULATION SOFTWARE

a. Simulation language Vs Simulators

A simulation language is a computer package that is general nature but may have special features for certain types of applications. A simulator is a computer package that allows one to simulate a system contained in a specific class of systems with little or no programming.

b. Modelling approaches

Event-scheduling approach is whereby a system is modelled by identifying its characteristics events and the writing a set of event routines that give a detailed description of the state changes taking place at the time of each event. The event scheduling approach is available in SIMLIB.

Process approach. A process is a Describe what happens to *entities* over simulated time. Use these descriptions to build event routines (behind the scenes)

c. Common modelling elements

- Entities: a person or an object that arrives to a system and gets serviced in some manner and the usually departs e.g Job (shop), customer (bank).
- Attributes: is a piece of information that describes or characterizes an entity.
- Resources: is a person or machine that provides service to an entity which it is present at the system.
- Queues: a collection of entities with some common characteristics.
- Sources of randomness.

DESIRABLE SOFTWARE QUALITIES

1. General features such as:
 - a. Modelling flexibility
 - b. Ease of model development
 - c. Fast model execution speed
 - d. The maximum model size is allowed by the simulation package
 - e. In some applications, it is convenient for the software to have capabilities for combined discrete continuous simulation.
2. Animation: easy to use animation is one of the main reasons for the increased popularity of simulation modelling.
3. Statistical capabilities: since most real world systems exhibit some sort of random behaviour, a simulation package must contain good statistical capabilities that should actually be used.
4. Customer support: most users of simulation software require some level of ongoing support from the vendor.

5. Output reports: a simulation package should provide time-saving standard reports for commonly occurring performance statistics.

Module 5: CONFIDENCE INTERVALS

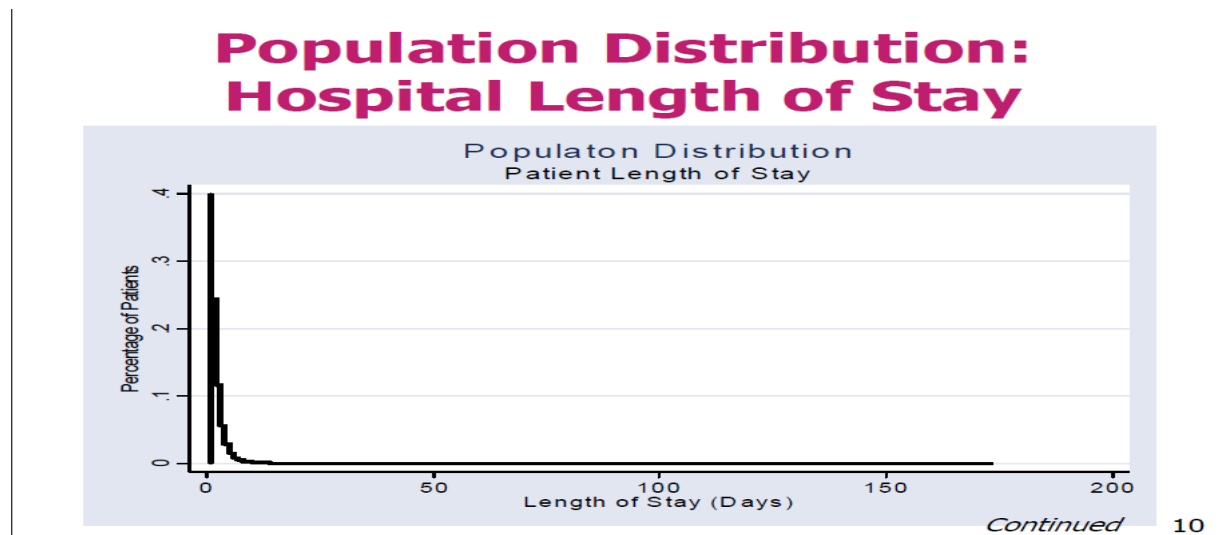
RANDOM SAMPLE

When a sample is randomly selected from a population, it is called a **random sample**. In a simple random sample, each individual in the population has an equal chance of being chosen for the sample. Random sampling helps control systematic bias. But even with random sampling, there is still sampling variability or error. If we repeatedly choose samples from the same population, a statistic will take different values in different samples. If the statistic does not change much if you repeated the study (you get the similar answers each time), then it is fairly reliable (not a lot of variability)

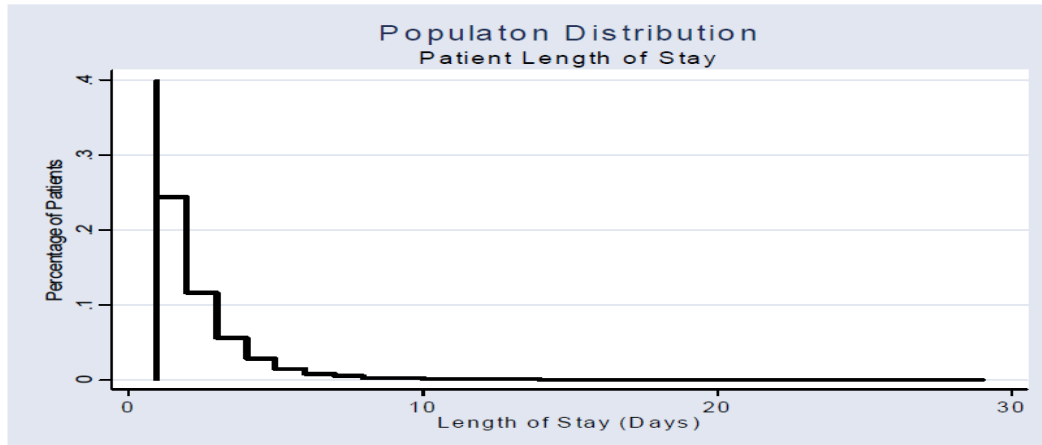
Example: Hospital Length of Stay

The distribution of the length of stay information for the population of patients discharged from a major teaching hospital in a one year period is a heavily right skewed distribution

- Mean, 5.0 days, SD 6.9 days
- Median, 3 days
- Range 1 to 173 days



Population Distribution: Hospital Length of Stay



Hospital Length of Stay

Suppose I have a random sample of 10 patients discharged from this hospital. I wish to use the sample information to estimate average length of stay at the hospital. The sample mean is 5.7 days. How “good” an estimate is this of the population mean?

