Process Analysis in Operations Management
A Concise, Book-Style Chapter

1 Why Process Analysis Matters in Operations Management

Operations Management (OM) studies how organizations design and run the systems that create goods and services. A useful way to see almost any organization is through its *process*: a set of activities, performed by resources, that transforms inputs into outputs. Factories transform raw materials into products, hospitals transform patients into recovered patients, universities transform students into graduates, and logistics networks transform bulk shipments into delivered parcels.

Managers care about processes for a practical reason: a process view makes performance measurable. If you can measure performance, you can compare alternatives and improve the system. At the same time, OM emphasizes *modeling and abstraction*. Real life is detailed and messy; a good OM model keeps the details that matter for decisions and temporarily ignores details that do not.

Patient view vs. manager view: a motivating example

Consider an operating room (OR) in a hospital. A single patient might go through registration, preparation, surgery, recovery, and discharge. If we follow one patient closely, a *Gantt chart* can map the sequence of activities, their durations, and which steps must happen before others. This is informative for that patient.

However, an OR manager cannot rely on one Gantt chart per patient when hundreds of patients come every month. The manager needs a macro-level representation that supports systematic analysis. This is where the *process view* becomes powerful.

2 The Process View: Activities, Buffers, Resources, and Flow Units

Business process

A **business process** is a network of **activities** performed by **resources** that transform **inputs** into **outputs**. Activities can be physical (cutting fabric) or informational (reviewing an insurance claim). Resources include labor, equipment, space, and capital.

Most processes also contain **buffers**, which are places where flow units wait between activities. A buffer can be a storage area in a factory or a queue in a service system. Buffers are not "mistakes" in themselves; they are often an unavoidable consequence of limited capacity and variability. But buffers are costly because they create waiting, congestion, and sometimes quality or safety risks.

Flow unit

To measure process performance you must define a **flow unit**, the item that moves through the process and represents the output of interest. The flow unit depends on what you want to measure.

For example, in a university you could choose "a student" as the flow unit if you care about how many students graduate. In a milk processing plant you might choose "pounds of milk powder." In a blood donation center you might choose "pints of type AB blood" if that blood type is the output of interest. If your objective changes (for instance, you now care about *all* blood types rather than AB only), the flow unit should change accordingly.

Process flow charts

A **process flow chart** is a diagram showing the major elements of a process: the activities, the buffers, the flow direction, and sometimes decision points (places where a unit may take different routes). Flow charts are useful because they force clarity: what happens first, what happens next, where does waiting occur, and which resources do the work.

Two design choices: boundaries and simplification

When describing a real process, there is no single "correct" flow chart. Two choices are always present:

Process boundary. You must decide where the process starts and ends. A surgeon managing a single operating room may define the boundary as "from entry into the OR to exit from the OR." A surgery-department manager may expand the boundary to include pre-op testing and post-op recovery units. A hospital administrator may expand further to include admissions and discharge planning. Boundaries should match the management question.

Level of simplification. You must decide how detailed your activity list should be. "Operation" could be one block, or it could be decomposed into incision, anesthesia, closure, and so on. More detail increases information granularity but reduces analytical convenience. A good OM model chooses the simplest representation that still answers the question.

3 Core Process Performance Measures

This section introduces five foundational measures. They are easy to define but commonly confused, so reading carefully pays off.

Flow time T

Flow time is the time a *given* flow unit spends in the system, from entry to exit. If you enter a cafeteria at 12:00 and leave at 12:40, your flow time is 40 minutes. If a T-shirt takes 10 minutes from start to finish in a production line, then $T = 10$ minutes.

Flow time is measured in units of time (minutes, hours, days).

Cycle time CT

Cycle time is the time between two successive completions (two consecutive outputs). If a system finishes one unit at 12:00 and the next at 12:05, then the cycle time is 5 minutes.

Cycle time is also measured in units of time. It is not the time a unit spends inside; it is the time gap between finished units.

Flow rate R (throughput)

Flow rate (also called throughput) is the rate at which the process delivers output. It is measured in "units per time," such as shirts per hour or claims per week. If the system produces 15 shirts per hour, then $R = 15$ shirts/hour.

In stable operations, the long-run average outflow rate matches the long-run average inflow rate.

Capacity

Capacity is the *maximum* flow rate the process can sustain given sufficient inputs and demand. Flow rate is what the system is actually producing; capacity is what it could produce at best under the assumed conditions. Always keep the inequality in mind:

$$R \leq \text{Capacity}.$$

Work-in-process inventory I

Work-in-process (WIP) inventory is the number of units currently in the process (in activities and in buffers). In a factory, WIP is partially completed products. In a cafeteria, WIP can be interpreted as the customers inside the system. High WIP often signals congestion and long waits, but there are contexts where higher inventory is desirable (for instance, a lecture benefits from more students attending).

A simple but crucial identity: cycle time and flow rate

By definition, if the system completes one unit every CT time units, then the long-run flow rate is the reciprocal:

$$CT = \frac{1}{R},$$

as long as CT and R use consistent time units (e.g., if CT is in hours/unit, then R is in units/hour).

4 Common Pitfalls and How to Avoid Them

Students often struggle with process measures not because the math is hard, but because the *interpretation* is subtle.

First, do not confuse **flow time** with **cycle time**. Flow time tracks one unit through the system; cycle time compares the completion times of two consecutive units. In processes with parallel work across stages (which is most real processes), flow time is typically larger than cycle time because multiple units can be in progress simultaneously.

Second, do not confuse **flow rate** with **capacity**. Capacity is a maximum; flow rate is what happens. A process can have high capacity but low realized flow rate due to low demand, missing inputs, downtime, or managerial choices.

Third, always check **time-unit consistency**. If R is in units per hour and T is in minutes, then you must convert minutes to hours before multiplying or dividing.

Finally, be careful about **process boundaries**. If one person reports flow time from "hospital admission to discharge" and another reports flow time from "enter OR to exit OR," both may be correct, but they are measuring different processes. Comparisons only make sense when boundaries match.

5 Little's Law: The Backbone of Process Analysis

Among all OM relationships, **Little's Law** is one of the most widely used because it is simple, intuitive, and remarkably general.

Statement of Little's Law

For a **stable** process, the long-run average work-in-process inventory equals the long-run average flow rate times the long-run average flow time:

$$I = R \times T,$$

where

$I =$ average inventory (units), $\quad R =$ average flow rate (units/time), $\quad T =$ average flow time (time/unit).

You can rearrange Little's Law to solve for any one measure given the other two:

$$I = RT, \qquad T = \frac{I}{R}, \qquad R = \frac{I}{T}.$$

Intuition

Imagine taking a snapshot of the process at a randomly chosen time. If the process is fast (small T), units leave quickly and fewer units remain inside on average (small I), all else equal. If the process outputs many units per hour (large R), more units are "in flight" at any moment, increasing I, all else equal.

This intuition is often visualized using cumulative inflow and outflow curves: the vertical gap between the curves at a time corresponds to inventory, while the horizontal gap for a particular unit corresponds to flow time. The average vertical gap is approximately the average slope (flow rate) times the average horizontal gap (flow time), which motivates $I = RT$.

When does it apply?

Little's Law holds for a very broad set of systems as long as the process is **stable**: the long-run inflow rate equals the long-run outflow rate and inventory does not grow without bound over time. It does not require FIFO service; it can hold under FIFO, LIFO, or other service rules. It can also hold when arrival and service times are random.

A practical way to remember the stability condition is this: if a line keeps growing longer forever, the system is not stable, and long-run averages may not exist in the usual sense.

A managerial use: understanding trade-offs

Little's Law shows an unavoidable trade-off among congestion (I), throughput (R), and time in system (T). If a museum can hold at most 100 visitors on average (an inventory cap), it cannot simultaneously have very high entry rate and long average visit times. If management wants to raise R while keeping capacity fixed, T must fall, I must rise, or both.

6 Worked Examples

6.1 Example 1: A three-stage T-shirt process

A T-shirt is produced by three sequential activities: cutting (3 min/unit), sewing (4 min/unit), and packing (3 min/unit). Assume demand and input supply are sufficient to keep the process running at its maximum sustainable rate.

Flow time. A single shirt must receive 3 minutes of cutting, 4 minutes of sewing, and 3 minutes of packing. If we ignore additional waiting inside buffers for this simplified example, the processing content sums to

$$T = 3 + 4 + 3 = 10 \text{ minutes}.$$

Capacity and flow rate. The slowest stage is sewing at 4 minutes per unit. This stage limits the long-run output rate (it is the bottleneck). Completing one unit every 4 minutes corresponds to

$$R = \frac{60}{4} = 15 \text{ shirts/hour}.$$

Under the assumption of full utilization with sufficient demand, the *flow rate equals capacity* here, so capacity is also 15 shirts/hour.

Cycle time. Since the system completes one shirt every 4 minutes in steady operation,

$$CT = 4 \text{ minutes} = \frac{1}{R},$$

which matches the reciprocal relationship.

Average WIP via Little's Law. Convert T into hours: 10 minutes $= \frac{1}{6}$ hour. Then

$$I = RT = 15 \times \frac{1}{6} = 2.5 \text{ shirts}.$$

Interpreting the result: on average, 2.5 shirts are inside the system at various stages (being cut, sewn, packed, or waiting).

6.2 Example 2: Insurance claims and the impact of faster processing

An insurance office processes 10,000 claims per year and operates 50 weeks per year. The average processing time per claim (from entry to completion) is 3 weeks.

Flow rate. A convenient unit is claims per week:

$$R = \frac{10{,}000}{50} = 200 \text{ claims/week}.$$

Inventory. With $T = 3$ weeks, Little's Law gives

$$I = RT = 200 \times 3 = 600 \text{ claims}.$$

So, on average, about 600 claims are in the system at any time (in review, waiting, or being processed).

Now suppose a technology improvement reduces average processing time by 80%, from 3 weeks to 0.6 weeks, while the flow rate remains 200 claims/week.

New inventory. Applying Little's Law again:

$$I_{\text{new}} = 200 \times 0.6 = 120 \text{ claims}.$$

This illustrates a key OM lesson: reducing flow time often dramatically reduces congestion and work-in-process, even if the throughput stays the same.

7 A Practical Measurement Idea: Customers in a Store

Suppose management wants to know the average time customers spend inside a supermarket (or a large store such as IKEA). Directly timing every customer can be expensive. Little's Law suggests a simpler approach if the system is stable.

If you can estimate the average number of customers inside the store at a moment in time (I)—for example, by periodic headcounts or entry-gate sensors—and you can estimate the average flow rate of customers leaving or entering (R)—for example, from transaction counts or door counters—then the average time in store is

$$T = \frac{I}{R}.$$

The key is to ensure the time window is reasonably stable (not during a sudden rush that is still building), and that you match units (customers/hour with hours, or customers/minute with minutes).

8 Closing Perspective

Process analysis starts with a deceptively simple move: define the flow unit and draw a process view that is appropriate for your managerial goal. From there, a small set of performance measures—flow time, cycle time, flow rate, capacity, and inventory—gives a language for describing performance and diagnosing problems. Little's Law then ties the measures together and reveals fundamental trade-offs that every manager must respect.

As you continue in OM, you will learn richer tools for variability, queues, bottlenecks, and process redesign. But the ideas in this chapter remain the foundation: clear process boundaries, careful definitions, consistent units, and the discipline to distinguish what is *possible* (capacity) from what is *actually happening* (flow rate).

Operations Management (ISOM 2700)
Session 3: Bottleneck Analysis
A Concise Textbook-Style Chapter

Based on course slides and transcript

Fall 2025

1 Why Bottlenecks Matter

Operations management studies how organizations design and run *processes*: repeated systems that transform inputs (materials, information, customers) into outputs (products, services, decisions). Many improvements sound attractive—buying faster equipment, hiring more people, adding new technology—but not every investment increases output. Bottleneck analysis provides a disciplined way to answer a practical question:

If we want higher throughput, where should we improve first?

A helpful intuition is the *law of the minimum*. An irregular pipe can only carry as much water as its narrowest gap allows. A wooden barrel holds water only up to its shortest stave. A chain breaks at its weakest link. Likewise, a process produces units only as fast as its most limiting resource allows.

This chapter develops the key definitions, core formulas, and a few common scenarios that occur frequently in homework, exams, and real organizations. The goal is not heavy math; the goal is accurate thinking.

2 Core Definitions and Process Measures

A **flow unit** is the item moving through the process. Depending on context, it could be a customer, a patient, an online order, a loan application, or a manufactured part.

A **resource** is what performs work in the process, such as a worker, a machine, a team, a server, or a shared tool.

A **processing time** is the time a resource needs to work on one unit (for a specified activity). Processing time is often measured in minutes per unit.

Flow rate, cycle time, and capacity

The **flow rate** (also called throughput) is the *actual* average output rate of the process, measured in units per time (e.g., units/hour). We denote it by R.

The **cycle time** is the time between consecutive completed units, so it is the inverse of flow rate:

$$\text{Cycle time} = \frac{1}{R}.$$

The **capacity** of a *resource* is the *maximum* rate that resource can process units (given it has enough work to do). If the workload per unit for a resource is w time/unit, then its capacity is:

$$\text{Resource capacity} = \frac{1}{w}.$$

If w is measured in minutes/unit, then capacity is units/minute; multiplying by 60 converts it to units/hour.

The **process capacity** is the maximum flow rate the entire process can achieve *given enough input and enough demand*. It is a property of the internal process design (resources and times), not of the market or supplier.

Inventory, flow time, and Little's Law (brief reminder)

Inventory I is the average number of flow units inside the process (including waiting). **Flow time** T is the average time a unit spends in the process from entry to exit. Little's Law connects these measures in steady state:

$$I = RT.$$

Although Little's Law is not the main focus here, bottlenecks often create waiting and thus increase inventory and flow time.

3 The Law of the Minimum and the Bottleneck

Most introductory bottleneck analysis begins with a *vanilla* (chain-type) process: each flow unit must visit every required resource in sequence. In such a process, every resource sees the same flow rate R, because every unit that enters must eventually be processed by each resource.

Law of the minimum (chain-type processes)

If resources $1, 2, \ldots, n$ are all required for each unit, and each has capacity $C_1, C_2, \ldots, C_n$, then:

$$C_{\text{process}} = \min\{C_1, C_2, \ldots, C_n\}.$$

The **bottleneck** is the resource with the smallest capacity. By the law of the minimum, the bottleneck capacity equals the process capacity:

$$C_{\text{process}} = C_{\text{bottleneck}}.$$

Utilization and a second way to recognize bottlenecks

The **utilization** of a resource is the fraction of its capacity currently used:

$$u_i = \frac{R}{C_i}.$$

Utilization has no unit and is usually expressed as a percentage.

In a chain-type process, the flow rate R is the same for all resources. Therefore, the smaller the capacity C_i, the larger u_i. This means that in the vanilla setting:

The bottleneck is also the resource with the highest utilization.

The utilization of the **process** is defined similarly:

$$u_{\text{process}} = \frac{R}{C_{\text{process}}}.$$

Because $C_{\text{process}} = C_{\text{bottleneck}}$, the process utilization equals the bottleneck utilization in the chain-type setting.

4 How to Identify the Bottleneck in Practice

A reliable procedure helps you avoid common mistakes:

Step 1. List activities, resources, and processing times. Be clear about which resource is needed for which activity.

Step 2. Recognize the structure. Is it one-to-one mapping? Does one resource do multiple activities? Can one activity be done by multiple resources (pooled resources)? Is there a random path?

Step 3. Compute workload per unit for each resource and convert to capacity.

Step 4. Identify the resource (or pooled resource) with the lowest capacity; that is the bottleneck, and its capacity is the process capacity (for the vanilla chain structure).

The next sections explain the most common scenarios.

5 Three Common Scenarios

Scenario A: One-to-one mapping between resource and activity

Each activity is performed by exactly one resource, and each resource performs exactly one activity. Then workload per unit is simply that activity's processing time for that resource.

Scenario B: One resource is needed in multiple activities

A resource may appear in more than one step for each unit. For example, a worker might both *prepare* and *verify* an order. In that case, that worker's workload per unit is the sum of the processing times across all activities the worker performs for one unit. If Worker A spends 2 minutes in activity 1 and 3 minutes in activity 2 for each unit, Worker A's workload is $2+3 = 5$ minutes/unit, not 2 and not 3.

Scenario C: One activity can be performed by multiple resources (pooled resources)

Sometimes an activity can be done by one of several resources. For example, a call can be answered by any available agent, or a job can be processed on either of two identical machines. In this case, the resources form a **pooled resource** for that activity, and their capacities add.

If an activity can be done by m identical machines, each with capacity C, then the pooled capacity is:

$$C_{\text{pooled}} = mC.$$

If they are not identical, you add their individual capacities:

$$C_{\text{pooled}} = \sum_{j=1}^{m} C_j.$$

The key managerial idea is simple: if demand is high enough, all resources in the pool can work simultaneously, so the pool can complete more units per hour than any single member of the pool.

6 Worked Example 1: One Resource Performs Multiple Tasks

Consider a three-step process. Each unit must pass through all steps (a chain). Processing times are:

Activity	Resource and processing time
1	Worker A: 2 min/unit
2	Worker A *and* Worker B: 3 min/unit for each
3	Worker C: 4 min/unit

A common incorrect instinct is to say "Activity 3 takes the longest time (4 minutes), so Worker C is the bottleneck." This is not a fair comparison because Worker A works in *two* activities per unit.

For each resource, compute workload per unit and capacity.

Resource	Activities used	Workload w (min/unit)	Capacity C (units/hour)
A	1 and 2	$2 + 3 = 5$	$60/5 = 12$
B	2	3	$60/3 = 20$
C	3	4	$60/4 = 15$

The process capacity is the minimum:

$$C_{\text{process}} = \min\{12, 20, 15\} = 12 \text{ units/hour}.$$

Thus Worker A is the bottleneck.

This example illustrates a general lesson: when a resource appears multiple times along the path, you must add up its workload before computing its capacity.

7 Worked Example 2: Pooled Resources Increase Capacity

Consider a two-activity process in sequence. Activity 1 uses Machine A, and Activity 2 can be done by either of two identical Machines B (a pool). Processing times are:

Activity	Resource and processing time
1	Machine A: 2 min/unit
2	Two identical Machines B: 5 min/unit per machine

First compute capacities.

Machine A capacity:

$$C_A = 60/2 = 30 \text{ units/hour}.$$

Each Machine B capacity:

$$C_{B,\text{each}} = 60/5 = 12 \text{ units/hour}.$$

Because there are two identical Machines B working in parallel on Activity 2, the pooled capacity is:

$$C_{B,\text{pooled}} = 2 \times 12 = 24 \text{ units/hour.}$$

Now compare stage capacities:

$$C_{\text{process}} = \min\{30, 24\} = 24 \text{ units/hour.}$$

The bottleneck is Activity 2's pooled resource (the two Machines B together).

This example shows why it is wrong to compute Activity 2's capacity as only 12 units/hour: the pool can handle two units at once.

8 Flow Rate with Input and Demand Constraints

Capacity tells you what the process *could* produce if it had unlimited inputs and unlimited demand. But real systems are also limited by supply and demand. A stable long-run flow rate is determined by the most limiting of three rates:

$$R = \min\{\text{Available input rate},\ \text{Potential demand rate},\ C_{\text{process}}\}.$$

This formula is powerful because it forces you to ask, "What is actually limiting output right now?" The answer may not be the bottleneck resource if the system is starved of input or if demand is weak.

It also leads to three important operating regimes:

An **input-constrained** process has insufficient input. Then R equals the input rate, and even the bottleneck has idle time.

A **demand-constrained** process has insufficient demand. Then R equals demand, and resources have idle time because customers are not arriving fast enough.

A **capacity-constrained** process has enough input and demand, and $R = C_{\text{process}}$. In that regime, the bottleneck is fully utilized (near 100% in the simplified model), while non-bottleneck resources may still be below 100% utilization.

9 Line Balancing: Improving Performance by Redistributing Work

Once a bottleneck is identified, managers often try to increase process capacity by **line balancing**, which means redistributing workload so resources are more evenly loaded.

The central intuition is that bottlenecks have "too much demand" for their time, while other resources have idle capacity. If tasks can be reassigned (workers have the right skills, machines can perform the tasks, quality is maintained), moving work away from the bottleneck or moving additional capacity toward it can raise the process capacity.

A short line-balancing illustration

Suppose a three-activity chain is set up so that Worker A does Activities 1 and 2, and Worker B does Activity 3.

Activity	Resource and processing time
1	Worker A: 3 min/unit
2	Worker A: 1 min/unit
3	Worker B: 2 min/unit

Worker A workload is $3 + 1 = 4$ min/unit, so $C_A = 60/4 = 15$ units/hour. Worker B capacity is $60/2 = 30$ units/hour. The bottleneck is Worker A and the process capacity is 15 units/hour.

If Activity 2 can be moved to Worker B (skill flexibility), then Worker A does 3 minutes/unit and Worker B does $1 + 2 = 3$ minutes/unit. Now both have capacity $60/3 = 20$ units/hour, and process capacity increases to 20 units/hour.

Line balancing often feels like "making the work fairer," but the operational meaning is sharper: it reduces the internal supply–demand mismatch across resources, which can increase throughput.

10 Utilization Profile Charts: Seeing Imbalance at a Glance

A **utilization profile chart** plots each resource's utilization as a bar. It is a visual way to identify the bottleneck (highest bar) and diagnose imbalance (gaps between the bottleneck and other resources).

Two readings are especially useful. First, if even the bottleneck utilization is well below 100%, the process likely has idle capacity overall and is probably input- or demand-constrained. Second, large gaps between non-bottleneck resources and the bottleneck suggest opportunities for line balancing, because some resources have time available that might be used to relieve the bottleneck.

11 Common Pitfalls and How to Avoid Them

Many errors in bottleneck problems come from mixing up *activities* and *resources*, or from comparing quantities that are not expressed on the same basis. The following issues appear repeatedly.

It is incorrect to identify the bottleneck as "the activity with the longest processing time" without checking whether some resource performs multiple activities. Bottlenecks are defined at the *resource* level in this framework.

It is incorrect to average workloads across pooled resources. For a pool, you should add capacities, not average processing times. Averaging only works in special symmetric cases and fails in general, especially when resources have different speeds.

It is easy to lose track of units. A workload might be minutes/unit, while capacity might be computed as units/hour. Converting carefully (often by multiplying or dividing by 60) prevents mistakes.

Finally, remember that *capacity is not always the limiting factor*. If input or demand is smaller than capacity, increasing capacity will not increase output. In that case, the bottleneck in the capacity sense exists, but it is not currently binding.

12 Extension: Random Paths and Effective Capacity (Optional but Useful)

Some processes do not send every unit to every resource. For example, after inspection, only 20% of units might require rework. If you compute capacity at the rework station based only on the units that arrive there, you might mistakenly declare it the bottleneck.

A simple correction is to convert a downstream resource's capacity into an **effective capacity** in terms of *original* flow units. If only a fraction p of units visit Resource C, and Resource C can handle C_C units/hour of the units that reach it, then in terms of original units:

$$C_{C,\text{effective}} = \frac{C_C}{p}.$$

This expresses "how many original units per hour the process can support before Resource C becomes overloaded," allowing fair comparison to upstream resources that see all units.

13 Extension: Implied Utilization for Complex Processes (Not Required)

In complex processes with multiple flow unit types (for example, high-severity and low-severity patients) and different paths, "utilization = flow rate / capacity" can be harder to apply directly because different resources see different mixes of work.

A general tool is **implied utilization**:

$$u^{\text{implied}} = \frac{\text{Demand}}{\text{Capacity}}.$$

Here, "demand" is interpreted as how much workload is required from the resource per unit time, often measured in minutes/hour. Implied utilization can exceed 100%, signaling that the resource cannot meet demand. In many complex settings, the bottleneck is identified as the resource with the highest implied utilization.

14 Summary of Key Formulas

For quick reference, the central formulas in process analysis are:

$$\text{Cycle time} = \frac{1}{R}, \qquad I = RT,$$

$$C_{\text{process}} = \min\{C_1, C_2, \ldots, C_n\} \quad \text{(chain-type processes)},$$

$$R = \min\{\text{Input rate},\ \text{Demand rate},\ C_{\text{process}}\},$$

$$u_i = \frac{R}{C_i}, \qquad u_{\text{process}} = \frac{R}{C_{\text{process}}}.$$

Bottleneck analysis is not just a calculation technique. It is a way of thinking that keeps improvement efforts honest: to increase output, you must relieve the system's most binding constraint, whether that constraint is capacity, input, or demand.

Operations Management, Finance Linkages, and Basic Statistics (A Concise Chapter for Junior Undergraduates)

1 Why Operations Management Shows Up in Finance

Operations management (OM) studies how organizations design, run, and improve processes that produce goods and services. Even when a course focuses on processes and flows, its ideas show up directly in finance and accounting because processes tie up money. Every extra day that a product sits in a warehouse is an extra day that capital is trapped, exposed to risk, and unable to be used elsewhere.

This chapter develops two central bridges between OM and finance.

First, inventory is both an operational buffer and a major balance-sheet asset for many firms, especially retailers. OM helps us quantify how quickly inventory becomes sales and how costly it is to hold that inventory. Second, firm-level financial performance often depends on operational quantities such as *flow rate* (the volume sold per unit time). We will see this connection through *Return on Invested Capital (ROIC)* and a simple decomposition called the *DuPont model*.

Because many operational decisions involve uncertainty (for example, uncertain demand), the chapter ends with the basic statistics ideas needed to describe and manage variability.

2 Inventory: What It Is, Why We Hold It, and Why It Costs Money

Definition and intuition

Inventory is the stock of items that are in the system at a point in time. In a retail store, inventory includes goods sitting on shelves and in the back room waiting to be sold. In a factory, it might include raw materials, work-in-process, and finished goods.

Firms hold inventory for practical reasons. Inventory helps a business meet variation in demand: if customers arrive unpredictably, having stock on hand prevents empty shelves. Inventory also acts as a safeguard against disruptions, such as shipping delays, supplier failures, or sudden spikes in demand. Finally, inventory can support economies of scale: if producing or purchasing in large batches lowers the cost per unit, a firm may intentionally buy or make more than it needs immediately, and then store the excess for later sale.

Costs of inventory

Inventory is useful, but it is never free. Inventory generates:

- **Holding (carrying) costs**, such as storage space, insurance, shrinkage, spoilage, and the *opportunity cost of capital* (money tied up in inventory cannot earn returns elsewhere).

- **Ordering or setup costs**, such as administrative costs of placing orders and switching production lines.
- **Shortage costs**, such as lost sales, customer dissatisfaction, or expensive emergency replenishment when inventory is too low.

A key OM theme is that inventory creates a trade-off between *robustness* and *efficiency*. More inventory makes operations more robust to shocks, but it also increases holding costs and ties up capital.

3 Inventory Turnover, Flow Time, and Weeks of Supply

Flow time and inventory turnover

To connect inventory to performance, we need a time-based view.

Flow time is the average time a unit spends in a process. In the retail context emphasized here, it is the average time an item remains in the firm before it is sold.

Inventory turnover measures how fast inventory is converted into sales. It is defined as the inverse of flow time:

$$\text{Inventory Turnover} = \frac{1}{\text{Flow Time}}.$$

A higher turnover generally suggests a firm sells inventory quickly, reducing the time capital is tied up.

Little's Law: the key bridge

Little's Law is one of the most useful identities in OM. It states:

$$\text{Inventory} = \text{Flow Time} \times \text{Flow Rate}.$$

Rearranging gives:

$$\text{Flow Time} = \frac{\text{Inventory}}{\text{Flow Rate}}.$$

In a sales setting, a natural interpretation of **flow rate** is the number of units sold per unit time (for example, units per year).

From "units" to dollars: using financial statements

In practice, firms sell many different products, so it is difficult to count inventory and sales in a single unit. A supermarket sells water bottles, snacks, rice, shampoo, and thousands of other items; the concept of "one unit" is not comparable across products. Financial statements solve this by using monetary values.

To connect Little's Law to accounting, we convert both the numerator and denominator into dollars by multiplying by purchase cost per unit. After aggregating across products, the result is:

$$\text{Flow Time} \approx \frac{\text{Average Inventory Value}}{\text{COGS}},$$

where:

- **Average Inventory Value** is the (average) dollar value of inventory held over a period.

- **COGS** (Cost of Goods Sold) is the purchasing-cost-based dollar value of goods sold over the period.

This leads to the widely used accounting-style estimate of inventory turnover:

$$\text{Inventory Turnover} \approx \frac{\text{COGS}}{\text{Average Inventory Value}}.$$

Weeks of supply

Another common metric is **weeks of supply**, which expresses inventory in the language of "how long can we keep selling if replenishment stops?" It is:

$$\text{Weeks of Supply} = \frac{\text{Average Inventory Value}}{\text{COGS}} \times 52.$$

Because Average Inventory Value/COGS is a time fraction of a year, multiplying by 52 converts it to weeks.

A larger weeks-of-supply number indicates more robustness against supply disruptions, but it also indicates higher inventory investment.

Per-unit percentage inventory holding cost

Holding cost is often expressed as an annual percentage of inventory value. For example, a firm may estimate that holding inventory for a year costs 20% of its value, accounting for warehousing, shrinkage, and capital cost.

A simple operational insight is that faster turnover reduces the holding burden per unit. The **per-unit percentage inventory holding cost** (per batch turned) is approximated by:

$$\text{Per-unit \% Holding Cost} \approx \frac{\text{Annual \% Holding Cost}}{\text{Annual Inventory Turnover}}.$$

The intuition is straightforward: if the annual holding burden is spread across more "turns" per year, each unit (or each batch) effectively carries less holding cost.

4 Worked Example 1: Turning Inventory into Numbers

Consider a retailer that buys a batch of inventory at the beginning of each quarter and sells it by the end of the quarter. Suppose each batch is worth \$100,000 at purchase cost.

The flow time is one quarter, or 1/4 year. Therefore:

$$\text{Inventory Turnover} = \frac{1}{1/4} = 4 \text{ turns/year}.$$

Weeks of supply is:

$$\text{Weeks of Supply} = \frac{1}{4} \times 52 = 13 \text{ weeks}.$$

Now assume the annual holding cost rate is 16% of inventory value. The per-unit percentage holding cost (per turn) is:

$$\text{Per-unit \% Holding Cost} = \frac{16\%}{4} = 4\%.$$

This calculation captures the core OM message: if you can safely increase turnover (without causing stockouts), holding cost per unit declines.

5 Common Pitfalls in Inventory Metrics

Inventory metrics look simple, but several mistakes occur repeatedly.

First, students often mix up **revenue** and **COGS**. Inventory turnover derived from financial statements uses *COGS*, not revenue, because both inventory value and COGS are measured at purchase cost. Using revenue in the numerator would silently change the meaning and typically inflate turnover.

Second, the phrase "average inventory" matters. Inventory fluctuates over time, so using an end-of-year inventory number can be misleading. In practice, analysts approximate the average using an average of beginning and ending inventory, or a more frequent time average if data are available.

Third, comparing turnover across unrelated industries can be misleading. Turnover depends strongly on the business model and product type. High-value durable goods (for example, cars or luxury items) often have low turnover, while financial firms can appear to have extremely high "turnover" because they can redeploy capital quickly.

Finally, it is a mistake to believe "higher turnover is always better." Very high turnover may reflect lean operations, but it can also indicate chronic understocking and lost sales. OM focuses on optimizing the trade-off, not maximizing a single metric.

6 Return on Invested Capital (ROIC) and the DuPont View

ROIC and economic value

Return on Invested Capital (ROIC) is a central financial measure of how effectively a firm generates returns from capital invested in the business:

$$\text{ROIC} = \frac{\text{Return}}{\text{Invested Capital}}.$$

In value-based thinking, the economic value created is often expressed as:

$$\text{Economic Value Created} = \text{Capital} \times (\text{ROIC} - \text{WACC}),$$

where **WACC** is the weighted average cost of capital (the firm's average cost of financing through debt and equity). If ROIC exceeds WACC, the firm creates value; if it is below WACC, the firm destroys value.

Why OM affects ROIC

At first glance, ROIC is purely financial. However, OM variables such as flow rate (units sold per year) affect both the return generated and the revenue produced, which in turn affects ROIC. To make this connection explicit, we use a DuPont-style decomposition.

Start by multiplying and dividing by revenue:

$$\text{ROIC} = \frac{\text{Return}}{\text{Invested Capital}} = \frac{\text{Return}}{\text{Revenue}} \times \frac{\text{Revenue}}{\text{Invested Capital}}.$$

The first term is a **margin** (profit as a fraction of sales). The second term is an **asset turnover** (how much revenue is generated per dollar of invested capital).

Now express return and revenue with a simple cost structure:

$$\text{Return} = \text{Revenue} - \text{Fixed Costs} - (\text{Flow Rate}) \times (\text{Variable Cost per Unit}),$$

$$\text{Revenue} = (\text{Flow Rate}) \times (\text{Price per Unit}).$$

Substituting yields an operationally meaningful expression:

$$\text{ROIC} = \left[1 - \frac{\text{Fixed Costs}}{\text{Flow Rate} \times \text{Price}} - \frac{\text{Variable Costs}}{\text{Price}}\right] \times \frac{\text{Flow Rate} \times \text{Price}}{\text{Invested Capital}}.$$

This equation makes a key point visible: holding other variables constant, increasing flow rate tends to increase ROIC in two ways. It raises asset turnover because revenue rises relative to invested capital, and it raises margin because fixed costs are spread across more units, lowering fixed cost per unit.

7 Worked Example 2: A Simple ROIC What-if

Suppose a firm sells one product with:

$$\text{Price} = \$100, \quad \text{Variable Cost} = \$80, \quad \text{Fixed Costs} = \$1000/\text{year},$$

$$\text{Flow Rate} = 100 \text{ units/year}, \quad \text{Invested Capital} = \$12{,}000.$$

Compute ROIC using the formula above:

$$\text{ROIC} = \left[1 - \frac{1000}{100 \times 100} - \frac{80}{100}\right] \times \frac{100 \times 100}{12000}.$$

First calculate the margin component:

$$1 - \frac{1000}{10000} - 0.8 = 1 - 0.1 - 0.8 = 0.1.$$

Then the asset turnover component:

$$\frac{10000}{12000} \approx 0.8333.$$

So:

$$\text{ROIC} \approx 0.1 \times 0.8333 = 0.08333 = 8.33\%.$$

Now increase the flow rate by 10% to 110 units/year:

$$\text{ROIC}_{\text{new}} = \left[1 - \frac{1000}{110 \times 100} - \frac{80}{100}\right] \times \frac{110 \times 100}{12000}.$$

Compute the margin:

$$1 - \frac{1000}{11000} - 0.8 = 1 - 0.0909 - 0.8 \approx 0.1091.$$

Compute turnover:

$$\frac{11000}{12000} \approx 0.9167.$$

So:

$$\text{ROIC}_{\text{new}} \approx 0.1091 \times 0.9167 \approx 0.10 = 10\%.$$

The ROIC increases from 8.33% to 10%. This illustrates the OM logic: increasing flow rate (selling more per year) increases ROIC because fixed costs are spread out and invested capital generates more revenue.

