オペレーションズ・リサーチ

Operations Research: Models, Algorithms, and Implementations

劉子昂

2025-12-10

目次

Preface 5

講義 7

記号 13

第 I 部　在庫モデル 15

第 1 章　在庫管理とは 17

第 2 章　経済的発注量 25

第 3 章　安全在庫 37

第 4 章　Wagner-Whitin モデル 47

第 5 章　新聞売り子問題 49

第 6 章　まとめ 65

第 II 部　待ち行列理論 67

第 7 章　待ち行列モデルの基礎 69

第 8 章　ポアソン過程 77

第 III 部　OR キャリア 85

第 9 章　キャリア 87

参考文献 89

付録　91

付録 A　微分積分　91

付録 B　確率　93

付録 C　標準正規分布表　95

付録 D　線形代数　99

付録 E　最適化問題　101

Preface

OR を学ぶための教科書です.

注意点

随時更新していきますので, 読む際は必ずブラウザをリロードしてください.

Python

すべてのモデル, アルゴリズムについて，Python での実装例を示します. 講義では Python プログラミングについては扱いませんが，Python の基礎的な知識があると理解が深まります.

以下の手順で, 教材に掲載されているコードを Google Colab 上で簡単に実行できます. ぜひ試してみてください.

1. Google アカウントにログインする
2. Google Colab にアクセスする
3. 「+新しいノート」をクリックする
4. コードをコピーして, セルに貼り付ける
5. セルを実行する

フィードバック

継続的に改善していきますので, 誤字脱字, 内容の不備, わかりにくい箇所などを見つけたら，劉子昂までご連絡ください.

内容

回	内容
1	ガイダンス

回	内容
2	在庫モデル：EOQ モデル
3	在庫モデル：新聞売り子問題
4	在庫モデル：安全在庫
5	AHP
6	PERT 1
7	PERT 2
8	PERT 3
9	包絡分析法
10	動的計画法
11	待ち行列 1
12	待ち行列 2
13	待ち行列 3

講義

本授業では、オペレーションズ・リサーチ（OR）の中の代表的な手法である PERT、在庫理論、待ち行列理論、動的計画法、階層分析法、及び包絡分析法の数理を理解し、具体的な問題への応用を学ぶ。

講義情報

- 講義名：オペレーションズ・リサーチ B
- 曜日：水曜日
- 時限：2 時限目（10:50〜12:30）
- 教室：西館 W202
- 担当教員：劉 子昂
- Google Classroom: Link

出席

不定期に出席を取ります。

出席は WebClass 上で取ります。以下の時間に注意してください。

- 出席扱い：10:50 - 11:09
- 遅刻扱い：11:10 - 11:29
- 欠席扱い：11:30 -

成績評価

- 期末試験（100%）
- 4 回以上の欠席は単位取得不可。
- 講義中の加点問題に正解した場合、試験の点数に加点します。

到達目標

各分野について、下記の事項を目標して講義を行う。

- **PERT** の計算と解析方法を理解し、プロジェクトの評価を行うことができる。
- **在庫モデル**を理解し、自分で式を構築及び解析することができる。
- **待ち行列理論**の重要な式や定理を理論的に導出し、それらを適切に解釈することができる。
- **動的計画法**の基本的な考え方を理解し、簡単な問題への適用ができる。
- **階層分析法**による意思決定の手法を理解し、一対比較行列からウェイトと整合性を計算することができる。
- **包絡分析法**における CCR モデルを理解し、得られた結果を解釈することができる。

さらに、これらの手法を用いて比較的簡単な現象をモデル化し、解析することができる。

OR は重要

経営工学において、最も重要な学問分野の一つ。

日本経営工学会によると、「解決すべき課題の数理モデルを構築し、最適な手法を求めるオペレーションズ・リサーチ（OR）という分野は、経営工学の主要なテーマとなっています」。

海外では、管理科学（Management Science）と OR は同義語として使われることもよくある。

OR の全体像

- 線形計画法
- 整数計画法
- 非線形計画法
- **動的計画法**
- グラフ理論・ネットワーク
- シミュレーション
- **在庫モデル**
- **待ち行列**
- **多基準意思決定分析**
- **プロジェクトマネジメント**

- …

OR は難しい？

基礎知識が必要

微分積分、線形代数、確率、統計の基礎知識が必要です。これらの基礎が不十分な場合、授業についていけないです。基礎知識が不十分な場合、必ず復習してください。

この講義では、以下の工夫をしています。

- 付録に必要な基礎知識のまとめがあります。随時更新しますので、参考にしてください。
- 講義資料には例題、図、演習問題を多く用意しています。
- プログラミングの実装例も示します。

数式が多い

OR は、問題を数理的にモデル化し、解析する学問です。数式をたくさん使い、証明も多いです。数式を読むのが苦手な人は、慣れるまで大変かもしれません。

この講義では、以下の工夫をしています。

- 証明は省略なく丁寧に行います。
- わかりにくいところをコラムで補足します。

何を学ぶ？

- モデル：現実の問題を数理的に表現したもの
- モデリング：現実の問題を数理モデルで表現すること
- 解：問題の答え
- 最適化：最適解を見つけること
- アルゴリズム：問題を解く手順

単に，結論・定理を覚えるだけでなく，モデリング，証明，アルゴリズムの理解が重要です。

Python について

- すべてのアルゴリズム、モデルを Python で実装します。
- Python の基礎については、この講義では説明しません。
- Python のコードを理解できなくても、授業内容の理解には支障ありません。
- Youtube などでは、2-3 時間で Python 入門の動画がたくさんあります。事前に学習しておくことをお勧めします。

授業時間外の学習

本授業の準備・復習等の授業時間外学習は、4 時間を標準とする

私語

- 講義中の私語は厳禁です.
- 注意してもやめない場合は、減点を行います。

学生による授業改善アンケート

第 14 回の授業中に, 以下の URL からアクセスして, 授業改善アンケートに回答してください.

学生による授業改善アンケート

- このアンケートの目的は、授業の内容と方法を向上させることにあります。授業は学生のみなさんと教員とが協力してこそ、実り多いものとなります。
- 回答内容が成績評価に影響することはありません。教員は回答の集計結果と自由記述回答のみを確認します。みなさんの学生証番号と個人名が担当教員に知られることはありません。
- 責任のある回答にご協力ください。
- アクセスしにくい場合は、授業時間外でご回答ください。
- アンケート提出前に回答内容をよく確認してください。

期末試験（2025）

- 試験時間：90 分
- 試験範囲：講義で扱ったすべての内容

- 参照・使用できるもの
  - 電卓
    - 平方根, 立方根の計算方法を確認しておいてください。
  - 公認カンニングペーパー(自筆, 表面のみ)

注意事項

- 学生証：
  - 試験を受けるためには学生証の提示が必要です。
- 携帯電話・スマートウォッチ等の通信機器：
  - いかなる形態でも通信機器の使用は禁止しますので、電源を切り必ずかばん等にしまってください。
  - 時計や電卓の代用として使用することも出来ません。

記号

集合

A 集合。大文字で表す。
A B 集合 A と集合 B の差集合。

線形代数

記号	意味
$\mathbf{x}$, $\mathbf{y}$	ベクトル。太字の小文字で表す。
$\mathbf{A}$, $\mathbf{B}$	行列。太字の大文字で表す。
$\mathbf{I}$	単位行列。

確率統計

X: 確率変数。大文字で表す。

$\mathbb{P}(X = x)$: 確率変数 X が x をとる確率

$\mathbb{E}[X]$: 確率変数 X の期待値

$F_X(x)$: 確率変数 X の累積分布関数

$f_X(x)$: 確率変数 X の確率密度関数

$p_X(x)$: 確率変数 X の確率質量関数

$\phi(z)$: 標準正規分布の確率密度関数

関数

A は要素の間に順序が定義された集合とする。

$\max A$: 集合 A の最大値。

$\min A$: 集合 A の最小値。

$(x)^+$: x と 0 のうち大きい方。すなわち、$(x)^+ = \max(x, 0)$。

$(x)^-$: x と 0 のうち小さい方の絶対値。すなわち、$(x)^- = \max(-x, 0) = -\min(x, 0)$。

例題

例 0.1 (差集合)**.** $A = \{1, 2, 3\}$、$B = \{2, 3, 4\}$ とする。このとき、$A \setminus B = \{1\}$ である。

例 0.2 (最大値と最小値)**.** $A = \{1, 0, -1\}$ とする。このとき、$\max A = 1$、$\min A = -1$ である。

例 0.3 (正の部分)**.** $(10)^+ = 10$、$(-30)^+ = 0$ である。

例 0.4 (負の部分)**.** $(10)^- = 0$、$(-30)^- = 30$ である。

第I部

在庫モデル

第1章　在庫管理とは

予備知識

- 微分積分
- 確率統計

商店・工場・倉庫などで、原材料・部品・製品などを適切に管理することを**在庫管理**（Inventory Management）という。一般的に、在庫管理の目的は、顧客の需要を満たしつつ、在庫に関わる費用を最小化することである。