Suppose I take another random sample of 10 patients . . . and the sample mean length of stay for this sample is 3.9 days.

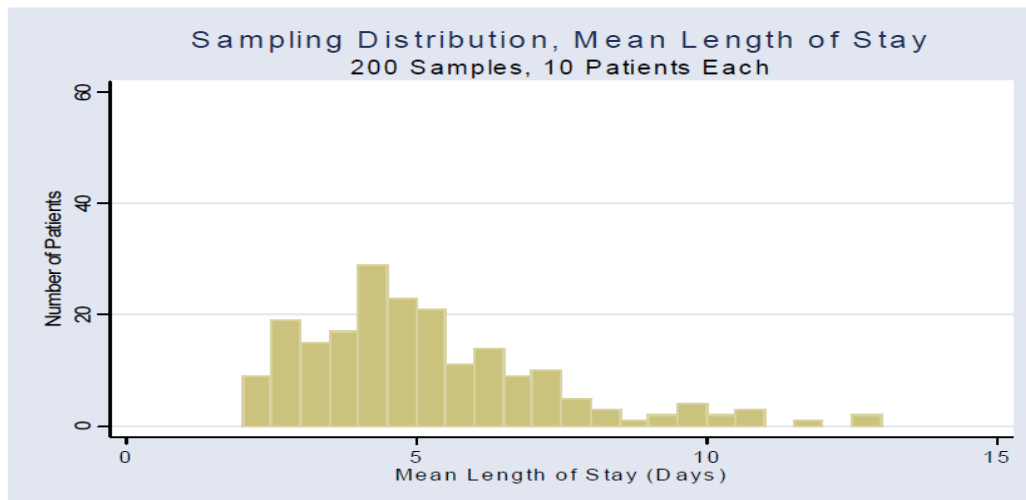
I do this a third time, and get a sample mean of 4.6 days.

Suppose I did this 200 times. If I want to get a handle on the behavior of my sample mean estimate from sample to sample is to plot a histogram of my 200 sample mean values.

The Sampling Distribution

The sampling distribution of the sample mean refers to what the distribution of the sample means would look like if we were to choose a large number of samples, each of the same size from the same population, and compute a mean for each sample.

Sampling Distribution, $n = 10$

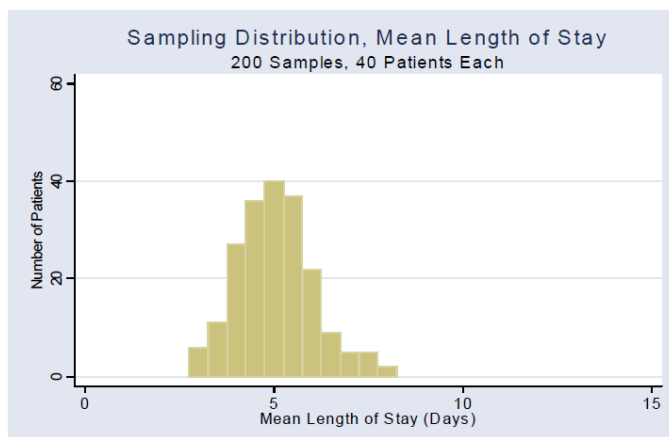


16

Sampling Distribution, $n = 40$

Suppose I again took 200 random samples, but this time, each sample had 40 patients. Again, I plot a histogram of the 200 sample mean values.

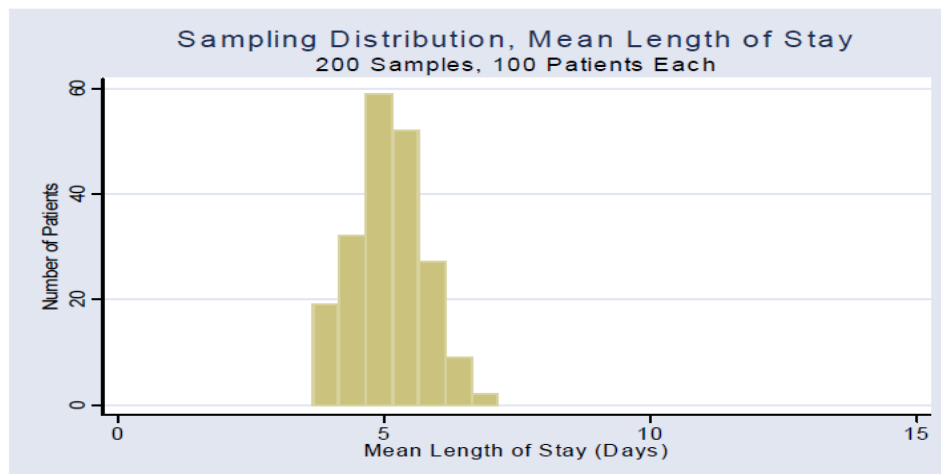
Sampling Distribution, $n = 40$



Mean length of stay from 200 samples, each of size $n = 40$

Continued 10

Central Limit Theorem



Mean length of stay from 200 samples, each of size $n = 100$

Comparing Sampling Distributions

Did you notice any pattern regarding the sampling distributions and the size of the samples from which the means were computed?

- Distribution gets “tighter” when means is based on larger samples.
- Distribution looks less like distribution of individual data, more like a “normal” curve.