8 Basic Statistics for OM: Describing Uncertainty and Variability

Operational decisions are made under uncertainty. A restaurant does not know exactly how many customers will arrive tomorrow. An e-commerce warehouse cannot perfectly predict daily orders. To analyze such systems, we treat key quantities (especially demand) as random variables.

Random variables, mean, and variance

A **random variable** X is a variable whose value is uncertain before it is observed. Two basic summaries are:

$$\mathbb{E}[X] \quad \text{(mean or expected value)}, \qquad \mathrm{Var}(X) \quad \text{(variance)}.$$

The **standard deviation** is $\mathrm{Sd}(X) = \sqrt{\mathrm{Var}(X)}$ and measures typical fluctuation around the mean.

Probability mass/density and cumulative probability

Two related probability descriptions are often used.

The **cumulative distribution function** (CDF) is:

$$F(q) = \Pr(X \leq q).$$

It tells you the probability that the random variable is at most q.

For discrete X, the probability that X equals an exact value q is:

$$\Pr(X = q).$$

(For continuous variables, exact equality probabilities are 0, and we use a density function instead; for this chapter, the key conceptual distinction is that $\Pr(X \leq q)$ accumulates probabilities up to q.)

Expected value in discrete form

If X takes values $x_1, x_2, \ldots, x_n$ with probabilities $p_1, p_2, \ldots, p_n$, then:

$$\mathbb{E}[X] = \sum_{i=1}^{n} x_i p_i.$$

For example, if X equals -1 with probability 0.3, 0 with probability 0.6, and 1 with probability 0.1, then:

$$\mathbb{E}[X] = (-1)(0.3) + 0(0.6) + 1(0.1) = -0.2.$$

Coefficient of variation: a scale-free variability measure

In OM, we often want to compare variability across different scales. A standard deviation of 10 is small relative to a mean of 1000, but large relative to a mean of 20. The **coefficient of variation** (CV) captures this idea:

$$CV(X) = \frac{\mathrm{Sd}(X)}{\mathbb{E}[X]}.$$

A larger CV means more variability relative to the average level.

I.I.D. samples and what averaging does

Suppose $X_1, X_2, \ldots, X_n$ are **i.i.d.** (independent and identically distributed), each with mean $\mathbb{E}[X]$ and variance $\mathrm{Var}(X)$. Define the sample average:

$$\bar{X} = \frac{X_1 + \cdots + X_n}{n}.$$

Then:

$$\mathbb{E}[\bar{X}] = \mathbb{E}[X], \qquad \mathrm{Var}(\bar{X}) = \frac{\mathrm{Var}(X)}{n}, \qquad \mathrm{Sd}(\bar{X}) = \frac{\mathrm{Sd}(X)}{\sqrt{n}}.$$

Averaging reduces variability: the more observations you average, the more stable the average becomes. This is one mathematical reason forecasting and planning often use averages, and why pooling information across time or locations can stabilize decision-making.

Similarly, for the sum $S_n = \sum_{i=1}^n X_i$:

$$\mathbb{E}[S_n] = n\mathbb{E}[X], \qquad \mathrm{Var}(S_n) = n\mathrm{Var}(X), \qquad \mathrm{Sd}(S_n) = \sqrt{n}\,\mathrm{Sd}(X).$$

Pooling reduces relative variability

Pooling is the idea of combining multiple uncertain demands (for example, aggregating demand from many stores into a central distribution center). If demands are roughly independent, pooling can reduce *relative* variability.

For i.i.d. X_i, the coefficient of variation of the sum satisfies:

$$CV\left(\sum_{i=1}^n X_i\right) = \frac{\mathrm{Sd}\left(\sum_{i=1}^n X_i\right)}{\mathbb{E}\left[\sum_{i=1}^n X_i\right]} = \frac{\sqrt{n}\,\mathrm{Sd}(X)}{n\,\mathbb{E}[X]} = \frac{CV(X)}{\sqrt{n}}.$$

The key insight is that the sum becomes more predictable *relative to its size* as n grows. This is one reason pooled queues can reduce waiting variability and consolidated distribution can reduce safety stock needs in supply chains.

9 Normal Distribution and Standardization

The **normal distribution** is commonly used to model aggregated uncertainty. We write:

$$X \sim \mathcal{N}(\mu, \sigma^2),$$

where μ is the mean and σ is the standard deviation.

A particularly useful trick is converting a general normal variable into a **standard normal** variable. Define:

$$Z = \frac{X - \mu}{\sigma}.$$

Then:

$$Z \sim \mathcal{N}(0, 1).$$

This matters because probabilities about X can be computed using standard normal tables (or software):

$$\Pr(X \le Q) = \Pr\left(\frac{X - \mu}{\sigma} \le \frac{Q - \mu}{\sigma}\right) = \Pr\left(Z \le \frac{Q - \mu}{\sigma}\right).$$

A short example

Suppose $X \sim \mathcal{N}(10, 10^2)$. Find $\Pr(X \leq 12.8)$. Standardize:

$$\Pr(X \leq 12.8) = \Pr\left(Z \leq \frac{12.8 - 10}{10}\right) = \Pr(Z \leq 0.28).$$

Using a standard normal table, $\Pr(Z \leq 0.28) \approx 0.6103$, so the probability is about 61.03%.

10 Final Remarks: The OM Mindset Behind the Formulas

The formulas in this chapter are deliberately simple, but the mindset is powerful.

Inventory turnover and weeks of supply translate operational flow into financial language: they quantify how quickly capital tied in inventory becomes sales and how long the firm can operate under disruption. ROIC connects operations to value creation by showing how flow rate can raise both margin (through fixed-cost spreading) and asset turnover (through higher revenue per capital). Basic statistics concepts explain why uncertainty is unavoidable and why pooling and averaging often make systems more stable.

When applying these tools, remember that OM is not about optimizing a single metric in isolation. It is about managing trade-offs: speed versus robustness, efficiency versus service, and short-term financial ratios versus long-term operational capability.

Operations Management: Variability and Service Systems
A Concise Chapter on Queues

1 Why do we wait? The central role of variability

Operations management (OM) studies how organizations design and run processes so that goods and services are delivered efficiently and reliably. A recurring puzzle in service operations is that customers can experience long waits even when average demand seems comfortably below average capacity. This chapter explains that puzzle using queueing models, which are the standard OM abstraction for waiting lines.

A motivating example comes from a busy canteen. Suppose a manager argues as follows: "We operate 10 hours per day and serve 500 students per day, so the average flow rate is 50 students/hour. Our service capacity is 100 students/hour, so utilization is only 50%, so we are doing well." The arithmetic is correct, yet students still observe severe congestion around noon.

The missing ingredient is *variability*: students do not arrive uniformly over the day, and service times are not identical across customers. In real service systems, demand often comes in bursts (e.g., lunch rush), and service can take longer for some customers (e.g., someone struggling with a self-ordering machine). Even if average demand is below average capacity, short-term demand can temporarily exceed capacity, creating queues.

Two simple facts make service systems especially vulnerable to variability.

First, *service can start only when capacity and demand are both present.* If no customers arrive at 3pm, the servers cannot "pre-serve" customers for the 12pm rush.

Second, *service capacity generally cannot be stored.* An empty seat on a flight, an idle cashier minute, or an unused doctor appointment slot cannot be saved and carried over to later periods. This non-storability converts randomness in arrivals and service times into waiting.

A call center illustrates the same logic. Suppose there is one staff member. On average, there are 10 incoming calls per hour, and each call takes 5 minutes on average. The capacity is

$$\mu = \frac{60}{5} = 12 \text{ calls/hour},$$

so utilization is $10/12 = 83.33\%$, which is below 100%. If calls arrived exactly every 6 minutes and each call took exactly 5 minutes, no one would ever wait. But if call arrivals are sometimes bunched together, or if some calls last longer than 5 minutes, calls can overlap and force others to queue. This is the operational meaning of "variability creates waiting."

2 Queueing models: a language for service operations

Queueing models describe a process in which customers arrive, possibly wait, receive service, and depart. The basic structure has two physical components.

A *queue* (or buffer) is where customers wait when service is unavailable.

A *server* is the collection of resources required to serve *one* customer independently at a time. Importantly, a server is not always one person or one machine. If a coffee shop needs two staff members working together to serve one customer (for example, one takes payment and one prepares the drink, always as a pair), then two staff may constitute one server for modeling purposes. In general, having s servers means that at most s customers can be served simultaneously.

To specify a queueing model, we need three ingredients.

Customer arrivals are modeled as a random process. Rather than modeling absolute arrival times (which grow without bound), we model *inter-arrival times*, the random time between consecutive arrivals.

Service times are also modeled as random. Different customers may require different amounts of service time.

System configuration includes the number of servers s, the number of lines (one common queue versus separate queues), and the *queue discipline* (the rule used to pick the next customer, such as first-come-first-served).

Once a model is specified, we can compute or estimate performance measures such as average waiting time and average number of customers in the system. Managers use these measures to assess current performance and to design better systems (for example, by adjusting staffing).

3 Key rates and utilization: connecting queues to process analysis

Queueing models use two fundamental rates.

The *arrival rate* λ is the average number of arrivals per unit time. If one customer arrives every 20 minutes on average, then

$$\lambda = \frac{60}{20} = 3 \text{ customers/hour.}$$

The *service rate* μ is the average number of customers served per unit time *per server*. If the average service time is 15 minutes per customer for each server, then

$$\mu = \frac{60}{15} = 4 \text{ customers/hour.}$$

In OM process terms, λ corresponds to the *flow rate* (throughput) of the system in steady state, and μ corresponds to the *capacity* of each server.

If there are s identical servers working in parallel on different customers, the total service capacity is the pooled-resource capacity

$$\text{Total capacity} = s\mu.$$

The *server utilization* is the fraction of time that server capacity is busy on average. It is defined by

$$\rho = \frac{\lambda}{s\mu}.$$

A basic stability requirement is

$$\rho < 1 \qquad \text{equivalently, } \lambda < s\mu.$$

If $\lambda \geq s\mu$, then customers arrive as fast as (or faster than) they can be served, so the queue tends to grow without bound.

A quick conversion example. Suppose there are $s = 2$ servers. The arrival rate is $\lambda = 10$ customers/hour. The average service time is 10 minutes/customer per server, so $\mu = 60/10 = 6$ customers/hour. Then

$$\rho = \frac{10}{2 \times 6} = 0.8333.$$

Even though the system is stable ($\rho < 1$), queues can still form due to variability.

4 Performance measures and universal relationships

Common queueing performance measures include the following.

L_q is the average number of customers waiting in the queue (not including those being served).

L_s is the average number of customers in the *system*, meaning queue plus service.

W_q is the average time a customer spends waiting in the queue.

W_s is the average time a customer spends in the system (waiting plus service).

Queueing measures have direct analogs in process analysis. The number of customers in system is *inventory* (work-in-process), and time in system is *flow time*. These quantities are connected by relationships that hold for very general queueing systems.

First, time decomposes into waiting plus service:

$$W_s = W_q + \frac{1}{\mu}.$$

Here $1/\mu$ is the mean service time per customer for one server.

Second, the number in system decomposes into queue plus those currently in service. If there are s servers and average utilization is ρ, then on average ρs customers are being served (each server is busy a fraction ρ of the time and serves one customer when busy). Therefore,

$$L_s = L_q + \rho s.$$

Third, Little's Law links inventory, flow rate, and flow time. Applied to the queue and to the entire system, it gives

$$L_q = \lambda W_q, \qquad L_s = \lambda W_s.$$

Little's Law is powerful because it does not require exponential assumptions; it holds under broad conditions as long as the system is stable and the averages exist.

5 The M/M/s family and the special M/M/1 model

Many service systems are approximated by the *M/M/s* model, where "M" stands for "memoryless," meaning exponential distributions.

In an M/M/s queue, inter-arrival times are exponential with mean rate λ, service times are exponential with mean rate μ per server, there are s identical servers, customers wait in one common queue, and service follows first-come-first-served (FCFS). Stability requires $\lambda < s\mu$.

The exponential arrival assumption implies a Poisson arrival count over time intervals: the number of arrivals in a period of length T has mean λT. This fact is useful intuition but is not required to apply the main formulas in this chapter.

The simplest case is *M/M/1*, where $s = 1$. This model is especially important because it has closed-form formulas for many performance measures.

Closed-form formulas for M/M/1

Assume an M/M/1 queue with arrival rate λ, service rate μ, and stability $\lambda < \mu$. Define utilization

$$\rho = \frac{\lambda}{\mu}.$$

The probability that an arriving customer sees exactly n customers in the system is

$$P_n = (1-\rho)\rho^n, \qquad n = 0, 1, 2, \ldots$$

Because there is only one server, an arriving customer must wait if and only if the server is busy. In M/M/1, the probability of delay equals utilization:

$$\mathbb{P}(\text{delay}) = \rho.$$

The mean number of customers in the system is

$$L_s = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1-\rho}.$$

The mean number waiting in queue is

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{\rho^2}{1-\rho}.$$

The mean time in the system is

$$W_s = \frac{1}{\mu - \lambda},$$

and the mean waiting time in queue is

$$W_q = \frac{\rho}{\mu - \lambda}.$$

These formulas are consistent with the universal relationships given earlier. For example, $W_s = W_q + 1/\mu$ and $L_s = \lambda W_s$ can be verified by substitution.

6 Why high utilization is "expensive": congestion grows nonlinearly

A key lesson in service operations is that congestion grows very rapidly as utilization approaches 1. For M/M/1, the average number in system is

$$L_s = \frac{\rho}{1-\rho}.$$

This function is not linear in ρ. For instance, $\rho = 0.5$ gives $L_s = 1$, but $\rho = 0.9$ gives $L_s = 9$, and $\rho = 0.99$ gives $L_s = 99$. This is why systems with high utilization feel fragile: small increases in load or small bursts of demand can cause dramatic increases in delay.

It is also why the naive argument "utilization is only 50%, so waiting should be small" can fail in practice. Even when long-run utilization is moderate, short-run utilization during peak periods can be near 100% or higher. Since capacity cannot be stored, the peak periods dominate customers' experience.

7 Worked examples

Example 1: Take-away restaurant (M/M/1)

Consider a take-away store with one employee. Customers arrive randomly at rate $\lambda = 25$ per hour. The employee serves one customer every 2 minutes on average, so

$$\mu = \frac{60}{2} = 30 \text{ customers/hour.}$$

This is an M/M/1 model with $\lambda < \mu$.

Utilization.

$$\rho = \frac{\lambda}{\mu} = \frac{25}{30} = 0.8333.$$

Average queue length.

$$L_q = \frac{\rho^2}{1-\rho} = \frac{0.8333^2}{1-0.8333} \approx 4.167.$$

Average number in the system.

$$L_s = \frac{\rho}{1-\rho} = \frac{0.8333}{0.1667} \approx 5.$$

This includes the customer being served.

Average waiting time in line. Using Little's Law $L_q = \lambda W_q$,

$$W_q = \frac{L_q}{\lambda} = \frac{4.167}{25} \text{ hours} = 0.1667 \text{ hours} \approx 10 \text{ minutes.}$$

Average time in the system.

$$W_s = \frac{L_s}{\lambda} = \frac{5}{25} \text{ hours} = 0.2 \text{ hours} = 12 \text{ minutes.}$$

Notice that $W_s = W_q + 1/\mu = 10 + 2 = 12$ minutes.

Probability of finding at least one person waiting in line. "At least one waiting in line" means the queue is nonempty, i.e., at least two customers are in the system (one in service and at least one in queue). Thus,

$$\mathbb{P}(\text{at least one waiting}) = 1 - (P_0 + P_1).$$

Compute $P_0 = (1-\rho)\rho^0 = 1-\rho = 0.1667$, and $P_1 = (1-\rho)\rho = 0.1667 \times 0.8333 \approx 0.1389$. Therefore,

$$1 - (P_0 + P_1) \approx 1 - (0.1667 + 0.1389) = 0.6944.$$

Example 2: Inter-arrival times versus arrival times

Suppose the inter-arrival times (in minutes) for five customers are:

$$2,\ 5,\ 1.5,\ 4,\ 3.$$

The arrival time of the first customer is 2. The second arrives at $2 + 5 = 7$. The third arrives at $7 + 1.5 = 8.5$. The fourth arrives at $8.5 + 4 = 12.5$. The fifth arrives at $12.5 + 3 = 15.5$.

This example highlights a practical modeling point: knowing all inter-arrival times is equivalent to knowing all arrival times, and either description can be converted into counts of total arrivals in time windows (for example, how many customers arrived in the first 10 minutes). In queueing, inter-arrival times are often easier to model statistically because they fluctuate around a stable scale rather than drifting upward like absolute arrival times.

8 Common pitfalls and how to avoid them

Queueing models are easy to misuse if you ignore definitions or assumptions.

A frequent pitfall is confusing μ (a *rate*) with service time (a *duration*). If service time is 10 minutes/customer, then $\mu = 6$ customers/hour. Always convert time-per-customer to customers-per-time before using formulas.

Another common mistake is mixing up "in the system" and "in the queue." In an M/M/1 system, if you see 4 people waiting in the queue, then there are 5 customers in the system because one is in service. Formulas such as $P_n = (1 - \rho)\rho^n$ refer to the number in the *system*.

It is also easy to compute utilization incorrectly when there are multiple servers. In M/M/s, utilization is $\rho = \lambda/(s\mu)$, not λ/μ. The latter would be correct only when $s = 1$.

Students also sometimes conclude that "utilization below 100% implies no waiting." The call-center and canteen examples show why this is false: variability can create temporary overload even when long-run average load is below capacity.

Finally, closed-form formulas in this chapter apply specifically to M/M/1. If arrivals are not well approximated as Poisson/exponential, if service times are not exponential, if there are multiple separate lines rather than one common queue, or if the queue discipline is not FCFS (such as hospital triage), then M/M/1 formulas may not be valid. In such cases, you may need an M/M/s model, a different analytic model, or simulation.

9 Conclusion

Variability is a core driver of waiting in service systems because capacity cannot be stored and demand can arrive in bursts. Queueing models translate these ideas into a small set of definitions, rates, and performance measures. Even simple models yield powerful managerial insights: utilization must be below 1 for stability, and as utilization increases, congestion grows sharply and nonlinearly. These lessons help explain everyday experiences—from lunch rushes and call centers to clinics and theme parks—and provide a starting point for designing better service operations.

Queueing Models in Operations Management: A Concise Chapter for Junior Undergraduates

1 Why Queues Matter in Operations Management

A large share of operations management is about *flow*: customers, jobs, calls, patients, data packets, or trucks arrive over time and require processing by limited resources such as staff, machines, or service counters. Whenever demand and capacity interact over time, *waiting* becomes a central performance and cost driver. Queueing models provide a simple, disciplined language for predicting congestion and for making decisions such as how many servers to staff, whether to redesign service layouts, and when pooling resources is beneficial.

A key lesson is that congestion is not only about average rates. Two systems can have the same average arrival rate and the same average service rate, yet have very different waiting times depending on *variability*. Even when average capacity exceeds average demand, random variation in inter-arrival and service times can create queues.

2 Core Concepts and Definitions

Arrival rate and service rate

Queueing models describe flow using rates.

- The **arrival rate** λ is the average number of customers (or jobs) arriving per unit time (e.g., customers/hour).
- The **service rate** μ is the average number of customers that one server can complete per unit time.

If the average service time is t hours per customer, then $\mu = 1/t$. For example, if service takes 4 minutes on average, then $t = 4/60$ hours and $\mu = 60/4 = 15$ customers/hour.

Utilization

The most important single number in queueing is **utilization**, the fraction of capacity that is used on average.

For a single-server system,

$$\rho = \frac{\lambda}{\mu}.$$

For s identical parallel servers (each with rate μ),

$$\rho = \frac{\lambda}{s\mu}.$$

Utilization is intuitive: it is average flow rate divided by average capacity. High utilization means resources are busy most of the time, leaving little slack to absorb randomness.

System measures: L and W

Queueing performance is commonly summarized by four averages:

- L_q: mean number *waiting in queue* (not being served).
- L_s: mean number *in the system* (waiting + in service).
- W_q: mean time spent *waiting in queue*.
- W_s: mean time spent *in the system* (waiting + service).

They are connected by **Little's Law**, a fundamental relationship that holds very broadly:

$$L_s = \lambda W_s, \qquad L_q = \lambda W_q.$$

Little's Law is not specific to any distributional assumptions; it is a conservation-of-flow principle.

Delay probabilities

Two useful probabilities in service design are:

- $P(0)$: probability that there are zero customers in the system.
- $P(\text{delay})$: probability an arriving customer must wait before service begins.

In a one-server setting, these connect tightly to utilization. With multiple servers, they do not.

3 Variability and the Logic of Waiting

If arrivals and services were perfectly regular (no randomness), then as long as capacity exceeds demand on average, waiting could be essentially eliminated: jobs would arrive "just in time" for service. Real operations are not so smooth. When either arrivals or service times fluctuate, clusters of arrivals or occasional long service times create temporary overloads. Those overloads produce a queue, and the queue can persist even when average capacity is sufficient.

This explains a common managerial surprise: *"Our average capacity is bigger than our average demand, so why are people waiting?"* The answer is variability and the lack of slack.

4 The M/M/1 Model (One Server)

What M/M/1 means

The notation M/M/1 describes a queue where:

- arrivals follow a Poisson process (equivalently, exponential inter-arrival times),
- service times are exponential,
- there is 1 server.

The key **stability condition** is:

$$\lambda < \mu \quad \text{(equivalently, } \rho < 1\text{)}.$$

If $\lambda \geq \mu$, average waiting times and average queue lengths grow without bound.

Main formulas for M/M/1

Given $\rho = \lambda/\mu$ with $\rho < 1$, the classic closed-form results are:

Steady-state probabilities. The probability of exactly n customers in the system is

$$P_n = (1-\rho)\rho^n, \qquad n = 0, 1, 2, \ldots$$

In particular, $P(0) = 1 - \rho$.

Average number in system and queue.

$$L_s = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1-\rho}, \qquad L_q = \frac{\lambda^2}{\mu(\mu-\lambda)} = \frac{\rho^2}{1-\rho}.$$

Average time in system and queue.

$$W_s = \frac{L_s}{\lambda} = \frac{1}{\mu - \lambda}, \qquad W_q = \frac{L_q}{\lambda} = \frac{\rho}{\mu - \lambda}.$$

Probability of delay in M/M/1. In a one-server system, an arriving customer waits if and only if the server is busy. Therefore,

$$P(\text{delay}) = \rho.$$

A useful intuition: congestion is nonlinear in ρ

Notice that $W_s = 1/(\mu - \lambda)$. As λ approaches μ, the denominator $\mu - \lambda$ becomes small and waiting increases very rapidly. This "blow-up" is a quantitative statement of a practical truth: operating near 100% utilization is extremely fragile when randomness exists.

5 The M/M/s Model (Multiple Parallel Servers)

When to use M/M/s

The M/M/s model applies when:

- customers arrive according to a Poisson process with rate λ,
- there are s identical servers,
- each server has exponential service time with mean rate μ,
- customers wait in a *single shared queue* and take the first available server.

The stability condition generalizes to:

$$\rho = \frac{\lambda}{s\mu} < 1, \quad \text{or equivalently} \quad \lambda < s\mu.$$

Why we often use a spreadsheet

Unlike M/M/1, M/M/s formulas are more algebraically involved. In many operations courses and in practice, you compute M/M/s performance metrics using a queueing spreadsheet or software. The inputs remain simple:

$$(\lambda, \mu, s),$$

and the outputs typically include ρ, L_q, L_s, W_q, W_s, $P(0)$, and $P(\text{delay})$.

Even if software does the calculation, managers must still interpret the results correctly. Two conceptual points matter a lot:

Utilization is not the same as probability of delay when $s > 1$. With multiple servers, a customer waits only when *all* servers are busy. Utilization ρ reflects the average busy fraction of capacity, but $P(\text{delay})$ is the probability that the system is fully occupied at an arrival instant. Because "all servers busy" is a stricter event than "a typical server is busy," we usually have

$$P(\text{delay}) < \rho \quad \text{for } s > 1.$$

In contrast, for $s = 1$, these two coincide.

Adding one server can reduce waiting by far more than it reduces utilization. Because waiting increases *convexly* as utilization rises, a modest reduction in utilization (by adding capacity) can create a dramatic reduction in queueing. This is why hiring one extra employee can sometimes cut waiting time by 90% even though utilization falls by only 50%.

6 Cost Trade-Off: Staffing as an Optimization Problem

Two types of costs

A service system often faces two competing cost components:

Waiting cost. Let C_w be the cost of waiting per customer per unit time (e.g., dollars per customer-hour). This can represent goodwill loss, abandonment, reduced sales, or penalties. If the average number waiting is L_q, then the average waiting cost per unit time is

$$\text{Waiting cost per unit time} = C_w L_q.$$

Service cost. Let C_s be the cost of operating one server per unit time (e.g., dollars per server-hour). With s servers, service cost per unit time is

$$\text{Service cost per unit time} = C_s s.$$

Total cost and the staffing decision

Total cost per unit time becomes

$$\text{Total cost}(s) = C_s s + C_w L_q(s).$$

Here $L_q(s)$ depends on s through the queueing model. Service cost increases linearly in s, while waiting cost typically decreases quickly at first and then more slowly as s grows. The result is often a U-shaped total cost curve, and the best staffing level is the s that minimizes total cost.

7 Pooled Queue versus Separate Queues

Two designs

Consider s servers that can perform the same type of work.

Pooled (shared) queue. All arrivals join a single queue, and the next available server takes the next customer. This is modeled as one M/M/s system.

Separate queues. Each server has its own line. If customers are assigned to a line (e.g., randomly, or by choice) and do not switch, then the system can be approximated as s independent M/M/1 queues, each with arrival rate approximately λ/s and service rate μ.

Utilization is the same, but waiting differs

In the pooled system,

$$\rho_{\text{pooled}} = \frac{\lambda}{s\mu}.$$

In the separate-queues approximation, each line sees arrival rate λ/s, so each server's utilization is

$$\rho_{\text{separate}} = \frac{\lambda/s}{\mu} = \frac{\lambda}{s\mu}.$$

Thus utilization matches. Yet waiting time is typically *lower* with pooling, especially when utilization is high.

Why pooling reduces waiting

Pooling helps in two reinforcing ways.

First, pooling improves the match between supply and demand. In a pooled queue, it is essentially impossible to have customers waiting while a server is idle, because the next waiting customer immediately takes the idle server. In separate queues, that inefficiency can occur: one line can be long while another server finishes and becomes idle.

Second, pooling reduces effective variability. Combining multiple random arrival streams and "sharing" capacity tends to smooth fluctuations, making the system more stable from the perspective of any individual server.

Practical trade-offs

Pooling is not universally best in every dimension. A pooled queue can look longer and may require more space and better layout management. Also, behavioral effects can matter: in some settings, separate queues increase worker ownership and accountability, while pooling may create opportunities for free riding if monitoring is weak. Many real operations adopt hybrid designs, such as partial pooling (pool two adjacent counters rather than all counters).

8 Common Pitfalls and How to Avoid Them

Pitfall 1: Confusing time units. If λ is in customers per hour, then μ must also be in customers per hour, and W_q and W_s will be in hours. Always convert minutes to hours (or vice versa) before computing.

Pitfall 2: Using an unstable model. For M/M/1 you must have $\lambda < \mu$. For M/M/s you must have $\lambda < s\mu$. If not, reported "infinite" values are not errors; they indicate the system cannot keep up on average.

Pitfall 3: Treating utilization as the probability of waiting in multi-server systems. For $s = 1$, $P(\text{delay}) = \rho$. For $s > 1$, $P(\text{delay})$ is typically smaller than ρ. Do not substitute ρ for $P(\text{delay})$ when interpreting multi-server results.

Pitfall 4: Misinterpreting separate-queue totals. In separate queues, a customer waits in only one line, so W_q is computed per line and does not get multiplied by s. But the *total* number of customers waiting across the whole system is the sum across lines, so total waiting customers is approximately $s \times L_q^{(\text{one line})}$ if the lines are symmetric.

Pitfall 5: Believing "adding one server won't help much" after looking only at utilization. Utilization may decrease modestly, yet waiting can drop dramatically because congestion is nonlinear in utilization. Always evaluate waiting metrics or total cost, not just ρ.

9 Worked Example 1: M/M/1 at a Student Service Counter

Students arrive at an average of one every 15 minutes, and service takes 10 minutes on average. Assume Poisson arrivals and exponential service times. There is one clerk.

Convert to hourly rates:

$$\lambda = \frac{60}{15} = 4 \text{ per hour}, \qquad \mu = \frac{60}{10} = 6 \text{ per hour}.$$

Utilization:

$$\rho = \frac{\lambda}{\mu} = \frac{4}{6} = \frac{2}{3} \approx 0.667.$$

The clerk is idle a fraction $1 - \rho \approx 0.333$, or about 33.3% of the time.

Average waiting time in queue:

$$W_q = \frac{\rho}{\mu - \lambda} = \frac{2/3}{6 - 4} = \frac{2/3}{2} = \frac{1}{3} \text{ hour} = 20 \text{ minutes}.$$

Average queue length:

$$L_q = \lambda W_q = 4 \times \frac{1}{3} = \frac{4}{3} \approx 1.33 \text{ students}.$$

This example illustrates a striking fact: even with the clerk idle one-third of the time, average queueing delay can still be 20 minutes because of randomness and limited slack.

10 Worked Example 2: Pooled versus Separate Queues (Same Utilization, Different Waiting)

A store has four identical servers. Total arrivals are $\lambda = 4.73$ customers/minute. Each server has $\mu = 1.33$ customers/minute.

Utilization in either design is

$$\rho = \frac{\lambda}{s\mu} = \frac{4.73}{4 \times 1.33} \approx 0.889.$$

So both systems are heavily utilized.

If the store uses a *pooled* queue, it is an M/M/4 system with $(\lambda, \mu, s) = (4.73, 1.33, 4)$. A standard M/M/s spreadsheet (or software) gives an average waiting time of about

$$W_q^{\text{pooled}} \approx 1.30 \text{ minutes.}$$

If the store uses *four separate queues* and customers are assigned evenly, each line receives $\lambda/4 = 1.1825$ customers/minute and has one server. Each line is approximated as M/M/1 with utilization

$$\rho_{\text{line}} = \frac{1.1825}{1.33} \approx 0.889,$$

the same as before. However, the M/M/1 waiting time becomes much larger; the spreadsheet (or the M/M/1 formula) yields approximately

$$W_q^{\text{separate}} \approx 6.03 \text{ minutes.}$$

This example highlights the operational value of pooling: it can drastically reduce waiting without changing utilization, especially under high load.

11 Closing Perspective

Queueing models translate the everyday experience of waiting into quantitative predictions that support better operational decisions. The major managerial insights of this chapter are simple but powerful. Variability creates waiting even when average capacity exceeds average demand. Utilization is important, but congestion grows rapidly as utilization approaches one. Multi-server systems require care in interpretation: the probability of delay differs from utilization, and pooling resources can reduce waiting dramatically with the same overall utilization. Finally, good operations decisions come from balancing service cost and waiting cost rather than chasing either extreme.

Operations Management Chapter: Variability, Queues, and Simulation

Concise textbook-style notes for junior undergraduates

1 Why Waiting Happens: Variability Meets Limited Capacity

Operations management is largely about designing and running processes so that customers (or jobs, patients, calls, orders, or parts) flow smoothly through a system. A basic service process has three ingredients: arrivals (customers show up), a buffer (a waiting line or virtual queue), and service (a server or resource processes customers). If the server can always work faster than customers arrive, we expect little waiting. Yet in real life, even well-designed systems experience queues.

The key reason is *variability*. Customers do not arrive at perfectly regular intervals, and service times are not perfectly identical. Some customers take longer; some arrive in bursts. When variability interacts with a system that is highly utilized, waiting can explode. This chapter develops that intuition using a single-server queue as the main teaching example, and then explains how *simulation* becomes essential when clean formulas are not available.

2 Queueing Models and the Meaning of "M" and "G"

A widely used notation describes a queue by the arrival-time distribution, service-time distribution, and number of servers. In this chapter we focus on one server and the first-come-first-served (FCFS) rule.

The M/M/1 model

In an **M/M/1** queue:

- Interarrival times are **exponentially distributed**.
- Service times are **exponentially distributed**.
- There is **1** server, and service is **FCFS**.

Historically, "M" is often read as "memoryless" (the exponential distribution has the memoryless property). The advantage of M/M/1 is mathematical convenience: many performance measures have explicit formulas. The downside is realism: exponential assumptions may not fit many service systems (e.g., appointments, batch arrivals, or tasks with a predictable minimum service time).

The G/G/1 model

In a **G/G/1** queue:

- Interarrival times follow a **general** distribution.

- Service times follow a **general** distribution.
- There is **1** server, and service is **FCFS**.

The M/M/1 model is a *special case* of G/G/1: if the "general" distributions happen to be exponential, G/G/1 reduces to M/M/1.

3 Core Quantities: Rates, Utilization, and Coefficient of Variation

Queueing performance depends on both *average rates* and *variability*. To describe them, we define:

Arrival rate and service rate

Let

$\lambda =$ average arrival rate (customers per unit time), $\quad \mu =$ average service rate (customers per unit time).

If the average interarrival time is $\mathbb{E}[A]$, then $\lambda \approx 1/\mathbb{E}[A]$. If the average service time is $\mathbb{E}[P]$, then $\mu \approx 1/\mathbb{E}[P]$.

In the lecture notation, the **average service time** is written as

$$T_p = \mathbb{E}[P] = \frac{1}{\mu}.$$

Utilization

The **utilization** (also called traffic intensity) is

$$\rho = \frac{\lambda}{\mu}.$$

Interpretation: ρ is the long-run fraction of time the server is busy (in a stable single-server system). When ρ approaches 1, the system is operating near full capacity, leaving little "slack" to absorb randomness.

A useful equivalent expression comes from inverting means:

$$\rho = \frac{\lambda}{\mu} = \frac{1/\mathbb{E}[A]}{1/\mathbb{E}[P]} = \frac{\mathbb{E}[P]}{\mathbb{E}[A]}.$$

Coefficient of variation (CV)

Variability is often summarized by the **coefficient of variation**:

$$CV = \frac{\text{standard deviation}}{\text{mean}}.$$

For interarrival times A and service times P, define

$$CV_a = \frac{\sigma_A}{\mathbb{E}[A]}, \qquad CV_p = \frac{\sigma_P}{\mathbb{E}[P]}.$$

These are dimensionless measures: a larger CV means "more variability relative to the mean."

