“...making simulations of what you're going to build is tremendously useful if you can get feedback from them that will tell you where you've gone wrong and what you can do about it”, C. Alexander
Outline

- Algorithms and Pseudo-Codes
- Principles of Simulation
- Simulating Coin Tosses
- Inventory Simulation on Excel Spread Sheet
Review of Previous Lecture

- We established system, system boundary, system environment and components of systems:
  - Entity:
  - Attribute
  - State:

- A bank branch example with two teller
  - If arrivals, service times and random choice sequence are known, we obtain an artificial history.
  - But can we know these information in reality?? So?
Algorithms

- Algorithm is a set of logical expressions for computations that will be executed by computer.

- Pseudo-code gives an implementation of a computer program without any syntax specifications...
The word *algorithm* comes from Persian mathematician Muhammad ibn Musa al-Khwarizmi (780-850 BC) who was a scholar in the House of Wisdom in Baghdad.

Algorithms are used to convey the general structure of a computer program without going too much into detail.
Main components of Algorithms:

- Flow line (connector)
- Input/output node
- Processor node
- Decision node
- Terminal node
Example 1: Write an algorithm that calculates \(3^2\).

Example 2: Calculate \(\sin(x)\) for \(x = k\pi/4, k \in \mathbb{Z}, -4 \leq k \leq 4\)

Example 3: For given \(a, b, c > 0\), write an algorithm that solves

\[ax^3 + bx + c = 0\]

by trying all integers in \([-c, \infty)\).
Bank Teller Example:

- As an industrial engineer, you are assigned to make improvements on a bank branch whose customers are complaining over long waiting times...

- There are two tellers:
  - Teller#1: Experienced
  - Teller#2: Young, fresh graduate.

- 2 Types of customers:
  - complicated transactions that take a long time (L),
  - standard transactions that take a short time (S).
The algorithm that simulates Bank Teller example for $N$ customers:

Two fundamental processes:
1) Creating arrival and service times for each customer
2) Calculation of service beginning and departure times according to the status of bank tellers and queue length.

-state variables:
- $Y_1$: 1 if the queue 1 is empty, 0 otherwise.
- $Y_2$: 1 if the queue 2 is empty, 0 otherwise.
- $X_1$: number of people in queue 1.
- $X_2$: number of people in queue 2.

-flowchart:
- START
- If customer number is less than $N$, create customer interarrival times.
- If teller 1 or teller 2 is empty, enter the empty teller.
- If queue lengths are equal, choose one of the queues randomly.
- Join the shortest queue.
- Calculate service start time for each customer.
- Calculate customers' departure times.
- Calculate average waiting times.
- Print average waiting times.
- END.
Start
For i=1 to N
  Create Arrival and Service Times for Customer(i)
  If Teller#1 or Teller#2 is empty?
    Yes: {Start Service; Calculate Departure Time;}
    No:
      {Calculate # of People in Queues;
      If lengths of Queues are Equal?
        Yes: Choose a Random Queue;
        No: Join the shortest Queue;
      Calculate Service Start Time;
      Calculate Departure Time;}
End For;
Take Average Waiting Time;
Bitir;
Bank Teller Example (N Replications)

ALGORITHM

1. Set Simulation Parameters
2. For i=1 to M,
   a. For N Customers Create Artificial History
   b. Waiting Avg Waiting Time(i);
3. End For
4. Calculate Average Waiting Time

PSEUDO-CODE

1. Set Simulation Parameters
2. For i=1 to M,
   a. For N Customers Create Artificial History
   b. Waiting Avg Waiting Time(i);
3. End For
4. Calculate Average Waiting Time
How to Simulate Randomness

- The unknown features of a system are covered with random variables in models, e.g. arrivals of customers, service times of tellers, outcomes of coin tosses.

- In Grocery Store example we used a sequence of numbers to generate random arrivals and services.

- Similarly, we use random numbers to generate randomness in systems.
How to Simulate Randomness

- We need random variables with the following properties:
  - The numbers should be *uniformly* distributed in \([0,1]\)
  - Subsequent numbers should be *statistically independent* of all previous numbers
How to Simulate Randomness

10000 simulated random variables in [0,1].

Uniform means:

\[ \Pr\{a \leq x \leq b\} = \frac{1}{b - a}, \quad 1 \geq b \geq a \geq 0 \]
How to Simulate Randomness

- What if we generate 10000 random variates with a ‘pattern’?

- Is the second set of random variables *uniform* or *random* in any sense?

- We want our random variables to be statistically independent.

- Statistical independence means history of a random variable cannot be used to estimate the future values.
Principles of Simulation

- How about histograms???

As $N \to \infty$, histogram of the sample approaches to the theoretical distr.

In Excel we obtain random numbers with RAND()
How to Simulate Randomness

- At the beginning of a simulation study, the developer should define
  - inputs (random variables and distributions),
  - events,
  - system states and
  - outputs of the model (measures of system performance).

- Also the simulation table must be designed.

- Simulation table provides the structure and the steps of simulation.