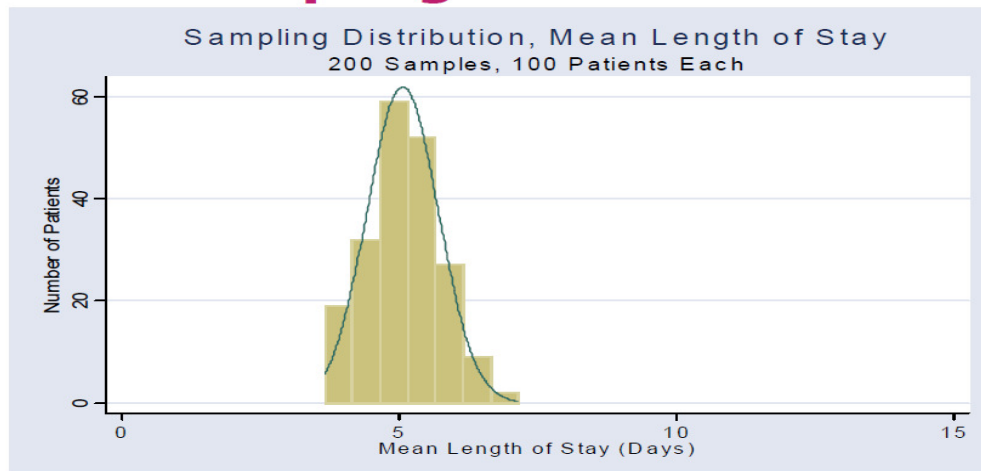
Mathematical statisticians have figured out how to predict what the sampling distribution will look like without actually repeating the study numerous times and having to choose a sample each time.

Often, the sampling distribution of a sample statistics will look “normally” distributed.

- This happens for sample means and sample proportions,
- This happens for sample mean differences and differences in sample proportions.

It’s not practical to keep repeating a study to evaluate sampling variability and to determine the sampling distribution. The sampling distribution of a statistic is often a normal distribution.

Sampling Distribution



200 samples of size 100; a normal probability density is superimposed on the histogram

27

This mathematical result comes from the **central limit theorem**:

- For the theorem to work, it requires the sample size (n) to be large.
- “Large sample size” means different things for different sample statistics.
 - For sample means, the standard rule is $n > 60$ for the Central Limit Theorem to kick in.

Statisticians have derived formulas to calculate the standard deviation of the sampling distribution, it's called the standard error of the statistic.

Central Limit Theorem

If the sample size is large, the distribution of sample means approximates a normal distribution. The central limit theorem (CLT) works even when the population is not normally distributed (or even continuous!!)

Estimate the proportion of persons in a population who have health insurance; choose a sample of size $n = 100$. The true proportion of individuals in this population is **.80**.

Population Density



Example

♦ Sample 1

$$n = 100 \quad \hat{p} = \frac{83}{100} = .83$$

Is the sample proportion reliable?

–If we took another sample of another 100 persons, would the answer bounce around a lot?