4 The G/G/1 Waiting Time Formula (and How to Read It)

A practical approximation for the **average waiting time in queue** (excluding service time), denoted W_q, is

$$W_q = \left(\frac{CV_a^2 + CV_p^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)T_p. \tag{1}$$

This formula is powerful because it separates waiting into three intuitive factors:

Variability factor.

$$\frac{CV_a^2 + CV_p^2}{2}.$$

More randomness in arrivals or service increases waiting. Squaring emphasizes that variability grows fast in its effect.

Utilization factor.

$$\frac{\rho}{1-\rho}.$$

This is the "danger term." As $\rho \to 1$, the denominator shrinks and waiting time grows rapidly. The curve is convex: waiting increases slowly at first, then accelerates dramatically near high utilization.

Service-time scale factor.

$$T_p.$$

If service takes longer on average, waiting grows proportionally, all else equal. This is a simple but often overlooked point: faster service shortens both the direct service time and the queueing delay.

A special case: how G/G/1 connects to M/M/1

In an M/M/1 queue, both interarrival and service times are exponential. A key mathematical fact is that for an exponential random variable, the standard deviation equals the mean. Therefore,

$$CV_a = 1, \qquad CV_p = 1.$$

Plugging into (1) gives the variability factor:

$$\frac{1^2 + 1^2}{2} = 1.$$

So the G/G/1 formula reduces to

$$W_q = \left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu} = \frac{\rho}{\mu - \lambda},$$

which matches the well-known M/M/1 waiting-time expression.

A striking managerial implication: reducing variability can halve waiting

Start with an M/M/1 system (so $CV_a = CV_p = 1$). If you can make *either* interarrival times almost constant (so $CV_a \approx 0$) while keeping service exponential, then the variability factor becomes

$$\frac{0^2 + 1^2}{2} = \frac{1}{2},$$

so average waiting is roughly cut in half. The same happens if service times become nearly constant while arrivals remain exponential. If *both* arrivals and service become perfectly regular ($CV_a = CV_p = 0$), then the formula suggests $W_q = 0$: with no variability, the line disappears because the system becomes perfectly synchronized.

5 From Waiting Time to "How Many Are Waiting": Little's Law and Related Measures

Once W_q is known, other measures follow from definitions and Little's Law.

Time in system

Average time in system equals waiting plus service:

$$W_s = W_q + T_p = W_q + \frac{1}{\mu}.$$

Average number in queue and in system

Little's Law states (for stable systems) that

$$L = \lambda W,$$

where L is average number in the system and W is average time in the system. Applying it to the queue only:

$$L_q = \lambda W_q.$$

For the whole system:

$$L_s = \lambda W_s.$$

Alternatively, since in a single-server system the average number in service equals the busy fraction ρ, one can also write

$$L_s = L_q + \rho.$$

Both expressions agree.

6 Common Pitfalls (and How to Avoid Them)

Queueing formulas are simple to write but easy to misuse. The most frequent mistakes come from confusing what is assumed, mixing units, or mixing variables.

First, do not use M/M/1 formulas unless exponential assumptions are justified. If you are given means and standard deviations (as in the lecture example), that is often a sign you should use the G/G/1 formula because variability matters explicitly.

Second, do not mix time units. If interarrival times are in minutes, then service times must also be in minutes when computing T_p, W_q, and W_s. Likewise, if you convert to rates per hour, convert everything consistently.

Third, be careful about what ρ means. Utilization is $\rho = \lambda/\mu$, not μ/λ. A utilization above 1 indicates an unstable system (arrivals exceed capacity), in which case long-run averages like W_q do not remain finite.

Fourth, do not confuse *interarrival time* with *arrival rate*. The arrival rate is not "10 minutes"; it is 1/10 per minute (or 6 per hour). Many arithmetic errors come from skipping this conversion.

Finally, remember that in simulation tables, waiting time is typically *computed*, not generated. Interarrival and service times are generated random inputs; waiting time is a performance outcome driven by those inputs.

7 Worked Example 1: Computing W_q in a G/G/1 Queue

A restaurant has one server. The interarrival time has mean 10 minutes and standard deviation 8 minutes. The service time has mean 8 minutes and standard deviation 10 minutes. Compute the average waiting time in queue.

Step 1: Compute coefficients of variation

$$CV_a = \frac{8}{10} = 0.8, \qquad CV_p = \frac{10}{8} = 1.25.$$

Step 2: Compute utilization

The arrival rate is $\lambda = 1/10$ customers per minute, and the service rate is $\mu = 1/8$ customers per minute. Thus

$$\rho = \frac{\lambda}{\mu} = \frac{1/10}{1/8} = \frac{8}{10} = 0.8.$$

The average service time is $T_p = 8$ minutes.

Step 3: Apply the G/G/1 waiting-time formula

$$W_q = \left(\frac{0.8^2 + 1.25^2}{2}\right)\left(\frac{0.8}{1-0.8}\right)(8).$$

Compute each part:

$$\frac{0.8^2 + 1.25^2}{2} = \frac{0.64 + 1.5625}{2} = 1.10125, \qquad \frac{0.8}{0.2} = 4.$$

So

$$W_q = 1.10125 \times 4 \times 8 = 35.24 \text{ minutes} \approx 35.2 \text{ minutes}.$$

This example illustrates two major drivers of delay: utilization at 80% is high enough to magnify variability, and service-time variability is substantial ($CV_p = 1.25$).

8 Psychology of Waiting: Why "Feels Long" Can Matter as Much as "Is Long"

Operations managers should care about actual waiting times because they influence abandonment, productivity, and cost. But customers respond to *perceived* waiting as well. Two systems with identical average W_q can create very different experiences.

Several robust psychological effects are especially relevant in service design. Unoccupied time feels longer than occupied time, so distractions (information screens, small tasks, or engaging environments) can reduce perceived delay even if actual delay is unchanged. Pre-process waiting (waiting before anything starts) often feels longer than in-process waiting, which is why many services try to start something early (e.g., giving a menu to browse or collecting basic information before service begins). Anxiety increases perceived waiting, and uncertainty tends to feel worse than a known finite wait; providing reliable estimates can improve satisfaction. Unexplained delays feel longer than explained delays, so transparent communication can matter. Perceived unfairness also increases dissatisfaction; for that reason, FCFS policies and single-line designs (when feasible) often feel more equitable. Comfort and social context matter too: uncomfortable waiting and solo waiting tend to feel longer.

A practical lesson is that queue management has both a *physics* side (rates and variability) and a *psychology* side (perception and fairness). Good operations design often improves both.

9 When Formulas Are Not Enough: The Idea of Simulation

Queueing formulas are valuable, but many real systems are too complex for closed-form analysis. Examples include multiple service stages (triage, then treatment), multiple servers with different speeds, customer abandonment (leaving the line), priority classes, time-varying arrival rates, and complicated routing rules.

Simulation is a numerical approach that helps in exactly these settings. The central idea is straightforward:

> Build a simplified but faithful model of the real system on a computer, generate random inputs that mimic real variability, run the model many times (or for a long time), and use the resulting averages to estimate system performance.

Because simulation imitates the system's dynamics, it is especially useful for "what-if" analysis: you can test staffing rules, appointment schedules, or routing policies without disrupting the real operation.

Strengths and limitations

Simulation can handle large, complex problems and supports counterfactual experiments. It does not interfere with real operations and can be inexpensive compared to real-world trials.

However, simulation does not automatically produce an optimal policy, and results depend on modeling choices. If the simulated system fails to capture key real-world features, the recommendations may be misleading. Simulation also requires careful managerial input: what rules govern the process, what randomness should be included, and what performance measures should be tracked.

10 Monte Carlo Simulation: Representative vs. Performance Variables

A basic Monte Carlo simulation follows a common logic.

First, you define the problem and decide what performance you care about (for example, the average waiting time in queue). Next, you identify **representative random variables**: the external random inputs that drive the system. Then you run the model and compute **performance variables**: outputs generated by the system dynamics.

The distinction matters. Representative variables are what you must *generate* from distributions. Performance variables are typically *computed* given the generated inputs.

A simple illustration: the dice game

In the lecture's dice game, the representative random variable is the dice roll at each step. Wealth levels are random too, but they are not generated directly; they are computed from the dice outcomes and the game's rules. The final performance variable might be "who wins" or "probability you win."

11 Simulating a Single-Server Queue: The "Physics" of the Service System

To simulate a single-server FCFS queue, the representative random variables are:

$$A_i = \text{interarrival time between customer } i-1 \text{ and } i, \qquad E_i = \text{service time of customer } i.$$

The performance variables include waiting times C_i, and their average W_q.

A convenient way to run the simulation is to build a time table. For customer i, define:

$$B_i = \text{arrival time}, \quad C_i = \text{waiting time in queue}, \quad D_i = \text{service start time}, \quad F_i = \text{service completion time}.$$

The dynamics are:

Arrival times. Assuming customer 1 arrives at time $B_1 = A_1$, subsequent arrivals satisfy

$$B_i = B_{i-1} + A_i, \qquad i \geq 2.$$

Waiting time. Customer i waits if the previous customer has not finished when i arrives:

$$C_i = \max\{F_{i-1} - B_i,\, 0\}, \qquad i \geq 2,$$

and $C_1 = 0$.

Service start and completion.

$$D_i = B_i + C_i, \qquad F_i = D_i + E_i.$$

After simulating many customers, the estimated average waiting time is

$$\widehat{W}_q = \frac{1}{n}\sum_{i=1}^{n} C_i,$$

where n is the number of customers simulated. The estimate becomes more stable as n grows.

12 Worked Example 2: A Short Hand Simulation of Three Customers

Suppose a single-server system operates FCFS. The first customer's arrival is at $A_1 = 2$ minutes, so $B_1 = 2$. Let service times be $E_1 = 3$, $E_2 = 4$, $E_3 = 2$ minutes. Let interarrival times be $A_2 = 2$ and $A_3 = 1$ minutes.

Customer 1

$$B_1 = 2, \quad C_1 = 0, \quad D_1 = B_1 + C_1 = 2, \quad F_1 = D_1 + E_1 = 5.$$

Customer 2

Arrival:

$$B_2 = B_1 + A_2 = 2 + 2 = 4.$$

Waiting:

$$C_2 = \max\{F_1 - B_2, 0\} = \max\{5 - 4, 0\} = 1.$$

Start and completion:

$$D_2 = B_2 + C_2 = 5, \qquad F_2 = D_2 + E_2 = 9.$$

Customer 3

Arrival:

$$B_3 = B_2 + A_3 = 4 + 1 = 5.$$

Waiting:

$$C_3 = \max\{F_2 - B_3, 0\} = \max\{9 - 5, 0\} = 4.$$

Start and completion:

$$D_3 = B_3 + C_3 = 9, \qquad F_3 = D_3 + E_3 = 11.$$

The average waiting time across these three customers is

$$\widehat{W}_q = \frac{0 + 1 + 4}{3} = \frac{5}{3} \approx 1.67 \text{ minutes}.$$

This toy run is not meant to be accurate for the long run; it simply demonstrates the mechanics of how waiting emerges from the interaction of arrival timing and service completion.

13 Closing Perspective: A Practical Toolkit

For many single-server settings, the G/G/1 waiting-time formula (1) provides a compact, insightful estimate that highlights three levers: reduce variability, reduce utilization, and reduce average service time. It also reminds us that pushing utilization close to 100% is risky because waiting grows superlinearly.

When systems become too complex for reliable closed-form formulas, simulation becomes a flexible alternative. In operations management practice, analysts often use both: formulas to build intuition and sanity checks, and simulation to evaluate realistic policies under realistic variability. The best designs usually combine "physics" (the mathematics of flow) with "psychology" (how waiting is experienced).

Quality Management in Operations Management: Capability and Conformance

Concise textbook-style chapter for junior undergraduates

1 Why Quality Belongs in Operations Management

Operations management (OM) studies how organizations transform inputs into outputs: raw materials into products, patient time into health outcomes, or data into financial services. In all these settings, one fact is unavoidable: *performance varies*. Two burgers cooked on the same grill will not be identical; two phone screens from the same production line will not have exactly the same brightness; two customer-service calls will not take the same amount of time.

From an OM perspective, **quality is tightly linked to variation**. If a process could produce exactly the same output every time, quality problems would largely disappear. Since perfect sameness is impossible, the practical goal is to understand variation, reduce the harmful part of it, and manage the consequences.

This chapter introduces two complementary lenses:

- **Capability analysis** answers a *static* question: given today's process variation, how likely are we to produce defects relative to design specifications?
- **Conformance analysis** (often called *statistical process control*) answers a *dynamic* question: as time passes, is the process stable, or has something abnormal changed that requires action?

We keep the math light and focus on intuition and correct use of key formulas.

2 What Do We Mean by "Quality"?

People use the word *quality* in different ways, and it helps to distinguish them.

First, quality can mean **"goodness" or excellence**, which is partly subjective and tied to design choices. For example, a luxury watch may be considered high quality because of craftsmanship, durability, and brand prestige. This is sometimes called *design quality*: the intended level of performance and features.

Second, and most important in OM, quality often means **consistency or conformance**. Here, a product is high quality when it *conforms to standards* and shows *small variability*. Under this definition, quality is not only about a high average performance; it is also about predictable performance.

Third, quality can be **stakeholder-relative**: satisfaction depends on expectations versus perceptions. A two-day delivery may feel "high quality" in one context and disappointing in another. Also, different parties define quality differently: customers, producers, competitors, and regulators often emphasize different outcomes.

In this chapter we mainly adopt the OM definition: **quality as conformance to specifications in the presence of variation**.

3 The Cost of Quality: Why Managers Care

Quality matters because it affects cost, risk, and reputation. OM typically groups costs into two categories.

Cost of achieving good quality

Firms spend money to prevent and detect problems: *prevention* costs include training, process design, and machine maintenance; *appraisal* costs include inspection and testing.

Cost of poor quality

Defects create losses in several ways. *Internal failure* costs occur before products reach customers, such as scrap and rework. *External failure* costs occur after delivery, such as warranty claims, recalls, and liability. There are also *hidden* costs, including customer dissatisfaction, loss of market share, and long-term brand damage.

A key OM lesson is that poor quality is not merely a technical issue; it is often a major economic issue.

4 Variation: The Root Cause of Quality Problems

A simple process view is:

$$(x_1, x_2, \ldots, x_k) \xrightarrow{\text{process}} (y_1, y_2, \ldots, y_m),$$

where inputs x (materials, labor effort, settings) are transformed into outputs y (dimensions, strength, waiting time, defect rate). The process is influenced by **environmental variables** (temperature, humidity, demand surges, supplier disruptions).

Variation in outputs can arise from: (1) variation in inputs, (2) variation in the environment, and (3) variation inside the process itself (human variability, machine variability, random perturbations). Even with strict standardization, some variability remains. Therefore, it is natural to model a measured product characteristic (such as length or weight) as a **random variable**.

A probabilistic view

If you measure the same characteristic repeatedly (say, the weight of cereal boxes), the values form a distribution. In many OM applications, the distribution is approximately normal (bell-shaped), especially when many small factors contribute to the final outcome. This does not mean outputs are "random with no structure"; it means variability has a *pattern* that can be studied and managed.

5 Two Types of Variation: Natural vs. Assignable

Quality management distinguishes two broad sources of variation.

Natural variation (common-cause variation)

Natural variation consists of random errors inherent to the process. It affects all outputs and is hard to eliminate without changing the process itself (for example, using a different machine or redesigning the workflow). When only natural variation is present, the output distribution tends to be **stable over time** and reasonably predictable.

Assignable variation (special-cause variation)

Assignable variation comes from specific, discoverable causes such as a machine fault, an untrained operator, incorrect material, or a calibration error. Unlike natural variation, it can often be reduced or eliminated through management action. When assignable variation appears, the output distribution can become **unstable over time**.

A subtle point

The boundary between "natural" and "assignable" is not absolute. If management decides a machine is "fixed and untouchable," its quirks may be treated as natural variation; if management is willing to replace or upgrade the machine, those same quirks become something one can assign and fix. In exams and routine calculations, you usually apply the textbook definitions directly.

6 Statistical Process Control: "Capable" vs. "In Control"

Statistical process control (SPC) is a broader framework that connects capability, conformance monitoring, and improvement actions.

A process is called **capable** if its output meets design specifications with sufficiently high probability. Capability is about *how wide the distribution is relative to specification limits.*

A process is called **in control** if it operates without assignable-cause variation and remains stable over time. Being in control is about *stability.*

These ideas are related but not identical: a process can be stable (in control) but still produce too many defects if natural variation is large or specifications are tight; conversely, a process might sometimes meet specs but be unstable due to assignable causes.

7 Specification Limits: Turning "Quality" into a Number

To measure conformance, engineers specify acceptable ranges.

Definition (specification limits)

Let a product characteristic be measured as X (e.g., length in cm). The **lower specification limit** (LSL) and **upper specification limit** (USL) are predetermined by design requirements. A unit is:

$$\text{acceptable if } \text{LSL} \leq X \leq \text{USL}, \quad \text{a defect otherwise.}$$

A **target value** often lies between LSL and USL, but "higher is better" is *not* always true. For a part that must fit into an assembly, being too large can be just as defective as being too small.

What drives defect probability?

Two forces matter most:

1. **Tightness of specifications:** the width $\text{USL} - \text{LSL}$. Wider limits (more tolerance) reduce the chance of defects.
2. **Process variability:** how spread out the distribution of X is. Larger variability increases the chance of defects.

In many OM settings, specifications are determined by engineering or regulation, so managers often have more control over *variability* than over *USL/LSL.*

8 Capability Analysis and the Capability Index C_{pk}

Capability analysis answers: *Given the current mean and variability, how well can we meet specifications?*

Asymmetry: why we look at the "narrow" side

If the process mean is not centered between LSL and USL, the process is more likely to violate the closer limit. Quality management is therefore conservative: it focuses on the worse side.

Definition (process capability index)

Let $\bar{X}$ be the sample average of measured outputs and let σ be the sample standard deviation.[1] The **capability index** is

$$C_{pk} = \min\left\{ \frac{\bar{X} - \text{LSL}}{3\sigma},\ \frac{\text{USL} - \bar{X}}{3\sigma} \right\}.$$

The two fractions measure how many *three-standard-deviation blocks* fit between the mean and each specification limit. Taking the minimum means we judge the process by its more vulnerable side.

How to interpret C_{pk}

A larger C_{pk} indicates better capability (lower defect likelihood), because it means the mean is farther from the nearest specification limit *in units of standard deviation*. Reducing σ generally increases C_{pk}, improving capability.

Connecting C_{pk} to "σ-quality"

An **N-sigma quality** process (in this simplified introductory sense) means the smaller of the distances $\bar{X} - \text{LSL}$ and $\text{USL} - \bar{X}$ equals $N\sigma$ or more. Because C_{pk} divides distances by 3σ,

$$C_{pk} = \frac{N}{3} \quad \text{(in the symmetric case, or for the limiting narrow side).}$$

Therefore:

$$C_{pk} \geq 1 \iff \text{at least 3-sigma quality}, \qquad C_{pk} \geq 2 \iff \text{at least 6-sigma quality.}$$

The normal distribution and 3-sigma intuition

If X is approximately normal and the process is centered, then roughly 99.73% of outcomes lie within $\pm 3\sigma$ of the mean. This is the origin of the "3-sigma" benchmark:

$$\mathbb{P}(\text{within specs}) \approx 0.9973, \qquad \mathbb{P}(\text{defect}) \approx 0.0027.$$

Under the same assumptions, 6-sigma quality corresponds to an extremely small defect probability (on the order of parts per billion).

[1] In practice, different estimators for σ appear depending on the sampling plan. In this chapter we use the simplified version consistent with introductory OM lectures.

9 Six Sigma: Why Chase Such a Tiny Defect Rate?

Six Sigma as a management initiative

Six Sigma is both an ambition (very low defect rates) and a toolkit for process improvement. Historically, it was developed in manufacturing and later widely adopted across industries. A common improvement cycle is DMAIC: *Define, Measure, Analyze, Improve, Control*. The philosophy is that variation drives many quality problems, variation can be identified and controlled, and reducing variation improves performance.

Two reasons 6-sigma can be necessary

First, some industries have extremely high safety and reliability requirements: aviation, medical devices, pharmaceuticals, semiconductors, and certain financial systems. A defect rate that sounds "small" can still be unacceptable when the consequences are severe.

Second, **errors accumulate** across complex systems. A final product may require many components to be within spec. Even if each component has a low defect probability, the probability that *every* component is good can drop substantially as the number of components increases.

To see the idea, suppose a product needs n independent components, and each component is good with probability p. Then:

$$\mathbb{P}(\text{final product is good}) = p^n, \qquad \mathbb{P}(\text{final product defective}) = 1 - p^n.$$

If $p = 0.9973$ (about 3-sigma) and $n = 100$, then p^{100} is noticeably below 1, producing a much higher system-level defect probability. This is why "pretty good" component quality may be insufficient in complex assemblies.

10 Handling Defects: Scrap vs. Rework

When defects occur, operations typically choose between:

Scrapping: removing defective units from the process. This wastes material and capacity already spent, and it creates uncertainty in usable output (you only learn how many good units you have after inspection).

Rework: fixing defective units and sending them back through part of the process. Rework increases workload and can create "bounce-backs" that clog capacity. In services, rework is common and costly: for example, discharging a patient too early may appear to save capacity today, but it can lead to readmission tomorrow, increasing total workload.

This trade-off connects quality directly to capacity, throughput, and cost.

11 Conformance Analysis: Monitoring Stability with Control Charts

Capability analysis is largely static: you take a sample, estimate $\bar{X}$ and σ, and compute C_{pk}. Conformance analysis is dynamic: it asks whether the process remains stable over time.

Definition (in control)

A process is **in control** if it operates without assignable-cause variation. If assignable causes are present, the process is **out of control**, and corrective action should be taken.

Core idea of a control chart

A control chart plots a sample statistic over time (often the sample mean for continuous measurements, or the fraction defective for binary outcomes). The chart includes an upper control limit (UCL) and lower control limit (LCL). If the process is stable, the probability of crossing those limits is very small. Therefore, when you *observe* a point beyond the limits, it is a signal that something unusual may have happened.

Two common chart families are:

1. $\bar{X}$**-charts** for continuous variables (e.g., length, waiting time).
2. p**-charts** for attribute/binary variables (e.g., defective vs. non-defective).

The detailed formulas for UCL/LCL depend on the chart type and sampling plan, and are typically covered in the next lecture/topic. For now, the key message is that control charts support *detection and response*: identify assignable causes and eliminate them, thereby keeping the process stable.

12 Common Pitfalls and How to Avoid Them

Students (and sometimes practitioners) make predictable mistakes in quality analysis.

First, do not confuse **specification limits** with **control limits**. Specification limits (LSL/USL) come from engineering design and define what customers will accept. Control limits (UCL/LCL) come from process data and define what the process typically produces when stable. A unit can be within control limits but still outside specifications if the process is consistently off-target.

Second, do not treat $\bar{X}$ as "the truth." The capability index is based on a *sample*, so a small or biased sample can give misleading capability estimates.

Third, remember the **asymmetric case**. If the mean drifts toward one specification boundary, defects increase even if variability is unchanged. The $\min\{\cdot,\cdot\}$ in C_{pk} is not a mathematical decoration; it encodes the conservative "narrow side" logic.

Fourth, be careful with the meaning of "3-sigma" and "6-sigma." These benchmarks rely on approximate normality and a stable process. If assignable causes are present (out of control), the distribution can shift, and sigma-based defect calculations can become unreliable.

13 Worked Examples

Example 1: Computing capability index and sigma quality

A factory produces computer tables. The required table length must be between 94 cm and 106 cm. A sample of recent output has sample mean $\bar{X} = 100$ cm and sample standard deviation $\sigma = 2$ cm.

Compute the capability index:

$$C_{pk} = \min\left\{\frac{100-94}{3(2)}, \frac{106-100}{3(2)}\right\} = \min\left\{\frac{6}{6}, \frac{6}{6}\right\} = 1.$$

Interpretation: $C_{pk} = 1$ corresponds to **3-sigma quality** in this symmetric case. The process is capable at approximately the 3-sigma level.

Example 2: How much must variability drop to reach 6-sigma?

Using the same specifications and mean, what is the largest σ that would achieve 6-sigma quality?

Six-sigma quality corresponds to $C_{pk} \geq 2$. Here,

$$C_{pk} = \min\left\{\frac{100-94}{3\sigma}, \frac{106-100}{3\sigma}\right\} = \min\left\{\frac{6}{3\sigma}, \frac{6}{3\sigma}\right\} = \frac{6}{3\sigma} = \frac{2}{\sigma}.$$

We require $\frac{2}{\sigma} \geq 2$, so $\sigma \leq 1$ cm.

Interpretation: to improve from 3-sigma to 6-sigma under the same specs and centered mean, the standard deviation must be reduced from 2 cm to 1 cm. This illustrates the central operational lever of capability improvement: **reduce process variability**.

14 Summary: What to Remember

Quality management in OM is largely the management of variation. Natural variation is unavoidable without changing the process; assignable variation is abnormal and should trigger investigation and action. Capability analysis uses specification limits and process variability to quantify how likely defects are, with the key formula

$$C_{pk} = \min\left\{\frac{\bar{X} - \text{LSL}}{3\sigma}, \frac{\text{USL} - \bar{X}}{3\sigma}\right\}.$$

Conformance analysis uses control charts to monitor stability over time and detect assignable causes. Finally, high sigma levels matter not only for safety-critical products but also because errors can accumulate in complex systems with many components.

Quality Management in Operations: Control Charts and Acceptance Sampling

Concise chapter for junior undergraduate students

1 Why quality management matters in operations

Operations management is about designing and running processes that reliably create value. "Quality" is central because poor quality creates rework, scrap, warranty claims, safety incidents, and damaged reputation. Even when a process is well-designed, its outputs vary. Quality management provides a disciplined way to (i) detect when a process has *changed* and needs corrective action, and (ii) make sensible accept/reject decisions about incoming or outgoing lots without inspecting every single unit.

This chapter focuses on two tools that appear everywhere in manufacturing and service operations: *control charts* for monitoring processes over time (conformance analysis) and *acceptance sampling* for deciding whether to accept an entire lot based on a sample.

2 Variation and the meaning of "in control"

Every process shows variation. Some variation is natural and unavoidable, such as small fluctuations in raw material properties, ambient temperature, or measurement noise. Other variation is due to a specific, *assignable cause*, such as a worn-out tool, a miscalibrated sensor, an untrained operator, or a change in supplier.

Definition (In control vs. out of control)

A process is **in control** if it is operating without assignable cause variation, so its output distribution is stable over time. A process is **out of control** if an assignable cause is present and the output distribution shifts or becomes more variable over time. When a process is out of control, corrective action should be taken.

The practical problem is that we never observe a full "distribution" directly. Instead, we observe *samples* over time and use statistics to decide whether the process is likely stable.

3 Conformance analysis and control charts

Conformance analysis controls output quality by controlling the process that produces the output. It uses statistics and **control charts** to tell *when to adjust* a process. The workflow is conceptually simple: create standards (control limits), measure sample outputs, and take corrective action when the chart signals unusual behavior.

A control chart plots a sample statistic over time (for example, a sample mean). If the process were stable, then the probability that the statistic violates appropriately chosen control limits is very small. Therefore, a violation is evidence that the process may be unstable.

Why we need a series of samples over time

A single sample taken at one time point cannot tell you whether the process is stable *over time*. Conformance analysis is about detecting shifts, drifts, or changes in variability. That requires repeated sampling: sample 1, sample 2, ..., sample m.

Two common charts

Control charts come in different forms depending on the type of quality characteristic:

- **Continuous variables**: measurable on a continuum (weight, length, thickness, temperature). A common tool is the $\bar{X}$ **chart**.
- **Attribute (binary) variables**: outcomes like defective/non-defective or yes/no. A common tool is the p**-chart**.

We keep bullet points minimal in this chapter; these two categories are the one place where a short list helps clarity.

4 The $\bar{X}$ chart for continuous characteristics

Suppose each time period you take a sample of size n (for example, $n = 3$ circuit boards each day) and measure a continuous characteristic (for example, thickness). Let m denote the number of samples collected over time (for example, $m = 25$ days). It is crucial to distinguish:

$$\text{sample size } n \neq \text{number of samples } m.$$

Step 1: Compute within-sample statistics

For sample $j \in \{1, \ldots, m\}$ with observations $x_{j1}, x_{j2}, \ldots, x_{jn}$, compute:

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ji}, \qquad R_j = \max_i x_{ji} - \min_i x_{ji}.$$

Here $\bar{x}_j$ summarizes the sample's center, and R_j summarizes its spread.

Step 2: Compute averages across samples

Compute the "grand mean" (mean of sample means) and the average range:

$$\bar{\bar{x}} = \frac{1}{m}\sum_{j=1}^{m} \bar{x}_j, \qquad \bar{R} = \frac{1}{m}\sum_{j=1}^{m} R_j.$$

Step 3: Compute control limits

The $\bar{X}$ chart uses control limits:

$$\text{LCL}_{\bar{X}} = \bar{\bar{x}} - A_2\bar{R}, \qquad \text{UCL}_{\bar{X}} = \bar{\bar{x}} + A_2\bar{R},$$

where A_2 is a constant that depends on the sample size n (and is typically given in a table). For example:

n	2	3	4	5	6
A_2	1.880	1.023	0.729	0.577	0.483

A key intuition from operations is that larger samples produce more stable averages. As n increases, A_2 decreases, making the control limits *narrower*. This is economically useful: when the mean is naturally less variable, you can impose stricter limits to detect smaller shifts.

How to read a $\bar{X}$ chart: common anomaly signals

If a process is in control, $\bar{x}_j$ should typically stay within control limits and fluctuate around $\bar{\bar{x}}$. In this course-level approach, three common "signals" of assignable cause variation are:

1. A point falls outside $\text{LCL}_{\bar{X}}$ or $\text{UCL}_{\bar{X}}$.
2. Seven consecutive points fall on the same side of the center line $\bar{\bar{x}}$.
3. Seven points in a row move upward or downward (a trend/drift).

These signals are designed to flag patterns that are unlikely under stable random variation.

Worked example 1: computing $\bar{X}$ chart limits (circuit board thickness)

A process produces circuit boards. Each day you sample $n = 3$ boards and measure thickness. After collecting $m = 25$ daily samples, you compute:

$$\bar{\bar{x}} = 0.0630, \qquad \bar{R} = 0.0009 \approx 0.0010, \qquad n = 3 \Rightarrow A_2 = 1.023.$$

Then:

$$\text{UCL}_{\bar{X}} = 0.0630 + 1.023(0.0010) \approx 0.0640,$$
$$\text{LCL}_{\bar{X}} = 0.0630 - 1.023(0.0010) \approx 0.0620.$$

Interpretation is straightforward: plot each day's $\bar{x}_j$ against time, add horizontal lines at $\text{LCL}_{\bar{X}}$, $\bar{\bar{x}}$, and $\text{UCL}_{\bar{X}}$. Any point outside limits or suspicious run/trend suggests the process may be out of control and should be investigated (e.g., tool wear, calibration issues, or shifts in raw material).

5 The p-chart for attribute (binary) outcomes

Sometimes the quality characteristic is not continuous. A board is either defective or not; a transaction is either correct or incorrect; a call is either answered within 30 seconds or not. In these cases, we track a *fraction defective*.

Suppose each time period you inspect n items and count d_j defectives in sample j. Define the sample fraction defective:

$$p_j = \frac{d_j}{n}.$$

Then compute the average fraction defective across m samples:

$$\bar{p} = \frac{1}{m}\sum_{j=1}^{m} p_j.$$

Control limits for the p-chart

A commonly used approximation treats the sampling variation in p_j using a standard deviation:

$$\sigma_p = \sqrt{\frac{\bar{p}(1-\bar{p})}{n}}.$$

Then the control limits are:

$$\mathrm{UCL}_p = \bar{p} + 3\sigma_p = \bar{p} + 3\sqrt{\frac{\bar{p}(1-\bar{p})}{n}},$$

$$\mathrm{LCL}_p = \max\{\bar{p} - 3\sigma_p,\ 0\} = \max\left\{\bar{p} - 3\sqrt{\frac{\bar{p}(1-\bar{p})}{n}},\ 0\right\}.$$

The $\max\{\cdot, 0\}$ appears because a negative fraction defective is meaningless.

Just as with the $\bar{X}$ chart, increasing n reduces sampling variation (the denominator n becomes larger), which tightens the limits and makes the chart more sensitive.

Worked example 2: computing p-chart limits (defective boards)

A process samples $n = 50$ boards each period and records the number of defective boards. After $m = 25$ samples, suppose:

$$\bar{p} = 0.1072.$$

Compute:

$$\sigma_p = \sqrt{\frac{0.1072(1-0.1072)}{50}} \approx 0.0438.$$

Then:

$$\mathrm{UCL}_p = 0.1072 + 3(0.0438) = 0.2386 \approx 0.2384 \text{ (rounding differences are normal)},$$

$$\mathrm{LCL}_p = \max\{0.1072 - 3(0.0438), 0\} = \max\{-0.0242, 0\} = 0.$$

You would plot p_j over time with these horizontal reference lines. Even if no point exceeds UCL_p, a persistent upward trend can still be an anomaly signal, suggesting deterioration in the process.

6 Specification limits vs. control limits: a common confusion

Operations courses often teach two types of "limits" that sound similar but serve different purposes.

Specification limits (LSL/USL)

Lower and upper specification limits, LSL and USL, are engineering or regulatory requirements. They define what counts as a defect: if a unit's measured value lies outside $[\mathrm{LSL}, \mathrm{USL}]$, it is defective. Specification limits are *exogenous*: they do not depend on the sample you collected today.

Control limits (LCL/UCL)

Control limits, such as $\mathrm{LCL}_{\bar{X}}$ and $\mathrm{UCL}_{\bar{X}}$ or LCL_p and UCL_p, are computed from process data. Their purpose is to detect instability (out-of-control behavior), not to decide whether an individual unit meets specs.

Pitfall

A process can be *in control but not capable*: it is stable, yet its stable distribution frequently violates specifications. Conversely, a process can temporarily produce many in-spec units but still be *out of control* if it is drifting or showing assignable-cause variation.

7 From control to improvement

Quality management is not only about detection; it is also about learning and improvement. Control charts help you identify and eliminate anomalies so the process returns to in-control operation. After that, you typically try to reduce variation further. Reduced variation usually lowers the defect rate (because fewer outputs fall outside specification limits) and improves customer experience.