- Each column should be either an activity, a random variable, a system state, an event or a model output.
Coin Tossing Game

- Charlie tosses a coin for 100 times. Outcomes
  - Heads: Tom -> Harry $1
  - Tails: Tom <- Harry $1

- Possible questions:
  - How often Harry or Tom is ahead during the game?
  - What is the chance of Harry winning at the end of a game?

- Several similar questions might be asked.
Coin Tossing Game

- **How to simulate:**
  - Generate 100 random variables in a column, say Column A
  - Transform random variables into coin tosses using the formula =If(Ax<0.5,”H”, ”T”) for the cell Bx.
  - In the third column (Column C) calculate Harry’s winnings for each coin toss.
  - Do this simulation for many times and calculate output measure for each simulation?

Let’s apply this recipe together...
Coin Tossing Game

<table>
<thead>
<tr>
<th>Toss#</th>
<th>Rand.</th>
<th>Coin Toss</th>
<th>Harry's Winning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.266959</td>
<td>H</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.709649</td>
<td>T</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.959649</td>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>0.429037</td>
<td>H</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.235177</td>
<td>H</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0.518273</td>
<td>T</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0.783652</td>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td>8</td>
<td>0.894311</td>
<td>T</td>
<td>-2</td>
</tr>
<tr>
<td>9</td>
<td>0.137429</td>
<td>H</td>
<td>-1</td>
</tr>
<tr>
<td>10</td>
<td>0.462031</td>
<td>H</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0.669757</td>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td>12</td>
<td>0.667056</td>
<td>T</td>
<td>-2</td>
</tr>
<tr>
<td>13</td>
<td>0.813984</td>
<td>T</td>
<td>-3</td>
</tr>
<tr>
<td>14</td>
<td>0.976344</td>
<td>T</td>
<td>-4</td>
</tr>
<tr>
<td>15</td>
<td>0.934739</td>
<td>T</td>
<td>-5</td>
</tr>
</tbody>
</table>

Formula of the cell C2:

=IF(B2<0.5,"H","T")
The # of times Harry is ahead is 11.

If we run the simulation for 20 times...
Simulation is widely used in inventory management problems.

- Demand is unknown
- Mostly we have varying lead times
- Stock-outs have undesired consequences...

Examples:
Newsvendor, grocery store, aircraft maintenance
Inventory Simulation

Periodic Review Systems:
Review every $N$ period and order-up-to $M$.

Continuous Review Systems:
Review continuously, and order-up-to $M$

Output measures:
Total profit
Total cost
Cost of lost sales
Salvage value

It is also important **what happens to remaining inventory** and **customers reaction to stock-out**.
Inventory Simulation

- **Inventory systems use following inputs**
  - Demand distr.
  - Lead time (distr.)
  - Purchase cost
  - Selling price
  - Fixed ordering cost
  - Holding cost and cost of lost sales

- **Inventory systems have the following parameters:**
  - Maximum inventory level
  - Review period
  - Order quantity
  - Lead time
Inventory Management Simulation

- Consider a newsvendor.
  - Lead time is zero.
  - Purchase cost = 0.33
  - Selling Price = 0.5
  - Salvage val. = 0.05

<table>
<thead>
<tr>
<th>Distribution of types of days:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
</tr>
<tr>
<td>Fair</td>
</tr>
<tr>
<td>Poor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution of demand for different days:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td>70</td>
</tr>
<tr>
<td>80</td>
</tr>
</tbody>
</table>

How to simulate in an Excel spreadsheet?
Inventory Management Simulation

- Simulated the system using excel:
  - Generate random numbers
  - Calculate demand
  - Calculate revenue
  - Calculate total costs
  - Calculate daily and monthly profit

<table>
<thead>
<tr>
<th>Day#</th>
<th>Rand Num</th>
<th>Demand</th>
<th>Pruchase Cost</th>
<th>Holding Cost</th>
<th>Lost Sale Cost</th>
<th>Order Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.19592</td>
<td>0.79479</td>
<td>80</td>
<td>0</td>
<td>0.5</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>0.80114</td>
<td>0.27995</td>
<td>40</td>
<td>0</td>
<td>0.15</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>0.54998</td>
<td>0.04875</td>
<td>50</td>
<td>0.05</td>
<td>0.3</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>0.24643</td>
<td>0.2175</td>
<td>60</td>
<td>0.4</td>
<td>0.15</td>
<td>90</td>
</tr>
<tr>
<td>5</td>
<td>0.69534</td>
<td>0.38893</td>
<td>70</td>
<td>0.1</td>
<td>0.15</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>0.49239</td>
<td>0.72189</td>
<td>80</td>
<td>0.35</td>
<td>0</td>
<td>90</td>
</tr>
</tbody>
</table>

Results of Replic.1
Average Daily Profit= -0.961
Monthly Total Profit= -28.3
Is this information enough??

Histogram summarizes the information of monthly profit...

What can we say about profitability of this newsvendor??
END OF LECTURE 2

Next Lecture: Chapter 3, General Principles of Simulation