Example

Sample 1

$$\hat{p} = \frac{83}{100} = .83$$

Sample 3

$$\hat{p} = .79$$

Sample 2

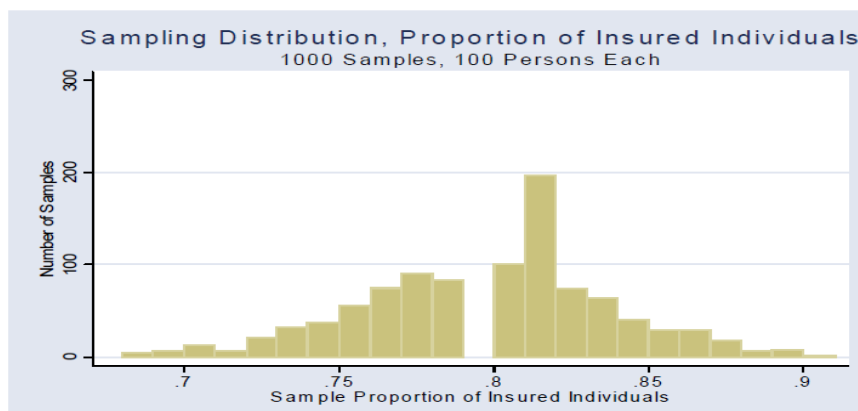
$$\hat{p} = \frac{81}{100} = .81$$

Sample 4

$$\hat{p} = .86$$

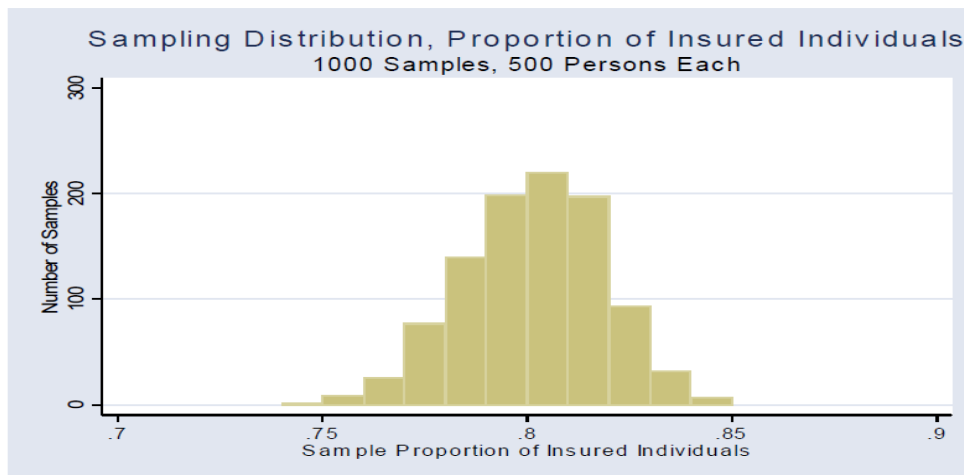
Sampling Distribution for p-hat

From 1,000 Samples of Size n = 100



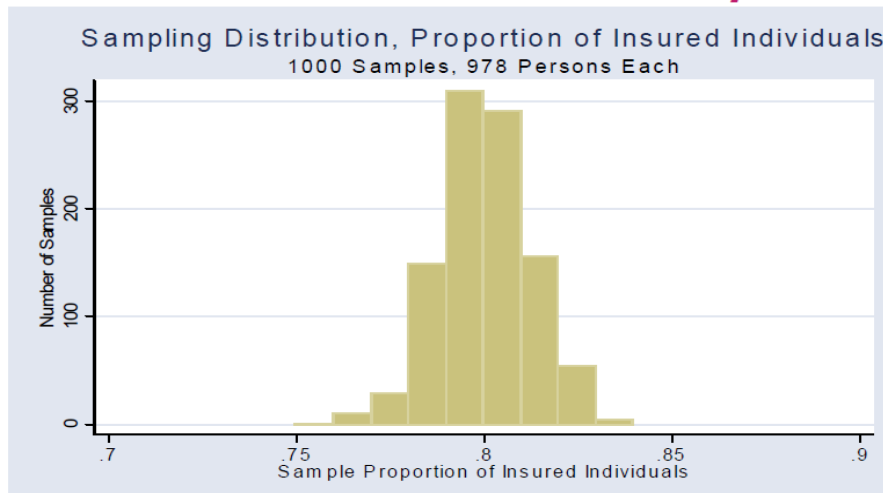
This is the sampling distribution of the sample proportion, based on 1,000 samples of size 100, when the population value is $p = .8$

Sampling Distribution for \hat{p} From 1,000 Samples of Size $n = 500$



41

Results of 1,000 Random Samples Each of Size 978 from the Same Population



This is the sampling distribution of the sample proportion \hat{p} when the population value is $p = .8$

42

Normal Distribution

Why is the normal distribution so important in the study of statistics?

It's not because things in nature are always normally distributed (although sometimes they are). It's because of the central limit theorem: The sampling distribution of statistics—like a sample mean—often follows a normal distribution if the sample sizes are large

Sampling Distribution

Why is the sampling distribution important?

If a sampling distribution has a lot of variability (that is, a big standard error), then if you took another sample, it's likely you would get a very different result.

About 95% of the time, the sample mean (or proportion) will be within two standard errors of the population mean (or proportion).

–This tells us how “close” the sample statistic should be to the population parameter.

Standard errors (SE) measure the precision of your sample statistic.

A small SE means it is more precise.

The SE is the standard deviation of the sampling distribution of the statistic.

Mathematical statisticians have come up with formulas for the standard error; there are different formulas for:

–Standard error of the mean (SEM).

Standard error of a proportion.

These formulas always involve the sample size n . As the sample size gets bigger, the standard error gets smaller.

Standard deviation measures the variability in the population, while Standard error measures the precision of a statistic—such as the sample mean or proportion—as an estimate of the population mean or population proportion.

Practice Problems

If the income data of nine doctors is given as below (in thousands of \$):

37 102 34 12 111 56 72 17 33

The mean is 52.67 and the standard deviation is 35.6.

1. How sure are we about our estimate of μ , the true mean income among the doctors? Give an estimate of the standard error on our best estimate of μ .
2. Suppose we took a random sample of 40 students, instead of nine. What is a sensible estimate for the standard deviation in this sample of 40?
3. What is a sensible estimate for the standard error of \bar{X} , the sample mean from the sample of 40 people?

Solution

1.

- ♦ So, the standard error of the mean—SEM, or $se(\bar{X})$ —estimate is . . .

$$SEM = \frac{s}{\sqrt{n}} = \frac{35.6}{\sqrt{9}} = \frac{35.6}{3} = 11.86$$

- Recall, our sample standard deviation, s , is just an estimate of the population standard deviation.
 - This should not change too much with a change in sample size.
 - We have no other information about the sample of size 40, so our “guesstimate” of s is the value from the sample of size 9: 35.6.
-

Again, we have a “guesstimate” for s , and know the sample size:

$$n = 40$$

- The best estimate for the SEM would be:

$$SEM = \frac{s}{\sqrt{n}} = \frac{35.6}{\sqrt{40}} = \frac{35.6}{6.3} = 5.65$$

Remember s and SEM are not the same thing! They are estimating variability for two different distributions:

S—An estimate of the overall variability in the entire population.

SEM—An estimate of the variability of the value of the sample mean among samples of equal size.

Confidence Intervals for the Population Mean μ

Standard Error of the Mean

- ♦ The *standard error of the mean (SEM)* is a measure of the precision of the sample mean

$$SEM = \frac{s}{\sqrt{n}}$$

Example

- ♦ **Measure systolic blood pressure on random sample of 100 students**

Sample size $n = 100$
 Sample mean $\bar{X} = 123.4$ mm Hg
 Sample SD $s = 14.0$ mm Hg

$$SEM = \frac{14}{\sqrt{100}} = 1.4 \text{ mmHg}$$

Population Mean and Sample Mean

- ♦ **How close to the population mean (μ) is the sample mean (\bar{X})?**
- ♦ The standard error of the sample mean tells us!

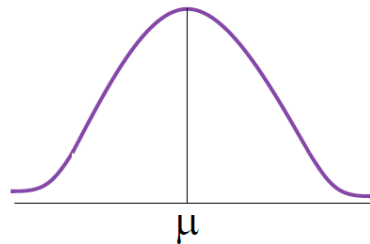
If we can calculate the sample mean and estimate its standard error, can that help us make a statement about the population mean?

The central limit theorem tells us that the sampling distribution for is approximately normal given enough data. Additionally, the theorem tells us this sampling distribution should be centred about the true value of the population mean.