A practical way to phrase this is:

$$\text{Quality} = \text{Process control} + \text{Improvement}.$$

Control brings stability; improvement makes the stable process better.

8 Acceptance sampling: deciding whether to accept a lot

Control charts view the world from the perspective of the process owner monitoring a running process. Acceptance sampling changes the role: you are often a *customer* (or downstream operation) receiving a *lot* (a batch) from a supplier and deciding whether to accept it.

The basic idea

Suppose a lot contains N items (often N is large, e.g., 10,000 chips). The lot has an unknown true population defective rate P. Testing every item (100% inspection) may be too costly or even destructive (e.g., testing some medical products). Testing nothing is risky. Acceptance sampling sits in between: you test a sample of size n and decide accept/reject for the entire lot.

Decision variables and rule

An **acceptance sampling plan** is defined by (n, c):

- n is the sample size.
- c is the acceptance number, the maximum number of defectives allowed in the sample for the lot to be accepted.

Inspect n randomly chosen items from the lot. Let d be the number of defectives found. The decision rule is:

$$\text{Accept the lot if } d \le c; \qquad \text{Reject the lot if } d > c.$$

If you increase c, you accept more often. That reduces the chance of rejecting a good lot, but increases the chance of accepting a bad lot. This is the fundamental trade-off in acceptance sampling.

9 Producer's risk and consumer's risk

A sampling plan cannot be perfect because a sample may not represent the entire lot. Therefore, two kinds of mistakes are possible:

1. **Producer's risk (Type I error)**: rejecting a good lot. Its probability is denoted α.
2. **Consumer's risk (Type II error)**: accepting a bad lot. Its probability is denoted β.

In many applications, values like $\alpha = 0.05$ and $\beta = 0.10$ are used as conventional benchmarks, but the right choice depends on consequences. For high-stakes products (airplane engines, vaccines, medicines), accepting a bad lot is extremely costly, so one aims for a very small β.

10 What counts as "good" or "bad"? AQL and LTPD

To define these error probabilities, we must define what we mean by "good" and "bad" lots in terms of the (unknown) true defective rate P.

Definition (AQL)

The **Acceptance Quality Level (AQL)** is a defective rate at or below which the lot is considered acceptable (a "good" lot). Conceptually, if $P \leq \text{AQL}$, the lot is good.

Definition (LTPD)

The **Lot Tolerance Percent Defective (LTPD)** is a defective rate at or above which the lot is considered unacceptable (a "bad" lot). Conceptually, if $P \geq \text{LTPD}$, the lot is bad.

Clarification

The true P is unknown unless you inspect the entire lot. AQL and LTPD are not computed from the sample; they are quality thresholds used to define which lots *should* be accepted or rejected in principle. The sampling plan then aims to keep:

$$\alpha = \Pr(\text{reject} \mid P \leq \text{AQL}), \qquad \beta = \Pr(\text{accept} \mid P \geq \text{LTPD})$$

small enough.

If P lies between AQL and LTPD, accepting or rejecting is not counted as either Type I or Type II error in this simplified framework. This middle region is a "gray area" where either decision may be tolerated, and the sampling plan is not designed to tightly control errors there.

11 Designing a sampling plan using a sampling table

In practice, many courses use precomputed sampling plan tables. For a given (α, β), such a table typically lists rows indexed by c, and provides:

$$\text{Column 2: } \frac{\text{LTPD}}{\text{AQL}}, \qquad \text{Column 3: } n \times \text{AQL}.$$

The table depends on α and β, so you must select the correct table.

Worked example 3: finding (n, c) from the table

A bulb manufacturer uses:

$$\text{AQL} = 0.01, \qquad \alpha = 0.05, \qquad \text{LTPD} = 0.06, \qquad \beta = 0.10.$$

Step 1: Compute the ratio

$$\frac{\text{LTPD}}{\text{AQL}} = \frac{0.06}{0.01} = 6.$$

Step 2: In the sampling table for $\alpha = 0.05$ and $\beta = 0.10$, find the *smallest* value in the LTPD/AQL column that is ≥ 6. Suppose the table gives 6.509 on the row with $c = 2$. Then set:

$$c = 2.$$

Step 3: In the same row, read $n \times \text{AQL} = 0.818$. Solve:

$$n = \frac{0.818}{0.01} = 81.8 \Rightarrow n = 82 \text{ (round up)}.$$

So the plan is $(n, c) = (82, 2)$. You sample 82 bulbs. If you find $d \leq 2$ defectives, you accept the lot; if $d \geq 3$, you reject. By construction (using the table), this plan keeps producer's risk near 5% at the AQL boundary and consumer's risk near 10% at the LTPD boundary.

12 Common pitfalls and how to avoid them

The most frequent mistakes in applying these tools are conceptual rather than computational.

First, do not confuse sample size n (units per sample) with the number of samples m (time points). In control charts, n affects the width of control limits because larger within-period samples produce more precise statistics.

Second, do not confuse control limits with specification limits. Control limits are about process stability; specification limits are about product acceptability. One monitors the *process*; the other classifies the *output* as conforming or defective.

Third, do not treat "out-of-control" as proof of defects. A control chart signal indicates unusual process behavior and calls for investigation. It does not automatically mean the product is defective relative to specifications; it means the process may have changed.

Fourth, in acceptance sampling, remember that AQL and LTPD define what is "good" and "bad" in terms of the unknown P. You cannot look at the sample fraction defective p and declare the true P with certainty. Sampling plans manage risk, not certainty.

Finally, be cautious when rounding. In sampling plans, rounding up n is typical because using fewer samples than designed can increase risk. In control charts, rounding intermediate values (like $\bar{R}$) too aggressively can noticeably change limits; keep reasonable precision.

13 Summary

Control charts and acceptance sampling address two complementary quality questions in operations. Control charts ask whether a process remains stable over time and help trigger corrective action when anomalies appear. The $\bar{X}$ chart monitors continuous characteristics via sample means and ranges, while the p-chart monitors defect proportions for binary outcomes. Acceptance sampling asks whether to accept an entire lot using a sample, balancing producer's risk α and consumer's risk β with quality thresholds AQL and LTPD. Together, these tools support a practical quality cycle: stabilize the process, eliminate assignable causes, and then reduce variation to improve capability.

Operations Management Chapter: Capacity Planning and Decision Trees

For junior undergraduate students

Introduction: Why "resource allocation" is an operations problem

Operations Management (OM) studies how organizations turn *inputs* (labor, machines, materials, information, and cash) into *outputs* (goods or services) in a reliable, efficient way. Many OM decisions can be summarized by one question: *How should we allocate scarce resources over time, while facing uncertainty?*

This chapter focuses on a particularly important resource allocation topic: **capacity planning** and a simple but powerful method for making capacity-related decisions under uncertainty: **decision trees**. The math is intentionally light: the key idea is to compare alternatives using *expected value*, while recognizing what that leaves out (especially risk).

1 A brief OM connection: Just-in-Time (JIT) manufacturing

Before turning to decision trees, it helps to see the bigger OM picture. One widely used philosophy in manufacturing and services is **Just-in-Time (JIT) manufacturing**, also called the **Toyota Production System (TPS)**.

Definition (JIT/TPS). JIT is a set of practices designed to match supply with demand smoothly and efficiently, aiming for production that is **cheaper, faster, and better**. It is commonly described as standing on two pillars: **production flow management** (getting work to move through the system without delays and excess inventory) and **quality improvement** (preventing defects and learning continuously).

The relevance to capacity is intuitive: when flow is smooth and quality is high, the same resources can produce more usable output, and planning becomes easier. When flow is disrupted or quality is poor, capacity is effectively reduced.

2 Capacity planning: what it is and why it is hard

What is capacity?

Definition (Capacity). Capacity is *the amount of output that a system is capable of achieving over a specific period of time*, assuming enough inputs and enough demand.

The "period of time" matters. A restaurant might describe capacity as meals per hour during lunch, while a factory might describe it as units per week. Capacity decisions are central because they shape many other choices. If you plan to open a restaurant on campus, deciding *how many customers it should be able to serve* drives how many tables you buy, how many employees you hire, how much kitchen equipment you need, and how much capital you must invest.

Three levels of capacity planning

Capacity planning decisions come at three time horizons.

Strategic capacity planning is long-term and hard to reverse. It includes major facility and equipment investments such as building a new plant or purchasing very expensive machinery.

Tactical capacity planning is mid-term. It includes decisions like hiring, layoffs, adding a shift, purchasing tools, or making minor equipment upgrades. These are adjustable, but not instantly.

Operational capacity planning is short-term. It includes detailed scheduling, job assignment, and day-to-day staffing adjustments (for example, staffing more workers on weekends than weekdays).

A useful rule of thumb is that longer-term capacity decisions are often more difficult because they are both more consequential and more uncertain.

Economies and diseconomies of scale

A classic capacity-planning tradeoff is captured by **average unit cost** versus **volume** (the scale of production).

Definition (Average unit cost). If total cost for producing Q units is $C(Q)$, then average unit cost is

$$AC(Q) = \frac{C(Q)}{Q}.$$

Economies of scale occur when average unit cost decreases as volume increases. This can happen because fixed costs are spread over more units, suppliers give quantity discounts, or processes become more efficient at larger scale.

Diseconomies of scale occur when average unit cost increases as volume increases beyond some point. This can happen due to congestion, coordination problems, management complexity, or quality and rework issues in overly large or tightly stretched systems.

Many systems exhibit a U-shaped pattern for $AC(Q)$, decreasing first (economies) and increasing later (diseconomies).

Definition (Best operating level). The best operating level is the capacity/volume level associated with the *lowest* average unit cost:

$$Q^\star \in \arg\min_Q AC(Q).$$

A common misconception is that a firm should *always* operate at $Q^\star$. In practice, firms may produce above this level to gain market share or respond to high demand, or below it to reduce risk, avoid overwork, maintain quality, or preserve flexibility. Cost is important, but it is not the only objective.

Uncertainty: the central challenge

Capacity planning is difficult largely because of uncertainty. Typical sources include uncertainty in supply (availability, lead times, prices), demand (how many customers will buy and when), production (breakdowns, quality variation, disruptions), and extreme events (pandemics, natural disasters, wars).

This uncertainty is especially damaging in strategic planning because long-term investments are slow and costly to reverse. A plant built for high demand becomes expensive excess capacity if demand falls; too little capacity can produce long backlogs and lost sales if demand rises.

3 Decision making under uncertainty and the decision tree method

A general decision process

When people or firms make decisions under uncertainty, a reasonable structure is: (1) specify the objective, (2) list alternatives, (3) analyze and compare alternatives, (4) choose, then (5) implement and monitor.

Decision trees provide a structured way to perform the analysis step when uncertainty can be described in a specific way.

When can decision trees be used?

Decision trees require more information than many real-world situations provide.

Key requirements. Before building and solving a decision tree, you must know:

- The possible future scenarios (often called *states of nature*) and their probabilities.
- The actions available at each decision point.
- The payoff (profit or cost) resulting from each action under each scenario.

In other words, uncertainty must be *imposed in a certain way*: you do not merely know that the future is uncertain, you must be able to list outcomes, attach probabilities, and quantify payoffs. This is why decision trees are often more plausible in mature industries with historical data and stable patterns than in brand new markets where the set of scenarios and probabilities is unclear.

Core idea: maximize expected value

In this chapter's framework, the decision criterion is to choose the alternative with the highest **expected value** (EV).

Definition (Expected value). If an action yields payoff X_i under scenario i, and scenario i occurs with probability p_i, then the expected payoff is

$$EV = \sum_i p_i X_i, \qquad \text{where } \sum_i p_i = 1.$$

The expected value is the long-run average payoff if the same decision were repeated many times under the same probability model.

Decision trees: the picture

A decision tree is a visual representation of a multi-stage decision problem.

A **decision point** (often drawn as a square) is where the decision-maker chooses an action.

An **event point** (often drawn as a circle) is where a random outcome occurs, branching into scenarios labeled with probabilities.

The tree is solved by computing expected values and working *backward* when decisions occur in multiple periods.

4 Worked Example 1: Glass factory capacity choice

A glass factory faces a backlog and considers three actions: subcontracting (A), constructing a new facility (B), or doing nothing (C). Demand next period may be low, medium, or high with probabilities 0.1, 0.5, and 0.4. Profits (in thousands of dollars) are given below.

	Low (0.1)	Medium (0.5)	High (0.4)
Subcontracting (A)	10	50	90
New facility (B)	-120	25	200
Do nothing (C)	20	40	60

Step 1: Compute expected value for each action

For subcontracting (A),

$$EV_A = 0.1(10) + 0.5(50) + 0.4(90) = 1 + 25 + 36 = 62.$$

For a new facility (B),

$$EV_B = 0.1(-120) + 0.5(25) + 0.4(200) = -12 + 12.5 + 80 = 80.5.$$

For doing nothing (C),

$$EV_C = 0.1(20) + 0.5(40) + 0.4(60) = 2 + 20 + 24 = 46.$$

Step 2: Choose the largest expected value

Since $EV_B = 80.5$ is the highest, the expected-value criterion selects **construct a new facility**.

A crucial interpretation: EV is not the same as "safe"

Even though building a new facility has the largest expected profit, it has a 10% chance of a large loss (-120). This highlights a limitation: maximizing expected value does not explicitly penalize risk. In the course framework, EV is the decision rule, but in practice managers often consider risk tolerance, cash constraints, and the potential for catastrophic outcomes.

5 The value of information: EVPI

Why information can be valuable

In many problems, you must choose an action *before* you observe the true demand scenario. This can be frustrating because, in hindsight, you would like to choose the best action for each scenario.

This motivates the **Expected Value of Perfect Information (EVPI)**.

Definition (Perfect information). Perfect information means you learn which scenario will occur *before* you choose an action. The scenario is still random, but it is revealed early enough to adapt your decision.

Definition (EVPI). EVPI is the improvement in expected payoff from having perfect information:

$$EVPI = EV(\text{with perfect information}) - EV(\text{without perfect information}).$$

It can be interpreted as the *maximum* amount you should be willing to pay for that information.

EVPI for the glass factory

Without perfect information, we choose action B and obtain $EV(\text{without PI}) = 80.5$.

With perfect information, we choose the best action in each demand scenario: low demand: choose C (20), medium demand: choose A (50), high demand: choose B (200). So,

$$EV(\text{with PI}) = 0.1(20) + 0.5(50) + 0.4(200) = 2 + 25 + 80 = 107.$$

Therefore,

$$EVPI = 107 - 80.5 = 26.5,$$

in thousands of dollars. This means perfect demand information would be worth up to \$26,500 in expected profit.

When is EVPI equal to zero?

EVPI is never negative: having more information cannot hurt if you are free to ignore it. EVPI becomes 0 when the same action is optimal no matter which scenario occurs. In that case, learning the scenario does not change your decision, so it creates no additional value.

6 Multi-period decisions and backward induction

Some decisions are staged. You may decide now, then decide again later after observing early signals about demand. Decision trees handle this naturally by solving from the end of the tree backward.

Backward induction (idea). In a multi-period decision tree, you first solve the last decision point (choose the best action there), replace that future decision with its resulting value, and then move backward to earlier decisions. This works because a rational decision-maker will choose optimally when that future decision point is reached.

7 Worked Example 2: A computer store with an option to wait

A computer store wants to maximize total profit over the next five years. Demand growth will be strong with probability 0.55 and weak with probability 0.45. Management considers:

A. Move to a new location (cost \$210k). Annual revenue is \$195k if demand is strong, \$115k if weak.

B. Expand the current store (cost \$87k). Annual revenue is \$190k if strong, \$100k if weak.

C. Do nothing now, but if demand is strong next year, consider expansion then.

Step 1: Convert costs and annual revenues into five-year profits

We treat all numbers as thousands of dollars to keep notation simple. If you move and demand is strong, five-year revenue is 195×5, and profit is $195 \times 5 - 210 = 765$. If you move and demand is weak, profit is $115 \times 5 - 210 = 365$.

If you expand now, profit is $190 \times 5 - 87 = 863$ under strong demand, and $100 \times 5 - 87 = 413$ under weak demand.

For option C (do nothing now), the weak-demand profit is simply $105 \times 5 = 525$. If demand is strong, you have a second decision after one year: either expand then or keep doing nothing. Expanding after one year yields one year at \$170k plus four years at \$190k, minus the \$87k expansion cost:

$$170 \times 1 + 190 \times 4 - 87 = 843.$$

If you do not expand even after seeing strong demand, profit is $170 \times 5 = 850$. Notice that even with strong demand, *waiting and then not expanding* can be better than *waiting and then expanding.*

Step 2: Solve by backward induction

At the second decision point (after observing strong demand under option C), compare 843 (expand then) versus 850 (do nothing). You choose **do nothing**, so the strong-demand branch under option C becomes 850.

Now compute expected values for the first-stage options:

$$EV(\text{Move}) = 0.55(765) + 0.45(365) = 585,$$

$$EV(\text{Expand now}) = 0.55(863) + 0.45(413) = 660.5,$$

$$EV(\text{Do nothing now}) = 0.55(850) + 0.45(525) = 703.75.$$

Thus the expected-value criterion chooses **do nothing now**, keeping the option to react later (but in this case, the best reaction is still to do nothing even if demand is strong).

EVPI for the computer store

Without perfect information, the best expected profit is 703.75.

With perfect information, you choose the best action knowing demand in advance. If demand will be strong, the best profit is 863 (expand now). If demand will be weak, the best profit is 525 (do nothing). So,

$$EV(\text{with PI}) = 0.55(863) + 0.45(525) = 710.9.$$

Therefore,

$$EVPI = 710.9 - 703.75 = 7.15,$$

again in thousands of dollars. Perfect information is worth only \$7,150 in expected value here because the best action does not change the payoff dramatically relative to the "do nothing" strategy.

8 Common pitfalls and practical cautions

A frequent mistake is mixing up *average unit cost* and *total cost*. Economies of scale refer to the behavior of $AC(Q) = C(Q)/Q$, not necessarily $C(Q)$ itself.

Another common error is using decision trees when the required inputs are not credible. If the scenarios are incomplete, probabilities are guesses without grounding, or payoffs are highly uncertain, then the apparent precision of an expected value calculation can be misleading.

Students also often forget that probabilities across scenarios must sum to 1, or they misread time horizons when converting annual revenues and one-time costs into multi-year profits.

Finally, expected value is not the only decision criterion. EV treats a 10% chance of a very large loss as just another term in an average. In real operations, cash constraints, bankruptcy risk, reputation damage, and risk attitudes can make managers prefer a lower-EV option that is more robust. The decision-tree structure remains useful, but the evaluation rule may need to incorporate risk measures or additional constraints.

Conclusion: what you should remember

Capacity planning determines how much a system can produce over time and influences many downstream OM decisions. Costs often exhibit economies and diseconomies of scale, leading to a best operating level that minimizes average unit cost, but real firms balance cost with strategy and risk.

Decision trees help allocate resources under uncertainty when scenarios, probabilities, actions, and payoffs can be specified. The central calculation is expected value:

$$EV = \sum_i p_i X_i.$$

When decisions occur in multiple periods, solve by backward induction. Finally, the expected value of perfect information,

$$EVPI = EV(\text{with PI}) - EV(\text{without PI}),$$

quantifies how much information about the future is worth, in expectation, and clarifies why forecasting, market research, feasibility studies, and data analytics can create operational value.

Operations Management Chapter: Resource Allocation with Linear Programming

For junior undergraduate students

1 Why Operations Managers Need Linear Programming

Operations management is fundamentally about *making good decisions with limited resources*. A manager may need to decide how much of each product to produce, how to schedule labor, how to route shipments, or how to allocate a budget across projects. These decisions are linked by a common structure: each decision consumes resources (time, materials, capacity, money), and the organization has only so much of each resource available.

When the relationships between decisions, resources, and profit (or cost) can be reasonably approximated as *linear*, the problem can often be solved using *linear programming* (LP). Linear programming is one of the most widely used optimization tools in business and engineering because it is both flexible and computationally efficient: problems with thousands of decision variables and constraints can often be solved quickly.

2 The Core Idea: Optimize Subject to Constraints

A linear program has three ingredients.

First, it has **decision variables**, which represent the quantities the manager can choose (for example, how many units to produce, how many workers to hire, or how much to ship).

Second, it has an **objective function**, which represents what the manager wants to maximize or minimize (profit, cost, time, emissions, and so on).

Third, it has **constraints**, which encode limitations and requirements. The most common constraints in operations are *availability constraints*: you cannot use more of a resource than you have. Another common type is the *non-negativity constraint*: you cannot produce a negative amount.

A convenient way to write an LP is:

$$\begin{aligned} \text{maximize or minimize} \quad & \text{(linear objective in the decision variables)} \\ \text{subject to} \quad & \text{(linear constraints)} \\ & \text{(non-negativity constraints).} \end{aligned}$$

The phrase *subject to* means that any acceptable solution must satisfy every constraint.

3 Key Definitions (with Intuition)

Decision variables

Decision variables are the unknown quantities the manager chooses. In a product-mix problem, a natural decision variable is the production quantity of each product.

Objective function and objective coefficients

The objective function expresses total profit or total cost as a function of the decision variables. When the objective is linear, it takes the form

$$Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

where the numbers c_i are called *objective coefficients*. In a profit-maximization model, c_i is often the profit per unit of product i.

Constraints and right-hand side

A typical availability constraint looks like

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b,$$

where the left-hand side represents resource usage (what you need), and the right-hand side b represents resource availability (what you have). The numbers a_i are often called *resource consumption coefficients*.

Feasible and infeasible solutions

A solution is a specific assignment of values to the decision variables, such as $(x_1, x_2) = (10, 10)$. A solution is **feasible** if it satisfies *all* constraints. If it violates at least one constraint, it is **infeasible**. Feasibility is about whether a plan can be executed in reality, not about whether it is good.

Optimal solution vs. optimal objective value

An **optimal solution** is a feasible solution that produces the best objective value (largest for maximization, smallest for minimization). The **optimal objective value** is the numerical value of the objective function at the optimal solution. Confusing these two is a common exam mistake: the solution is a vector (a set of decision-variable values), while the objective value is one number.

Binding vs. non-binding constraints and slack

At a chosen solution, a constraint of the form $\leq$ can either be "tight" or "loose."

If the solution makes the constraint hold with equality, the constraint is **binding**. If the inequality is strict, it is **non-binding**.

For an availability constraint

$$a_1x_1 + \cdots + a_nx_n \leq b,$$

the **slack** is defined as

$$\text{slack} = b - (a_1x_1 + \cdots + a_nx_n).$$

Slack is always ≥ 0 for feasible solutions. Slack equals 0 for binding constraints and is positive for non-binding constraints.

Economically, slack represents unused capacity. If a resource has positive slack at the optimum, increasing that resource slightly will not improve the objective, because the resource is not currently limiting the best plan. In contrast, a binding constraint identifies a bottleneck resource that may be worth expanding.

4 What Makes a Problem "Linear"?

Linear programming applies when *both* the objective function and all constraints are **linear** in the decision variables.

Linear functions

A function is linear if each variable appears only to the first power, variables are not multiplied together, and coefficients are constants. Examples:

$$13A + 23B \quad \text{and} \quad 0.5A + \frac{2}{3}B$$

are linear.

Nonlinear examples include:

$$13A^2 + 23AB, \qquad \frac{1}{A} + B, \qquad \log(A) + \cos(B), \qquad \max(A, 0).$$

Linear constraints

A linear constraint has a linear function on the left-hand side and a constant on the right-hand side, connected by $\leq$, $\geq$, or $=$. In LP, constraints are written with *weak* inequalities ($\leq$ or $\geq$) or equality, not strict inequalities.

Sometimes a constraint may look nonlinear but can be rearranged into a linear form under extra conditions. For example, $A/B \geq 2$ is not linear as written, but if $B \geq 0$ is guaranteed, it can be rewritten as $A - 2B \geq 0$.

5 How to Formulate a Linear Program

A reliable LP formulation process is short but disciplined.

First, define decision variables carefully, including units (barrels, units per week, number of employees, etc.). Poorly defined variables lead to wrong constraints.

Second, write the objective in terms of the decision variables. Verify that the coefficients represent the correct *per-unit* contribution (profit or cost).

Third, write constraints in terms of the decision variables. In resource allocation, each resource typically generates one availability constraint of the form "usage $\leq$ availability." Also add non-negativity constraints for each decision variable.

Finally, check linearity and unit consistency. If "minutes" appear in one constraint, every term in that constraint must be measured in minutes.

6 Implicit Assumptions Behind LP (and Why They Matter)

Using LP means accepting several approximations.

Continuity

LP allows decision variables to take fractional values. This is appropriate for divisible quantities (kilograms of material, hours of machine time), but not for inherently integer decisions (number of trucks, number of nurses). If integrality is essential, the correct tool is *integer programming*. In practice, LP solutions are sometimes rounded, but rounding can break feasibility.

Proportionality

Producing twice as much output uses exactly twice as much resource. This assumption fails with economies of scale, setup times, learning effects, or quantity discounts.

Additivity

Total profit (or cost) is the sum of the contributions of each unit produced, and the per-unit contribution does not depend on volume. This can fail when prices drop with higher supply, when overtime changes labor cost, or when producing product A affects demand for product B (cannibalization).

These assumptions are not always realistic, but they often provide a good first-order approximation and enable fast and insightful optimization.

7 Worked Example 1: The Bland Brewery Product-Mix Model

A brewery produces two products: ale and beer. Let

$$A = \text{number of barrels of ale}, \qquad B = \text{number of barrels of beer}.$$

Profit per barrel is \$13 for ale and \$23 for beer, so profit is

$$Z = 13A + 23B.$$

Three resources are required: corn, hops, and malt. The per-barrel requirements and total availability are:

	Corn (lbs)	Hops (oz)	Malt (lbs)
1 barrel of ale	5	4	35
1 barrel of beer	15	4	20
Availability	480	160	1190

This yields the LP:

$$\begin{aligned} \max \quad & Z = 13A + 23B \\ \text{s.t.} \quad & 5A + 15B \le 480 \qquad \text{(corn)} \\ & 4A + 4B \le 160 \qquad \text{(hops)} \\ & 35A + 20B \le 1190 \qquad \text{(malt)} \\ & A \ge 0,\ B \ge 0. \end{aligned}$$

Feasibility check (quick practice)

Consider $(A, B) = (10, 10)$. Resource usage is:

corn: $5(10)+15(10) = 200 \le 480$, hops: $4(10)+4(10) = 80 \le 160$, malt: $35(10)+20(10) = 550 \le 1190$.

This plan is feasible.

Now consider $(A, B) = (40, 10)$. Usage includes:

$$\text{hops: } 4(40) + 4(10) = 200 > 160, \quad \text{malt: } 35(40) + 20(10) = 1600 > 1190,$$

so this plan is infeasible.

Optimal solution and slack interpretation

For this example, the optimal solution (given in the session materials) is

$$(A^*, B^*) = (12, 28), \qquad Z^* = 13(12) + 23(28) = 800.$$

At the optimum, compute slack for each resource:

$$\text{corn slack} = 480 - (5 \cdot 12 + 15 \cdot 28) = 480 - 480 = 0,$$

$$\text{hops slack} = 160 - (4 \cdot 12 + 4 \cdot 28) = 160 - 160 = 0,$$

$$\text{malt slack} = 1190 - (35 \cdot 12 + 20 \cdot 28) = 1190 - 980 = 210.$$

Corn and hops constraints are binding; malt is non-binding with 210 lbs left over. This suggests that expanding malt availability alone would not improve profit near the current optimum, while expanding corn or hops might.

8 Worked Example 2: Solving a Two-Variable LP Graphically

Graphical solution is mainly a learning tool because it works only when there are two decision variables. Still, it builds strong intuition for feasibility, corner points, and binding constraints.

A manager chooses weekly production of products A and B. Each unit requires processing time (in minutes):

	Molding	Painting	Cutting
Product A	1	2	0
Product B	1	1	1
Availability	300	400	250

Profit margins are \$3 per unit of A and \$2 per unit of B. The LP is

$$\begin{aligned} \max \quad & Z = 3A + 2B \\ \text{s.t.} \quad & A + B \le 300 \qquad \text{(molding)} \\ & 2A + B \le 400 \qquad \text{(painting)} \\ & B \le 250 \qquad \text{(cutting)} \\ & A \ge 0,\ B \ge 0. \end{aligned}$$

Graphical method intuition

Each inequality describes a half-plane. The feasible region is the intersection of all half-planes in the first quadrant. In LP, the optimum occurs at a *corner point* (also called a vertex) of the feasible region. A practical way to "search" graphically is the *iso-profit line* method.

An iso-profit line is the set of points with the same objective value:

$$3A + 2B = k,$$

for some constant k. Rearranging gives

$$B = -\frac{3}{2}A + \frac{k}{2},$$

so all iso-profit lines have slope $-3/2$. Larger k shifts the line up and to the right, corresponding to higher profit. To maximize profit, push the iso-profit line as far up-right as possible while still touching the feasible region; the last point of contact will be an optimal corner point.

From the session materials, the optimal solution is

$$(A^*, B^*) = (100, 200), \qquad Z^* = 3(100) + 2(200) = 700.$$

Binding constraints and idle time

A constraint is binding if the optimal point lies on its boundary (the equality line). Check:

$$A + B = 100 + 200 = 300 \quad \Rightarrow \quad \text{molding is binding, slack } = 0,$$

$$2A + B = 2(100) + 200 = 400 \quad \Rightarrow \quad \text{painting is binding, slack } = 0,$$

$$B = 200 \leq 250 \quad \Rightarrow \quad \text{cutting is non-binding, slack } = 250 - 200 = 50.$$

So cutting has 50 minutes of idle time; molding and painting have none.

9 Common Pitfalls (and How to Avoid Them)

Students (and practitioners) often struggle not with solving LPs, but with writing correct models. The following mistakes are especially common.

First, many formulations fail because decision variables are not defined precisely. Always state what each variable represents and include a time frame and unit (for example, "units per week").

Second, unit mismatches quietly ruin constraints. If a constraint represents minutes of painting time, every term must be in minutes. Mixing hours and minutes or pounds and ounces without conversion causes incorrect results.

Third, it is easy to confuse "optimal solution" with "optimal objective value." Remember that $(A, B) = (12, 28)$ is a solution, while $Z = 800$ is the corresponding objective value.

Fourth, do not forget non-negativity constraints. Without $A \geq 0$ and $B \geq 0$, the model can produce mathematically optimal but physically meaningless negative production.

Fifth, avoid nonlinear expressions unless you are sure they can be converted to linear form. Terms like AB, A^2, $\max(\cdot)$, and $\log(\cdot)$ are not allowed in a standard LP.

Finally, be cautious with integer requirements. If the decision is indivisible (hiring people, buying machines), an LP solution that contains fractions is not directly implementable. Rounding can violate constraints; if integrality matters, use an integer programming model.

10 Where LP Appears in Real Operations

Linear programming appears across operations management, often under different names.

In *production planning*, LP helps determine how much to produce over time given capacity and inventory limits. In *transportation and logistics*, LP determines shipping quantities from multiple sources to multiple destinations to minimize total cost. In *blending* problems, LP chooses a recipe that meets quality requirements at minimum cost. In *workforce planning*, LP can allocate labor hours to meet staffing constraints while minimizing payroll.

As a concrete everyday example beyond the classroom, consider a coffee shop deciding how many barista hours to schedule in the morning versus afternoon. Each hour of labor costs money, but too few hours create long waiting times (a service-level constraint). If service requirements can be approximated by linear constraints (for example, "at least x labor-hours per expected customer volume"), staffing can be formulated and solved as an LP.

11 Summary

Linear programming is a structured way to make the best resource allocation decision when profit (or cost) and resource usage can be modeled linearly. You define decision variables, write a linear objective, add linear constraints for resources and requirements, and enforce non-negativity. The concepts of feasibility, optimality, binding constraints, and slack provide both mathematical clarity and managerial insight: they reveal bottlenecks and identify where additional resources are most valuable.

Operations Management Chapter: Linear Programming, Shadow Prices, and Sensitivity Analysis

ISOM 2700 (Session 12-inspired), concise textbook-style notes

1 Why Operations Managers Use Optimization

Operations management is fundamentally about making good decisions with limited resources. A manager might ask: How many units of each product should we produce this month? Which orders should we prioritize? Should we buy extra material, add overtime, or invest in more capacity?

These questions are difficult because resources are constrained. You may have limited labor hours, machine time, raw materials, or storage space. When resources are scarce, *trade-offs* appear: producing more of one item often forces you to produce less of another. Linear programming (LP) is one of the simplest and most widely used tools to handle such trade-offs in a disciplined way.

LP is especially common in *capacity planning* and *resource allocation*, where the goal is to choose decision variables (such as production quantities) to maximize profit (or minimize cost) while obeying resource limits. The power of LP is not only that it finds an optimal plan, but also that it explains *why* the plan is optimal and how sensitive it is to changes in the business environment.

2 Linear Programming: A Modeling Language for Constrained Decisions

Key definitions

A **linear programming problem** chooses values of **decision variables** to optimize a **linear objective function** subject to **linear constraints**.

Decision variables represent choices under the manager's control. In a product-mix problem, typical decision variables are production quantities:

$$x_1 = \text{units of product 1}, \qquad x_2 = \text{units of product 2}, \quad \ldots$$

The **objective function** measures what you want to optimize, often profit or revenue:

$$\max \; z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

where c_j is the profit (or revenue contribution) per unit of decision variable x_j.

A **constraint** represents a limitation, often a resource availability constraint:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i.$$

Here, a_{ij} is the amount of resource i consumed per unit of activity j, and b_i is the available amount of resource i.

Finally, most production models include **non-negativity constraints**:

$$x_j \geq 0 \quad \text{for all } j,$$

because negative production is not meaningful.

Core intuition: "Use scarce resources where they create the most value"

LP formalizes a simple idea: if resources are limited, allocate them to the combination of activities that creates the most objective value without violating constraints. If a resource is plentiful, it should not affect the plan; if a resource is scarce and fully used, it shapes the optimal plan.

The rest of this chapter makes those ideas precise using three concepts that appear in Solver output and in practice: *binding vs. non-binding constraints, slack, and shadow prices*, and then *sensitivity analysis*.

3 Binding, Non-binding, and Slack: Reading Constraints Like a Manager

Consider a resource availability constraint written as

$$(\text{LHS}) \leq (\text{RHS}).$$

The **left-hand side (LHS)** is how much of the resource is used under your chosen plan. The **right-hand side (RHS)** is how much of the resource is available.

A constraint is **binding** if it is exactly tight at the solution:

$$\text{LHS} = \text{RHS}.$$

Binding means the resource is fully used. In operations terms, this is a bottleneck candidate: you are right up against capacity.

A constraint is **non-binding** if it is not tight:

$$\text{LHS} < \text{RHS}.$$

Non-binding means there is leftover resource.