ノート

豊田自動車が提唱した**ジャストインタイム**（Just In Time, JIT）は、生産方式としてよく知られている。
JIT とは必要なものを、必要な時に、必要な量だけ生産することである。
JIT の目的は、在庫を最小限に抑え、効率的な生産を実現することである。
アメリカの研究者らは、その生産方式を体系化し、**リーン生産方式**（Lean Manufacturing）という概念を提唱した。

在庫量が多すぎると、保管費用がかかる。逆に、在庫量が少なすぎると、欠品が発生し、顧客の需要を満たせなくなる。在庫管理は次の二つの問題を決定する。

1. どのくらいの量を発注するか？（発注量）
2. いつ発注するか？（発注時期）

科学的在庫管理（Scientific Inventory Management）では、これらの問題に答えるために、次の手順で在庫管理を行う。

1. 在庫システムを数学モデルとして定式化する。
2. 最適な発注量と発注時期を決定する。

練習 1.1. 前回スーパーに行ったときに買った商品（例えば、牛乳、卵など）について考える。需要と在庫の観点から、次の質問に答えよ。

1. どのくらいの量を買ったか？
2. なぜその量を買ったのか？
3. どのタイミング・頻度でその商品を買うか？

1.1　在庫量

需要（demand） ある期間に顧客が購入したい商品の量。通常、d で表す。
手持ち在庫（on-hand inventory） ある時点で、実際に手元にある在庫の量。OH で表す。
バックオーダー（backorder） 手持ち在庫がなく、満たせない需要。BO で表す。
在庫量（inventory level） ある時点での在庫の量。I で表す。

一般、BO と OH は次のように表される。

$$OH = I^+ = \max(0, I)$$

$$BO = I^- = \max(0, -I)$$

例 1.1. 手持ち在庫が $OH = 50$、需要が $d = 30$ のとき、在庫量は次のように計算される。

$$I = 50 - 30 = 20$$

手持ち在庫が $OH = 50$、需要が $d = 70$ のとき、在庫量は次のように計算される。

$$I = 50 - 70 = -20$$

このとき、手持ち在庫は

$$OH = I^+ = \max(0, I) = 0$$

となる。バックオーダーは

$$BO = I^- = \max(0, -I) = 20$$

となる。

1.2 在庫モデルの分類

在庫モデルは、次のような要素で分類される。

需要（demand） 需要が決定論的 (Deterministic) か確率的（Stochastic）か。

観測（review） 在庫量を連続観測 (Continuous Review) するか、周期観測 (Periodic Review) するか。連続観測の場合、在庫量が連続的に観測でき、いつでも発注が可能である。周期観測の場合、一定の期間（例えば 1 週間）ごとに在庫量を観測する。

リードタイム（lead time） 発注から納品までの期間。調達期間とも呼ばれる。リードタイムが決定論的か確率的か。また、リードタイムが 0 かどうか。在庫モデルを単純化するために、リードタイムを 0 とし、発注から納品までの期間を無視することもある。

バックオーダー（backorder） バックオーダーが許容されるかどうか。需要が手持ち在庫を上回った場合、バックオーダーが許容されると、欠品が発生しても、後で需要を満たすことができる。バックオーダーが許容されない場合、欠品が発生すると、上回った需要は失われ、機会損失が発生する。

計画期間（planning horizon） 単一期間 (Single Period) か、複数期間 (Multi Period) か、無限 (Infinite) か。

以下の表に、需要と観測に基づく、古典的な在庫モデルを示す。

在庫モデル	需要	観測
EOQ モデル	決定論的	連続観測
Wagner-Whitin	決定論的	周期観測
安全在庫	確率的	連続観測
新聞売り子問題	確率的	周期観測

1.3 在庫の費用

ここでは、在庫に関わる費用を紹介する。

発注費用（ordering cost） 発注量に関わらず、1 回の発注にかかる費用。調達費用、固定費用（fixed cost）などとも呼ばれる。通常、1 回の発注にかかる費用を K とする。

購入費用（purchase cost） 商品を購入するためにかかる費用。通常、単位あたりの購入費用を c とする。

欠品費用（stockout cost） 需要が手持ち在庫を上回った場合に発生する費用。通常、単位あたりの欠品費用を p とする。

保管費用（holding cost） 在庫を保管するためにかかる倉庫費用、保険費用、税金、機会費用など。通常、単位時間あたりの 1 単位あたりの保管費用を h とする。

例 1.2 (発注費用と購入費用)**.** 毎回の発注量を Q、1 回の発注にかかる費用を K、単位あたりの購入費用を c とする。1 回の発注にかかる総費用は、次のように計算される。

$$K + cQ$$

となる。

例 1.3 (欠品費用)**.** 在庫量を I、需要を d、単位あたりの欠品費用を p とする。欠品費用は次のように計算される。

$$p(d - I)^+$$

$p = 10$、$I = 50$、$d = 70$ のとき、欠品費用は次のように計算される。

$$p(d - I)^+ = 10(70 - 50)^+ = 200$$

$p = 10$、$I = 50$、$d = 20$ のとき、欠品費用は次のように計算される。

$$p(d - I)^+ = 10(20 - 50)^+ = 0$$

例 1.4 (在庫量が一定の保管費用)**.** 1 日あたり 1 単位の在庫を保管するために、h の費用がかかるとする。30 日間、50 単位の在庫を保管するための総保管費用を計算せよ。

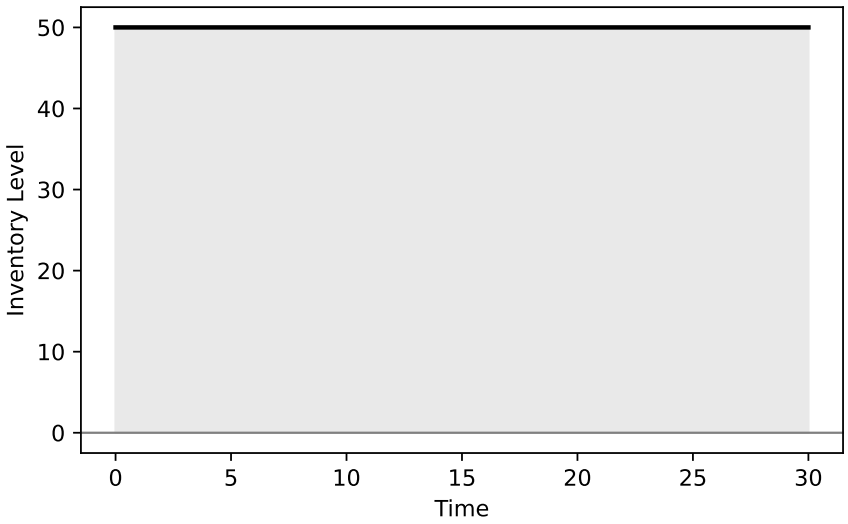
保管費用は次のように計算される。

$$30 \times 50 \times h = 1500h$$

下の図では、横軸が時間、縦軸が在庫量を表す。

```
import matplotlib.pyplot as plt
import numpy as np

t = np.linspace(0, 30, 1000)
inventory = np.full_like(t, 50)
```

```
# Plotting the inventory level
plt.fill_between(t, inventory, color="lightgray", alpha=0.5, label="Inventory Level")
plt.plot(t, inventory, label="Inventory Level", color="black", linewidth=2)
plt.xlabel("Time")
plt.ylabel("Inventory Level")
plt.axhline(0, color="gray", linewidth=1)
plt.tight_layout()
plt.show()
```



一般的に、保管費用は次の式で計算される。

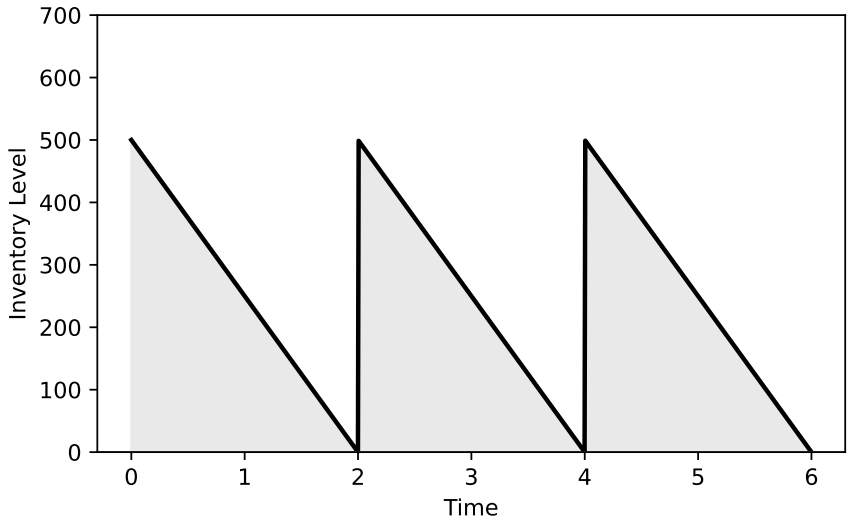
$$\text{保管費用} = \text{面積} \times h$$

例 1.5 (在庫が時間とともに変化する保管費用). 通常、在庫量が定数ではなく、時間とともに変化する。ここでは、在庫量が時間とともに線形に減少し、0 になると在庫が補充される場合を考える。毎回の発注量を 500 とする。