The standard error of gives us a measure of variability in the sampling distribution. We can then use properties of the normal distribution to make a statement about μ .

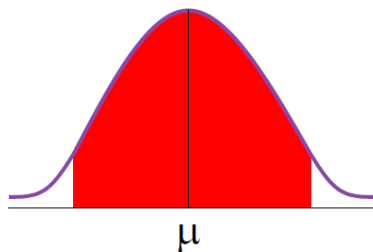
Sampling Distribution

- ♦ *Sampling distribution* is the distribution of all possible values of \bar{x} from samples of same size, n



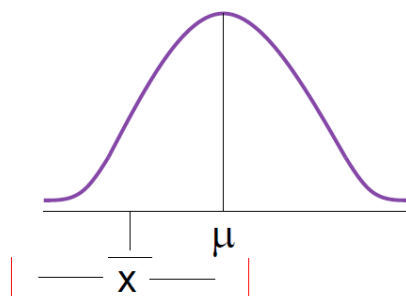
Sampling Distribution

- ♦ 95% of possible values for \bar{x} will fall within approximately two standard errors of μ



Sampling Distribution

- ♦ The “reverse” is also true—95% of the time μ will fall within two standard errors of a given \bar{x}



Continued

Sampling Distribution

- ♦ 95% of the time, the population mean will lie within about two standard errors of the sample mean
 - $\bar{x} \pm 2 SEM$
- ♦ Why is this true?
 - Because of the central limit theorem

Interpretation

We are 95% confident that the sample mean is within two standard errors of the population mean.

Confidence Interval


- ♦ A 95% confidence interval for population mean μ is

$$\bar{x} \pm 2 SEM$$

- ♦ The confidence interval gives the range of plausible values for μ

Example

- ♦ Blood pressure
 $n = 100$, $\bar{X} = 125$ mm Hg, $s = 14$
- ♦ 95% CI for μ (mean blood pressure in the population) is . . .

 $125 \pm 2 \times 1.4$
 125 ± 2.8

Ways to Write a Confidence Interval

- 122.2 to 127.8

- (122.2, 127.8)
- (122.2–127.8)

We are highly confident that the population mean falls in the range 122.2 to 127.8.
The 95% error bound on the mean is 2.8.

Notes on Confidence Intervals

Interpretation

–Plausible values for the population mean μ with high confidence

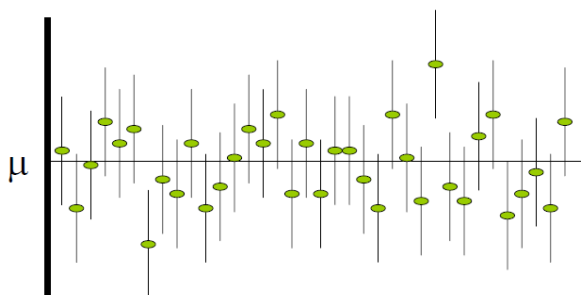
Are all CIs 95%?

- No
- It is the most commonly used
- A 99% CI is wider.
- A 90% CI is narrower.

The length of CI decreases when . . .

- n increases
- s decreases
- Level of confidence decreases—for example, 90%, 80% vs 95%.

Confidence Interval



Each bar represents a 95% CI created from a random sample of size n

Underlying Assumptions

- ♦ In order to be able to use the formula

$$\bar{x} \pm 2SEM$$
- ♦ The data must meet a few conditions that satisfy the underlying assumptions necessary to use this result
- Random sample of population—important!

- Observations in sample independent
- Sample size n is at least 60 to use ± 2 SEM
- Central limit theorem requires large n !

If sample size is smaller than 60

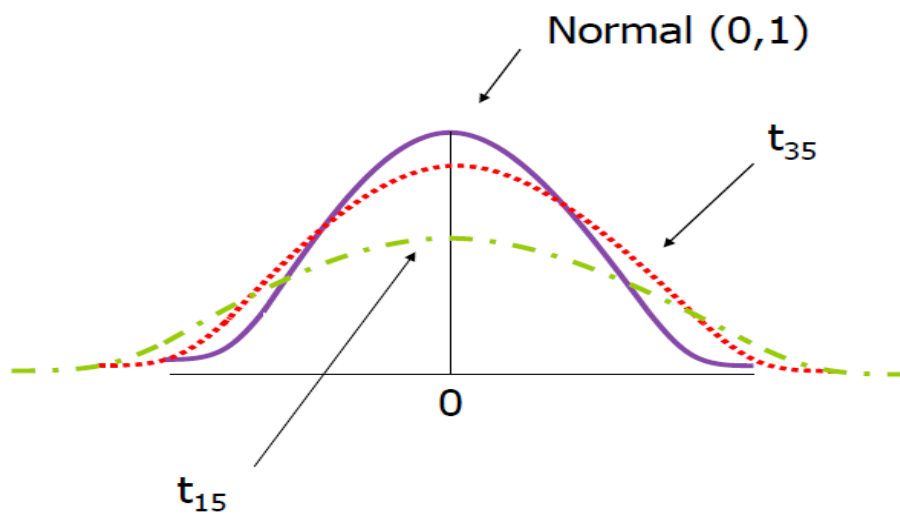
- The sampling distribution is not quite normally distributed
- The sampling distribution instead approximates a “t-distribution”

The t-distribution

The t-distribution looks like a standard normal curve that has been “stepped on”—it’s a little flatter and fatter.

A t-distribution is solely determined by its degrees of freedom—the lower the degrees of freedom, the flatter and fatter it is.

The t-distribution



Underlying Assumptions

- ♦ If sample size is smaller than 60
 - There needs to be a small correction—called the **t-correction**
 - The number 2 in the formula $\bar{x} \pm 2SEM$ needs to be slightly bigger

How much bigger the 2needs to be depends on the sample size.
You can look up the correct number in a “t-table” or “t-distribution” with $n-1$ degrees of freedom.