The leftover amount is called **slack**:

$$\text{Slack} = \text{RHS} - \text{LHS}.$$

Slack is always ≥ 0 for a feasible solution. Slack tells you "how much room you still have" in that constraint. Slack is a physical quantity (grams, hours, units of capacity), so it is usually easy to interpret.

4 Shadow Price: The Marginal Value of a Resource

Definition and meaning

A **shadow price** (also called a dual value) is defined for each *resource constraint*, not for each product. It answers a practical question:

> If we increase the availability of resource i by one unit, by how much would the optimal objective value improve (approximately, and exactly within a valid range)?

Formally, let the original optimal objective value be $z^\star$. Increase the RHS of resource i by 1 (for example from b_i to $b_i + 1$), re-solve the LP, and call the new optimal value $z^\star_{(+1 \text{ in } i)}$. The shadow price of resource i is

$$\pi_i = z^\star_{(+1 \text{ in } i)} - z^\star.$$

In product-mix settings where the objective is revenue or profit and constraints represent "$\leq$" limitations, shadow prices are typically $\pi_i \geq 0$. Intuitively, having more of a resource cannot make you worse off, because your old plan remains feasible.

Shadow price can be zero. A zero shadow price means that increasing this resource slightly does not improve your best possible objective value, usually because the resource is not currently limiting.

Shadow price and slack: complementary slackness

Slack and shadow price are tightly linked.

If a constraint has Slack > 0, then you are not using all of that resource. Giving you one more unit does not change what you can do optimally, because you already have leftover. Therefore, the shadow price must be 0.

If a constraint has $\pi_i > 0$, then one more unit *does* improve the objective. This is only possible when the resource is fully used, so slack must be 0.

This logic is captured by **complementary slackness**:

$$\pi_i \times \text{Slack}_i = 0 \quad \text{for each resource } i.$$

At the optimum, you cannot have both positive slack and a positive shadow price for the same resource.

A subtle (but important) pitfall is that Slack $= 0$ does not *guarantee* $\pi_i > 0$ in every possible LP; degeneracy can create "rare" cases where slack is zero and shadow price is also zero. In typical introductory product-mix problems, however, a binding resource is usually valuable, so π_i is often positive.

5 Sensitivity Analysis: How Robust Is the Optimal Plan?

In operations, parameters change. Material deliveries are delayed. Demand shifts. Selling prices move. A manager needs to know when the current plan is robust and when it must be re-optimized.

A **sensitivity report** (as produced by Excel Solver for LP) helps answer two broad questions:

1. How does the optimal solution and objective value respond to changes in *product prices* (objective coefficients)?
2. How does the optimal objective value respond to changes in *resource availability* (constraint RHS values)?

1) Changes in objective coefficients (prices/profits)

Suppose the objective is

$$\max\ z = \sum_{j=1}^{n} c_j x_j.$$

A sensitivity report gives an *allowable increase* and *allowable decrease* for each coefficient c_j. The key conclusion is:

> If a coefficient c_j changes within its allowable range, then the *optimal solution* (the optimal x-values) stays the same.

However, even if the optimal solution stays the same, the *optimal objective value changes* because you are evaluating the same production plan under new unit profits. If only c_j changes by Δc_j and the optimal production quantity stays $x_j^\star$, then the objective changes by

$$\Delta z = (\Delta c_j)\, x_j^\star.$$

This is a managerial statement about robustness: "Small price changes do not necessarily require re-planning, but they do change how much money we make."

2) Changes in RHS (resource availability)

For each constraint RHS b_i, the sensitivity report provides an allowable increase and decrease. The central rule is:

> If a RHS changes within its allowable range, then the shadow price for that constraint remains valid, and the change in the optimal objective value is
>
> $$\Delta z = \pi_i\, \Delta b_i.$$

This formula is extremely practical because it converts capacity changes into dollar values. It is the foundation for decisions like whether to pay for overtime, buy additional material, or accept a contract to expand capacity.

One must always check whether Δb_i lies within the allowable increase/decrease. If the change is outside the allowable range, the shadow price can change, the structure of the optimum can change, and you generally need to re-solve the LP.

6 Worked Example 1: ST Powder Product Mix (Revenue Maximization)

Problem description and LP model

ST sells two products: cologne and perfume.

Let

$$C = \text{ounces of cologne produced}, \qquad P = \text{ounces of perfume produced}.$$

Revenue per ounce is \$3 for cologne and \$8 for perfume. Resource requirements are:

- Cologne uses 2 grams of fragrance and 6 grams of intensifier per ounce.
- Perfume uses 4 grams of fragrance, 2 grams of intensifier, and 1 gram of stabilizer per ounce.

Resource availabilities are 1600 grams fragrance, 1800 grams intensifier, and 350 grams stabilizer.

The LP is:

$$\begin{aligned} \max \quad & 3C + 8P \\ \text{s.t.} \quad & 2C + 4P \le 1600 \qquad \text{(fragrance)} \\ & 6C + 2P \le 1800 \qquad \text{(intensifier)} \\ & P \le 350 \qquad \text{(stabilizer)} \\ & C \ge 0,\ P \ge 0. \end{aligned}$$

From Solver output in the session, the optimal solution is

$$(C^\star, P^\star) = (100, 350).$$

The optimal revenue is

$$z^\star = 3(100) + 8(350) = 300 + 2800 = 3100.$$

Interpreting slack and shadow prices

At $(C^\star, P^\star) = (100, 350)$, resource usage is:

$$\text{Fragrance used} = 2(100) + 4(350) = 200 + 1400 = 1600,$$

$$\text{Intensifier used} = 6(100) + 2(350) = 600 + 700 = 1300,$$

$$\text{Stabilizer used} = 350.$$

So slack values are:

$$\text{Slack}_{\text{fragrance}} = 1600 - 1600 = 0, \quad \text{Slack}_{\text{intensifier}} = 1800 - 1300 = 500, \quad \text{Slack}_{\text{stabilizer}} = 350 - 350 = 0.$$

The sensitivity report indicates shadow prices:

$$\pi_{\text{intensifier}} = 0, \quad \pi_{\text{stabilizer}} = 2, \quad \pi_{\text{fragrance}} = 1.5.$$

Complementary slackness is visible immediately: intensifier has positive slack and zero shadow price; fragrance and stabilizer are binding (zero slack) and have positive shadow prices.

A short sensitivity question: what is 100 grams of extra fragrance worth?

Suppose a supplier offers 100 more grams of fragrance. Is it valuable?

We check that 100 is within the allowable increase for fragrance (reported as about 166.67). Then the shadow price is valid, and the revenue increase is

$$\Delta z = \pi_{\text{fragrance}} \cdot \Delta b = 1.5 \times 100 = 150.$$

So ST should be willing to pay *at most \$150* for 100 extra grams of fragrance; paying more would reduce profit.

A common pitfall illustrated: 60 grams extra stabilizer

If stabilizer increases by 60 grams but the allowable increase is only 50, then $\pi_{\text{stabilizer}} = 2$ is *not guaranteed* to apply. It might still be close, but the correct operations answer is: you cannot determine the exact impact from the current shadow price; you should re-solve the LP with the new RHS.

7 Worked Example 2: Shelby Shelving (Profit Maximization and Resource Value)

LP model and solution

A firm produces two shelf models, S and LX. Profits are \$260 per unit of S and \$245 per unit of LX. Capacity constraints:

$$S \le 1900, \qquad LX \le 1400,$$

$$0.3S + 0.3LX \le 800 \quad \text{(stamping hours)},$$

$$0.25S + 0.5LX \le 800 \quad \text{(forming hours)},$$

$$S \ge 0,\ LX \ge 0.$$

The session reports the optimal solution:

$$S^\star = 1900, \qquad LX^\star = 650, \qquad z^\star = 653{,}250.$$

Shadow price as "how much to pay for capacity"

The sensitivity report gives the shadow price for the S-assembly constraint as $\pi_{S\text{-assembly}} = 137.5$ (dollars per unit of assembly capacity), valid within a certain range.

If the S-assembly capacity increases from 1900 to 1902 (an increase of 2, within the allowable increase), then the profit improvement is

$$\Delta z = 137.5 \times 2 = 275.$$

This number has a direct managerial meaning: if you can buy two extra units of S-assembly capacity (for example, through overtime or outsourcing) for less than \$275 total, it is worth doing; if it costs more, it is not.

Pricing a proposed new product using shadow prices

Suppose a new product GX yields revenue \$1000, variable cost \$850, so profit contribution is

$$\text{Profit}(GX) = 1000 - 850 = 150.$$

However, it consumes 2 hours of stamping and 0.5 hours of forming.

If stamping has shadow price 0, its opportunity cost is $2 \times 0 = 0$. If forming has shadow price 490 dollars per hour, then 0.5 hours costs

$$0.5 \times 490 = 245$$

in lost optimal profit from the current plan.

The net impact of producing one unit of GX is approximately

$$+150 - 245 = -95,$$

so it should *not* be produced. The deep intuition is that "profit contribution" is not enough; what matters is *profit after paying for scarce resources at their shadow prices.* Shadow prices convert physical resource consumption into economic opportunity costs.

8 Common Pitfalls and How to Avoid Them

Students and practitioners often make predictable mistakes when first using LP and sensitivity reports. Most can be avoided by slowing down and asking what each number *means*.

First, do not confuse decision variables with resources. Decision variables represent what you choose (production quantities); shadow prices belong to constraints (resources), not products.

Second, always check feasibility and units. The LHS and RHS of a constraint must be in the same units (grams with grams, hours with hours). Mixing ounces, grams, and hours without consistent conversion is one of the fastest ways to produce nonsense models.

Third, do not apply a shadow price outside its allowable range. Shadow prices are local marginal values: they are reliable for RHS changes within the report's allowable increase/decrease. Outside that region, the bottleneck structure can change.

Fourth, remember the difference between stability of the *solution* and changes in the *objective*. When an objective coefficient changes within its allowable range, the optimal quantities stay fixed, but profit/revenue changes because the same quantities are now valued differently.

Finally, do not overinterpret slack. A positive slack means "unused resource," but it does not necessarily mean "the resource is unimportant" forever. It may become important under different prices, different product mix options, or different constraints. Slack is a statement about the current optimal plan, not about all possible plans.

9 A Practical Note on Solving LPs with Excel Solver

Graphical solution methods are useful for intuition when there are at most two decision variables, but real operations problems often have many variables and constraints. Excel Solver is a convenient tool for small to medium models. For larger-scale models, dedicated solvers through Python, MATLAB, or specialized optimization software are typically used, but the modeling logic remains the same.

In Excel Solver for LP, the standard approach is: write formulas for the objective and each constraint LHS in terms of decision-variable cells, specify the objective cell to maximize (or minimize), identify the changing-variable cells, add constraints including non-negativity, and select `Simplex LP`. The `Answer` and `Sensitivity` reports provide the managerial insights discussed above.

10 Chapter Summary

Linear programming is a powerful yet accessible framework for operations management decisions under constraints. Binding constraints and slack reveal which resources limit the plan. Shadow prices quantify the marginal economic value of relaxing a resource constraint. Sensitivity analysis explains when an optimal plan is robust to changes in prices or capacities and provides simple, actionable formulas such as

$$\Delta z = \pi_i \, \Delta b_i$$

(when within allowable ranges). Together, these ideas turn a static optimization result into a flexible decision-support tool for real operations settings, where change is the norm rather than the exception.

Demand Forecasting in Operations Management

A Concise, Book-Style Chapter for Junior Undergraduates

1 Why Demand Forecasting Matters in Operations Management

Operations management (OM) is about designing and running processes that deliver goods and services efficiently and reliably. Almost every operational decision needs an assumption about how much customers will want in the future. *Demand forecasting* is the practice of estimating future demand using available information, so that managers can plan capacity, inventory, staffing, and other resources.

Forecasting is an indispensable input to many OM models. When a firm uses linear programming to plan production, queueing models to predict waiting times, or decision trees to evaluate investments, it still needs an estimate of demand. A hospital needs a forecast of patient arrivals and the likely severity mix to schedule staff and beds. Airlines and hotels forecast passenger volumes and length of stay to set prices and allocate rooms or seats. Even a university implicitly forecasts applicant numbers and enrollment yield to set admission quotas and plan course sections.

A key message from practice is that forecasting is not an isolated analytics task. It is a bridge between business reality (customer behavior, marketing plans, competitors, macroeconomy) and operational commitments (machines, workers, contracts, inventory). Good forecasts therefore balance quantitative analysis with business judgment.

2 Core Definitions: Demand Types and Forecasting Types

Independent vs. dependent demand

In OM, it is useful to distinguish two kinds of demand.

Independent demand is demand determined by final customers. It cannot be mechanically derived from the demand of another item. For example, the demand for a finished chair in a store is independent: customers decide whether to buy it.

Dependent demand is demand that is derived from the demand for another product or service. If you plan to produce 1,000 chairs, you can derive the demand for components such as legs, seats, and screws from the bill of materials. Dependent demand is often handled by planning systems (such as material requirements planning), but it is the independent demand that typically must be forecasted first because it drives everything downstream.

Strategic vs. tactical forecasts

Forecasts can also be classified by decision horizon.

Strategic forecasts support medium- and long-term decisions, such as entering or exiting a market, building a new factory, or making heavy capacity investments. Errors can be costly because these decisions are hard to reverse.

Tactical forecasts support short-term, day-to-day operational decisions, such as staffing a restaurant next week, scheduling machines for tomorrow, or setting reorder quantities for the next delivery cycle.

Qualitative vs. quantitative forecasting

Qualitative forecasting uses judgment, expert opinion, and market information. It is flexible and can incorporate forward-looking considerations, but it is vulnerable to behavioral biases such as overconfidence, anchoring, and groupthink.

Quantitative forecasting relies on data and explicit mathematical models. It is objective, repeatable, and can be automated, but it may fail when the world changes in ways not reflected in historical data.

This chapter focuses on the simplest quantitative methods commonly taught early in OM: time-series forecasting methods that use past demand to predict future demand.

3 Principles and Intuition: What Forecasts Can and Cannot Do

Several practical principles guide forecasting in OM.

First, **forecasting is only forecasting**. In most real systems, the future contains randomness: weather, traffic, competitor promotions, news events, and many other factors. Perfect forecasts are generally impossible.

Second, **the longer the forecast horizon, the worse the forecast**. Uncertainty compounds over time. Predicting tomorrow's sales is usually easier than predicting sales one year from now.

Third, **aggregate forecasts are usually more accurate**. Aggregation smooths out noise. For instance, forecasting total weekly demand for a product category is often easier than forecasting demand for each individual SKU.

Finally, **good forecasting balances business acumen and quantitative analysis**. A purely data-driven model may miss upcoming events (a marketing campaign, a new competitor, regulatory changes), while a purely judgmental forecast may be inconsistent or biased. Many organizations combine both.

4 Time-Series Thinking: Trend, Seasonality, and Random Variation

A **time series** is data observed sequentially over time (daily demand, weekly sales, monthly patient arrivals). Time-series forecasting is a form of **extrapolation**: it uses patterns in the past to infer the near future.

This approach implicitly assumes that the past contains useful information about the future and that at least part of the past pattern will continue. That is a reasonable assumption in many mature settings (stable products, stable customer base), but it can be risky for new products or during structural shifts.

A time series often contains three intuitive components.

Trend is a long-run increase or decrease over time. A growing user base for a mature technology platform often shows an upward trend.

Seasonality is a repeating periodic pattern. Retail sales commonly rise at the end of the year due to holidays and fall afterward.

Random variation is the remaining irregular fluctuation after trend and seasonality. Even if you know that Fridays are busier than Mondays, a particular Friday can still be unusually quiet because of heavy rain.

In a full time-series course, you would learn explicit models for trend and seasonality. In many introductory OM courses, the focus is instead on simple methods that *reduce random noise* and provide usable short-term forecasts.

5 A Fundamental Trade-Off: Responsiveness vs. Robustness

Before learning formulas, it helps to understand the central design tension in forecasting.

A forecast should be **responsive**, meaning it reacts quickly when demand truly changes (for example, when a product becomes more popular).

A forecast should also be **robust** or **stable**, meaning it is not overly affected by random noise. Stability matters because managers use forecasts to make decisions; if the forecast swings wildly from period to period, staffing and inventory plans become erratic.

The problem is that responsiveness and robustness tend to conflict. If you react strongly to the most recent observation, you will also react strongly to noise. If you average over many past observations, you smooth noise but respond slowly to real changes. The methods below can be understood as different ways of choosing where to sit on this trade-off.

6 Forecasting Methods for Short-Term Demand

Throughout, let A_t denote the **actual realized demand** in period t, and let F_t denote the **forecast for period** t (made at the end of period $t-1$). This notation is consistent with the lecture materials: the forecast for period t uses information available up to period $t-1$.

Naïve forecast

The simplest possible method sets next period's forecast equal to last period's demand:

$$F_t = A_{t-1}.$$

This method is maximally responsive, but it is also maximally sensitive to noise. Because every random shock in A_{t-1} is copied into F_t, the forecast can be unstable and often inaccurate.

Simple moving average

A **simple moving average (SMA)** sets the forecast equal to the average of the most recent n actual demands:

$$F_t = \frac{A_{t-1} + A_{t-2} + \cdots + A_{t-n}}{n}.$$

The parameter n is the **look-back window length**. Increasing n averages over more history, which tends to smooth random variation and produce a more stable forecast. However, a larger n also makes the forecast less responsive to recent changes.

As a qualitative rule, for products where recent information matters a lot (new products, markets with fast changes), a smaller n is often preferred. For stable products, a larger n may work well.

Weighted moving average

A **weighted moving average (WMA)** generalizes the simple moving average by allowing different weights for different lags:

$$F_t = w_1 A_{t-1} + w_2 A_{t-2} + \cdots + w_n A_{t-n}, \qquad \text{where } \sum_{i=1}^{n} w_i = 1.$$

If $w_i = 1/n$ for all i, the WMA becomes the SMA. The key design choice is how much weight to place on recent observations. Making w_1 larger increases responsiveness because the most recent demand influences the forecast more. But it also increases sensitivity to noise.

A practical drawback of moving-average methods is operational: to forecast many items, you must store and update many past observations. In addition, data older than n periods is dropped entirely, even though it might contain useful long-run information.

Exponential smoothing

Exponential smoothing produces a forecast using only the most recent actual demand and the most recent forecast. The basic (single) exponential smoothing formula is

$$F_t = \alpha A_{t-1} + (1-\alpha) F_{t-1}, \qquad 0 \le \alpha \le 1,$$

which can be rewritten as

$$F_t = F_{t-1} + \alpha \left(A_{t-1} - F_{t-1} \right).$$

The parameter α is the **smoothing constant**. The first equation shows that the new forecast is a weighted average of the last forecast and the last observed demand. The second equation offers a clear interpretation: you start with the old forecast F_{t-1} and then *revise it in the direction of the observed error* $A_{t-1} - F_{t-1}$, moving a fraction α of the way.

The trade-off appears again. A larger α puts more emphasis on recent demand, producing a forecast that is more responsive but less stable. A smaller α makes the forecast more stable but slower to react.

Exponential smoothing is popular in operations because it is simple, fast, and memory-light: you do not need to carry a long history to update the forecast.

7 Worked Examples

Example 1: Weighted moving average for weekly demand

Suppose weekly demand for a product is:

$$A_1 = 650, \quad A_2 = 678, \quad A_3 = 720.$$

You want to forecast week 4 using a weighted moving average with weights placed on the most recent weeks:

$$w_1 = 0.5, \quad w_2 = 0.3, \quad w_3 = 0.2,$$

where w_1 multiplies A_{t-1} (the most recent observation), w_2 multiplies A_{t-2}, and so on. Then the week 4 forecast is

$$F_4 = 0.5A_3 + 0.3A_2 + 0.2A_1 = 0.5(720) + 0.3(678) + 0.2(650).$$

Compute each term:

$$0.5(720) = 360, \qquad 0.3(678) = 203.4, \qquad 0.2(650) = 130.$$

So

$$F_4 = 360 + 203.4 + 130 = 693.4.$$

This forecast is below the most recent demand $A_3 = 720$ because it blends in earlier, lower weeks. If you increased w_1 further (for example, to 0.7) the forecast would move closer to 720, becoming more responsive but potentially noisier.

Example 2: Exponential smoothing update

Assume last month's forecast was $F_{t-1} = 1050$ units, and actual demand turned out to be $A_{t-1} = 1000$ units. With smoothing constant $\alpha = 0.20$, the new forecast is

$$F_t = F_{t-1} + \alpha(A_{t-1} - F_{t-1}) = 1050 + 0.2(1000 - 1050).$$

The error is $1000 - 1050 = -50$, so

$$F_t = 1050 + 0.2(-50) = 1050 - 10 = 1040.$$

The forecast decreases because actual demand was below forecast. Notice the logic: exponential smoothing does not jump all the way to 1000; it adjusts partially, which helps reduce overreaction to random shocks.

8 Measuring Forecast Accuracy: Error, MAD, and Tracking Signal

Forecasts should be evaluated, not just produced. Suppose you have actual demand A_t and forecasts F_t for periods $t = 1, 2, \ldots, T$.

Forecast error

The **forecast error** in period t is

$$e_t = A_t - F_t.$$

This can be positive or negative. A positive error means actual demand exceeded forecast (you under-forecasted). A negative error means actual demand was less than forecast (you over-forecasted).

Mean absolute deviation (MAD)

The **mean absolute deviation** up to time t is

$$\text{MAD}_t = \frac{1}{t}\sum_{i=1}^{t} |A_i - F_i| = \frac{1}{t}\sum_{i=1}^{t} |e_i|.$$

MAD measures the typical size of the forecast error without letting positive and negative errors cancel. Smaller MAD indicates forecasts that are, on average, closer to the actual demand.

Tracking signal (TS)

The **tracking signal** up to time t is defined as

$$\mathrm{TS}_t = \frac{\sum_{i=1}^{t}(A_i - F_i)}{\mathrm{MAD}_t} = \frac{\sum_{i=1}^{t} e_i}{\mathrm{MAD}_t}.$$

Unlike MAD, the numerator sums *signed* errors, so persistent over-forecasting or under-forecasting accumulates over time. Tracking signal is therefore a bias diagnostic: it helps detect whether the forecasting method is systematically high or low.

A common rule of thumb in operations practice is that if

$$|\mathrm{TS}_t| > 3.75,$$

then the forecast may be performing poorly and should be investigated or recalibrated. Importantly, "smaller is better" is not the right interpretation for TS because TS can be negative; what you typically want is TS fluctuating around zero rather than trending away.

A short worked accuracy example (from the lecture table)

Consider five months of actual demand and forecasts (in millions of dollars). For Method 1:

Month t	A_t	F_t (Method 1)	$e_t = A_t - F_t$
1	100	60	40
2	100	130	-30
3	200	200	0
4	200	270	-70
5	400	340	60

Compute cumulative absolute error:

$$\sum_{i=1}^{5} |e_i| = |40| + |-30| + |0| + |-70| + |60| = 40 + 30 + 0 + 70 + 60 = 200.$$

So

$$\mathrm{MAD}_5 = \frac{200}{5} = 40.$$

Compute cumulative signed error:

$$\sum_{i=1}^{5} e_i = 40 - 30 + 0 - 70 + 60 = 0,$$

so

$$\mathrm{TS}_5 = \frac{0}{40} = 0.$$

This illustrates the difference between the measures. MAD tells you the average magnitude of error is 40. TS tells you that, over these five months, Method 1 does not show persistent bias in one direction because positive and negative errors balance out in the long run.

9 Common Pitfalls and How to Avoid Them

Forecasting mistakes often come from interpretation and bookkeeping rather than difficult mathematics.

A frequent pitfall is mixing up indices. In moving averages and weighted moving averages, A_{t-1} is the most recent demand used to forecast period t. If you are forecasting Week 4, then A_3 is the most recent observation and must receive weight w_1 (the largest weight if you want responsiveness).

Another common pitfall is forgetting the responsiveness–robustness trade-off. Choosing a very small n in a moving average or a very large α in exponential smoothing can make forecasts jumpy. Choosing a very large n or a very small α can make forecasts too sluggish, causing systematic under-forecasting in a rising market or over-forecasting in a declining market.

Students also sometimes misread tracking signal. TS is not an "accuracy" measure in the same sense as MAD. A method can have a small MAD but a poor TS if it is consistently biased by a small amount (errors accumulate). Conversely, a method can have a large MAD but a TS near zero if it alternates between over-forecasting and under-forecasting, causing cancellation. The best practice is to examine both: MAD for typical error size, and TS for bias over time.

Finally, time-series methods can fail during structural changes. If a competitor enters, a regulation changes, or a product is replaced by a new technology, historical patterns may stop being informative. In such cases, relying only on extrapolation can be dangerous; managers often supplement quantitative forecasts with qualitative insights and scenario planning.

10 Conclusion: A Practical Forecasting Workflow

In introductory operations management, forecasting is best viewed as a disciplined routine rather than a search for perfection. You select a method appropriate for the context, tune its parameter(s) to balance responsiveness and robustness, and monitor performance using measures like MAD and tracking signal. When the environment changes, you revisit assumptions and update the model.

The three core methods in this chapter—simple moving average, weighted moving average, and exponential smoothing—are deliberately simple. Their value is that they teach the essential OM logic: forecasts are inputs to operational decisions, and the right forecast is the one that is accurate enough for the decision, stable enough to support planning, and responsive enough to keep up with real changes in demand.

Inventory Management in Operations Management: The EOQ Model (A Concise Chapter)

1 Why inventory matters in operations management

Operations management is about designing and running processes that deliver goods and services effectively. Few topics connect so many operational decisions as *inventory*. Inventory can be a firm's largest asset on the balance sheet, but it can also be a major source of waste if it is mismanaged. Too much inventory ties up cash, consumes space, and risks obsolescence. Too little inventory leads to stockouts, lost sales, and damaged customer trust.

At a high level, **inventory management** aims to match *(inventory) supply* with *(customer) demand*. This balancing act sits at the intersection of manufacturing, purchasing, logistics, marketing, and accounting. In practice, firms rely on inventory tracking, order management, reporting and analytics, and increasingly on tools such as IoT sensors, AI forecasting, and data-driven replenishment systems. The core managerial question remains simple:

> *How much should we order or produce, and when should we reorder, so that we meet demand at minimum cost?*

This chapter introduces the Economic Order Quantity (EOQ) model, one of the most widely taught and used foundational models in inventory management.

2 Key definitions and costs

Inventory

Inventory is a stock of goods awaiting consumption. It can appear at many points in a supply chain: supplier, manufacturer, distributor, retailer, and even in-transit between them. Inventory includes raw materials, work-in-process, supplies, and finished goods.

Demand rate and lead time

In many inventory models we distinguish:

- D: **demand rate** (units per unit time, such as units/week or units/year).
- L: **lead time** (time between placing an order and receiving it).

A central simplification in EOQ is that D and L are known and constant (no randomness).

Three major cost concepts

Inventory decisions are driven by a few recurring cost categories.

Ordering (setup) cost. Each time you place an order (or start a production run), you often incur a mostly fixed administrative or setup cost. We denote:

$$S = \text{ordering/setup cost per order (dollars/order).}$$

This can include paperwork, contracting, shipping arrangements, machine setup time, quality checks, and receiving/inspection.

Holding (carrying) cost. Keeping inventory is not free. Holding cost includes storage, insurance, shrinkage, spoilage, obsolescence, and most importantly the *opportunity cost of capital*. We denote:

$$H = \text{holding cost per unit per unit time (dollars/unit/time).}$$

If holding cost is stated as a percentage of item value (e.g., "1% per week"), it must be converted into a dollar amount per unit per week.

Purchasing cost. If each unit costs C to buy (or produce), then:

$$C = \text{purchase cost per unit (dollars/unit).}$$

EOQ will show an important fact: when unit cost is constant and demand is fixed, purchasing cost affects *total cost* but typically does *not* change the optimal EOQ order quantity.

Stockout (shortage) cost and uncertainty

Real systems can also have **stockout cost** (lost sales, penalty fees, and customer goodwill loss). EOQ in its basic form assumes stockouts do not occur and demand is deterministic. When demand is uncertain or product life is short, other models such as the *newsvendor* model are more appropriate.

3 Why firms hold inventory: types and intuition

Inventory is not always a mistake; it often has a purpose.

Pipeline inventory arises naturally because items spend time moving or being processed. Little's Law links these ideas:

$$\text{Inventory} = \text{Flow rate} \times \text{Flow time.}$$

If a process has positive throughput and positive flow time, it inevitably has some inventory (including in-transit goods).

Seasonal inventory occurs when demand fluctuates but capacity is relatively fixed. For example, a company selling holiday decorations cannot manufacture everything during the peak season, so it produces earlier and holds inventory.

Cycle inventory is created by economies of scale: ordering or producing in batches reduces ordering/setup frequency, but increases average inventory.

Buffer inventory sits between activities to reduce the impact of disruptions. If an upstream machine breaks down, a buffer can allow downstream work to continue temporarily.

Safety inventory protects against uncertainty in demand and/or lead time. EOQ focuses on a deterministic setting, so safety stock is not central here, but the concept matters in practice.

4 When EOQ is the right model

The **EOQ model** is intended for **long life-cycle** (non-perishable, non-obsolescent) items with **stable demand**. It is most appropriate when:

- Demand rate D is constant and known.
- Lead time L is constant and known.
- Each order has a fixed ordering cost S.
- Holding cost is proportional to average inventory, with constant rate H.
- Replenishment arrives as a batch (inventory jumps up when the shipment arrives).

Even when these assumptions are not perfectly true, EOQ can still provide a useful baseline policy and a way to reason about trade-offs.

5 The EOQ saw-tooth pattern

Suppose a firm orders a fixed quantity Q whenever it orders. Under constant demand D, inventory falls linearly over time because customers consume inventory at a constant rate. When a replenishment arrives, inventory jumps up by Q. Repeating this creates the classic *saw-tooth* inventory pattern.

Two facts from the saw-tooth picture drive the math:

First, the time between replenishments is

$$\text{cycle length} = \frac{Q}{D}.$$

Second, inventory decreases from Q down to 0 each cycle, so the average inventory level is

$$\text{average inventory} = \frac{Q}{2}.$$

In the deterministic EOQ setting, the firm times replenishment so that the new batch arrives exactly when the old inventory reaches zero. This avoids unnecessary holding cost and prevents stockouts.

6 Total cost in the EOQ model

Choose a time unit (week, month, year) and use it consistently for all quantities. Let:

$D =$ demand per unit time, $\quad S =$ ordering cost per order, $\quad H =$ holding cost per unit per unit time.

If the firm orders Q each time, it places $\frac{D}{Q}$ orders per unit time. Therefore:

Ordering cost per unit time.

$$\text{Ordering cost} = S \cdot \frac{D}{Q}.$$

Holding cost per unit time.

$$\text{Holding cost} = H \cdot \frac{Q}{2}.$$

Total cost per unit time (excluding purchasing cost).

$$TC(Q) = H\frac{Q}{2} + S\frac{D}{Q}.$$

This expression captures the central **trade-off**. Larger Q means fewer orders (lower ordering cost) but higher average inventory (higher holding cost). Smaller Q means the opposite.

7 Optimal order quantity (EOQ)

To minimize $TC(Q)$, EOQ sets the marginal increase in holding cost equal to the marginal decrease in ordering cost. Using calculus (or accepting the standard result), the optimal order quantity is:

$$Q^* = \sqrt{\frac{2DS}{H}}.$$

This formula is worth reading economically: Q^* increases with demand D and ordering cost S, and decreases with holding cost H. It also has a *square-root* structure: if D quadruples, Q^* roughly doubles (all else equal).

Optimal cost and an important property

Plugging Q^* into the cost expression yields the minimum total cost per unit time (excluding purchasing cost):

$$TC(Q^*) = \sqrt{2DSH}.$$

A useful and memorable property holds at the optimum:

$$\text{Ordering cost at } Q^* = \text{Holding cost at } Q^* = \frac{1}{2}TC(Q^*).$$

This means that if, under your current policy, ordering cost is much larger than holding cost, you are likely ordering too frequently (your Q is too small). If holding cost dominates, you are likely ordering too much each time (your Q is too large).

8 Extensions: purchasing cost and lead time

Adding per-unit purchasing cost

If unit purchase cost is C dollars per unit and demand is fixed at D units per unit time, then purchasing cost per unit time is simply:

$$\text{Purchasing cost} = DC.$$

Including purchasing cost, total cost becomes:

$$TC(Q) = DC + H\frac{Q}{2} + S\frac{D}{Q}.$$

Since DC does *not* depend on Q, it does not affect the minimizing Q. Therefore the EOQ remains:

$$Q^* = \sqrt{\frac{2DS}{H}}.$$

However, when asked for *total cost including purchasing*, you must include the DC term.

Lead time and the reorder point

Lead time L does not change the EOQ trade-off: it does not change average inventory $Q/2$ or order frequency D/Q. Instead, lead time determines *when* you must place an order.

Define the **reorder point** (ROP) as the on-hand inventory level that triggers a new order. Under constant demand and constant lead time:

$$ROP = DL.$$

The logic is simple: during the L time units you wait for delivery, demand consumes DL units. If you reorder when you have exactly DL units left, those units will last exactly until the new shipment arrives.

9 Worked example 1: A computer retailer (EOQ with percentage holding cost)

A retailer sells a stable $D = 100$ computers per week. Each order costs $S = \$5000$. Each computer costs $C = \$400$. Holding cost is approximately 1% of item value per week. Lead time is $L = 1$ week.

First convert holding cost to dollars per unit per week:

$$H = 0.01 \times 400 = 4 \text{ dollars per computer per week.}$$

Compute EOQ:

$$Q^* = \sqrt{\frac{2DS}{H}} = \sqrt{\frac{2(100)(5000)}{4}} = \sqrt{250000} = 500 \text{ computers.}$$

Time between orders (cycle length):

$$\frac{Q^*}{D} = \frac{500}{100} = 5 \text{ weeks.}$$

Ordering cost per week at Q^*:

$$S\frac{D}{Q^*} = 5000 \cdot \frac{100}{500} = 1000.$$

Holding cost per week at Q^*:

$$H\frac{Q^*}{2} = 4 \cdot \frac{500}{2} = 1000.$$

As predicted, they match at the optimum.

Reorder point:

$$ROP = DL = 100 \times 1 = 100 \text{ computers.}$$

So the operational policy is: *order 500 computers whenever on-hand inventory falls to 100; the shipment arrives one week later as inventory reaches zero.*

10 Worked example 2: Components with annual demand and daily lead time

A firm uses $D = 18{,}000$ units per year of a component. Each order costs $S = \$100$. Holding cost is $H = \$5$ per unit per year. The firm operates 360 days per year and lead time is $L = 4$ days.