下の図に示すように在庫量が時間とともに変化するとする。6 日間の保管費用を計算せよ。

```
# Parameters
d = 250  # Demand rate
Q = 500  # Order quantity
T = Q / d  # Cycle length
t = np.linspace(0, 2.999 * T, 1000)
```

```
# Inventory level over time
inventory = np.maximum(0, Q - (d * t) % Q)

# Plotting the inventory level
plt.fill_between(t, inventory, color="lightgray", alpha=0.5, label="Inventory Level
plt.plot(t, inventory, label="Inventory Level", color="black", linewidth=2)
plt.xlabel("Time")
plt.ylabel("Inventory Level")
plt.axhline(0, color="gray", linewidth=1)
plt.ylim(bottom=0, top=Q + 200)
plt.tight_layout()
plt.show()
```



保管費用は 面積 $\times\, h$ で計算される。それぞれの三角形の面積は $\frac{1}{2} \times 500 \times 2$ であるため、6 日間の保管費用は次のように計算される。

$$\frac{2 \times 500}{2} \times 3 \times h$$

1.4　在庫方策

確率的在庫モデルにおいて、一つ重要な概念は**在庫方策**（inventory policy）である。在庫方策は、在庫の状況に応じて、在庫管理のルールを定めるものである。代表的な在庫方策を以下に示す。

1. (r, Q) 方策：在庫量を連続的に観測し、在庫量が発注点 r 以下になったときに発注量 Q を発注する方式である。**発注点方式**とも呼ばれる。
2. BSP 方策(Base Stock Policy)：在庫量を定期的に観測し、在庫量が基準在庫 S 以下になったときに、在庫量を S まで補充する方式である。**定期発注方式**とも呼ばれる。
3. (s, S) 方策：在庫量を定期的に観測し、在庫量が発注点 s 以下になったときに、在庫量を補充点 S まで補充する方式である。

一部の確率的在庫モデルにに対し、これらの在庫方策は**最適**であることが知られている。その場合、在庫方策が持つパラメータを最適化することで、在庫の期待コストを最小化することができる。

1.5 練習問題

1. 在庫管理の目的は、_**を満たしつつ、**_を最小化することである。
2. 在庫管理の決定すべき二つの問題は、_**と**_である。
3. 在庫量 $I = -30$, 需要 $d = 10$ のとき、手持ち在庫 OH とバックオーダー BO を計算せよ。
4. 1 回の発注にかかる費用 $K = 1000$、単位あたりの購入費用 $c = 50$、発注量 $Q = 200$ のとき、3 回の発注にかかる総費用を計算せよ。

第2章　経済的発注量

経済的発注量（EOQ: Economic Order Quantity）モデルは，最も基本的な在庫管理モデルの一つである．Harris (1990) がこのモデルを最初に提案した．

EOQ モデルは，単位時間あたりの需要量は決定論的で，一定であると仮定する．すなわち，需要量は事前に分かっており，時間とともに変化しない．単位時間あたりの需要量は需要率（demand rate）と呼ばれ，記号 d で表される．リードタイムは 0 とし，発注から納品までの時間はないと仮定する．一回の発注量を Q とし，一定であるとする．欠品は許せないとする．全ての需要は満されなければならない．また，EOQ モデルでは，在庫量は連続的に観測され，いつでも発注が可能であるとする．

在庫に関わる費用は，1 回あたりの発注費用 K，単位時間あたりの 1 単位の保管費用 h と，購入単価 c がある．

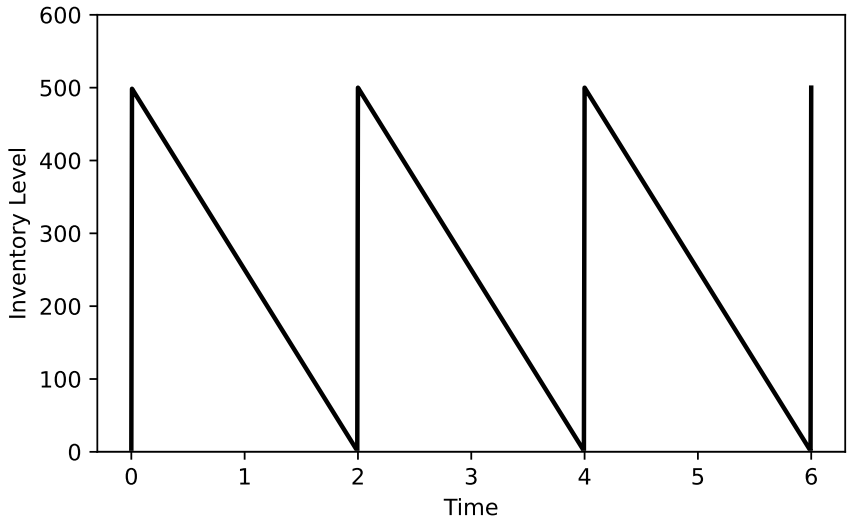
EOQ モデルの最適解は次の二つの性質を持つ (Snyder と Shen 2019)：

1. Zero-inventory ordering (ZIO). 在庫量が 0 のときに発注を行う．リードタイムは 0 であるため，在庫量が 0 でないときに発注すると，保管費用が発生する．
2. Constant order sizes. 発注量は一定である．需要率 d が一定であり，在庫量が 0 のときに発注を行うため，最適発注量も一定である．

以上の性質から，在庫量の時間的変化は下図のようになる．

```
import matplotlib.pyplot as plt
import numpy as np

# Parameters
d = 250  # Demand rate
Q = 500  # Order quantity
T = Q / d  # Cycle length
t = np.linspace(0, 3 * T, 1000)  # Time from 0 to 3 cycles

# Inventory level over time
```

```
inventory = np.maximum(0, Q - (d * t) % Q)
inventory[0] = 0

# Plotting the inventory level
plt.plot(t, inventory, label="Inventory Level", color="black", linewidth=2)
plt.xlabel("Time")
plt.ylabel("Inventory Level")
plt.axhline(0, color="gray", linewidth=1)
plt.ylim(bottom=0, top=Q + 100)
plt.tight_layout()
plt.show()
```



発注の間隔を**サイクル** (cycle) と呼び, サイクル期間は

$$T = \frac{Q}{d}$$

で与えられる.

例 2.1. A 社は, 毎月 250 個の需要がある商品を取り扱っている. 一回の発注量は 500 個とし, サイクル期間は

$$T = \frac{500}{250} = 2\text{ヶ月}$$

となる.

2.1 記号

記号	意味
d	単位時間あたりの需要量
K	1 回あたりの発注費用
h	単位時間あたりの 1 単位の保管費用
c	購入単価
Q	発注量
T	サイクル期間
$g(Q)$	平均コスト

2.2 コスト関数

ここでは，1 サイクルあたりのコストを考える．

発注費用：発注は 1 回だけ行うため, 発注費用は K である．

購入費用：Q 個の商品を単価 c で購入するため, 購入費用は cQ である．

保管費用：サイクル期間 T は $\frac{Q}{d}$ であるため，1 サイクルあたりの保管費用は

$$\frac{TQ}{2}h = \frac{hQ^2}{2d}$$

となる．

以上より，1 サイクルあたりのコストは次のように表される．

$$K + cQ + \frac{hQ^2}{2d}$$

平均コストは, これをサイクル期間 T で割ったものとして定義される. したがって, 平均コスト $g(Q)$ は次のように表される．

$$\begin{aligned} g(Q) &= \frac{1}{T}\left(K + cQ + \frac{hQ^2}{2d}\right) \\ &= \frac{d}{Q}\left(K + cQ + \frac{hQ^2}{2d}\right) \\ &= \frac{Kd}{Q} + cd + \frac{hQ}{2} \end{aligned}$$

以上より, 平均コストは発注量 Q の関数として次のように表される.

$$g(Q) = \frac{Kd}{Q} + cd + \frac{hQ}{2}$$

2.3　最適発注量

EOQ モデルの目的は, 平均コスト $g(Q)$ を最小化する発注量 Q を求めることである.

平均コストの導関数 $g'(Q)$ が 0 となる点を求めることで, 最適発注量 Q^* を求めることができる.

$$g'(Q) = -\frac{Kd}{Q^2} + \frac{h}{2} = 0$$

これを解くと, 最適発注量

$$Q^* = \sqrt{\frac{2Kd}{h}}$$

を得る. これを **EOQ 公式**（EOQ formula）と呼ぶ．Q^* を経済的発注量と呼ぶ(経済的は最適という意味である).

二階導関数 $g''(Q)$ を求めて, 最適発注量が最小値を与えることを確認する.

$$g''(Q) = \frac{2Kd}{Q^3} > 0$$

$g''(Q) > 0$ であるため，Q^* は最小値を与える.

最適発注量 Q^* を次の定理にまとめる.