Adjustment for Small Sample Sizes

$$\bar{X} \pm t^* (\text{SEM})$$

$$\bar{X} \pm t^* (s/\sqrt{n})$$

Adjustment for Small Sample Sizes

Value of $T_{.95}$ Used for 95% Confidence Interval for Mean

df	T	df	T
1	12.706	12	2.179
2	4.303	13	2.160
3	3.182	14	2.145
4	2.776	15	2.131
5	2.571	20	2.086
6	2.447	25	2.060
7	2.365	30	2.042
8	2.036	40	2.021
9	2.262	60	2.000
10	2.228	120	1.980
11	2.201	∞	1.960

You can use the above table or use the one below:

df	80%	90%	95%	99%	df	80%	90%	95%	99%
1	3.078	6.314	12.706	63.657	16	1.337	1.746	2.120	2.921
2	1.886	2.920	4.303	9.925	17	1.333	1.740	2.110	2.898
3	1.638	2.353	3.182	5.841	18	1.330	1.734	2.101	2.878

4	1.533	2.132	2.776	4.604	19	1.328	1.729	2.093	2.861
5	1.476	2.015	2.571	4.032	20	1.325	1.725	2.086	2.845
6	1.440	1.943	2.447	3.707	21	1.323	1.721	2.080	2.831
7	1.415	1.895	2.365	3.500	22	1.321	1.717	2.074	2.819
8	1.397	1.860	2.306	3.355	23	1.319	1.714	2.069	2.807
9	1.383	1.833	2.262	3.250	24	1.318	1.711	2.064	2.797
10	1.372	1.812	2.228	3.169	25	1.316	1.708	2.060	2.787
11	1.363	1.796	2.201	3.106	26	1.315	1.706	2.056	2.779
12	1.356	1.782	2.179	3.055	27	1.314	1.703	2.052	2.771
13	1.350	1.771	2.160	3.012	28	1.313	1.701	2.048	2.763
14	1.345	1.761	2.145	2.977	29	1.311	1.699	2.045	2.756
15	1.341	1.753	2.131	2.947	30	1.310	1.697	2.042	2.750

df	80%	90%	95%	99%
infinity	1.282	1.645	1.96	2.576

Example: Blood Pressure

- $n = 5$, $\bar{X} = 99$ mm Hg, $s = 16$
- 95% CI is $\bar{X} \pm 2.78$ SEM
 - 2.78 from t-distribution with 4 degrees of freedom

Example: Blood Pressure

- ♦ $n = 5$, $\bar{X} = 99$ mm Hg, $s = 16$
- ♦ 95% CI is $\bar{X} \pm 2.78$ SEM

→
$$99 \pm 2.78 \times \frac{16}{\sqrt{5}}$$

Example: Blood Pressure

- ♦ The 95% CI for mean blood pressure is . . .
 - (79.1, 118.9)
 - 79.1–118.9
- ♦ Rounding off is okay, too
 - (79, 119)

MODULE 6: PROBABILITY DISTRIBUTIONS

In order to carry out a simulation using random inputs such as interarrival times, we have to specify their probability distributions. Almost all real systems contain one or more sources of randomness.

Parameterization of Continuous distributions

For a given family or continuous distributions e.g normal distribution, there are usually several ways to define, or parameterize, the probability density function.

- Location parameter/shift parameters (on the x-axis) which specifies the location point of a distribution's range of values.
- Scale parameter which determines the scale or unit of measurements of the values in the range of the distribution.
- Shape parameter which determines the basic form or shape of a distribution within the general family of distributions of interest.

CONTINUOUS DISTRIBUTIONS

Uniform Distribution

- Flat valued, a straight line, $U(a,b)$, first approximation

$$\text{Density: } f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

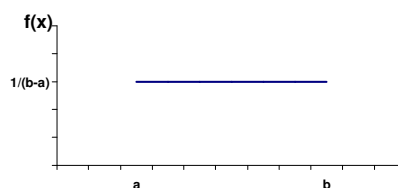
$$\text{Distribution: } F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } b < x \end{cases}$$

$$\text{Range: } [a, b]$$

$$\text{Mean: } \frac{a+b}{2}$$

$$\text{Mode: } \textit{not unique}$$

$$\text{Variance: } \frac{(b-a)^2}{12}$$



Exponential distribution

- A common and easy model
- Special case of gamma or Weibull distributions
- Inter-arrival times of 'customers'

Density :
$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

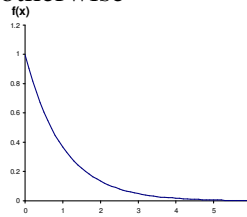
Distribution :
$$F(x) = \begin{cases} 1 - e^{-x/\beta} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Range : $[0, \infty]$

Mean : β

Mode : 0

Variance : β^2



Gamma distribution

- Task completion distribution
- Acquires some similarity to small sample distributions

Density :
$$f(x) = \begin{cases} \frac{\beta^{-\alpha} x^{\alpha-1} e^{-x/\beta}}{\Gamma(\alpha)} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

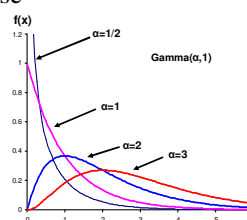
Distribution :
$$F(x) = \begin{cases} 1 - e^{-x/\beta} \sum_{j=0}^{\alpha-1} \frac{(x/\beta)^j}{j!} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Range : $[0, \infty]$

Mean : $\alpha\beta$

Mode : $\beta(\alpha - 1)$ if $\alpha \geq 1$, 0 if $\alpha < 1$

Variance : $\alpha\beta^2$



Weibull distribution

- Task completion distribution
- Like gamma, but gives emphasis to a mode value

$$\text{Density : } f(x) = \begin{cases} \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