EOQ:

$$Q^* = \sqrt{\frac{2DS}{H}} = \sqrt{\frac{2(18{,}000)(100)}{5}} = \sqrt{720{,}000} \approx 848.5 \text{ units.}$$

In practice you would order about 849 units (or round to a feasible pack size).

Annual ordering cost:

$$S\frac{D}{Q^*} \approx 100 \cdot \frac{18{,}000}{848.5} \approx 2121.3.$$

Annual holding cost:

$$H\frac{Q^*}{2} \approx 5 \cdot \frac{848.5}{2} \approx 2121.3,$$

again equal as expected.

To compute the reorder point, convert demand to a daily rate:

$$d = \frac{18{,}000}{360} = 50 \text{ units/day.}$$

Then:

$$ROP = dL = 50 \times 4 = 200 \text{ units.}$$

11 Common pitfalls and how to avoid them

EOQ is straightforward, but students (and sometimes managers) often make avoidable mistakes.

Mixing time units. If D is "per year" but H is "per week," your EOQ will be wrong. Pick one time unit and convert everything to it. For example, if holding cost is given as 1% per week, keep D in units/week.

Forgetting to convert percentage holding cost into dollars. Holding cost is H in dollars per unit per time, not a percentage. If holding cost is stated as a rate i per period and unit value is C, then typically $H = iC$ (as in the computer example).

Thinking lead time changes EOQ. In the basic EOQ model, lead time changes *reorder point* $ROP = DL$, not Q^*. Lead time affects *when* to order, not *how much* to order.

Including purchasing cost incorrectly. With constant unit price C and fixed demand D, the purchasing cost per period is DC, which does not depend on Q. It changes total cost accounting but does not change the EOQ decision.

Using EOQ when the product is short-life or demand is highly uncertain. EOQ assumes a stable, deterministic demand rate and no obsolescence. For seasonal fashion, fresh food, or one-shot selling seasons, a newsvendor-style model is usually more appropriate.

12 Managerial meaning and real-world intuition

The EOQ model is not only a formula; it is a way of thinking. It says that ordering policies should balance two forces: the fixed cost of placing orders and the ongoing cost of carrying inventory. The square-root structure also provides a simple sensitivity rule: large changes in demand or ordering cost produce smaller proportional changes in the optimal batch size, which is why EOQ-based policies can be reasonably robust.

Real firms rarely have perfectly constant demand or perfectly constant lead times, so EOQ is often used as a baseline. A grocery retailer might use EOQ-like reasoning for stable household items (e.g., bottled water) while using different logic for perishables. An industrial plant might apply EOQ to maintenance, repair, and operations (MRO) parts where demand is steady and stockouts are costly.

13 Summary

Inventory management seeks to match supply with demand while balancing ordering, holding, and (sometimes) shortage costs. In the deterministic long-life setting, the EOQ model provides a clean answer:

$$TC(Q) = H\frac{Q}{2} + S\frac{D}{Q}, \qquad Q^* = \sqrt{\frac{2DS}{H}}, \qquad ROP = DL.$$

At Q^*, holding and ordering costs are equal. Purchasing cost DC affects total cost accounting but not the optimal EOQ quantity when unit cost is constant. Lead time affects the reorder point but not the EOQ batch size.

Inventory Management II: The Newsvendor Model (A Concise Operations Management Chapter)

For junior undergraduate students

1 Why a "Newsvendor" model belongs in Operations Management

Operations Management (OM) studies how organizations design and run processes so that resources (money, time, labor, capacity, materials) are turned into valuable goods and services. A recurring OM question is deceptively simple: *How much should we prepare before we know what customers will want?*

Many decisions must be made *before* demand is observed. A campus must decide how many course sections to open before students finish registering. A hospital must reserve operating room (OR) time before the true duration of each surgery is known. A retailer must decide how many seasonal items to stock before the season starts. These problems share three features:

First, demand is uncertain. Second, there is a one-time (or very limited) chance to commit resources. Third, leftovers lose value, sometimes becoming nearly worthless. This is the setting of the **newsvendor model**, named after the classic situation of a newsstand buying newspapers in the morning, selling during the day, and facing unsold copies that become outdated by night.

The newsvendor model is one of the most important single-period inventory models in OM because it teaches a central trade-off: ordering too little creates *lost sales* (missed revenue), while ordering too much creates *leftovers* (wasted cost).

2 Contrasting two inventory mindsets: EOQ versus Newsvendor

Before learning newsvendor, students often see the EOQ (Economic Order Quantity) model. EOQ is useful when products have long lifecycles and you can order repeatedly over time. In that setting, demand is often approximated as deterministic, and the goal is usually to minimize costs such as ordering and holding.

The newsvendor setting is different. It focuses on a *single selling period* (a day, a season, a registration window) with random demand and costly leftovers. Here the decision is not "how often to reorder," but rather "what initial level to commit to," and the objective is typically to *maximize expected profit* rather than minimize holding-plus-ordering costs.

3 Model ingredients and key notation

We use the following notation throughout:

- D: random demand during the selling period.
- Q: the inventory level (order quantity) chosen before observing D.

- $\Pr(\cdot)$: probability.

To connect decisions to uncertainty, we describe demand with a probability distribution. In a **discrete** distribution, demand takes specific values (e.g., 8,000, 10,000, ...). In a **continuous** distribution, demand can take any value within an interval (e.g., modeled by a Normal distribution). In either case, we often use the **cumulative distribution function (CDF)**:

$$F(q) = \Pr(D \leq q).$$

The CDF answers the question: "What is the probability demand is at most q?"

A common pitfall: density/probability versus cumulative probability

In discrete settings, a table may provide $\Pr(D = q)$, the probability of demand equaling exactly q. But the newsvendor solution uses $\Pr(D \leq Q)$, a cumulative probability. Confusing these two is one of the most frequent errors.

4 Service level and shortage probability

Managers often track service performance, not only profit. A widely used measure is the **service level** (also called the in-stock probability in this context):

$$\text{Service level} = \Pr(D \leq Q).$$

This is the probability that the inventory is sufficient to satisfy all demand in the period. If Q increases, the event $\{D \leq Q\}$ becomes more likely, so the service level increases monotonically with Q from 0% toward 100%.

The **probability of shortage** is the probability that demand exceeds inventory:

$$\Pr(D > Q) = 1 - \Pr(D \leq Q).$$

As Q increases, the shortage probability decreases.

These relationships are intuitive: having more inventory makes it less likely that demand will exceed supply. However, stocking more also increases the risk of leftovers, which can reduce profit. The newsvendor model explains how to balance these goals.

5 Profit intuition: why "more inventory" is not always better

If a manager cared only about service level, the logic would be simple: order more. In the extreme, ordering an enormous amount would nearly guarantee no shortages.

But the newsvendor setting includes costly leftovers. Once the selling period ends, unsold units can often only be salvaged (returned, discounted, scrapped, or repurposed) at a lower value. Therefore, more inventory raises service level but can *lower expected profit* when leftovers become likely.

A crucial lesson is that profit typically behaves like a "hill" as a function of Q: at first it rises (because extra units are likely to sell), then it falls (because extra units are increasingly likely to remain unsold). Meanwhile, service level keeps rising. This creates a trade-off once inventory goes beyond the profit-maximizing level.

6 Marginal analysis: deciding whether to add one more unit

The newsvendor solution is most easily understood through **marginal analysis**. Imagine you have already decided to stock Q units. You now ask whether it is beneficial to stock one more unit, the $(Q+1)$st.

To compute the marginal expected profit, we consider two scenarios:

1. If demand is high enough to sell the extra unit (i.e., $D > Q$), you earn the per-unit profit margin.
2. If demand is not high enough (i.e., $D \leq Q$), the extra unit is unsold, and you incur a per-unit leftover loss relative to salvage.

To formalize this, introduce three per-unit values:

$r =$ retail price (selling price), $\quad c =$ purchase/production cost, $\quad s =$ salvage value of an unsold unit.

Typically, $r > c > s$ in classic retail settings, though variations are possible.

If the extra unit sells, its incremental profit is $r - c$. If it does not sell, you recover only s after paying c, so the incremental loss is $c - s$.

Therefore, the **marginal expected profit** of increasing inventory from Q to $Q+1$ is:

$$\Delta\Pi(Q) = (r-c)\Pr(D > Q) - (c-s)\Pr(D \leq Q).$$

If $\Delta\Pi(Q) > 0$, ordering one more unit increases expected profit, so you should increase Q. If $\Delta\Pi(Q) < 0$, ordering one more unit decreases expected profit, so you should not increase Q further. The best Q is reached when marginal expected profit is around zero.

7 Understocking and overstocking costs (the heart of the trade-off)

The model becomes cleaner when we name two fundamental costs.

Understocking cost (C_u). Understocking means having too little inventory when demand is sufficient. The penalty is the profit you *could have earned* from selling one more unit but did not.

$$C_u = r - c.$$

Overstocking cost (C_o). Overstocking means having too much inventory when demand is insufficient. The penalty is the loss from ending with one more unsold unit.

$$C_o = c - s.$$

Both are usually positive in standard settings. Substituting C_u and C_o into marginal expected profit yields:

$$\begin{aligned}\Delta\Pi(Q) &= C_u \Pr(D > Q) - C_o \Pr(D \leq Q) \\ &= C_u\,[1 - \Pr(D \leq Q)] - C_o \Pr(D \leq Q) \\ &= C_u - (C_u + C_o)\Pr(D \leq Q).\end{aligned}$$

This expression reveals a key monotonic pattern: as Q increases, $\Pr(D \leq Q)$ increases, so $\Delta\Pi(Q)$ decreases from approximately $+C_u$ (when Q is extremely small) down toward $-C_o$ (when Q is extremely large). Thus the marginal value of additional inventory naturally declines as inventory grows.

8 Critical fractile (critical ratio): the optimality condition

The profit-maximizing inventory level Q^* is obtained when the marginal expected profit is zero:

$$\Delta\Pi(Q^*) = 0.$$

Using the formula above,

$$C_u - (C_u + C_o)\Pr(D \le Q^*) = 0,$$

so the optimality condition becomes

$$\Pr(D \le Q^*) = \frac{C_u}{C_u + C_o}.$$

The right-hand side is called the **critical fractile** (or critical ratio). It is a number between 0 and 1 that depends only on the economic trade-off between understocking and overstocking.

Interpretation

You should choose Q^* so that the probability of meeting all demand equals the critical fractile. If understocking is very costly (large C_u), the fraction increases, pushing Q^* up. If overstocking is very costly (large C_o), the fraction decreases, pushing Q^* down.

9 Service level versus expected profit: where the trade-off appears

Because service level equals $\Pr(D \le Q)$, raising Q always raises service level. Expected profit, however, rises only until Q reaches Q^* and then falls.

This yields a practical managerial insight. If current inventory is *below* the newsvendor optimum, increasing Q improves both service level and expected profit because the marginal expected profit is positive. If inventory is *above* the newsvendor optimum, increasing Q still improves service level but reduces expected profit because the marginal expected profit is negative. The trade-off emerges only once you are above the profit-maximizing level.

10 Worked example 1: Pumpkins (marginal thinking in action)

A retailer sells pumpkins for a holiday. The information is:

$$\Pr(D \le 200) = 0.5, \qquad \Pr(D \le 300) = 0.9.$$

The retailer pays $c = \$2$ per pumpkin, sells at $r = \$5$, and salvages unsold pumpkins at $s = \$1$.

If one more pumpkin sells, profit is $r - c = 3$. If it does not sell, the loss is $c - s = 1$.

Should the retailer buy the 201st pumpkin if she already has $Q = 200$? The 201st pumpkin sells exactly when $D > 200$, which has probability $1 - 0.5 = 0.5$. It is unsold when $D \le 200$, probability 0.5.

$$\Delta\Pi(200) = 3(0.5) - 1(0.5) = 1 > 0.$$

So buying the 201st pumpkin increases expected profit.

Should the retailer buy the 301st pumpkin if she already has $Q = 300$? Now it sells when $D > 300$, probability $1 - 0.9 = 0.1$, and is unsold with probability 0.9.

$$\Delta\Pi(300) = 3(0.1) - 1(0.9) = -0.6 < 0.$$

So buying the 301st pumpkin *reduces* expected profit. This illustrates why the marginal value declines as inventory grows: selling an extra unit requires a higher and less likely demand level.

11 Worked example 2: Discrete demand and the critical fractile

A seasonal sportswear firm produces a winter item with:

$$c = \$80, \qquad r = \$125, \qquad s = \$20.$$

Therefore,

$$C_u = r - c = 45, \qquad C_o = c - s = 60.$$

The critical fractile is

$$\frac{C_u}{C_u + C_o} = \frac{45}{45 + 60} = \frac{45}{105} \approx 0.429.$$

Demand is discrete:

Demand d	$\Pr(D = d)$	Cumulative $\Pr(D \le d)$
8,000	0.11	0.11
10,000	0.11	0.22
12,000	0.28	0.50
14,000	0.22	0.72
16,000	0.18	0.90
18,000	0.10	1.00

We want $\Pr(D \le Q^*) \approx 0.429$. Looking at the cumulative probabilities, 0.429 lies between 0.22 (at 10,000) and 0.50 (at 12,000). Thus Q^* is between 10,000 and 12,000 in the sense of matching the target service level.

A simple course-level rule is to *round up* to the next available demand level, giving:

$$Q^* = 12{,}000.$$

This yields service level $\Pr(D \le 12{,}000) = 0.50$. A more rigorous approach (sometimes required in other courses) is to compute expected profit at the nearby candidate quantities (here 10,000 and 12,000) and select the higher-profit option. The round-up rule is a convenient shortcut when permitted.

12 Common pitfalls and how to avoid them

Students often understand the story but make mistakes in execution. The following issues are especially common.

Confusing $\Pr(D = Q)$ with $\Pr(D \le Q)$ is a frequent error in discrete problems. The critical fractile condition always uses the cumulative probability (a CDF value), not a point probability.

Mixing up understocking and overstocking costs leads to wrong critical fractiles. In the classic inventory setting, remember that $C_u = r - c$ is the missed margin from lost sales, while $C_o = c - s$ is the leftover loss.

Forgetting salvage value is another trap. Salvage does not mean "no cost"; it means you recover some value, which reduces the overstocking cost from c to $c - s$.

Interpreting Q^* as a guaranteed "high service level" is also misleading. The optimal service level is not necessarily close to 100%. It equals the critical fractile, which depends on economics. If leftovers are very costly (large C_o), the model may recommend a relatively low service level.

Finally, it is important to remember what the model assumes. Newsvendor is a single-period model with one pre-commitment decision and uncertain demand. If you can reorder during the period, if leftovers can be carried to the next period without loss, or if demand depends strongly on your inventory choice (e.g., customers are attracted by large displays), you may need a different model or additional extensions.

13 Summary: what you should take away

The newsvendor model explains how to set an inventory (or capacity) level when demand is random, commitment must be made before demand is known, and leftovers are costly. It introduces service level as $\Pr(D \le Q)$, shows how marginal analysis connects inventory to expected profit, and delivers a powerful, compact optimality condition:

$$\Pr(D \le Q^*) = \frac{C_u}{C_u + C_o}, \qquad C_u = r - c, \qquad C_o = c - s.$$

In words, the profit-maximizing service level is the critical fractile determined by the relative pain of understocking versus overstocking. Once you know this target probability, finding Q^* becomes a demand-distribution problem: locate the inventory level whose CDF matches that target as closely as possible.

A Concise Chapter on the Newsvendor Model under Normal Demand

(Operations Management: Inventory Management III)

For junior undergraduate students

1 Why the Newsvendor Model Matters

Many operations decisions must be made *before* demand is known. A bakery must decide how many croissants to bake for the morning. A fashion retailer must decide how many seasonal T-shirts to order before the trend fades. A clinic must decide how many flu shots to stock before the season starts. In all of these settings, ordering too little creates *shortages* (missed sales or dissatisfied customers), while ordering too much creates *leftovers* (waste, markdowns, storage, or disposal).

The **newsvendor model** is a classic one-period inventory model that captures this basic trade-off. It is especially useful when the item has a short selling season or a single selling opportunity, so leftover inventory has limited value after the period ends.

This chapter focuses on three practical questions when demand is normally distributed:

1. How to choose the optimal order quantity under normal demand.
2. How to compute key performance metrics (expected leftover, expected sales, expected lost sales, expected profit).
3. Why pooling demand and inventory across locations often increases profit (risk pooling / location pooling).

2 Model Setup and Key Definitions

Consider one product and one selling period.

Decision and uncertainty

Let Q be the order quantity (inventory available for the period). Demand is a random variable D. In this chapter we often assume:

$$D \sim \mathcal{N}(\mu, \sigma^2),$$

where μ is the mean demand and σ is the standard deviation (a measure of variability). In practice, demand cannot be negative; many courses "ignore" negative demand when μ is much larger than σ (so the probability of negative demand is negligible).

Economic inputs

Let

- p be the selling price per unit,

- c be the purchase cost per unit,
- v be the salvage value per leftover unit (what you can recover at the end of the period).

Typically $p > c$ and $v < c$. Salvage value could be a clearance price, a buyback price, or the value from reusing parts.

Underage and overage costs

The newsvendor model uses **marginal** costs of being short or being long by one unit.

Underage cost (also called understocking cost) is the profit you lose by ordering one unit too few:

$$C_u = p - c.$$

Intuition: if you had one more unit when demand was high, you could sell it and earn $p - c$ additional profit.

Overage cost (also called overstocking cost) is the loss you incur by ordering one unit too many:

$$C_o = c - v.$$

Intuition: you paid c for a unit but only recover v if it remains unsold, so the net loss is $c - v$.

A common pitfall is to confuse C_u and C_o with fixed costs. They are *per-unit marginal* quantities.

Service level

In this course session, the **service level** is the probability that inventory covers demand:

$$\text{Service level} = \mathbb{P}(D \leq Q).$$

Higher Q increases this probability, but also increases expected leftovers.

3 The Core Idea: The Critical Fractile

The newsvendor solution is famous because it reduces the optimal decision to a single probability target.

Critical fractile (target service level)

The profit-maximizing order quantity Q^* satisfies:

$$\mathbb{P}(D \leq Q^*) = \frac{C_u}{C_u + C_o}. \tag{1}$$

The ratio

$$\frac{C_u}{C_u + C_o}$$

is called the **critical fractile**. It is also the service level you should aim for at the optimum.

Intuition without heavy math

Imagine increasing Q by one unit.

- If demand is at least Q (you would have stocked out), that extra unit will be sold and earns you C_u.
- If demand is below Q (you would have leftover), that extra unit becomes leftover and costs you C_o.

So the *expected* marginal benefit balances the *expected* marginal cost at the optimum. Equation (1) is the probability statement of this balance.

Pitfall: critical fractile is not "90% by default"

Students sometimes think the goal is always a high service level. In reality, the best service level depends on the economics:

- If C_u is large (stockouts are very costly), the critical fractile is high, and you should carry more inventory.
- If C_o is large (leftovers are very costly), the critical fractile is low, and you should carry less inventory.

4 Working with Normal Demand: From Service Level to Q

When $D \sim \mathcal{N}(\mu, \sigma^2)$, we convert probabilities about D into probabilities about a standard normal random variable.

Standardization and the z-score

Define the **z-score** of an inventory level Q by

$$z = \frac{Q - \mu}{\sigma}.$$

Let $Z \sim \mathcal{N}(0, 1)$ be standard normal. Then:

$$\mathbb{P}(D \le Q) = \mathbb{P}\left(Z \le \frac{Q - \mu}{\sigma}\right) = \Phi(z), \tag{2}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function (CDF). In many exams, you read $\Phi(z)$ from a standard normal table.

Two directions you should be able to do

First, if you are given Q and want the service level:

$$\text{Service level} = \Phi\left(\frac{Q - \mu}{\sigma}\right).$$

Second, if you are given a desired service level α and want Q:

$$\alpha = \Phi(z) \quad \Rightarrow \quad Q = \mu + \sigma z.$$

In table-based calculations, if α lies between two table values, a common rule used in class is the **round-up rule**: choose the larger z (and thus the larger Q) for simplicity.

5 Optimal Order Quantity under Normal Demand

Combine the critical fractile with normal standardization.

Step 1: compute C_u, C_o, and the critical fractile

$$C_u = p - c, \qquad C_o = c - v, \qquad \alpha^* = \frac{C_u}{C_u + C_o}.$$

Step 2: find the optimal z^*

Find z^* satisfying:

$$\Phi(z^*) = \alpha^*.$$

Step 3: compute the optimal order quantity

$$Q^* = \mu + \sigma z^*. \tag{3}$$

A subtle point about variability

Because $Q^* = \mu + \sigma z^*$, increasing σ does not always increase Q^*. It depends on the sign of z^*:

- If $z^* > 0$ (critical fractile above 0.5), larger σ increases Q^*.
- If $z^* < 0$ (critical fractile below 0.5), larger σ decreases Q^*.

This is a common conceptual trap: "more uncertainty means more inventory" is not universally true in the newsvendor model.

6 Performance Metrics under Normal Demand

Choosing Q^* is only half the story. Managers also care about what happens *on average*: How many units will remain unsold? How many will be sold? How many sales will be lost? How much profit can we expect?

Realized metrics (for a given demand realization)

Given inventory Q and realized demand D, define:

$$\begin{aligned} \text{Leftover} &= \max\{Q - D, 0\}, \\ \text{Sales} &= \min\{Q, D\}, \\ \text{Lost sales} &= \max\{D - Q, 0\}. \end{aligned}$$

Two identities always hold (for every realized D):

$$\text{Sales} + \text{Leftover} = Q, \tag{4}$$

$$\text{Sales} + \text{Lost sales} = D. \tag{5}$$

They are simply "conservation laws" of the system. Every unit of inventory is either sold or leftover, and every unit of demand becomes either a sale or a lost sale.

Expected metrics (because demand is random)

When demand is random, leftover, sales, and lost sales are random too. We therefore focus on expectations:

$$\mathbb{E}[\text{Leftover}], \quad \mathbb{E}[\text{Sales}], \quad \mathbb{E}[\text{Lost sales}].$$

A very important distinction is that for a *given* realized demand D, you cannot have both leftover and lost sales positive at the same time. However, when you take expectations over many possible demand outcomes, it is normal that both $\mathbb{E}[\text{Leftover}] > 0$ and $\mathbb{E}[\text{Lost sales}] > 0$.

Expected leftover under normal demand and the inventory function

Under normally distributed demand, the expected leftover has a convenient form:

$$\mathbb{E}[\text{Leftover}] = \sigma\, I(z), \tag{6}$$

where $z = (Q - \mu)/\sigma$ and $I(z)$ is the **standard normal inventory function**. In the course materials, $I(z)$ is obtained from an *inventory function table*, which is different from the standard normal CDF table.

From expected leftover to expected sales and expected lost sales

Using identity (4) and taking expectations:

$$\mathbb{E}[\text{Sales}] = Q - \mathbb{E}[\text{Leftover}]. \tag{7}$$

Using identity (5) and taking expectations (noting $\mathbb{E}[D] = \mu$):

$$\mathbb{E}[\text{Lost sales}] = \mathbb{E}[D] - \mathbb{E}[\text{Sales}] = \mu - \mathbb{E}[\text{Sales}]. \tag{8}$$

Expected profit

Profit in a realization equals profit from sold units minus loss from leftover units:

$$\text{Profit} = (p - c) \cdot \text{Sales} - (c - v) \cdot \text{Leftover}.$$

Taking expectations gives:

$$\mathbb{E}[\text{Profit}] = (p - c)\,\mathbb{E}[\text{Sales}] - (c - v)\,\mathbb{E}[\text{Leftover}] = C_u\,\mathbb{E}[\text{Sales}] - C_o\,\mathbb{E}[\text{Leftover}]. \tag{9}$$

By construction, Q^* from the critical fractile maximizes $\mathbb{E}[\text{Profit}]$.

Pitfall: using the wrong table

The standard normal *distribution table* reports $\Phi(z) = \mathbb{P}(Z \le z)$, so its values must lie between 0 and 1.

The *inventory function table* reports $I(z)$, which is *not* a probability and can be larger than 1. Mixing these tables is one of the most common exam mistakes.

7 Worked Example 1: Ordering under Normal Demand (Service Level and Q)

A retailer faces demand $D \sim \mathcal{N}(500, 60^2)$.

(a) Service level if $Q = 580$

Compute the z-score:

$$z = \frac{580 - 500}{60} = 1.333.$$

From a standard normal table, $\Phi(1.33) \approx 0.9082$. Therefore the service level is about 90.82%.

(b) Inventory needed for a 90% service level

We want $\Phi(z) = 0.90$. From the table, $z \approx 1.29$ (using the round-up rule). Thus,

$$Q = \mu + \sigma z = 500 + 60(1.29) = 577.4 \approx 578.$$

8 Worked Example 2: A Fashion T-shirt Newsvendor (Decision and Performance)

A retailer buys a T-shirt for $c = \$60$, sells it for $p = \$120$, and salvages leftovers for $v = \$20$. Demand is $D \sim \mathcal{N}(100, 40^2)$.

Step 1: compute underage/overage costs

$$C_u = p - c = 120 - 60 = 60, \qquad C_o = c - v = 60 - 20 = 40.$$

Step 2: compute the critical fractile

$$\alpha^* = \frac{C_u}{C_u + C_o} = \frac{60}{60 + 40} = 0.6.$$

Step 3: find z^* and compute Q^*

From the standard normal table, $\Phi(z^*) = 0.60$ corresponds to $z^* \approx 0.26$ (round-up). Then:

$$Q^* = \mu + \sigma z^* = 100 + 40(0.26) = 110.4 \approx 111.$$

Step 4: compute expected leftover, sales, lost sales, and profit

First compute $z = (Q^* - \mu)/\sigma = 0.26$. From the inventory function table, suppose $I(0.26) = 0.5424$ (as given in the session). Then:

$$\mathbb{E}[\text{Leftover}] = \sigma I(z) = 40(0.5424) = 21.696 \approx 22.$$

Next:

$$\mathbb{E}[\text{Sales}] = Q^* - \mathbb{E}[\text{Leftover}] \approx 111 - 22 = 89.$$

Then:

$$\mathbb{E}[\text{Lost sales}] = \mu - \mathbb{E}[\text{Sales}] \approx 100 - 89 = 11.$$

Finally, expected profit:

$$\mathbb{E}[\text{Profit}] = C_u\,\mathbb{E}[\text{Sales}] - C_o\,\mathbb{E}[\text{Leftover}] \approx 60(89) - 40(22) = 5340 - 880 = \$4460.$$

This example shows the full workflow: economics → critical fractile → order quantity → expected performance.

9 Risk Pooling and Location Pooling: Why "Sharing Inventory" Helps

Firms often operate multiple stores, regions, or channels. A key operations question is whether each location should hold its own inventory (separated mode) or whether inventory can be pooled (pooled mode).

Definition: location pooling

Location pooling means combining inventory from multiple locations into a shared pool (for example, one warehouse serving many stores). This is a form of **risk pooling**: aggregating uncertain demands can reduce variability and improve system performance.

A familiar real-world example is e-commerce fulfillment. Instead of every neighborhood shop stocking every item, a platform may stock centrally and ship to customers. The tradeoff is that pooling can introduce delivery time, warehouse setup cost, and incentive conflicts between local sales units, but it can reduce inventory-related waste.

Why pooling changes the demand distribution

Suppose two stores have independent demands:

$$D_1 \sim \mathcal{N}(\mu, \sigma^2), \qquad D_2 \sim \mathcal{N}(\mu, \sigma^2),$$

and D_1 is independent of D_2. Then total demand

$$D_{\text{tot}} = D_1 + D_2$$

is also normal, with

$$\mathbb{E}[D_{\text{tot}}] = 2\mu, \qquad \text{SD}(D_{\text{tot}}) = \sqrt{2}\,\sigma. \tag{10}$$

Notice the mean doubles but the standard deviation grows only by $\sqrt{2}$, not by 2. This is why pooling often reduces *relative* variability.

A convenient measure of relative variability is the **coefficient of variation**:

$$\text{CV} = \frac{\sigma}{\mu}.$$

For two identical independent stores, separated CV is σ/μ, while pooled CV is

$$\text{CV}_{\text{pool}} = \frac{\sqrt{2}\sigma}{2\mu} = \frac{1}{\sqrt{2}} \cdot \frac{\sigma}{\mu},$$

which is smaller.

Intuition: better matching supply and demand

Pooling helps for two reinforcing reasons.

First, it reduces uncertainty in the sense that the pooled demand has smaller CV, so inventory can be set closer to expected demand.

Second, pooling allows one location's leftover to "rescue" another location's stockout. In separated mode, it is possible that Store 1 has leftover while Store 2 has lost sales on the same day; the system wastes inventory in one place and disappoints customers in another. In pooled mode, a shared inventory eliminates this mismatch, creating a hedging effect.

Factors that affect the benefit of pooling

The benefit of pooling is generally larger when demand variability is higher, and smaller when demands are strongly positively correlated. If two locations' demands move perfectly together (correlation 1), pooling provides little or no hedging benefit because both locations are high or low at the same time.

10 A Two-Store Illustration (Separated vs. Pooled)

Return to the T-shirt example. Each store has $D \sim \mathcal{N}(100, 40^2)$, and demands are independent.

Separated mode

Each store orders the single-store optimum $Q^* \approx 111$ and earns expected profit \$4460. Across two stores, totals double:

$$Q_{\text{sep}} = 222, \quad \mathbb{E}[\text{Profit}_{\text{sep}}] = 2(4460) = \$8920.$$

Pooled mode

Total demand is normal by (10):

$$D_{\text{tot}} \sim \mathcal{N}(200, (56.57)^2), \quad \text{since } 56.57 \approx 40\sqrt{2}.$$

Costs are unchanged, so the critical fractile is still 0.6, and $z^* \approx 0.26$ is unchanged. The pooled optimal order quantity is:

$$Q^*_{\text{pool}} = 200 + 56.57(0.26) \approx 214.7 \approx 215.$$

Using the same inventory function value $I(0.26) = 0.5424$:

$$\mathbb{E}[\text{Leftover}_{\text{pool}}] \approx 56.57(0.5424) \approx 31,$$

$$\mathbb{E}[\text{Sales}_{\text{pool}}] \approx 215 - 31 = 184,$$

$$\mathbb{E}[\text{Profit}_{\text{pool}}] \approx 60(184) - 40(31) = 11040 - 1240 = \$9800.$$

Thus pooling increases expected profit (here by about \$880) while reducing expected leftovers and lost sales.

11 Common Pitfalls and How to Avoid Them

Students often lose points not because the ideas are hard, but because they confuse definitions or apply the right idea to the wrong quantity.

First, always compute C_u and C_o correctly. Remember $C_u = p - c$ is *profit* per sold unit, and $C_o = c - v$ is *loss* per leftover unit. Swapping them flips the critical fractile and produces a wildly wrong Q^*.

Second, do not confuse service level $\mathbb{P}(D \le Q)$ with the expected fill rate (a different metric used in some inventory contexts). In this session, service level is the probability of no stockout.

Third, keep the two tables conceptually separate. Use the $\Phi(z)$ table to convert between z and probabilities, and use the $I(z)$ table to compute expected leftover. If you see a number outside $[0, 1]$, it cannot be a probability.

Finally, distinguish realized and expected metrics. In one realization, leftover and lost sales cannot both be positive, but their expectations can both be positive.

12 Summary

The newsvendor model provides a clear and powerful method for one-period inventory decisions. The key is the critical fractile condition

$$\mathbb{P}(D \leq Q^*) = \frac{C_u}{C_u + C_o},$$

which turns an economic tradeoff into a probability target. Under normal demand, the optimal order quantity becomes

$$Q^* = \mu + \sigma z^*, \qquad \Phi(z^*) = \frac{C_u}{C_u + C_o}.$$

Once Q is chosen, performance metrics follow a simple chain: expected leftover comes from $\sigma I(z)$, expected sales equal Q minus expected leftover, expected lost sales equal mean demand minus expected sales, and expected profit combines expected sales and expected leftover using C_u and C_o.

Finally, pooling inventory across independent locations often increases profit by reducing relative variability and improving the match between supply and demand. This is one of the most important operational intuitions that extends far beyond the newsvendor model.

Operations Management: Revenue Management I (Capacity-Based Revenue Management)

Concise textbook-style chapter for junior undergraduates

1 Why Revenue Management Belongs in Operations Management

Operations Management (OM) is often introduced as the study of how organizations design and run processes that produce goods and services. A natural question is: if OM is about processes, where does *revenue* fit? The answer is that many operations decisions determine how much value a firm can capture from a fixed set of resources.

Revenue Management (RM) is the practice of maximizing revenue by optimizing *product availability (capacity)* and *price*. The field was first developed in the airline industry around 1980, but RM now appears in hotels, car rentals, retailing, media and advertising auctions, e-commerce platforms, and ride-hailing.

A key motivation is that a *small* revenue improvement can create a *large* profit improvement when many costs are roughly fixed in the short run. Consider a simplified income statement. Suppose revenues are 1000, cost of goods sold (COGS) is 800, other expenses are 150, so net margin is 50. If RM increases revenue by only 2% (from 1000 to 1020) while COGS and other expenses remain unchanged, then net margin becomes 70. That is a 40% increase in net margin from only a 2% increase in revenue. The intuition is simple: revenue is the largest line item, and incremental revenue often flows disproportionately to the bottom line.

2 The Economic Engine Behind RM: Customer Segmentation

RM works because customers are not identical. **Customer segmentation** means dividing customers into groups with different willingness to pay and different purchasing behavior. Many RM practices are closely related to *price discrimination* in economics, but RM adds an OM perspective: limited capacity, uncertainty, and operational constraints.

Two common segmentation dimensions are:

Time-based differentiation. Some customers are willing to buy early (often at lower prices), while others buy late (often at higher prices). For example, retailers may sell at a regular price early in the season and mark down later.

Quality-based differentiation. Some customers pay for higher quality or convenience. Airlines sell multiple fare products; theme parks sell standard entry and premium "FastPass" options.

RM can be summarized as matching supply with demand in the "right" way: selling the right product to the right customer at the right time and price. Modern RM is also an intersection of demand data (collection and forecasting), supply design (capacity and service design), and analytics (capacity control, pricing, and algorithms).

3 Capacity-Based RM and the Role of Perishability

This chapter focuses on **capacity-based revenue management**, which is especially important for **fixed and perishable resources**. A resource is **perishable** if unused capacity cannot be stored for future sale. An airplane seat that departs empty cannot be sold tomorrow. A hotel room night that passes unsold is gone forever.

Airlines illustrate the operational pressure clearly: a typical airline may operate with about 73% of seats filled, while break-even might require about 70%. That thin margin means that the decision of allocating even a few seats across customer types can determine whether a flight is profitable.