定理 2.1. *EOQ* モデルにおいて, 最適発注量 Q^* は

$$Q^* = \sqrt{\frac{2Kd}{h}} \tag{2.1}$$

で与えられる.

Q^* を用いて, 最適なサイクル期間 T^* を求めることができる.

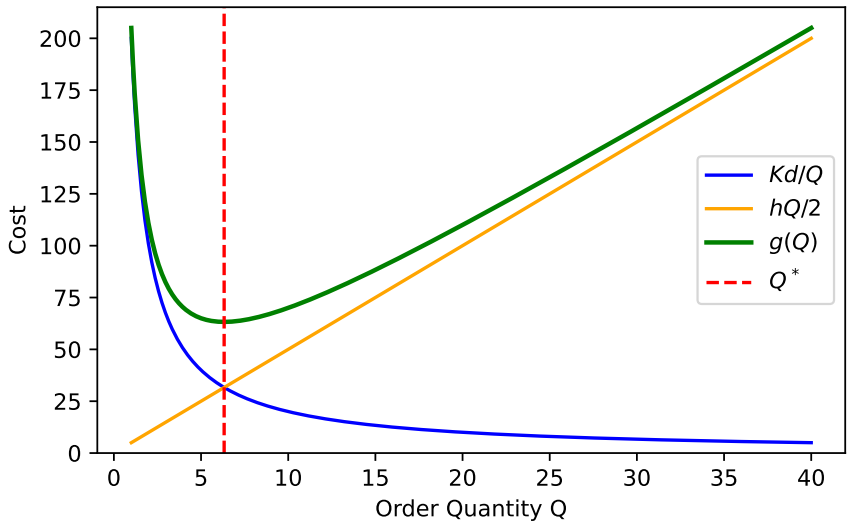
$$T^* = \frac{Q^*}{d} = \sqrt{\frac{2K}{hd}} \tag{2.2}$$

式 2.1 と 式 2.2 から, 以下の性質がわかる.

1. h の増加に伴い, Q^* は減少する. 保管費用が高い場合は, 少量で高い頻度で発注することが望ましい.
2. K の増加に伴い, Q^* は増加する. 発注費用が高い場合は, 多量で低い頻度で発注することが望ましい.
3. d の増加に伴い, Q^* は増加する.
4. c は Q^* に影響しない. 購入単価は最適発注量に影響しない.

次の図は, 購入単価を $c = 0$ とするとき, 平均コスト $g(Q)$, 発注コスト $\frac{Kd}{Q}$, 保管コスト $\frac{hQ}{2}$ のグラフを示す.

```
# Parameters
K = 40  # Order cost
h = 10  # Holding cost
d = 5  # Demand rate
Q = np.linspace(1, 40, 400)  # Order quantity from 1 to 40

# Cost calculations
order_cost = K * d / Q
holding_cost = (h * Q) / 2
average_cost = order_cost + holding_cost
optimal_Q = np.sqrt(2 * K * d / h)

# Plotting the costs
plt.plot(Q, order_cost, label=r"$Kd/Q$", color="blue")
plt.plot(Q, holding_cost, label=r"$hQ/2$", color="orange")
plt.plot(Q, average_cost, label=r"$g(Q)$", color="green", linewidth=2)
plt.axvline(optimal_Q, color="red", linestyle="--", label=r"$Q^*$")
plt.xlabel("Order Quantity Q")
plt.ylabel("Cost")
plt.ylim(bottom=0)
plt.legend()
plt.tight_layout()
plt.show()
```



平均コストが最小となる発注量 Q^* は, 発注コストと保管コストの交差点である. すなわち, 発注コストと保管コストを等しくする発注量は最適な発注量 Q^* である. この性質は以下の式からわかる.

$$\frac{Kd}{Q^*} = \frac{hQ^*}{2} \Longrightarrow Q^* = \sqrt{\frac{2Kd}{h}}$$

また, この図からもわかるように, Q の増加に伴い, 平均発注コストは減少し, 平均保管コストは増加する. 逆もまた然りである.

例 2.2. ある電気量販店では, 毎月 250 台の PC が販売されている. 発注費用は 5000 円, 保管費用は 1 台あたり月 150 円, 購入単価は 10 万円とする. このとき, 最適発注量 Q^* は次のように求められる.

$$Q^* = \sqrt{\frac{2 \cdot 5000 \cdot 250}{150}}$$

Excel では, 下記のように計算できる.

```
=SQRT(2 * 5000 * 250 / 150)
```

Python では, 次のように `eoq(K, d, h)` 関数を定義し, 最適発注量を計算できる.

```
def eoq(K, d, h):
    """
    Calculate the Economic Order Quantity (EOQ).
```

```
    Parameters:
    K (float): Order cost
    d (float): Demand rate
    h (float): Holding cost

    Returns:
    float: Optimal order quantity Q*
    """
    return np.sqrt(2 * K * d / h)


if __name__ == "__main__":
    K = 5000  # Order cost
    d = 250  # Demand rate (units per month)
    h = 150  # Holding cost (per unit per month)

    Q_star = eoq(K, d, h)
    T_star = Q_star / d
    print(f"Optimal Order Quantity (Q*): {Q_star:.2f}")
    print(f"Optimal Cycle Time (T*): {T_star:.2f}")
```

```
Optimal Order Quantity (Q*): 129.10
Optimal Cycle Time (T*): 0.52
```

PC の場合は, 注文量が整数である必要があるため, $g(129)$ と $g(130)$ を比較して最適発注量を決定する.

2.4 リードタイム*

EOQ モデルでは, リードタイムは 0 と仮定している. リードタイムが $L > 0$ の場合も, 最適発注量 Q^* も変換せず, L 期間前に Q^* を発注すればよい.

ここでは, r を発注点(reorder point)とする. 在庫量が r になったときに発注を行う. リードタイム L の間に需要が dL 個あるため, 発注点は次のように表される.

$$r = dL$$

例 2.3. 上の例で, リードタイムが一週間とし, 一か月を 4 週間とすると, リードタイムは $L = 1/4$ となる. したがって, 発注点は次のように求められる.

$$r = dL = 250 \times \frac{1}{4} = 62.5$$

PC の在庫量が 63 台になったときに発注を行う.

2.5　他の EOQ モデル*

- バックオーダーを考慮した EOQ モデル
- 数量割引(quantity discount)を考慮した EOQ モデル
  - 総量割引(all-units discount)
  - 増分割引(incremental discount)

2.6　文献案内

オペレーションズ・リサーチに関する教科書の多くは，EOQ モデルを取り扱っている. モデルの分類から,「deterministic continuous-review inventory models」などの章で説明されていることが多い.

リードタイムを考慮した EOQ モデルについては，Snyder と Shen (2019) で説明されている.

数量割引を考慮した EOQ モデルについては, Snyder と Shen (2019) , Camm ほか (2022) で説明されている.

バックオーダーを考慮した EOQ モデルについては，Snyder と Shen (2019) , Camm ほか (2022) , Hillier と Lieberman (2025) で説明されている.

2.7　練習問題

練習 2.1. ある工場は, 鋼材を毎日 16 トン消費し, 年間 250 日稼働している. 鋼材の購入単価は 1 トンあたり 1100 ドル, 1 回の発注にかかる固定費は 5500 ドル, 保管費は鋼材 1 トンあたり年間は 275 ドルである. EOQ モデルを用いて, 次の問いに答えよ.

1. 最適な発注量を求めよ.
2. 最適なサイクル期間を求めよ.
3. 年間の平均費用を求めよ.

解答 2.1. 需要率は $d = 4000$ で, 発注費用は $K = 5500$, 保管費用は $h = 275$, 購入単価は $c = 1100$ である.

経済的発注量 Q^* は

$$Q^* = \sqrt{\frac{2Kd}{h}} = \sqrt{\frac{2 \cdot 5500 \cdot 4000}{275}} = 400 \text{ トン}$$

となる. 式 2.2 より, サイクル期間 T^* は

$$T^* = \frac{Q^*}{d} = \frac{400}{4000} = 0.1 \text{ 年}$$

となる. 年間平均コスト $g(Q^*)$ は

$$g(Q^*) = \frac{Kd}{Q^*} + cd + \frac{hQ^*}{2} = 4510000 \text{ ドル}$$

となる.

```python
import numpy as np

K = 5500  # Order cost
d = 4000  # Demand rate (units per year)
h = 275  # Holding cost (per unit per year)
c = 1100  # Purchase cost (per unit)

Q_star = np.sqrt(2 * K * d / h)
T_star = Q_star / d
g_avg = (K * d / Q_star) + (c * d) + (h * Q_star / 2)
print(f"Optimal Order Quantity (Q*): {Q_star:.2f} tons")
print(f"Optimal Cycle Time (T*): {T_star:.2f} years")
print(f"Average Annual Cost (g(Q*)): {g_avg:.2f} dollars")
```

```
Optimal Order Quantity (Q*): 400.00 tons
Optimal Cycle Time (T*): 0.10 years
Average Annual Cost (g(Q*)): 4510000.00 dollars
```

練習 2.2. ある家電量販店では, あるスマートフォンを毎週 50 台販売している. 店舗はスマートフォンを 1 台あたりメーカーに 10 万円で購入している. 毎回の発注には発注処理や配送などで 5000 円の固定費がかかる. また, スマートフォン 1 台あたりの保管費用は 1 週間で 100 円である. EOQ モデルを用いて, 次の問いに答えよ.