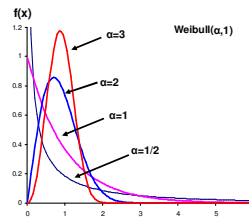
$$\text{Distribution : } F(x) = \begin{cases} 1 - e^{-(x/\beta)^\alpha} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Range : } [0, \infty]$$

$$\text{Mean : } \frac{\beta}{\alpha} \Gamma\left(\frac{1}{\alpha}\right)$$

$$\text{Mode : } \begin{cases} \beta \left(\frac{\alpha-1}{\alpha}\right)^{1/\alpha} & \text{if } \alpha \geq 1 \\ 0 & \text{if } \alpha < 1 \end{cases}$$

$$\text{Variance : } \frac{\beta^2}{\alpha} \left\{ 2\Gamma\left(\frac{2}{\alpha}\right) - \frac{1}{\alpha} \left[\Gamma\left(\frac{1}{\alpha}\right) \right]^2 \right\}$$



Normal distribution

- Error terms of various types (uncertainty about a mean)

$$\text{Density : } f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2 / 2\sigma^2}$$

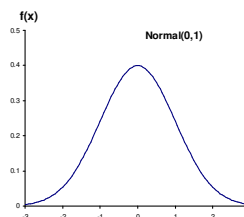
$$\text{Distribution : } \text{no closed form}$$

$$\text{Range : } [-\infty, \infty]$$

$$\text{Mean : } \mu$$

$$\text{Mode : } \mu$$

$$\text{Variance : } \sigma^2$$



Lognormal distribution

- Found to represent many natural phenomena such as infant mortality vs. age.

Density : $f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-(\ln(x)-\mu)^2/2\sigma^2}$

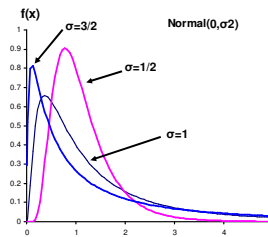
Distribution : no closed form

Range : $[0, \infty]$

Mean : $e^{\mu+\sigma^2/2}$

Mode : $e^{\mu-\sigma^2}$

Variance : $e^{2\mu+2\sigma^2/2}(e^{\sigma^2}-1)$



Triangular Distribution

- Rough model in absence of data

Density : $f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{if } a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{if } c \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$

Distribution : $F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{(x-a)^2}{(b-a)(c-a)} & \text{if } a \leq x \leq c; \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & \text{if } c < x \leq b; \\ 1 & \text{if } b < x \end{cases}$

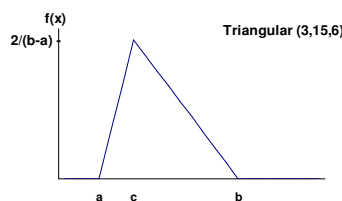
$F'(x) = \begin{cases} \frac{x(b-a)}{(c-a)} & \text{if } a \leq x \leq c; \\ 1 - \frac{x}{1 - \frac{(c-a)}{(b-a)}} & \text{if } c < x \leq b; \end{cases}$

Range : $[a, b]$

Mean : $\frac{a+b+c}{3}$

Mode : c

Variance : $\frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}$



OTHER DISTRIBUTIONS

Discrete distributions: These include Bernoulli, Discrete Uniform, Binomial, Geometric, negative Binomial and Poisson distribution.

Empirical distributions: In some situations, we might want to use the observed data themselves to specify directly a distribution, called empirical distribution.

TECHNIQUES FOR ASSESING SAMPLE INDEPENDENCE

An important assumption made by many of the statistical technique is that the observations of the variables are independent. If the assumption of independence is not satisfied, then these statistical techniques may not be valid, but heuristic techniques such as histograms can still be used.

There are two graphical presentations for informally assessing whether the data is independent.

- a. Correlation plot
- b. Scatter diagram

I am leaving the details of these to you, the reader, to research.

MODULE 7: RANDOM NUMBER GENERATORS

A simulation of any system or process in which there are inherently random components requires a method of generating or obtaining numbers that are random.

Numerical or arithmetic ways to generate random numbers are sequential methods, with each new number being determined by one of several of its predecessors according to a fixed mathematical formula. An example is the **mid-square method**. This method is illustrated by the following example:

Start with a four digit positive integer Z_0 , e.g 7182 and square it to obtain an integer with up to eight digits; if necessary, append zeros to the left to make it exactly eight digits. Squaring 7182 gives 51581124. Take the middle four digits of this eight-digit number as the next four digit number, Z_1 . In this case 5811 and then place a decimal point at the left of Z_1 to obtain 0.5811 which is the first $U(0,1)$ random number, U_1 . Then the next Z_1 will be 5811 then take the square and so on...

There are several objections to the mid-square method:

- It is not random at all, in the sense of being unpredictable. Indeed, if we know one number, then next is completely determined since the rule to obtain it is fixed.
- The mid-square method has a strong tendency to degenerate to zero, where it will stay forever. This illustrates the danger in assuming that a good random number generator will always be obtained by doing something to one number to obtain the next.

A good arithmetic random-number generator should possess the following properties:

- The numbers produced should appear to be distributed uniformly on $[0,1]$ and should not exhibit any correlation with each other.
- The generator should be fast and avoid the need for a lot of storage.
- It should be able to produce a stream of random numbers exactly. This will make debugging and verification of the computer program easier. We might also want to use identical random numbers in simulating different systems in order to obtain a more precise comparison.
- There should be provision in the generator to produce several separate “streams” of random numbers. A stream is simply a subsegment of numbers produced by the generator, with one stream beginning where the previous stream ended.