We study two canonical airline RM problems:

1. **Two-class seat allocation** (also called capacity protection or booking limits).
2. **Overbooking** (selling more tickets than seats to offset no-shows).

Both can be solved using the same OM workhorse: the **newsvendor model** (also called the single-period inventory model), once we carefully define what "inventory" and "demand" mean in each context.

4 The Newsvendor Logic You Will Reuse

The newsvendor model is a one-shot decision under uncertainty. You choose an *inventory level* Q before observing demand D. If Q is too high, you face **overage** (leftover inventory). If Q is too low, you face **underage** (lost sales or shortages).

Two cost concepts drive the optimal decision:

Overstocking (overage) cost C_o: the marginal penalty when you increase inventory by one unit and end up with inventory exceeding demand.

Understocking (underage) cost C_u: the marginal penalty when you decrease inventory by one unit and end up with demand exceeding inventory.

Under mild assumptions, the optimal Q^* satisfies the **critical fractile rule**:

$$\Pr(D \le Q^*) \;=\; \frac{C_u}{C_u + C_o}. \tag{1}$$

If D is Normal with mean μ and standard deviation σ, then

$$Q^* = \mu + \sigma z^*, \qquad \text{where } \Phi(z^*) = \frac{C_u}{C_u + C_o}, \tag{2}$$

and $\Phi(\cdot)$ is the standard normal CDF.

A practical OM lesson is that the difficult part is usually *not* the quantile calculation; it is identifying D, Q, C_o, and C_u correctly.

5 Application 1: Two-Class Allocation (Protection Levels and Booking Limits)

Setting and Key Definitions

An airline sells tickets for a flight with fixed capacity C (seats). Customers are of two types:

- **Leisure travelers** are price sensitive and book early if the fare is discounted.
- **Business travelers** are less price sensitive and tend to book late, often paying a higher fare.

Assume the airline offers two fares:

$$f_1 > f_2,$$

where f_1 is the full (high) fare for business travelers and f_2 is the discounted (low) fare for leisure travelers.

A common simplifying assumption (used for teaching and often approximately true) is:

1. There is sufficient leisure demand so that any discounted seats offered will sell.
2. Business demand is uncertain and limited, and can be modeled as a random variable D_b.

The operational decision is a **capacity control** decision:
Protection level Q: the number of seats reserved (protected) for business travelers.
Booking limit for leisure: $C - Q$, the maximum number of discounted seats sold in advance.
These satisfy:

$$(C - Q) + Q = C.$$

The central trade-off is:

- If Q is too small, you sell too many low-fare seats early and later reject high-fare business demand.
- If Q is too large, you protect too many seats and may fly with empty seats if business demand is low.

Newsvendor Reformulation

To apply the newsvendor model, we must decide what plays the role of "demand" and "inventory."
Here:

$$\text{Demand } D = D_b \quad \text{(business demand)}, \qquad \text{Inventory } Q = \text{protected seats for business.}$$

Now compute the costs by marginal reasoning.

If you protect *one more* seat and business demand is not enough to use it, that seat will go empty *instead of* being sold at the discount price (which would have sold for sure). The loss is f_2. Thus

$$C_o = f_2.$$

If you protect *one fewer* seat and business demand would have used it, you sell that seat at f_2 early but could have sold at f_1 later. The loss is the fare difference:

$$C_u = f_1 - f_2.$$

Therefore the optimal protection level Q^* satisfies

$$\Pr(D_b \le Q^*) = \frac{f_1 - f_2}{(f_1 - f_2) + f_2}. \tag{3}$$

Worked Example 1 (Two-Class Allocation)

A flight has capacity $C = 335$ seats. Fares are $f_1 = \$7950$ and $f_2 = \$5250$. Business demand is

$$D_b \sim \mathcal{N}(\mu = 25, \sigma = 5).$$

First compute costs:

$$C_o = f_2 = 5250, \qquad C_u = f_1 - f_2 = 7950 - 5250 = 2700.$$

Then the critical fractile is

$$\frac{C_u}{C_u + C_o} = \frac{2700}{2700 + 5250} \approx 0.339.$$

Find z^* such that $\Phi(z^*) \approx 0.339$. From a normal table, $z^* \approx -0.41$. Hence

$$Q^* = \mu + \sigma z^* = 25 + 5(-0.41) \approx 22.95 \approx 23 \text{ seats.}$$

So the airline should protect about 23 seats for business travelers, and sell

$$C - Q^* = 335 - 23 = 312$$

discounted seats in advance.

Performance Metrics and Interpretation

Once Q is chosen, three standard performance quantities help interpret outcomes:

Leftover:

$$L = (Q - D)^+ = \max\{Q - D, 0\}.$$

In this problem, leftover corresponds to *protected seats that go unused*, i.e., empty seats caused by over-protection.

Sales:

$$S = \min\{Q, D\}.$$

Here sales is the number of full-fare tickets actually sold to business travelers.

Lost sales:

$$LS = (D - Q)^+ = \max\{D - Q, 0\}.$$

Here lost sales is the number of business travelers turned away because protection was insufficient.

Expected revenue is the sum of expected high-fare revenue and certain low-fare revenue. If leisure demand is sufficient, then the airline sells $C - Q$ seats at f_2 for sure, and sells S seats at f_1.

6 Application 2: Overbooking (No-Shows as "Demand")

Setting and Key Definitions

Overbooking means accepting more reservations than the physical number of seats. Airlines do this because some ticketed passengers do not show up.

Define:

- Physical seat capacity: C.
- Overbooking level: $X \geq 0$, meaning the airline sells $C + X$ tickets.

- Random number of no-shows: N (ticketed passengers who do not appear).

If the airline sells $C + X$ tickets and N passengers do not show, then the number who show up is $(C + X) - N$. If this exceeds C, the airline must deny boarding to $((C + X) - N) - C = X - N$ passengers. In practice, airlines seek volunteers first, but when that fails, some passengers are involuntarily "bumped."

Let:

$$p = \text{ticket fare collected per passenger}, \qquad b = \text{total cost per bumped passenger}.$$

The bumping cost b can include direct compensation and rebooking, plus implicit customer ill-will.

Newsvendor Reformulation

This time the trick is conceptual: the uncertainty is in *no-shows*, and more no-shows are actually *good* for avoiding bumping and for filling seats when overbooking.

We set:

$$\text{Inventory } Q = X \quad \text{(overbooked tickets)}, \qquad \text{Demand } D = N \quad \text{(no-shows)}.$$

This choice follows the newsvendor "principle" that higher demand should improve revenue: here, higher N (more no-shows) reduces bumping and increases the feasibility of selling extra tickets.

Now identify marginal costs.

If X is *too large* relative to N, then $X > N$ and some passengers must be bumped. Increasing X by one increases bumping by one in that scenario. The net penalty of bumping an extra passenger is the bumping cost minus the fare already collected:

$$C_o = b - p.$$

If X is *too small* relative to N, then there are more no-shows than overbooked tickets, meaning empty seats remain that could have been filled by selling additional tickets without any bumping. The marginal opportunity loss is the fare p:

$$C_u = p.$$

Thus, the optimal overbooking level X^* satisfies

$$\Pr(N \le X^*) = \frac{p}{p + (b - p)} = \frac{p}{b}. \tag{4}$$

Worked Example 2 (Overbooking)

A flight has $C = 100$ seats. The number of no-shows is

$$N \sim \mathcal{N}(\mu = 20, \sigma = 10).$$

Ticket fare is $p = \$105$. Total cost of denied boarding is $b = \$405$.

Compute costs:

$$C_o = b - p = 405 - 105 = 300, \qquad C_u = p = 105.$$

Critical fractile:

$$\frac{C_u}{C_u + C_o} = \frac{105}{105 + 300} \approx 0.259.$$

Find z^* with $\Phi(z^*) \approx 0.259$, giving $z^* \approx -0.65$. Then

$$X^* = \mu + \sigma z^* = 20 + 10(-0.65) = 13.5 \approx 14.$$

So the airline should overbook by about 14 tickets, selling $C + X^* \approx 114$ tickets.

Service Level and Performance Metrics

In overbooking, the standard newsvendor performance quantities have concrete meanings.

Let X be overbooking and N be no-shows.

Leftover inventory:

$$L = (X - N)^+.$$

Here L equals the number of **bumped** passengers, because bumping occurs when overbooking exceeds no-shows.

Sales:

$$S = \min\{X, N\}.$$

Here S equals the number of **additional passengers served** due to overbooking. Intuitively, you can only fill extra seats up to the number of no-shows, but if no-shows are very low, you cannot serve all extra ticket holders because you must bump some.

Lost sales:

$$LS = (N - X)^+.$$

In this reframing, lost sales correspond to **empty seats** that remain because no-shows exceeded the overbooking level.

A commonly reported service measure is the probability of bumping at least one passenger. Because N is integer-valued in reality, a careful expression is

$$\Pr(\text{bump}) = \Pr(N \leq X - 1),$$

since if $N = X$, then exactly C passengers show and nobody is bumped.

7 Common Pitfalls (and How to Avoid Them)

The same mistakes appear repeatedly when students first learn RM with the newsvendor framework.

First, many errors come from confusing what is "demand" and what is "inventory." In two-class allocation, demand is business demand D_b and inventory is protected seats Q. In overbooking, demand is no-shows N and inventory is overbooking level X. If you pick these incorrectly, your costs and critical fractile will be wrong even if your algebra is perfect.

Second, students often compute C_o and C_u using intuition rather than the formal definition. A safer method is to ask: "What is the marginal penalty if inventory ends up one unit above demand?" for C_o, and "What is the marginal penalty if demand ends up one unit above inventory?" for C_u.

Third, be cautious about "revenue" versus "profit." In these RM settings, we often maximize expected *revenue* because many costs are fixed or not modeled. In overbooking, we explicitly include bumping costs, so the objective behaves more like expected profit from the overbooking decision. Always read what the objective includes.

Fourth, do not forget integer effects. Bumping probability often uses $X - 1$ rather than X due to discrete passenger counts, even when a normal approximation is used for convenience.

Finally, a normal model is a modeling convenience, not a law of nature. Real demand and no-show distributions can be skewed, truncated, or event-driven. In practice, analysts validate distributions, use richer forecasting, and add policy constraints. The OM value of this chapter is that the *structure* of the trade-off remains the same.

8 A Unifying Summary

Capacity-based revenue management is an OM application of the newsvendor principle: make a one-shot capacity decision under uncertainty to balance the cost of being too aggressive against the cost of being too conservative.

In two-class allocation, the decision is a *protection level* for high-fare demand, with overage cost equal to the forgone low fare and underage cost equal to the fare difference.

In overbooking, the decision is an *overbooking level*, where the random driver is no-shows, overage cost is the net bumping penalty, and underage cost is the lost fare opportunity.

In both cases, the optimal policy is a quantile:

$$\Pr(D \le Q^*) = \frac{C_u}{C_u + C_o},$$

and the most important OM skill is translating the real operational situation into correct definitions of D, Q, C_u, and C_o.

Operations Management: Price-Based Revenue Management (A Concise Chapter for Junior Undergraduates)

1 Why Revenue Management Belongs in Operations Management

Operations management studies how an organization designs and runs processes to deliver goods and services effectively. Although pricing is often associated with marketing, it is also a core *operations* lever because price directly shapes demand, and demand must be matched with operational capacity (staffing, inventory, production slots, delivery windows, and service time).

Revenue Management (RM) is the practice of maximizing revenue by optimizing *product availability (capacity)* and *price*. Historically, RM was first developed in the airline industry in the 1980s, where seats are perishable: a seat unsold at takeoff becomes worthless. Today RM is used far beyond airlines, including retail markdowns, e-commerce dynamic pricing, and ride-hailing surge pricing.

It is useful to distinguish two broad families:

- **Capacity-based RM**: Decide which customers/products to accept when capacity is fixed and perishable (common in airlines, hotels, car rentals).
- **Price-based RM**: Adjust prices to influence demand and maximize revenue (common in retail, e-commerce, and on-demand services).

This chapter focuses on *price-based* RM.

2 Price-Based Revenue Management: Core Ideas and Definitions

Price optimization

Price optimization means choosing prices to maximize a performance metric, typically revenue (and sometimes profit). In its simplest form for a single product, we choose a price p to maximize

$$R(p) = p \cdot D(p),$$

where $D(p)$ is the demand at price p.

Price optimization has become more prevalent because firms now have more data for demand forecasting, lower computing cost for analytics, and greater flexibility to change prices quickly (for example, electronic shelf labels and online price updates).

Dynamic pricing

Dynamic pricing is the practice of changing prices over time (sometimes automatically) in response to changing market conditions such as demand, cost, time, location, inventory position, and

competitors' prices. Airlines, ride-hailing platforms, fuel stations, and large e-commerce retailers are well-known users of dynamic pricing.

A key intuition is that demand and supply conditions change over time. If prices are fixed while conditions change, the firm may experience either waste (idle capacity, excess inventory) or shortages (stockouts, long queues). Dynamic pricing aims to better match demand with supply, thus improving revenue.

Willingness-to-pay

Willingness-to-pay (WTP) is the maximum amount a customer is willing to pay for one unit of a product or service. A customer buys if and only if the price does not exceed their WTP.

Customers often have different WTP for the same product due to differences in income, preferences, urgency, perceived quality, brand value, and the availability of substitutes. This variation makes pricing powerful: a firm can sometimes increase revenue by choosing a price that balances *margin* (price per unit) and *volume* (units sold).

Substitutes and complements (in multi-product pricing)

When multiple products interact, the demand for one product can depend on the price of another.

- Two products are **substitutes** if a higher price of product 2 increases demand for product 1.
- Two products are **complements** if a higher price of product 2 decreases demand for product 1.

This interaction matters because pricing decisions can no longer be made independently product-by-product.

3 Pricing from Willingness-to-Pay Data

Sometimes a firm does not start with a smooth demand curve $D(p)$. Instead, it may have a distribution of willingness-to-pay levels in a market segment, obtained from surveys, past transactions, or experiments.

Suppose a firm must set a *single posted price* p. A customer with WTP $\geq p$ buys; otherwise, they do not. Therefore, demand at price p equals the number of customers whose WTP is at least p.

Procedure for discrete WTP data

If WTP takes discrete levels, a simple method is: (1) Sort WTP from high to low. (2) For each candidate price equal to a WTP level, compute demand as the cumulative frequency of customers with WTP at least that price. (3) Compute revenue $R = p \times$ demand for each candidate price and choose the maximum.

The reason we only need to check candidate prices at observed WTP levels is that between two adjacent WTP levels, demand does not change, so revenue increases linearly with price until hitting the next WTP threshold.

Worked Example 1: Pricing a textbook from WTP frequencies

A publisher is pricing a textbook. The estimated WTP distribution is shown below.

WTP (candidate price)	Frequency	Demand if price = WTP	Revenue = price × demand
80	10	10	800
70	12	22	1540
60	14	36	2160
50	28	64	3200
40	20	84	3360
30	10	94	2820
20	6	100	2000

The revenue-maximizing price is $p^* = 40$, yielding demand 84 and revenue 3360. The example illustrates the classic trade-off: high prices give high revenue per unit but fewer buyers; low prices attract more buyers but reduce revenue per unit.

Common pitfalls with WTP-based pricing

A frequent mistake is forgetting to sort WTP from high to low before taking cumulative sums. Another mistake is using *frequency at exactly that WTP* as demand, rather than using *all customers with WTP at least the price*. Finally, remember that this method assumes one posted price and no capacity constraints; if capacity is limited (for example, a concert hall), the optimal price may change because demand above capacity does not translate into additional sales.

4 Pricing with Demand Models: The Linear Demand Case

When historical sales data are available, it is common to estimate a functional relationship between price and demand. A widely used approximation is **linear demand**:

$$D(p) = a - bp,$$

where a and b are constants and typically $b > 0$. Here, b is the *price sensitivity*: increasing price reduces demand.

Revenue and the optimal price

Revenue is

$$R(p) = pD(p) = p(a - bp) = ap - bp^2.$$

This is a concave quadratic function in p when $b > 0$. A concave quadratic of the form

$$f(x) = -bx^2 + ax + c \quad \text{with } b > 0$$

is maximized at

$$x^* = \frac{a}{2b}.$$

Applying this to revenue gives the **revenue-maximizing price**:

$$p^* = \frac{a}{2b}.$$

This formula is especially valuable in operations settings because it gives a quick decision rule without advanced calculus.

Worked Example 2: A T-shirt with linear demand

A store estimates demand for a T-shirt as

$$D(p) = 120 - 20p.$$

Here $a = 120$ and $b = 20$, so

$$p^* = \frac{120}{2 \cdot 20} = 3.$$

Demand at this price is $D(3) = 120 - 60 = 60$ units, and revenue is

$$R(3) = 3 \times 60 = 180.$$

Why linear demand is popular (and when it fails)

Linear demand is popular because it is easy to estimate and communicate, often requiring limited data. For instance, simple regression on past price and sales can provide an estimated a and b.

However, real demand is not always linear. A key limitation is that linear demand implies *constant marginal effect*: every \$1 price increase reduces demand by the same amount b, regardless of the price level. In reality, customers may be more price sensitive at some price ranges than others. When the marginal impact changes with price (for example, demand drops sharply after a psychological threshold such as \$9.99 to \$10.99), a nonlinear model may be more appropriate.

A practical pitfall is using a linear model outside the price range where it was estimated. Linear models can predict negative demand at high prices or unrealistically high demand at very low prices. In practice, managers often restrict prices to a realistic interval.

5 Multi-Product Pricing with Interacting Demands

Many operational decisions involve multiple products whose demands interact. For example, two nearby beverage shops may compete, or a retailer may sell two similar product variants.

Consider two products with prices p_1 and p_2. A simple linear demand system is

$$D_1 = a_1 - b_{11}p_1 - b_{12}p_2, \qquad D_2 = a_2 - b_{21}p_1 - b_{22}p_2,$$

where the a's and b's are constants. Notice that the sign of cross-price terms matters. If increasing p_2 increases D_1, then D_1 must contain a *positive* coefficient on p_2, which corresponds to a *negative* b_{12} in the standardized form above (because the model is written with a minus sign).

Revenue from each product is

$$R_1(p_1, p_2) = p_1 D_1, \qquad R_2(p_1, p_2) = p_2 D_2.$$

Two decision modes: individual vs. joint pricing

There are two economically distinct ways to set prices.

Individual pricing means each product is priced to maximize its own revenue. This represents two separate firms (or two business units) acting independently. When demands interact, each firm reacts to the other's price, and the final outcome is an *equilibrium*: neither party wants to deviate given the other's price.

Joint pricing means a single decision-maker chooses both p_1 and p_2 to maximize total revenue $R_1 + R_2$. This can represent a merger, centralized pricing within one firm, or a coordinated agreement. Joint pricing typically increases *total* revenue because it accounts for cross-effects, but it may change how revenue is split across products or divisions.

A concrete example: coffee and tea on the same street

Suppose daily demands are estimated as

$$D_1(p_1, p_2) = 100 - p_1 + 0.5p_2 \quad \text{(coffee)},$$

$$D_2(p_1, p_2) = 120 + 0.5p_1 - p_2 \quad \text{(tea)}.$$

Coffee demand decreases with its own price p_1, and increases with tea price p_2. That is the signature of *substitutes*: when tea becomes more expensive, some customers switch to coffee, raising coffee demand.

Individual pricing and equilibrium intuition

Under individual pricing, each store chooses its price to maximize its own revenue given the other's price. The optimal "best response" prices can be shown (using the concave quadratic maximization rule) to satisfy

$$p_1^* = 50 + \frac{p_2^*}{4}, \qquad p_2^* = 60 + \frac{p_1^*}{4}.$$

Solving these simultaneously yields approximately

$$(p_1^*, p_2^*) \approx (69.3, 77.3),$$

with total revenue about 10,785.

The key intuition is strategic: because products are substitutes, each store has an incentive to lower price to attract demand away from the other. This competitive pressure tends to keep prices lower.

Joint pricing and coordination intuition

Under joint pricing, the merged entity maximizes

$$R_1(p_1, p_2) + R_2(p_1, p_2).$$

For the coffee-tea model above, total revenue simplifies to

$$R_1 + R_2 = -p_1^2 - p_2^2 + p_1p_2 + 100p_1 + 120p_2.$$

Optimizing yields the conditions

$$p_1^* = 50 + \frac{p_2^*}{2}, \qquad p_2^* = 60 + \frac{p_1^*}{2},$$

which lead to

$$(p_1^*, p_2^*) \approx (106.7, 113.3),$$

and total revenue about 12,133.3.

Compared to individual pricing, joint pricing produces *higher prices* and *higher total revenue*. This is not a paradox: coordination reduces internal price competition. Even if demand becomes smaller at higher prices, the combined revenue can increase because the firm is no longer "stealing customers from itself."

6 Revenue Distribution and Coordination: Total Gain Does Not Guarantee Everyone Wins

A subtle but important operations point is that maximizing *total* revenue does not automatically make every party better off. This matters in supply chains, alliances, franchising, and multi-division firms where separate stakeholders must agree to coordinate.

Consider two firms. Under separate pricing, revenues are $(1500, 1200)$ with total 2700. Under joint pricing, revenues become $(1400, 1600)$ with total 3000. Total revenue increases, but Firm 1 is worse off and may refuse to coordinate.

A standard remedy is **transfer payments** (revenue sharing or side payments). If Firm 2 transfers an amount x to Firm 1 under joint pricing, then Firm 1 receives $1400 + x$ and Firm 2 receives $1600 - x$. For both to accept the switch, we need

$$1400 + x \geq 1500 \quad \Rightarrow \quad x \geq 100,$$

$$1600 - x \geq 1200 \quad \Rightarrow \quad x \leq 400.$$

Any transfer $x \in [100, 400]$ makes both firms at least as well off as before, while preserving the higher total revenue. Operationally, this illustrates why contracts and coordination mechanisms matter: good pricing policies sometimes require *organizational* solutions, not just mathematical ones.

7 A Handy Formula Sheet for Two-Product Linear Demands

For two products with linear demands

$$D_1 = a_1 - b_{11}p_1 - b_{12}p_2, \qquad D_2 = a_2 - b_{21}p_1 - b_{22}p_2,$$

there are convenient equation systems that characterize optimal prices in the two modes.

Individual pricing

If each product is priced to maximize its own revenue, optimal prices satisfy

$$2b_{11}p_1 + b_{12}p_2 = a_1, \qquad b_{21}p_1 + 2b_{22}p_2 = a_2.$$

Joint pricing

If prices are chosen to maximize total revenue $R_1 + R_2$, optimal prices satisfy

$$2b_{11}p_1 + (b_{12} + b_{21})p_2 = a_1, \qquad (b_{12} + b_{21})p_1 + 2b_{22}p_2 = a_2.$$

A common pitfall is sign confusion when mapping a demand equation into the standard form. For example, if the estimated demand is $D_1 = 100 - p_1 + 0.5p_2$, then in the standardized form $D_1 = a_1 - b_{11}p_1 - b_{12}p_2$ we have $a_1 = 100$, $b_{11} = 1$, and $b_{12} = -0.5$ because $-b_{12}p_2$ must equal $+0.5p_2$.

8 Common Pitfalls and Good Habits

Students and practitioners often make the same types of mistakes when first learning price-based RM.

First, it is easy to forget what objective is being optimized. This chapter mostly maximizes *revenue*, not profit. If unit costs are significant, the firm may want to maximize profit $\pi(p) = (p - c)D(p)$ instead of $pD(p)$. The optimal price can differ.

Second, equilibrium in individual pricing is not the same as joint optimality. Equilibrium reflects decentralized incentives; joint optimality reflects centralized coordination.

Third, models are only as good as their assumptions. Linear demand is helpful, but it must be used within reasonable price ranges, and it should be re-estimated when the environment changes (new competitors, new substitutes, seasonality, promotions, or changes in customer mix).

Finally, dynamic pricing can create customer pushback if customers perceive the system as unfair or unpredictable. In practice, firms often add guardrails (maximum change per hour/day, transparent reasons for changes, or loyalty guarantees) to balance revenue with customer trust.

9 Conclusion

Price-based revenue management is an operations tool for matching supply with demand by choosing prices intelligently. The two main approaches in this chapter are (i) pricing from willingness-to-pay distributions by checking candidate prices and (ii) pricing from demand models, especially linear demand, where revenue becomes a simple concave quadratic.

In multi-product settings, demand interactions create strategic effects. Pricing individually can lead to competitive pressure and lower prices, while pricing jointly can increase total revenue by accounting for cross-effects. However, coordination raises an additional operational issue: how the total gains are distributed, which often requires contracts and transfer mechanisms.

These ideas form a foundation for more advanced topics such as markdown optimization, dynamic pricing algorithms, bundling, and behavioral effects (for example, the decoy effect), all of which extend the same core principle: use data and models to set prices that improve operational and financial performance.

Operations Management: Pricing and Supply Chain Management (A Concise Chapter)

ISOM 2700-inspired notes (junior undergraduate level)

1 Why Operations Management Cares About Pricing and Supply Chains

Operations management (OM) studies how organizations design and run processes that transform inputs (materials, labor, information) into outputs (products and services). A useful way to summarize OM is: *make good decisions under constraints*. Those constraints include limited capacity, uncertain demand, time pressure, and the fact that firms rarely operate alone.

Two OM themes come together naturally:

1. **Revenue management and pricing:** choosing prices to influence demand and maximize revenue (or profit).
2. **Supply chain management (SCM):** coordinating multiple firms and activities so that supply meets demand profitably.

This chapter introduces a small but important toolkit from each theme and shows how the ideas connect.

2 Multi-Product Pricing with Linear Demand

Pricing is an operational decision because it directly shapes demand, which then determines how much you need to produce, stock, ship, and staff. When a firm sells multiple products, pricing decisions interact: lowering the price of one product may increase or decrease the demand for another.

A simple demand model

A widely used first model is the **linear demand model** for two products:

$$D_1 = a_1 - b_{11}p_1 - b_{12}p_2, \tag{1}$$

$$D_2 = a_2 - b_{21}p_1 - b_{22}p_2. \tag{2}$$

Here:

- p_1, p_2 are the decision variables (prices).
- D_1, D_2 are the resulting demands (quantities sold).
- a_1, a_2 are **base demand** terms (how much demand would be if prices were zero in this simplified model).

- b_{11}, b_{22} are **own-price sensitivity** coefficients; usually positive so that higher price reduces demand.
- b_{12}, b_{21} are **cross-price sensitivity** coefficients; they capture substitution or complementarity.

Interpreting cross effects. Because the model is written as "minus $b_{12}p_2$", the *sign of* b_{12} *matters*:

- If $b_{12} > 0$, increasing p_2 decreases D_1 (product 2 is a *complement* for product 1 in this sign convention).
- If $b_{12} < 0$, increasing p_2 increases D_1 (product 2 is a *substitute* for product 1: making product 2 more expensive pushes customers to product 1).

The same logic applies to b_{21}.

Revenue and the core decision problem

Revenue from each product is

$$R_1(p_1, p_2) = p_1 D_1, \qquad R_2(p_1, p_2) = p_2 D_2,$$

and total revenue is $R(p_1, p_2) = R_1 + R_2$.

Two settings matter operationally:

1. **Individual pricing (non-coordinated):** each product (or each manager) chooses its price to maximize its own revenue, treating the other price as "given."
2. **Joint pricing (coordinated):** prices are chosen to maximize total revenue of the firm (or supply chain) as a whole.

Although the words sound similar, the outcomes can differ because cross-price effects create externalities: one product's price changes can help or hurt the other product's revenue.

Main formulas: first-order conditions as linear systems

The linear demand structure makes the optimality conditions especially neat.

Individual pricing. If each product sets its own price to maximize its own revenue, the optimal prices solve the system

$$2b_{11}p_1 + b_{12}p_2 = a_1, \tag{3}$$
$$b_{21}p_1 + 2b_{22}p_2 = a_2. \tag{4}$$

Joint pricing. If prices are set jointly to maximize $R_1 + R_2$, the optimal prices solve

$$2b_{11}p_1 + (b_{12} + b_{21})p_2 = a_1, \tag{5}$$
$$(b_{12} + b_{21})p_1 + 2b_{22}p_2 = a_2. \tag{6}$$

These are linear equations in (p_1, p_2), so the solution is straightforward using substitution or matrix methods.

Worked Example 1: Tea and coffee (individual vs. joint pricing)

Suppose demands are

$$D_1(p_1, p_2) = 800 - 2p_1 + 0.5p_2, \qquad D_2(p_1, p_2) = 600 + 0.5p_1 - 2p_2.$$

To match the form in (1)–(2), rewrite D_1 as

$$D_1 = 800 - 2p_1 - (-0.5)p_2,$$

so $a_1 = 800$, $b_{11} = 2$, $b_{12} = -0.5$. Similarly, $a_2 = 600$, $b_{21} = -0.5$, $b_{22} = 2$.

Individual pricing. Using (3)–(4):

$$2(2)p_1 + (-0.5)p_2 = 800, \qquad (-0.5)p_1 + 2(2)p_2 = 600,$$

which simplifies to

$$4p_1 - 0.5p_2 = 800, \qquad -0.5p_1 + 4p_2 = 600.$$

Solving gives approximately

$$p_1 \approx 222.2, \qquad p_2 \approx 177.8.$$

Demands become

$$D_1 \approx 800 - 2(222.2) + 0.5(177.8) = 444.5, \qquad D_2 \approx 600 + 0.5(222.2) - 2(177.8) = 355.5.$$

Revenues:

$$R_1 \approx 222.2 \times 444.5 \approx 9.88 \times 10^4, \qquad R_2 \approx 177.8 \times 355.5 \approx 6.32 \times 10^4.$$

Joint pricing. Using (5)–(6), we replace the cross term by $b_{12} + b_{21} = -1$:

$$4p_1 - p_2 = 800, \qquad -p_1 + 4p_2 = 600.$$

Solving gives approximately

$$p_1 \approx 253.3, \qquad p_2 \approx 213.3.$$

Demands become

$$D_1 \approx 800 - 2(253.3) + 0.5(213.3) = 400, \qquad D_2 \approx 600 + 0.5(253.3) - 2(213.3) = 300.$$

Revenues:

$$R_1 \approx 253.3 \times 400 \approx 1.01 \times 10^5, \qquad R_2 \approx 213.3 \times 300 \approx 6.40 \times 10^4.$$

Total revenue increases under joint pricing. The operational intuition is that coordinated pricing internalizes cross-product effects, while individual pricing ignores them.

Common pitfalls in two-product pricing

A frequent mistake is **mismatching coefficients** when translating a problem statement into the template (1)–(2). In particular, if a demand equation contains "$+0.5p_2$", then b_{12} is *negative* because the model subtracts $b_{12}p_2$.

Another common pitfall is **forgetting to check the implied demands**. A linear model can produce negative demand for extreme prices. In real applications one would add constraints (e.g., $D_i \geq 0$) or interpret negative demand as "zero sales." In introductory problems, the computed optimal prices usually keep demands positive, but it is still good practice to verify.

Finally, it is easy to confuse **revenue** with **profit**. Revenue ignores costs. In many operations settings (especially with meaningful unit costs or capacity costs), profit maximization is the correct objective. Revenue management models often begin with revenue because it isolates the pricing–demand relationship, but the next step is typically to incorporate costs.

3 Beyond Optimization: Practical Pricing Tactics

Not all pricing decisions are solved by calculus or linear algebra. Many everyday pricing tactics are operationally motivated: they are ways to manage inventory, capacity, and customer behavior.

Markdown pricing

A **markdown** is a permanent reduction in price (unlike a temporary promotion). Markdowns help when inventory risks losing value. There are three classic drivers:

1. **Obsolescence:** technology products lose value when a new generation arrives.
2. **Seasonality:** winter items are hard to sell once the season ends.
3. **Deterioration:** food and other perishables must be sold quickly.

Cannibalization effect (a key concern). Markdowns can backfire if customers learn to wait. If many customers delay purchases because they expect a future discount, the firm loses full-price sales. Operationally, this is a demand-timing problem: your current pricing policy shapes not only *how much* people buy, but *when* they buy.

Bundle pricing

Bundle pricing means grouping multiple products and selling the bundle at one price instead of pricing each item separately. The core intuition is **pooling**: when customers value different items differently, bundling can reduce the risk that any one customer refuses the purchase.

Suppose each customer has a **willingness-to-pay** (WTP), the maximum amount they would pay. If a customer pays price p for an item they value at v, their **consumer surplus** is

$$\text{Surplus} = v - p.$$

Worked Example 2: Two-channel cable bundle

Two customers have these WTP values:

	ESPN	History
Sports lover	\$10	\$3
History lover	\$3	\$10

Separate pricing at \$9 per channel. The sports lover buys ESPN (surplus $10-9=1$) but not History. The history lover buys History (surplus $10-9=1$) but not ESPN. Revenue is $9+9=18$ and total surplus is $1+1=2$.

Bundling at \$11 for both channels. Each customer values the bundle at $10+3=13$, so both buy. Revenue is $11+11=22$. Each customer surplus is $13-11=2$, so total surplus is 4.

This example shows a "win–win" possibility: properly designed bundling can increase both firm revenue and consumer surplus.

Psychological pricing: the decoy effect and the number 9

Real customers do not always behave like perfectly rational calculators, so firms often use behavioral patterns in pricing.

The magic number 9. Prices like \$49, \$79, or \$99 can sell better than nearby prices because consumers overweight the leftmost digit. The difference between \$299 and \$300 feels larger than one dollar to many shoppers.

Decoy effect. The **decoy effect** occurs when adding a third option changes preferences between two existing options, even if the third option is rarely chosen. The typical decoy is **asymmetrically dominated**: it is clearly worse than one option, but only partially worse than the other. It acts like a comparison anchor that makes the intended option look better.

Operationally, these tactics matter because product menus and price lists affect demand composition, which then affects forecasting, inventory, and capacity decisions.

4 Supply Chain Management: Thinking Beyond One Firm

Most of the earlier OM tools can be applied within a single organization, such as a retailer optimizing inventory or a manufacturer planning production. In reality, however, a product reaches a customer through many connected entities.

Definition and structure

A **supply chain** is the system of organizations, people, activities, information, and resources involved in moving a product or service from supplier to customer.

A typical chain includes suppliers, manufacturers, distributors, retailers, and end customers. The direction toward raw materials is called **upstream**; the direction toward end customers is **downstream**.

The goal: match supply and demand profitably

At a high level, supply chains aim to match supply and demand in a profitable way. In practical terms, this means delivering:

the right product, at the right price, in the right store, in the right quantity, to the right customer, at the right t

This statement looks like a slogan, but it is operationally meaningful: each "right" corresponds to a decision area (product design, pricing, distribution, inventory, segmentation, and timing).

Three flows: material, cash, and information

Supply chains have three main flows:

1. **Material flow:** usually travels downstream (raw materials → customer).
2. **Cash flow:** usually travels upstream (customer payment → suppliers).
3. **Information flow:** travels both ways (forecasts, orders, production plans, delays, and quality signals).