1. 最適な発注量を求めよ.
2. 最適なサイクル期間を求めよ.

解答 2.2. 需要率は $d = 50$ で, 発注費用は $K = 5000$, 保管費用は $h = 100$ である.

経済的発注量 Q^* は

$$Q^* = \sqrt{\frac{2Kd}{h}} = \sqrt{\frac{2 \cdot 5000 \cdot 50}{100}} \approx 70.71 \text{ 台}$$

となる. スマートフォンの場合は, 注文量が整数である必要があるため, $g(70)$ と $g(71)$ を比較して最適発注量を決定する.

$$g(70) = \frac{5000 \cdot 50}{70} + 100 \cdot 70/2 \approx 7071.43$$
$$g(71) = \frac{5000 \cdot 50}{71} + 100 \cdot 71/2 \approx 7071.13$$

$g(71) < g(70)$ であるため, 最適発注量は $Q^* = 71$ 台となる.

サイクル期間 T^* は

$$T^* = \frac{Q^*}{d} = \frac{71}{50} = 1.42 \text{ 週間}$$

となる.

```python
import numpy as np

K = 5000  # Order cost
d = 50  # Demand rate (units per week)
h = 100  # Holding cost (per unit per week)
Q_star = np.sqrt(2 * K * d / h)


# Since order quantity must be an integer, compare g(70) and g(71)
def g(Q):
    return (K * d / Q) + (h * Q / 2)


g_70 = g(70)
g_71 = g(71)
optimal_Q = 70 if g_70 < g_71 else 71
T_star = optimal_Q / d
print(f"Optimal Order Quantity (Q*): {optimal_Q} units")
print(f"Optimal Cycle Time (T*): {T_star:.2f} weeks")
```

```
Optimal Order Quantity (Q*): 71 units
Optimal Cycle Time (T*): 1.42 weeks
```

第3章　安全在庫

需要 D がある確率分布に従うと仮定する。リードタイムを L とし、既知の定数とする。発注費用を K、単位あたりの保管費用を h とする。在庫量が連続的に観測され、いつでも発注が可能であるとする**連続観測**の場合を考える。

在庫管理には，(r, Q) 方策が用いられるとする。在庫量が発注点 r 以下になったときに、発注量 Q を発注する。この場合、与えられたサービスレベルを満たすように、発注点 r と発注量 Q を決定することが目的である。

> **ノート**
>
> リードタイム $L = 0$, 需要 d が一定の場合，EOQ モデルに帰着する。最適発注量は $Q^* = \sqrt{2Kd/h}$, 最適な発注点は $r^* = 0$ である。

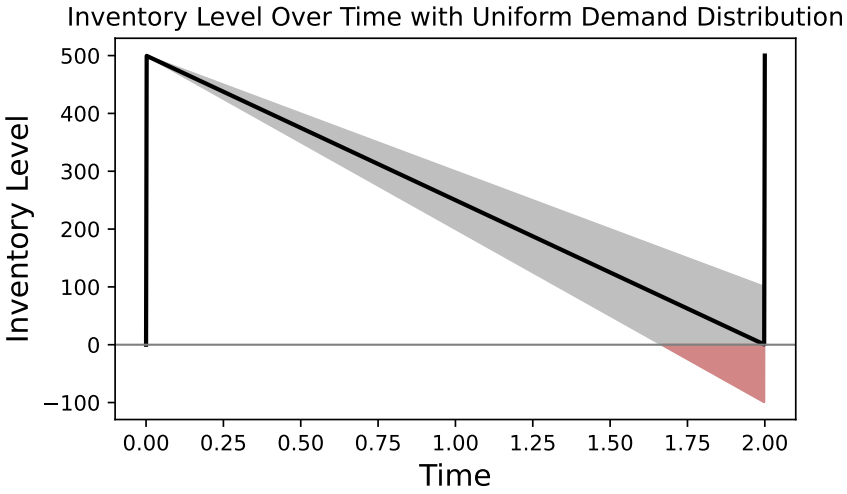
例 3.1 (思考実験)**.** 毎日の需要 D が連続一様分布 $U(200, 300)$ に従うと仮定する。毎日の平均需要は $\mu = 250$ である。以下の問題を考えよ。

1. 二日間の需要が従う分布を求めよ。
2. 初期在庫量は 500 であるとき, 欠品が発生する確率を求めよ。
3. 100% のサービスレベルを達成するために必要な初期在庫量を求めよ。

次の図は、在庫量の時間的変化を示す。灰色の領域は、需要の範囲を示す。赤い領域は、在庫量が 0 以下になったときの欠品を示す。黒い線は平均需要に基づく在庫量の変化を示す。

```
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt

# Parameters
d_mean = 250  # Mean demand rate
d_max = 300  # Max demand rate
d_min = 200  # Min demand rate
Q = 500  # Order quantity
T = Q / d_mean  # Average cycle length
```

```
# Simulate over multiple cycles to show repeated pattern
n_cycles = 1
t = np.linspace(0, n_cycles * T, 1000)

# Inventory levels: linear depletion over time
inventory_mean = Q - d_mean * (t % T)
inventory_mean[0] = 0
inventory_max = Q - d_max * (t % T)
inventory_min = Q - d_min * (t % T)

# Plotting
plt.plot(t, inventory_mean, label="Mean Demand", color="black", linewidth=2)
plt.fill_between(
    t, inventory_min, inventory_max, color="gray", alpha=0.5, label="Demand Range"
)

# Highlight when inventory drops below 0 (shortage)
plt.fill_between(
    t,
    inventory_max,
    0,
    where=(inventory_max < 0),
    color="red",
    alpha=0.3,
    label="Shortage",
)

# Aesthetics
plt.axhline(0, color="gray", linewidth=1)
plt.xlabel("Time", fontsize=14)
plt.ylabel("Inventory Level", fontsize=14)
plt.title("Inventory Level Over Time with Uniform Demand Distribution")
plt.tight_layout()
plt.show()
```



3.1 近似解法

> **ノート**
>
> この問題の定式化および厳密解法は、ここでは説明しない。Snyder と Shen (2019) の「Fundamentals of Supply Chain Theory」などの文献を参照されたい。以下は (r, Q) の近似解法を紹介する。

単位期間あたりの需要を D とし、D は正規分布 $N(\mu, \sigma^2)$ に従うと仮定する。

3.1.1 発注量 Q

D の平均需要 μ を EOQ モデルの需要率とみなすと、発注量 Q は次のように求めることができる (Camm ほか 2022)。

$$Q = \sqrt{\frac{2K\mu}{h}}$$

得られた発注量 Q は、最適解ではなく、近似解であることに注意されたい。

例 3.2. 単位期間あたりの需要 D が正規分布 $N(100, 20^2)$ に従うと仮定する。発注費用 $K = 200$、単位あたりの保管費用 $h = 5$ のとき、発注量 Q を求めよ。

$$Q = \sqrt{\frac{2K\mu}{h}} = \sqrt{\frac{2 \cdot 200 \cdot 100}{5}} \approx 89.44$$

Python では、以下のように計算できる。

```
import math

K = 200
h = 5
mu = 100
Q = math.sqrt((2 * K * mu) / h)
print(f"order quantity: {Q:.2f}")
```

```
order quantity: 89.44
```

3.1.2　発注点 r

リードタイム期間中に発生する需要は $D_L \sim N(\mu_L, \sigma_L^2)$ とし、正規分布の再生性により、

$$\mu_L = \mu L, \quad \sigma_L^2 = \sigma^2 L$$

になる。すなわち、リードタイム期間中の平均需要は $\mu_L = \mu L$、標準偏差は $\sigma_L = \sigma\sqrt{L}$ である。

> **正規分布の再生性**
>
> $X_1, X_2, \ldots, X_n$ が独立に同一の正規分布 $N(\mu, \sigma^2)$ に従うならば、$Y = X_1 + X_2 + \cdots + X_n$ は正規分布 $N(n\mu, n\sigma^2)$ に従う。

例 3.3. 単位期間あたりの需要 D が正規分布 $N(100, 20^2)$ に従うと仮定する。リードタイム $L = 4$ のとき、リードタイム期間中の平均需要 μ_L と標準偏差 σ_L を求めよ。

$$\mu_L = \mu L = 100 \cdot 4 = 400$$

$$\sigma_L = \sigma\sqrt{L} = 20\sqrt{4} = 40$$

これにより, リードタイム期間中の需要 D_L は正規分布 $N(400, 40^2)$ に従うことがわかる。

発注点 r を決めるためには、**サービスレベル**（service level）を考える。ここでは、サービスレベルを、リードタイム期間中に需要を満たす確率と定義する。サービスレベルを α とし、$0 < \alpha < 1$ とする。