LINEAR CONGRUENTIAL GENERATORS (LCGs)

The great majority of random-number generators are LCGs, in which a sequence of integers Z_1, Z_2, \dots is defined by the recursive formula

$$Z_i = (aZ_{i-1} + c) \pmod{m}$$

Where m (the modulus), a (the multiplier), c (the increment), and Z_0 (the seed or starting value) are nonnegative integers. We obtain the desired random numbers U_i (for $i=1,2,\dots$) on $[0,1]$, by letting $U_i = Z_i/m$.

Consider the LCG defined by $m=16$, $a=5$, $c=3$ and $Z_0=7$.

Then you get the formula: $Z_i = (5Z_{i-1} + 3) \pmod{16}$ with $Z_0=7$.

So at Z_1 , $Z_1 = (5*7+3) \pmod{16} = 38 \pmod{16} = 6$ and $U_1 = 6/16 = 0.375\dots$ and so on for Z_2 onwards.

The length of a cycle is called the *period* of the generator. When the period is in fact m , the LCG is said to have full period.

Other examples of LCGs are mixed generators and multiplicative generators.

Other types of generators include:

- More general congruences which define a general function

$Z_i = g(Z_{i-1}, Z_{i-2}, \dots) \pmod{m}$ where g is a fixed deterministic function of previous Z_i s.

- Composite generators that take two or more separate generators and combine them in some way to generate the final random numbers. A common example is the use of a second LCG to shuffle the output of the first LCG.
- Tausworthe and related generators which are related to cryptographic methods and operate directly with bits to form random numbers.

TESTING RANDOM-NUMBER GENERATORS

There are two kinds of tests:

a. Empirical tests

These are tests based on statistical tests based on the actual U_i s produced by a generator.

- i. Chi-square test which tests to see whether the U_i s appear to be uniformly distributed between 0 and 1.

- Basic chi squared (χ^2)
 - Divide (0,1) IID into k subintervals (≥ 100)
 - Generate U_1, U_2, \dots, U_n , for $n/k \geq 5$

$$\chi^2 = \frac{k}{n} \sum_{j=1}^k (f_j - \frac{n}{k})^2;$$

$$f_j = \text{count}(U_i \text{'s in } j^{\text{th}} \text{ subinterval})$$

- Reject null hypothesis if:

$$\chi^2 > \chi_{k-1, 1-\alpha}^2$$

- ii. Serial test which is a generalization of the chi-square test to higher dimensions.

- Multi-dimensional chi squared (χ^2)
 - Divide (0,1) IID into k subintervals (≥ 100)
 - Generate non-overlapping d -tuples
 - $\mathbf{U}_1 = (U_1, U_2, \dots, U_d), \mathbf{U}_2 = (U_{d+1}, U_{d+2}, \dots, U_{2d}), \dots, \mathbf{U}_{n/k^d}$ for $n/k^d \geq 5$
- Compute and test

$$\chi^2 = \frac{k^d}{n} \sum_{j_1=1}^k \sum_{j_2=1}^k \dots \sum_{j_d=1}^k (f_{j_1 j_2 \dots j_d} - \frac{n}{k^d})^2$$

- iii. The runs test which is a more direct test of the independence assumption.
- iv. A direct test which is a way of assessing whether the generated UiS exhibit discernable correlation.

b. Theoretical tests

These are not tests in the statistical sense but they use numerical parameters. These tests indicate how well an LCG can perform by looking at its defining constants m , a , and c . They differ from empirical tests in that they are global that is, an LCG's behaviour over its entire cycle is examined.

MODULE 8: OUTPUT DATA ANALYSIS FOR A SINGLE SYSTEM

TYPES OF SIMULATIONS WITH REGARD TO OUTPUT ANALYSIS

A **terminating simulation** is one for which there are natural events E that specifies the length of each run. Since the initial condition for a terminating simulation generally affects the desired measures of performance, these conditions should be representative of those for the actual system.

Example 1: A bank closes each evening. If the establishment is open from 9 to 5, the objective of the simulation might be to estimate some measure of the quality of customer service over the period beginning at 9 a.m. and ending when the last customer who entered before the doors closed at 5 p.m. has been served. In this case $E = \{\text{at least 8 hours of simulated time have elapsed and the system is empty}\}$, and the initial conditions for the simulation are the number of customers present at time 0.

A **non-terminating** simulation is one for which there is no natural event E to specify the length of a run. A measure of performance for such a simulation is said to be a steady-state parameter.

Example 2: Consider a company that is going to build a new manufacturing system and would like to determine the long-run (steady-state) mean hourly throughput of their system after it has been running long enough for the workers to know their jobs and for mechanical difficulties to have been work out. Assume that:

- a. The system will operate 16 hours a day for 5 days a week.
- b. There is negligible loss of production at the end of one shift or at the beginning of the next shift.
- c. There are no breaks that shut down production at specified times each day.

This system could be simulated by “pasting together” 16-hour days, thus ignoring the system idle time at the end of each day and on the weekend. Let N_i be the number of parts manufactured in the i th hour. If the stochastic process N_1, N_2, \dots has a steady state distribution with corresponding random variable N , then we are interested in estimating the mean $E(N)$.

STATISTICAL ANALYSIS FOR TERMINATING SIMULATIONS

a. Estimating means:

Suppose we wish to obtain a point estimate and an approximate 90% confidence interval for the expected average delay of a customer over a day, in a bank that opens at 9 am and closes at 5 pm, but waits until the last customer has left. You can calculate the mean, variance, standard deviation and confidence interval from data you observed or given.

b. Estimating other measures of performance

Depending on the particular situation, other measures of performance can be estimated. For example, in the bank system, we looked at other measures such as the average delay of customers, the average number in queue, etc.

c. Choosing initial conditions

The measures of performance for a terminating simulation depend explicitly on the state of the system at time 0, thus care must be taken in choosing appropriate initial conditions. The first approach to doing this is to estimate the initial conditions. The second approach is to collect data.