A central lesson of SCM is that improving *information flow* can drastically improve the other two.

5 Coping with Complexity: The AAA Strategies

Modern supply chains are challenged by shocks (pandemics, natural disasters, strikes, wars), policy changes (tariffs, export restrictions), and rapid technology cycles. A useful framework summarizes successful supply chain strategies as the **AAA strategies**:

Agility. **Agility** is the ability to respond quickly to short-term changes in supply chain conditions. A firm is agile if it can adjust procurement, inventory, and delivery rapidly when demand spikes or supply is disrupted.

Adaptability. **Adaptability** is the ability to adjust to long-term structural shifts, such as new technologies (e.g., electric vehicles), changing customer preferences, or permanent changes in supply markets. Adaptability is about redesigning the supply chain, not just reacting within it.

Alignment. **Alignment** is coordination among supply chain partners so that actions that are good for one party are not harmful to the chain as a whole. Alignment requires sharing information, risks, and rewards in a way that makes cooperation stable.

6 Coordination, Contracts, and Double Marginalization

When multiple firms interact, each firm naturally optimizes its own objective. Without careful coordination, this can harm overall performance.

Contracts

A **contract** in SCM is a set of rules that control or modify goods and cash flows between partners (for example, wholesale prices, revenue-sharing, buyback agreements, or quantity flexibility). For a contract to work, it should be **verifiable** (observable outcomes can be checked) and **enforceable** (there are consequences for breaking it).

Double marginalization

Double marginalization describes the inefficiency that can occur when each firm in a supply chain adds its own markup. Each stage sets decisions (often price or quantity) to maximize its own profit, which typically results in a final price that is too high and total supply chain profit that is too low compared with a coordinated solution.

The key intuition is simple: *what is optimal locally may be suboptimal globally.* This mirrors the difference between individual and joint pricing in multi-product problems: coordination internalizes externalities.

7 Variability and the Bullwhip Effect

Even when average demand is stable, supply chains can experience large swings in orders and production. This volatility increases costs because capacity and inventory must be sized for peaks, not averages.

Inventory balance equation

For any player in a supply chain, a basic accounting identity links inventory, ordering, and demand. Let

- I_t be the inventory at the start of period t,
- Q_t be the order (or production) quantity placed in period t,
- D_t be the demand faced in period t,
- I_{t+1} be ending inventory (start of next period).

Then

$$Q_t + I_t - D_t = I_{t+1}. \tag{7}$$

This is not an assumption; it is a definition.

A simple ordering rule and amplification

To illustrate variability, consider a simplistic rule:

$$I_{t+1} = D_t.$$

In words, each firm tries to end the period with inventory equal to current demand. Substituting into (7) gives:

$$Q_t = 2D_t - I_t. \tag{8}$$

This rule is not necessarily optimal, but it is intuitive enough that it captures an important mechanism: small demand changes can cause large order changes upstream.

Bullwhip effect (definition and intuition)

The **bullwhip effect** is the phenomenon that *demand variability increases as you move upstream* in the supply chain. Formally, the variability of orders at one level is greater than the variability of demand observed at the next downstream level.

Why it happens is partly mathematical (inventory corrections propagate and amplify) and partly behavioral/organizational (batch ordering, promotions, overreaction, and delays in information). The result is operational pain: unstable production schedules, overtime, idle capacity, excessive inventory, and stockouts.

A standard remedy is **collaborative planning, forecasting, and replenishment**: partners share more accurate demand information and coordinate replenishment policies so upstream decisions are based on true consumption rather than noisy order signals.

Common pitfalls when reasoning about the bullwhip

One mistake is to treat upstream demand as "the same as customer demand." In practice, upstream firms often see *orders*, not true customer sales, and orders are a distorted signal. Another mistake is to assume variability is purely caused by "random customers." In many cases, variability is *endogenous*: it is created by policies, incentives, and delays inside the chain.

8 Modern Context: Shocks, Tariffs, and Technology

Recent years have highlighted how supply chains interact with geopolitics and technology. The COVID-19 pandemic created disruptions in trade, logistics, and workforce availability, revealing vulnerabilities and motivating firms to hold more safety stock, diversify suppliers, and redesign networks.

Tariffs and trade policy changes add another layer: even products assembled domestically can contain components sourced globally, so costs and lead times depend on complex cross-border flows.

Technology is also reshaping SCM. Automation can reduce labor dependence in manufacturing; data analytics improves visibility; and AI tools can support forecasting, routing, and risk detection. These developments do not remove supply chain complexity, but they can make it more manageable when combined with sound operational principles.

9 Summary

Operations management connects pricing decisions and supply chain decisions because both determine how supply meets demand. Linear demand models provide a clean introduction to multi-product pricing and show why coordination (joint optimization) can outperform isolated decision-making. Practical pricing tactics such as markdowns, bundling, and menu design influence not only revenue but also operational outcomes like inventory risk.

Supply chain management broadens the OM viewpoint from a single firm to a network. The supply chain goal is to match supply and demand profitably, supported by material, cash, and information flows. Modern supply chains face uncertainty and rapid change, motivating the AAA strategies of agility, adaptability, and alignment. When alignment fails, double marginalization and the bullwhip effect illustrate how local optimization can harm system performance. The central OM lesson is that good operations require both smart models and smart coordination.

Coordinating a Supply Chain with Risk-Sharing Contracts

A Concise Operations Management Chapter (Junior Undergraduate Level)

1 Why Supply Chains Need Coordination

Operations management is not only about forecasting demand or setting inventory levels; it is also about designing *incentives* so that different firms in a supply chain make decisions that are good for the system as a whole. A classic coordination problem appears when a supplier sells to a retailer who faces uncertain demand. The retailer chooses how much to order before demand is realized. If the retailer orders too little, the supply chain misses sales; if the retailer orders too much, inventory is left over and must be salvaged at a low value.

The key challenge is that a supply chain is often *decentralized*: each firm maximizes its own expected profit. When incentives are misaligned, the resulting decisions can be inefficient even if everyone is behaving rationally. This chapter explains (i) why a simple *wholesale price contract* can lead to inefficiently low orders, and (ii) how a *revenue-sharing contract* can coordinate the supply chain by sharing risk and reward.

2 Core Setting and Key Definitions

Consider a two-stage supply chain:

$$\text{Supplier} \longrightarrow \text{Retailer} \longrightarrow \text{Customers}.$$

The retailer orders quantity Q before the selling season. Customer demand D is random. Units not sold are salvaged at a salvage value.

We will use the following parameters throughout:

- c: supplier's unit production cost.
- p: retail selling price (revenue per unit sold to customers).
- s: salvage value (revenue per leftover unit at the end of season), with $s < p$.
- w: wholesale price charged by the supplier to the retailer under a wholesale-price contract.
- y: revenue-sharing percentage paid to the supplier per unit sold under a revenue-sharing contract, with $0 \leq y \leq 1$.

Supply-chain optimum vs. retailer optimum

Supply-chain optimum is the order quantity Q^{SC} that maximizes the *total* expected profit of supplier plus retailer, treating the supply chain as one integrated firm and ignoring internal transfers (payments between supplier and retailer).

Retailer optimum is the order quantity Q^R that maximizes the retailer's expected profit given the contract terms.

These two generally differ under common contracts; coordination aims to make $Q^R = Q^{SC}$.

Double marginalization (intuition)

Double marginalization occurs when multiple firms in a vertical chain each add their own markup (profit margin). The supplier marks up above cost when charging w, and the retailer marks up above w when selling to customers. Because each layer optimizes locally, the final decision (here, the order quantity) can be too conservative, shrinking the "total profit pie".

3 The Newsvendor Logic: Critical Fractile and Order Quantity

The ordering decision in this chapter is a standard **newsvendor** decision: choose Q once, then random demand D occurs. The newsvendor solution can be explained using two costs:

Overstocking and understocking costs

Overstocking cost C_o is the economic loss from ordering one extra unit that ends up unsold. If one more unit is leftover, you paid the relevant acquisition cost but only recover salvage value.

Understocking cost C_u is the economic loss from ordering one unit too few when there is demand for it. If one more unit had been available, you would have earned the per-unit margin associated with a sale.

Different decision-makers face different C_o and C_u, because their payoffs depend on the contract.

Critical fractile rule

The optimal Q satisfies the **critical fractile** condition:

$$\Pr(D \le Q^*) \;=\; \frac{C_u}{C_u + C_o}. \tag{1}$$

Intuitively, if understocking is very painful (large C_u), you aim for a high service level $\Pr(D \le Q)$ and order more. If overstocking is very painful (large C_o), you order less.

Normal-demand shortcut

When $D \sim \mathcal{N}(\mu, \sigma)$, we can write the optimal order quantity as

$$Q^* \;=\; \mu + \sigma z, \tag{2}$$

where z is chosen so that

$$\Phi(z) \;=\; \frac{C_u}{C_u + C_o}. \tag{3}$$

Here, $\Phi(\cdot)$ is the standard normal cumulative distribution function. The practical workflow is simple: compute C_u and C_o, compute the critical fractile, find z from a normal table (or calculator), then compute Q^*.

4 Wholesale Price Contract: Why It Fails to Coordinate

Contract definition

Under a **wholesale price contract**, the supplier charges the retailer w per unit ordered. The retailer chooses Q and pays wQ to the supplier. The retailer then sells to customers at price p and salvages leftovers at s.

The decision-maker is the retailer, so the retailer's perceived costs determine Q.

Supply-chain optimum (benchmark)

If the supply chain were integrated, the relevant unit cost would be the true production cost c, and the relevant revenues would be p if sold and s if leftover. Therefore,

$$C_o^{SC} = c - s, \qquad C_u^{SC} = p - c, \qquad \Pr(D \le Q^{SC}) = \frac{p-c}{(p-c)+(c-s)} = \frac{p-c}{p-s}.$$

This benchmark depends only on (p, c, s) and the demand distribution, not on the contract.

Retailer's optimum under wholesale price

Under wholesale price w, the retailer effectively pays w to acquire a unit. Thus,

$$C_o^{R} = w - s, \qquad C_u^{R} = p - w, \qquad \Pr(D \le Q^{R}) = \frac{p-w}{(p-w)+(w-s)} = \frac{p-w}{p-s}.$$

Compare the critical fractiles:

$$\frac{p-w}{p-s} \quad \text{vs.} \quad \frac{p-c}{p-s}.$$

If the supplier wants nonnegative margin at the wholesale stage, then typically $w > c$. In that case $p - w < p - c$, so

$$\frac{p-w}{p-s} < \frac{p-c}{p-s}.$$

A smaller critical fractile implies a smaller z, and therefore a smaller order quantity Q^R. In words: **under a wholesale price contract, the retailer orders less than the supply-chain optimum, leading to underordering (underproduction) and lower total profit**.

This is the fundamental limitation emphasized in the lecture material: changing w cannot fully fix the problem as long as $w > c$ is required for the supplier to earn margin purely from wholesale pricing.

5 Revenue Sharing Contract: A Risk-Sharing Solution

Contract definition

A **revenue-sharing contract** uses two levers:

1. The retailer pays wholesale price w per unit ordered (as before).
2. For each unit sold to customers, the retailer pays the supplier a fraction y of the retail price p. Equivalently, the retailer keeps $p(1-y)$ per unit sold and the supplier gets py per unit sold.

The retailer still chooses Q, so coordination must come from changing the retailer's incentives.

Retailer's newsvendor costs under revenue sharing

From the retailer's perspective:

$$\text{Revenue per sold unit} = p(1-y), \qquad \text{Acquisition cost} = w, \qquad \text{Salvage per leftover} = s.$$

So,

$$C_o^R = w - s, \qquad C_u^R = p(1-y) - w.$$

The retailer's critical fractile becomes

$$\Pr(D \le Q^R(w,y)) = \frac{p(1-y)-w}{\big(p(1-y)-w\big)+(w-s)} = \frac{p(1-y)-w}{p(1-y)-s}. \tag{4}$$

This is the key: by adjusting y, we can change the retailer's perceived understocking cost (the gain from selling one more unit) without changing the salvage logic.

Coordination idea: match the critical fractile

To achieve the supply-chain optimum, we want the retailer to choose $Q^R(w,y) = Q^{SC}$. Under continuous demand, it is enough to match the critical fractile:

$$\frac{p(1-y)-w}{p(1-y)-s} \;=\; \frac{p-c}{p-s}. \tag{5}$$

Solving this equation for y in terms of w yields a family of coordinating contracts. One convenient closed form (from the slides) is:

$$y \;=\; \frac{c(p-s)-(p-s)w}{p(c-s)}. \tag{6}$$

Any (w,y) pair satisfying (5) (or equivalently (6)) makes the retailer order the supply-chain optimal quantity, so the decentralized supply chain achieves the centralized outcome.

Important structural feature

For y to be feasible (between 0 and 1), the lecture result emphasizes that we typically need

$$s < w < c. \tag{7}$$

This may look surprising: it means the supplier may *lose money* on the wholesale stage (selling below production cost c), but is compensated through the revenue share py from actual sales. Economically, the supplier is now sharing demand risk with the retailer: if sales are high, the supplier benefits; if sales are low, the supplier earns less.

6 Worked Example 1: Wholesale Contract vs. Supply-Chain Optimum (UV Sunglasses)

This example follows the session material closely.

Data

Supplier production cost: $c = 35$.
Retail price: $p = 115$.
Salvage value: $s = 25$.
Demand: $D \sim \mathcal{N}(\mu = 250, \sigma = 125)$.
Wholesale price contract: $w = 75$.

Step A: Supply-chain optimum

Compute newsvendor costs for the integrated supply chain:

$$C_o^{SC} = c - s = 35 - 25 = 10, \qquad C_u^{SC} = p - c = 115 - 35 = 80.$$

Critical fractile:

$$\frac{C_u^{SC}}{C_u^{SC} + C_o^{SC}} = \frac{80}{80 + 10} = \frac{80}{90} \approx 0.89.$$

From the standard normal table, $\Phi(z) = 0.89$ corresponds to approximately $z \approx 1.23$. Thus,

$$Q^{SC} = \mu + \sigma z = 250 + 125(1.23) \approx 404.$$

Step B: Retailer's optimum under wholesale price $w = 75$

From the retailer's perspective:

$$C_o^R = w - s = 75 - 25 = 50, \qquad C_u^R = p - w = 115 - 75 = 40.$$

Critical fractile:

$$\frac{C_u^R}{C_u^R + C_o^R} = \frac{40}{40 + 50} = \frac{40}{90} \approx 0.44.$$

This corresponds to a negative z (because $0.44 < 0.5$); the slides approximate $z \approx -0.15$. Therefore,

$$Q^R = 250 + 125(-0.15) \approx 232.$$

Interpretation

The retailer orders 232, far below the system-optimal 404. This is *underordering*. The reason is incentive misalignment: the retailer's "unit cost" is w, not the true production cost c. The supplier earns a margin through w, but the retailer becomes cautious because overstocking is expensive for the retailer.

7 Worked Example 2: A Coordinating Revenue-Sharing Contract

Using the same UV sunglasses data, suppose we choose a revenue-sharing contract with

$$w = 29, \qquad y \approx 0.47 \text{ (about 47\%)}.$$

The session notes state that this pair achieves the supply-chain optimum by matching critical fractiles.

Retailer's order quantity

Retailer's costs under revenue sharing:

$$C_o^R = w - s = 29 - 25 = 4, \qquad C_u^R = p(1-y) - w = 115(1-0.47) - 29 \approx 32.$$

Critical fractile:

$$\frac{C_u^R}{C_u^R + C_o^R} = \frac{32}{32+4} = \frac{32}{36} \approx 0.89,$$

which matches the supply-chain benchmark. Therefore, the retailer orders the same Q as the supply-chain optimum:

$$Q^R(29, 0.47) \approx 404.$$

Profits (conceptual structure)

The lecture computes expected profits using inventory-table quantities (expected sales and left-overs). The important accounting idea is that the supplier's profit has *two parts*:

$$\Pi_S = Q(w-c) + \big(\text{expected sales}\big) \cdot py.$$

The first term can be negative when $w < c$; the second term compensates the supplier when sales occur. The retailer's profit is computed like a newsvendor with its own C_u^R and C_o^R.

The slides report that, compared with the wholesale contract at $w = 75$, this coordinating revenue-sharing contract increases total supply-chain profit to the optimum and can raise both firms' expected profits (a "win–win" region exists).

8 Common Pitfalls and How to Avoid Them

Pitfall 1: Mixing up whose costs matter

In a decentralized supply chain, the order quantity is determined by the party who chooses Q. If the retailer chooses Q, then use the retailer's perceived C_u and C_o, not the supply chain's. Students often (incorrectly) plug production cost c into the retailer's problem under a wholesale contract. The retailer pays w, not c.

Pitfall 2: Confusing the critical fractile with the order quantity

The critical fractile is a probability level between 0 and 1:

$$\frac{C_u}{C_u + C_o}.$$

It is *not* the order quantity. Under normal demand you still need to convert it to a z-score using $\Phi(z)$, then compute $Q = \mu + \sigma z$.

Pitfall 3: Forgetting the feasibility conditions in revenue sharing

For a coordinating revenue-sharing contract, y must satisfy $0 \le y \le 1$. This often implies $s < w < c$. If you pick w too high (e.g., $w > c$), the required y to coordinate may become negative or otherwise infeasible.

Pitfall 4: Thinking coordination automatically means both firms earn more

Coordination maximizes the *total* expected profit. But how that larger pie is split depends on (w, y). Some coordinating contracts can make one party worse off than under the status quo, so they would refuse to switch. Contract design therefore has two goals:

$$\text{Efficiency (maximize total profit)} \quad \text{and} \quad \text{Acceptability (each party benefits).}$$

Pitfall 5: Misinterpreting supplier profit randomness

Under a pure wholesale-price contract, once Q is placed the supplier profit is often deterministic (approximately $Q(w-c)$) because it does not depend on realized demand. Under revenue sharing, supplier profit becomes demand-dependent because the supplier receives a share of realized sales.

9 Efficiency vs. Distribution: The Two Faces of Contract Design

Revenue-sharing contracts illustrate a powerful separation:

- **Efficiency** depends on inducing the right operational decision (here, the right order quantity). Matching the retailer's critical fractile to the supply-chain critical fractile is the coordination mechanism.
- **Distribution** depends on how contract parameters split the total profit between firms. Many (w, y) pairs can achieve Q^{SC}, but they yield different expected profits for supplier and retailer.

In practice, the final terms reflect bargaining power, outside options, and risk tolerance. A supplier with strong bargaining power may prefer a larger y (more share of sales revenue) or a higher w (though in coordinating revenue sharing, w often must stay below c). A retailer with strong power may push for terms that shift risk back to the supplier.

10 Brief Note: Other Risk-Sharing Contracts (Context)

Revenue sharing is one of many contract forms used in supply chains. Other common structures include return contracts (supplier buys back leftovers at an agreed price), quantity discounts, option contracts, quantity flexibility contracts, and price protection. The unifying idea is the same: modify who bears demand risk and how marginal incentives are formed, so that the decentralized decision resembles the centralized optimum.

11 Summary

A wholesale price contract is simple, but it often creates misalignment: the retailer orders based on its own margin $p-w$ and overstocking loss $w-s$, typically producing underordering relative to the system optimum. This is a manifestation of double marginalization.

A revenue-sharing contract adds a sales-based transfer y, which changes the retailer's effective margin and can be chosen to match the supply-chain critical fractile:

$$\frac{p(1-y)-w}{p(1-y)-s} = \frac{p-c}{p-s}.$$

When this holds, the retailer's profit-maximizing order quantity coincides with the supply-chain optimum, so the decentralized supply chain achieves centralized efficiency. Among the many coordinating contracts, the final choice of (w, y) determines profit distribution and must be acceptable to both parties.

Price-Based Revenue Management in a Supply Chain
and a Brief Introduction to Behavioral Operations Management

Concise chapter for junior undergraduate Operations Management

1 Why pricing belongs in Operations Management

Operations management (OM) is often introduced through decisions about capacity, inventory, and process flow. Pricing can look like a marketing topic, but in many industries pricing is also an operational control: a price can *shape* demand, which then shapes production, purchasing, staffing, and service capacity needs. This idea is central to *revenue management* (RM), where firms use prices (and sometimes capacity controls) to match demand with limited resources and to improve profitability.

This chapter studies a simple but powerful setting: a supply chain with a supplier and a retailer, where consumer demand is a deterministic function of the retail price. Even in this simple case, decentralized decision making can lead to inefficiency. We then show how a contract can fix the inefficiency and how behavioral ideas remind us that real decisions often deviate from the fully rational models.

2 A baseline model: supplier–retailer pricing with linear demand

The setting and key definitions

Consider a two-stage supply chain:

- A **supplier** produces a product at constant unit production cost c and sells it to a retailer at a **wholesale price** w.
- A **retailer** sets the **retail price** p charged to consumers. Demand depends on price.

In this chapter we assume **deterministic linear demand**:

$$D(p) = a - bp, \tag{1}$$

where $a > 0$ is the demand intercept and $b > 0$ is the slope (price sensitivity). Deterministic means that for a given p, demand is known exactly.

A common operational interpretation is that the retailer will order exactly what it plans to sell, so the order quantity is

$$q = D(p) = a - bp. \tag{2}$$

Profit functions

With no fixed costs and no salvage value, profits are margin times volume.

Supply chain (integrated firm). If the supplier and retailer were one integrated firm, the unit margin against production cost is $p - c$. Total supply chain profit is

$$\Pi^{SC}(p) = (p - c)\, D(p) = (p - c)(a - bp). \tag{3}$$

Retailer under a wholesale price contract. In the traditional **wholesale price contract**, the retailer pays w per unit, so its unit margin is $p - w$. Retailer profit is

$$\Pi^{R}(p \mid w) = (p - w)\, D(p) = (p - w)(a - bp). \tag{4}$$

Supplier under a wholesale price contract. The supplier earns $w - c$ per unit on the quantity ordered $q = D(p)$. Supplier profit is

$$\Pi^{S}(w) = (w - c)\, D(p(w)), \tag{5}$$

where $p(w)$ is the retailer's optimal response to w. This dependence matters: the supplier chooses w anticipating that the retailer will adjust p.

Core intuition: single vs. double marginalization

An integrated firm chooses p to maximize $\Pi^{SC}(p)$, putting *one* markup over cost c. In a decentralized chain, the supplier sets a markup $w - c$, and then the retailer sets a second markup $p - w$. This two-layer markup is called **double marginalization**. The typical outcome is:

$$p \text{ is too high}, \qquad q = D(p) \text{ is too low}, \qquad \Pi^{SC} \text{ is below the integrated optimum.}$$

The logic is not that either party is "wrong," but that each optimizes its own profit, and the incentives are not aligned.

3 Solving the model: main formulas

Because $D(p)$ is linear, profits become quadratic in p. The maximizing price is at the vertex of the parabola.

Supply chain optimal retail price

Maximize $\Pi^{SC}(p) = (p - c)(a - bp)$. Expanding:

$$\Pi^{SC}(p) = ap - abp^2 - ac + bcp = -bp^2 + (a + bc)p - ac.$$

The optimal price is

$$p^{SC} = \frac{a + bc}{2b}. \tag{6}$$

Then demand and profit are $q^{SC} = D(p^{SC})$ and $\Pi^{SC}(p^{SC})$.

Retailer best response under a wholesale price w

Given w, the retailer chooses p to maximize $\Pi^R(p \mid w) = (p - w)(a - bp)$. Expanding:

$$\Pi^R(p \mid w) = -bp^2 + (a + bw)p - aw.$$

Thus the retailer's optimal retail price is

$$p^R(w) = \frac{a + bw}{2b}. \tag{7}$$

A useful observation follows immediately:

$$p^R(w) = \frac{a}{2b} + \frac{w}{2}, \tag{8}$$

so an increase in wholesale price increases the retail price, but only by half of the increase in this linear model.

Given $p^R(w)$, the resulting order quantity is

$$q(w) = D(p^R(w)) = a - b\left(\frac{a + bw}{2b}\right) = \frac{a - bw}{2}. \tag{9}$$

Supplier optimal wholesale price

Anticipating $q(w) = (a - bw)/2$, the supplier chooses w to maximize

$$\Pi^S(w) = (w - c)\, q(w) = (w - c)\frac{a - bw}{2}.$$

This is a concave quadratic in w. The maximizing wholesale price is

$$w^* = \frac{a + bc}{2b}. \tag{10}$$

It is striking that $w^* = p^{SC}$ in this model; however, they represent different decisions: w^* is the supplier's optimal wholesale price in a decentralized chain, while p^{SC} is the integrated chain's optimal retail price.

4 Worked Example 1: the numbers from the slides

Suppose

$$D(p) = 1600 - 20p, \qquad a = 1600, \qquad b = 20, \qquad c = 10.$$

Supply chain optimum

Supply chain profit is

$$\Pi^{SC}(p) = (1600 - 20p)(p - 10) = -20p^2 + 1800p - 16000.$$

The optimal price is

$$p^{SC} = \frac{a + bc}{2b} = \frac{1600 + 20 \cdot 10}{2 \cdot 20} = \frac{1800}{40} = 45.$$

Demand is

$$q^{SC} = D(45) = 1600 - 20 \cdot 45 = 700.$$

Supply chain profit is

$$\Pi^{SC}(45) = 700 \cdot (45 - 10) = 700 \cdot 35 = 24500.$$

Wholesale price contract with $w = 30$

Retailer profit is $\Pi^R(p \mid 30) = (1600 - 20p)(p - 30)$, so the retailer chooses

$$p^R(30) = \frac{a + bw}{2b} = \frac{1600 + 20 \cdot 30}{40} = \frac{2200}{40} = 55.$$

Quantity is

$$q(30) = D(55) = 1600 - 20 \cdot 55 = 500.$$

Retailer profit is

$$\Pi^R = q(p - w) = 500(55 - 30) = 12500.$$

Supplier profit is

$$\Pi^S = q(w - c) = 500(30 - 10) = 10000.$$

Total decentralized profit is 22500, which is below the integrated optimum 24500. The gap is the efficiency loss from double marginalization.

5 Why a wholesale contract cannot coordinate this supply chain

Coordination means the decentralized decisions produce the same outcome as the integrated optimum, especially the same retail price p^{SC} and quantity q^{SC}.

Under a wholesale contract, the retailer always chooses $p^R(w) = \frac{a+bw}{2b}$. To force $p^R(w) = p^{SC}$, we would need

$$\frac{a + bw}{2b} = \frac{a + bc}{2b} \quad \Rightarrow \quad w = c.$$

So the wholesale price would have to equal production cost.

However, if $w = c$, then supplier profit $(w - c)q$ is zero. In many settings a supplier will not accept a contract that yields no profit. Therefore the wholesale price will typically satisfy $w > c$, which implies

$$p^R(w) = \frac{a}{2b} + \frac{w}{2} > \frac{a}{2b} + \frac{c}{2} = p^{SC}.$$

So the retail price is always too high relative to the supply chain optimum, and the quantity is too low. This is a clean algebraic explanation of why the traditional wholesale contract fails to coordinate this pricing problem.

6 Revenue sharing contract: aligning incentives (partially)

Definition

A **revenue sharing contract** adds one more term to the wholesale contract. The retailer still pays a wholesale price w per unit, but also shares a fraction $y \in [0, 1]$ of its revenue pq with the supplier.

There are many variations in practice; here we use the structure from the slides: the retailer keeps fraction $(1 - y)$ of the margin-based profit expression, which leads to the key coordination insight.

Retailer decision under revenue sharing

The retailer's profit is modeled as

$$\Pi_{RS}^{R}(p \mid w, y) = (1 - y)\,(p - w)\,D(p). \tag{11}$$

Because $(1 - y)$ is a positive constant multiplier, it does not change the maximizing p. Therefore the retailer still chooses

$$p_{RS}^{R}(w, y) = p^{R}(w) = \frac{a + bw}{2b}. \tag{12}$$

This is an important lesson: not every contract parameter affects operational decisions. Some parameters only redistribute money without changing behavior.

How to achieve the supply chain optimum

To achieve $p^{SC} = \frac{a+bc}{2b}$, we again need $w = c$. Under revenue sharing, the supplier can accept $w = c$ because it can earn money through the shared revenue portion controlled by y.

Thus, in this model:

$$\text{Coordination requires } w = c, \quad \text{while } y \text{ determines the split of the total profit.} \tag{13}$$

7 Worked Example 2: choosing a win–win sharing ratio

Continue the numerical example with $a = 1600$, $b = 20$, $c = 10$. Compare:

- Original wholesale contract with $w = 30$: retailer profit 12500, supplier profit 10000.
- Coordinating revenue sharing contract with $w = c = 10$: total profit equals the supply chain optimum 24500.

Under the revenue sharing contract (with coordination), profits are split as:

$$\Pi_{RS}^{R} = 24500(1 - y), \qquad \Pi_{RS}^{S} = 24500y.$$

For the retailer to be willing to switch:

$$24500(1 - y) > 12500 \quad \Rightarrow \quad 1 - y > \frac{12500}{24500} \quad \Rightarrow \quad y < 1 - \frac{12500}{24500} \approx 0.4898.$$

For the supplier to be willing to switch:

$$24500y > 10000 \quad \Rightarrow \quad y > \frac{10000}{24500} \approx 0.4082.$$

So a *win–win* range is

$$0.4082 < y < 0.4898,$$

or about 40.8% to 49.0%. Any y in this interval makes both parties better off than under $w = 30$, while also achieving the supply chain optimum price and quantity.

8 Common pitfalls and how to avoid them

Students often make mistakes in this topic for reasons that are easy to fix once you know where to look.

First, it is common to confuse **who chooses what**. In the wholesale and revenue sharing settings here, the supplier chooses w and the retailer chooses p (and therefore $q = D(p)$). When optimizing, always maximize the correct profit function with respect to the correct decision variable.

Second, it is easy to mix up the **supply chain optimum** with the **equilibrium under decentralization**. The integrated optimum maximizes $\Pi^{SC}(p)$. The decentralized outcome is a two-step logic: retailer chooses p given w, and then supplier chooses w anticipating that response. If you maximize the wrong objective, you will typically get the wrong price.

Third, pay attention to **what parameters affect behavior**. Under the revenue sharing profit form $(1-y)(p-w)D(p)$, y scales profit but does not change the maximizing p. Many learners try to differentiate with respect to p and keep y in the first-order condition; it cancels out.

Fourth, remember the **feasibility of demand**. In a linear demand $D(p) = a - bp$, demand becomes negative if $p > a/b$. In real applications, demand is truncated at zero. In this course-style model, we typically assume the relevant optimum lies in the region where demand is nonnegative. It is still good practice to check that your computed price satisfies $D(p) \geq 0$.

Finally, do not over-interpret the equality $w^* = p^{SC}$ in this linear model. It happens because of the symmetry of the algebra, not because wholesale price and retail price are "the same thing."

9 A brief introduction to Behavioral Operations Management

The pricing and contracting analysis above is an example of **normative** OM: it describes what fully rational agents *should* do to maximize profits given a model. Behavioral operations management (Behavioral OM) asks a complementary question: how do real people actually make operational decisions, and how do systematic deviations from rationality affect performance?

Traditional economics and many traditional OM models often assume that people are close to optimal decision makers with stable preferences, strong reasoning ability, and unbiased beliefs. Behavioral OM relaxes these assumptions and studies the operational consequences of human judgment limits, simple heuristics, social preferences such as fairness, and framing effects.

Judgment under uncertainty: three families of biases

A **judgment bias** is a systematic pattern of deviation from a normative benchmark (often probability theory or optimization). Three broad families are especially relevant.

Availability bias. Availability bias occurs when people judge frequency or probability using information that is easiest to recall or most vivid, rather than information that is statistically representative. Recent dramatic events can dominate memory. For instance, after seeing news about an earthquake, people may overestimate the probability of another earthquake soon and rush to buy insurance, even if the true long-run probability is not as high as their intuition suggests.

Representativeness bias. Representativeness bias occurs when people assess probability by similarity to a stereotype or "prototype." A classic operational implication is **base rate neglect**: people overreact to a signal (like a positive test) and underweight the underlying prevalence rate.

A simple Bayesian calculation shows why base rates matter. Let T be the event "test is positive" and D be "disease is present." Bayes' rule is

$$\mathbb{P}(D \mid T) = \frac{\mathbb{P}(T \mid D)\mathbb{P}(D)}{\mathbb{P}(T \mid D)\mathbb{P}(D) + \mathbb{P}(T \mid D^c)\mathbb{P}(D^c)}. \tag{14}$$

Even highly accurate tests can yield many false positives if the disease is rare, because the pool of healthy people is so large.

Confirmation-related biases. Confirmation bias refers to searching for and interpreting information in a way that supports prior beliefs, while discounting contradictory evidence. Two related ideas commonly discussed are **anchoring**, where an initial number (an "anchor") unduly influences estimates, and **over-precision**, where people are too confident about the accuracy of their judgments.

For OM practice, the message is practical: forecasts, quality decisions, and capacity plans can be distorted by what is most salient, what seems representative, or what confirms existing beliefs.

Risky choice and preference: framing, reference points, and utility curvature

Behavioral OM also studies how people evaluate risky outcomes. A key finding is that preferences depend on **framing** and **reference points**. Two options that are logically equivalent can feel different if described as gains versus losses.

One way to represent attitudes toward risk is with a **utility function** $u(x)$, where x is an outcome such as money gained. Under expected utility, a person prefers option 1 over option 2 if

$$\mathbb{E}[u(X_1)] > \mathbb{E}[u(X_2)].$$

A **concave** utility function (curving downward) captures **risk aversion** in gains: the average utility of a gamble is less than the utility of its expected value. A **convex** utility function (curving upward) captures **risk seeking**.

Behavioral evidence often suggests an **S-shaped** value function around a reference point: concave for gains and convex for losses. This helps explain why people may be cautious when trying to secure gains, yet gamble to avoid sure losses.

In operations, these patterns can matter in contexts like capacity investment, inventory decisions under shortage risk, and contract negotiations, where "loss frames" (e.g., avoiding stockouts) may trigger more aggressive or risk-seeking behavior than "gain frames" (e.g., achieving service targets).

10 Closing perspective

Price-based revenue management in a supply chain shows how a simple demand model can connect pricing, ordering, and contracting. The main operational lesson is that decentralized incentives can push prices above the system-optimal level, reducing total profit through double marginalization. A coordinating contract can restore the system optimum, and its parameters can then be used to distribute the gains so that all parties are willing to participate.

Behavioral OM reminds us that even when the mathematics is clear, real decision makers use heuristics, are influenced by salience and framing, and may not behave like perfect optimizers. Good operations management therefore combines clear models with careful attention to how people actually decide.