与えられたサービスレベル α に対して、D_L が発注点 r 以下になる確率（欠品が発生しない確率、つまり、サービスレベル）が α になるように発注点 r を決定する。

$$P(D_L \leq r) = \alpha$$

もし、発注点 $r = \mu_L$ とすると、$P(D_L \leq \mu_L) = 0.5$ となる。すなわち、50% の確率で欠品が発生することになる。

> **ノート**
>
> $$P(D_L \leq \mu_L) = P\left(\frac{D_L - \mu_L}{\sigma_L} \leq 0\right) = \Phi(0) = 0.5$$

したがって、サービスレベル $\alpha > 0.5$ の場合、発注点 r は平均需要 μ_L より大きくなる必要がある。$r - \mu_L$ を**安全在庫**（safety stock）と呼び、s と表す。

$$s = r - \mu_L$$

この式を変形すると、発注点 r は次のように表される。

$$r = \mu_L + s$$

従って、サービスレベル $P(D_L \leq r) = \alpha$ は次のように表される。

$$P(D_L \leq \mu_L + s) = \alpha$$

与えられたサービスレベル α に対して、安全在庫 s を求めることを考える。ここで、$s \geq 0$ とする。

$$P(D_L - \mu_L \le s) = \alpha \tag{3.1}$$

$$P\left(\frac{D_L - \mu_L}{\sigma_L} \le \frac{s}{\sigma_L}\right) = \alpha \tag{3.2}$$

$$\Phi\left(\frac{s}{\sigma_L}\right) = \alpha \tag{3.3}$$

$$\frac{s}{\sigma_L} = \Phi^{-1}(\alpha) \tag{3.4}$$

$$s = \sigma_L \Phi^{-1}(\alpha) \tag{3.5}$$

$$s = \sigma\sqrt{L}\Phi^{-1}(\alpha) \tag{3.6}$$

ここで、$\Phi(\cdot)$ は標準正規分布の累積分布関数であり、$\Phi^{-1}(\alpha)$ はその逆関数である。したがって、発注点 r は次のように表される。

$$r = \mu_L + s = \mu L + \sigma\sqrt{L}\Phi^{-1}(\alpha)$$

$\Phi^{-1}(\alpha)$ は標準正規分布表、Excel、Python などを用いて求めることができる。

例 3.4. 単位期間あたりの需要 D が正規分布 $N(100, 20^2)$ に従うと仮定する。リードタイム $L = 4$、発注点 $r = 500$ のとき、安全在庫 s とサービスレベル α を求めよ。

リードタイム期間中の平均需要は $\mu_L = 400$ である。したがって、安全在庫 s は次のように求められる。

$$s = r - \mu_L = 500 - 400 = 100$$

サービスレベル α は次のように求められる。

$$\alpha = P(D_L \le r) = \Phi\left(\frac{r - \mu_L}{\sigma_L}\right) = \Phi\left(\frac{100}{40}\right) = \Phi(2.5) \approx 0.99379$$

発注点 $r = 500$ のとき、安全在庫 s は 100、サービスレベル α は約 99.379% である。

例 3.5. リードタイム $L = 4$、平均需要 $\mu = 100$、需要の標準偏差 $\sigma = 20$、サービスレベル $\alpha = 0.95$ のとき、発注点 r と安全在庫 s を求める。

リードタイム期間中の平均需要と標準偏差は次のように計算される。

$$\mu_L = \mu L = 100 \cdot 4 = 400 \tag{3.7}$$
$$\sigma_L = \sigma\sqrt{L} = 20\sqrt{4} = 40 \tag{3.8}$$
$$\tag{3.9}$$

標準正規分布表から $\Phi^{-1}(0.95) \approx 1.64485$ を得る。これを用いて安全在庫 s と発注点 r を求める。

$$s = \sigma_L \Phi^{-1}(0.95) \approx 40 \cdot 1.64485 \approx 65.79 \tag{3.10}$$
$$r = \mu_L + s \approx 400 + 65.79 \approx 465.79 \tag{3.11}$$

したがって、発注点 r は約 465.79、必要な安全在庫 s は約 65.79 となる。

Python では、以下のように計算できる。

```python
from scipy.stats import norm

L = 4
mu = 100
sigma = 20
alpha = 0.95

mu_L = mu * L
sigma_L = sigma * (L ** 0.5)

s = sigma_L * norm.ppf(alpha)
r = mu_L + s

print(f"reorder point: {r:.2f},  safety stock: {s:.2f}")
```

```
reorder point: 465.79,  safety stock: 65.79
```

3.2 欠品費用を考慮する場合*

また、欠品費用も考慮する場合、バックオーダーを考慮した EOQ モデルを用いて、発注量 Q は次のように求めることができる (Hillier と Lieberman 2025)。

$$Q = \sqrt{\frac{2K\mu}{h}}\sqrt{\frac{p+h}{p}}$$

p は単位あたりの欠品費用である。

3.3　文献案内

3.4　練習問題

練習 3.1. ある会社は、A 商品を販売している。(r, Q) 方策に基づいて在庫管理を行っている。毎日の需要 D は正規分布 $N(150, 30^2)$ に従う。商品を発注するための固定費用は $K = 300$、単位あたりの保管費用は $h = 4$, 購入単価は $c = 20$ である。リードタイムは $L = 4$ 日である。この会社は，90% のサービスレベルを目標としている。このとき、発注量 Q、発注点 r、必要な安全在庫 s を求めよ。

解答 3.1. 発注量 Q は次のように求められる。

$$Q = \sqrt{\frac{2K\mu}{h}} = \sqrt{\frac{2 \cdot 300 \cdot 150}{4}} = 150$$

リードタイム期間中の平均需要と標準偏差は次のように計算される。

$$\mu_L = \mu L = 150 \cdot 4 = 600$$

$$\sigma_L = \sigma\sqrt{L} = 30\sqrt{4} = 60$$

標準正規分布表から $\Phi^{-1}(0.9) \approx 1.28155$ を得る。これを用いて安全在庫 s と発注点 r を求める。

$$s = \sigma_L \Phi^{-1}(0.9) \approx 60 \cdot 1.28155 \approx 76.89$$

$$r = \mu_L + s \approx 600 + 76.89 \approx 676.89$$

したがって、発注量 Q は 150、発注点 r は約 676.89、必要な安全在庫 s は約 76.89 となる。

```
import math
from scipy.stats import norm
K = 300
h = 4
mu = 150
sigma = 30
L = 4
alpha = 0.9
Q = math.sqrt((2 * K * mu) / h)
mu_L = mu * L
sigma_L = sigma * (L ** 0.5)
s = sigma_L * norm.ppf(alpha)
r = mu_L + s
print(f"order quantity: {Q:.2f}, reorder point: {r:.2f},  safety stock: {s:.2f}")
```

```
order quantity: 150.00, reorder point: 676.89,  safety stock: 76.89
```

第4章　Wagner-Whitin モデル

記号	意味
$\mathcal{T}$	期間の集合、$\mathcal{T} = \{1, 2, \dots, T\}$
K	1 回あたりの発注費用
h	単位あたりの保管費用
d_t	第 t 期の需要量
q_t	第 t 期の発注量
x_t	第 t 期の在庫量
y_t	第 t 期に発注する場合は 1、しない場合は 0
M	非負の大きな数

Wagner-Whitin モデルは次のように定式化される。

$$\text{minimize} \quad \sum_{t=1}^{T}(Ky_t + hx_t) \tag{4.1}$$

$$\text{subject to} \quad x_t = x_{t-1} + q_t - d_t \qquad \forall t \in \mathcal{T} \tag{4.2}$$

$$q_t \geq 0 \qquad \forall t \in \mathcal{T} \tag{4.3}$$

$$q_t \leq My_t \qquad \forall t \in \mathcal{T} \tag{4.4}$$

$$x_t \geq 0 \qquad \forall t \in \mathcal{T} \tag{4.5}$$

$$y_t \in \{0, 1\} \qquad \forall t \in \mathcal{T} \tag{4.6}$$

第5章　新聞売り子問題

これまで紹介した EOQ 在庫モデルは、需要が決定論的であると仮定していた。ここからは、需要が確率的であると仮定した在庫モデルを紹介する。

新聞売り子問題（Newsvendor Problem）は, 古典的な確率的在庫モデルの一つである。新聞は次の日には売れなくなるため, 新聞売り子問題は最も単純な perishable 在庫モデルとして知られている。新聞に限らず, 食品や花などの生鮮品の在庫管理にも応用される。

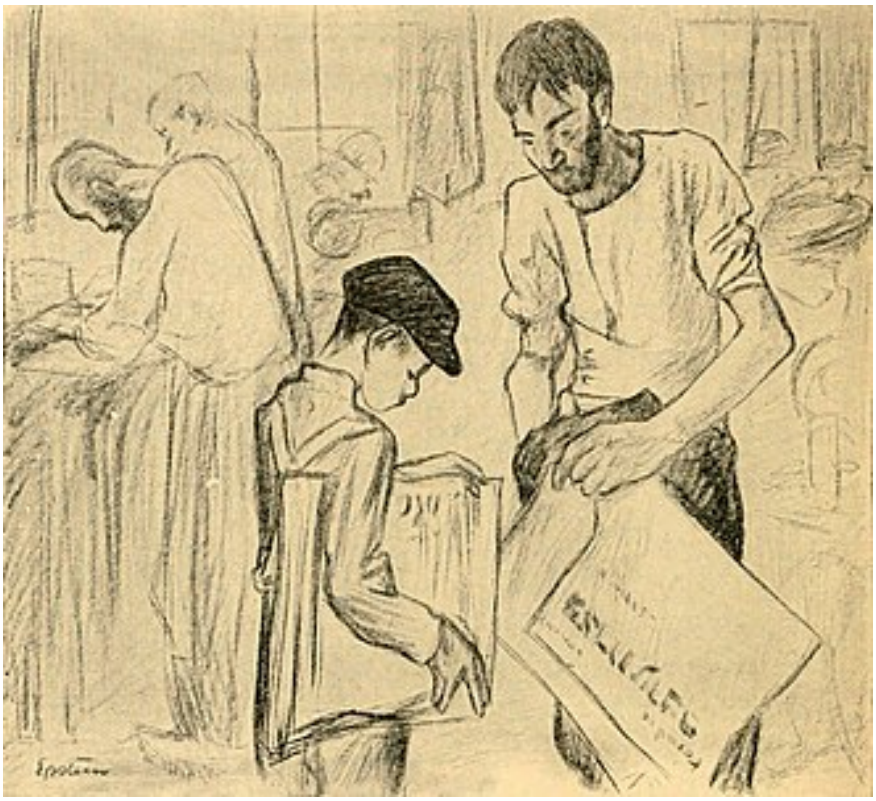


図 5.1: Jacob Epstein, Buying a Newspaper. The Spirit of the Ghetto, 1902, public domain, Wikimedia Commons

新聞売り子が新聞を仕入れ, 販売する問題を考える。新聞 1 部の欠品費用を p, 保管費用を h とする。新聞売子問題において, p を**在庫不足費用** (underage cost)，h を**在庫超過費用** (overage cost)とも呼ぶ。$p0$，$h0$ とし, 初期在庫は 0 とする。

新聞の需要 D を確率変数とするとき, 新聞売り子はどれだけの新聞を発注すればよいかという問題である。

例 5.1 (思考実験)**.** 需要 D が 一様分布 $U(100, 300)$ に従うとする。以下のそれぞれの場合, 新聞売り子がどれだけの新聞を発注するか考えよ。

- 在庫超過費用 $h = 1000$, 在庫不足費用 $p = 0.1$
- 在庫超過費用 $h = 0.1$, 在庫不足費用 $p = 1000$
- 在庫超過費用 $h = 10$, 在庫不足費用 $p = 10$

5.1　記号

記号	意味
h	1 個当たりの在庫超過費用
p	1 個当たりの在庫不足費用
D	需要(確率変数)
d	需要の観測値
$f_D(d)$	需要 D の確率密度関数
$F_D(d)$	需要 D の累積分布関数
S	発注量
$g(S, d)$	発注量 S, 需要の観測値 d に対するコスト
$g(S)$	$g(S, d)$ の期待値，$g(S) = \mathbb{E}[g(S, D)]$

5.2　定式化

新聞売り子が S 部の新聞を仕入れ, 需要 D が d であったとする。このとき, 新聞売り子のコスト $g(S, d)$ は以下のように表される。

$$g(S, d) = h(S - d)^+ + p(d - S)^+ \tag{5.1}$$

例 5.2. 発注量 $S = 100$, 需要 D の観測値 $d = 120$, 在庫超過費用が $h = 10$, 在庫不足費用が $p = 5$ のとき, コスト $g(S, d)$ は以下のように求められる。

$$g(100, 120) = 10(100 - 120)^+ + 5(120 - 100)^+ = 100$$

観測値 d が分かれば, コスト $g(S,d)$ を計算できる。しかし, 需要 D が確率変数であることを思い出そう。ここからは, 需要 D が連続型確率変数であると仮定する。

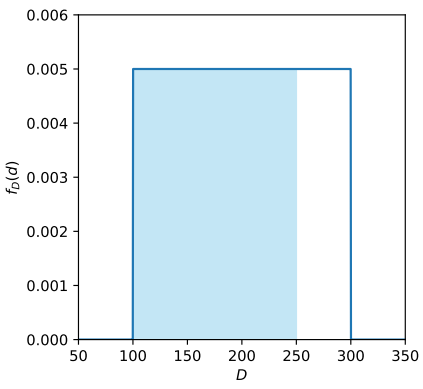
需要 D の確率密度関数を $f_D(d)$, 累積分布関数を $F_D(d)$ とする。発注量を S としたとき, 需要 $D \leq S$ である確率は, 累積分布関数 $F_D(S)$ で与えられる。

$$P(D \leq S) = F_D(S) = \int_0^S f_D(d)dd \tag{5.2}$$

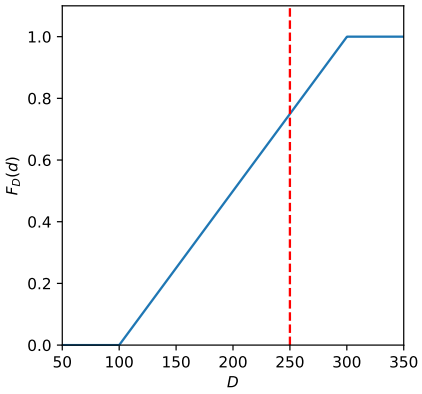
例 5.3. 需要 D が一様分布 $U(100, 300)$ に従うとき, 確率密度関数 $f_D(250)$ と累積分布関数 $F_D(250)$ は以下のように表される。

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import uniform

# 一様分布のパラメータ
a = 100
b = 300
# 需要の範囲
d = np.linspace(a - 50, b + 50, 1000)
# plot pdf with shaded area
plt.figure(figsize=(4, 4))
plt.plot(d, uniform.pdf(d, a, b - a))
plt.fill_between(
    d,
    0,
    uniform.pdf(d, a, b - a),
    where=(d <= 250),
    color="skyblue",
    alpha=0.5,
    label="F(250)",
)
plt.xlabel(r"$D$")
plt.ylabel(r"$f_D(d)$")
plt.xlim(a - 50, b + 50)
plt.ylim(0, 0.006)
plt.show()

# plot cdf
```

```
plt.figure(figsize=(4, 4))
plt.plot(d, uniform.cdf(d, a, b - a))
plt.axvline(250, color="r", linestyle="--", label="d=250")
plt.xlabel(r"$D$")
plt.ylabel(r"$F_D(d)$")
plt.xlim(a - 50, b + 50)
plt.ylim(0, 1.1)
plt.show()
```



(a) 確率密度関数 $f_D(d)$



(a) 累積分布関数 $F_D(d)$

需要 $D \leq 250$ である確率は, 累積分布関数 $F_D(250)$ で与えられる。

$$P(D \leq 250) = F_D(250) = \int_{100}^{250} f_D(d)dd = 0.75$$

需要 $D > 250$ である確率は $1 - F_D(250)$ で与えられる。

$$P(D > 250) = 1 - F_D(250) = 1 - \int_{100}^{250} f_D(d)dd = 0.25$$

D が確率変数であるため, コストも確率変数となる。新聞売り子問題の目的は, コストの期待値を最小化することである。

ここで, コストの期待値を $g(S) = \mathbb{E}[g(S, D)]$ とする。$g(S, D)$ は 式 5.1 で与えられるため, コストの期待値 $g(S)$ は以下のように表される。

$$
\begin{aligned}
g(S) &= \mathbb{E}[g(S, D)] && (5.3)\\
&= \mathbb{E}[h(S-D)^+ + p(D-S)^+] && (5.4)\\
&= \mathbb{E}[h(S-D)^+] + \mathbb{E}[p(D-S)^+] && (5.5)\\
&= h\mathbb{E}[(S-D)^+] + p\mathbb{E}[(D-S)^+] && (5.6)\\
& && (5.7)
\end{aligned}
$$

この式により, コストの期待値 $g(S)$ は，$(S-D)^+$ の期待値かける在庫超過費用 h と，$(D-S)^+$ の期待値かける在庫不足費用 p の和であることが分かる。

期待値の線形性(Linearity of Expectation)

X と Y を確率変数，a と b を定数とする。このとき，

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

$$\mathbb{E}[aX+b] = a\mathbb{E}[X] + b$$

が成り立つ。

需要 D が連続型確率変数であるため，$\mathbb{E}[(S-D)^+]$ と $\mathbb{E}[(D-S)^+]$ は以下のように表される。

$$\mathbb{E}[(S-D)^+] = \int_0^\infty (S-d)^+ f_D(d)dd = \int_0^S (S-d) f_D(d)dd \quad (5.8)$$

$$\mathbb{E}[(D-S)^+] = \int_0^\infty (d-S)^+ f_D(d)dd = \int_S^\infty (d-S) f_D(d)dd \quad (5.9)$$

連続型確率変数の期待値

X を連続型確率変数，$f_X(x)$ を X の確率密度関数とする。このとき，X の期待値 $\mathbb{E}[X]$ は以下のように表される。

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x)dx$$

$\mathbb{E}[g(X)]$ は以下のように表される。

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)dx$$

したがって, コストの期待値 $g(S)$ は以下のように表される。

$$g(S) = \mathbb{E}[g(S, D)] \tag{5.10}$$
$$= h\mathbb{E}[(S-D)^+] + p\mathbb{E}[(D-S)^+] \tag{5.11}$$
$$= h\int_0^S (S-d)f_D(d)dd + p\int_S^\infty (d-S)f_D(d)dd \tag{5.12}$$

例 5.4. 在庫超過費用が $h = 10$, 在庫不足費用が $p = 5$ のとき, 発注量 $S = 150$ に対するコストの期待値 $g(150)$ は以下のように求められる。

$$g(150) = 10\int_0^{150} (150-d)f_D(d)dd + 5\int_{150}^\infty (d-150)f_D(d)dd$$

5.3　最適化

以上の議論から, 新聞売り子問題は, コストの期待値 $g(S)$ を最小化する発注量 S を求める問題に帰着される。手順は以下の通りである。

1. $g(S)$ の 1 階微分を求める。
2. $dg(S)/dS = 0$ を解く。
3. 2 階微分を求め，$g(S)$ が凸関数であることを確認する。

式 5.2 を用い，$g(S)$ の 1 階微分は以下のように求める。

$$\frac{dg(S)}{dS} = h\int_0^S f_D(d)dd - p\int_S^\infty f_D(d)dd \tag{5.13}$$
$$= hF_D(S) - p(1-F_D(S)) \tag{5.14}$$
$$\tag{5.15}$$

よって，$dg(S)/dS = 0$ から，

$$hF_D(S) - p(1-F_D(S)) = 0 \tag{5.16}$$
$$F_D(S) = \frac{p}{h+p} \tag{5.17}$$

になる。2 階微分は

$$\frac{d^2g(S)}{dS^2} = hf_D(S) + pf_D(S) \tag{5.18}$$

$$= (h+p)f_D(S) > 0 \tag{5.19}$$

である。したがって, コスト関数 $g(S)$ は凸関数であり，1 階微分が 0 になる点は最小値を与える。

> **ヒント**
>
> $p > 0$，$h > 0$，$f_D(S) > 0$ より，$(h+p)f_D(S) > 0$ が成り立つ。

コスト関数 $g(S)$ を最小化するための最適発注量 S^* は

$$S^* = F_D^{-1}\left(\frac{p}{h+p}\right)$$

となる。ここで，F_D^{-1} は需要 D の累積分布関数の逆関数である。

> **逆関数(Inverse Function)**
>
> ある関数 $y = f(x)$ に対し, 次の条件を満たす関数 $x = f^{-1}(y)$ を $f(x)$ の逆関数と呼ぶ。
>
> $$f(f^{-1}(y)) = y, \quad f^{-1}(f(x)) = x$$

定理 5.1. 新聞売り子問題における最適発注量 S^* は,

$$S^* = F_D^{-1}\left(\frac{p}{h+p}\right)$$

で与えられる。

5.3.1 正規分布の場合

需要 D が正規分布 $N(\mu, \sigma^2)$ に従うとき, 新聞売り子問題の最適発注量 S^* は以下の式を満たす。

$$F_D(S^*) = \Phi\left(\frac{S^* - \mu}{\sigma}\right) = \frac{p}{h+p}$$

ここで,

$$z = \Phi^{-1}\left(\frac{p}{h+p}\right)$$

とおくと,

$$S^* = \sigma z + \mu$$

で与えられる。標準正規分布表を用いて z を調べることができる。

> **標準正規分布(Standard Normal Distribution)**
>
> 平均 0, 分散 1 の正規分布を**標準正規分布**（standard normal distribution)と呼ぶ。標準正規分布に従う確率変数を Y とすると，$Y \sim N(0,1)$ と表される。標準正規分布の確率密度関数を $\phi(z)$, 累積分布関数を $\Phi(z)$ と表す。
> 与えられた $X \sim N(\mu, \sigma^2)$ の累積分布関数 $F_X(x)$ の値を求めるには, 以下のように変換する。
>
> $$\begin{aligned} F_X(x) &= P(X \leq x) && (5.20)\\ &= P\left(\frac{X-\mu}{\sigma} \leq \frac{x-\mu}{\sigma}\right) && (5.21)\\ & && (5.22)\\ &= P\left(Y \leq \frac{x-\mu}{\sigma}\right) && (5.23)\\ & && (5.24)\\ &= \Phi\left(\frac{x-\mu}{\sigma}\right) && (5.25) \end{aligned}$$
>
> 与えられた z に対し，$\Phi(z)$ の値は**標準正規分布表**（standard normal table)を用いて調べることができる。

5.4　臨界率

以上の議論から, 最適発注量 S^* は以下の式を満たす。

$$F_D(S) = P(D \leq S) = \frac{p}{h+p}$$

ここで，$F_D(S) = P(D \leq S)$ は需要 D が発注量 S 以下である確率を表す。言い換えると, 欠品が発生しない確率を表す。この確率のことは**サービスレベ**

ル（service level）と呼ぶ。定理 5.1 は，サービスレベルを $p/(h+p)$ に等しくする発注量 S が最適であることを示している。

この $p/(h+p)$ は**臨界率**（critical ratio）と呼ばれる。

注釈 5.1.

- 在庫不足費用 p の増加に伴い，サービスレベル $p/(h+p)$ は増加し，最適発注量 $S^* = F_D^{-1}(p/(h+p))$ も増加する。
- 在庫超過費用 h の増加に伴い，サービスレベル $p/(h+p)$ は減少し，最適発注量 $S^* = F_D^{-1}(p/(h+p))$ も減少する。
- 直感的に，在庫不足費用が増加すると，欠品を避けるために発注量が増加し，サービスレベルも上昇する。一方，在庫超過費用が増加すると，過剰在庫を避けるために発注量が減少し，サービスレベルも低下する。

5.5 例題

例 5.5. 需要 D が正規分布 $N(100, 25)$ に従う新聞売り子問題を考える。在庫超過費用が $h = 10$，在庫不足費用が $p = 40$ のとき，最適発注量 S^* を求める。

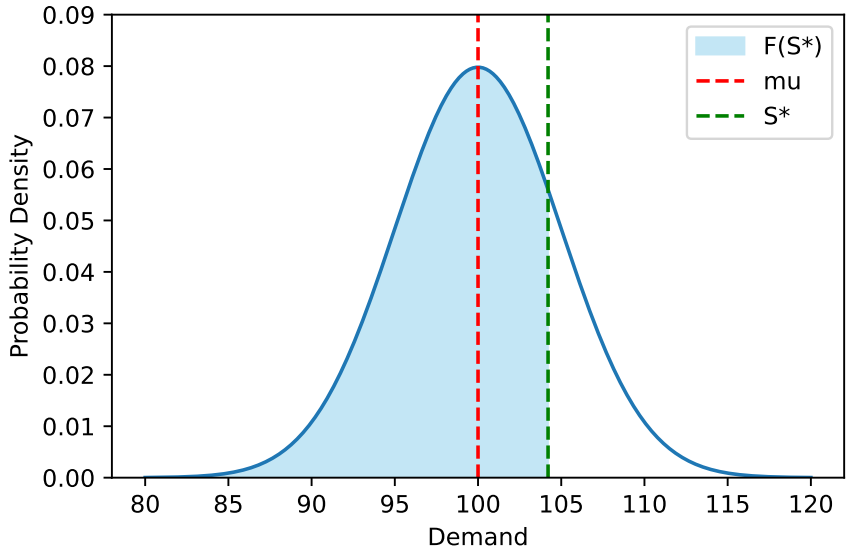
定理 5.1 より，S^* は以下の式で与えられる。

$$S^* = F_D^{-1}\left(\frac{p}{h+p}\right) = F_D^{-1}\left(\frac{40}{10+40}\right) = F_D^{-1}(0.8)$$

言い換えると，$F_D(S^*) = 0.8$ を満たす S^* を求めればよい。これを図で表すと，面積が 0.8 になるような S^* を求めることに相当する。

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm

# 正規分布のパラメータ
mu = 100
sigma = 5

# 需要の範囲
d = np.linspace(mu - 4 * sigma, mu + 4 * sigma, 1000)
S_star = norm.ppf(0.8, loc=mu, scale=sigma)

# plot the normal distribution
```

```
plt.plot(d, norm.pdf(d, mu, sigma))
plt.fill_between(
    d,
    0,
    norm.pdf(d, mu, sigma),
    where=(d <= S_star),
    color="skyblue",
    alpha=0.5,
    label="F(S*)",
)
plt.xlabel("Demand")
plt.ylabel("Probability Density")
plt.axvline(mu, color="r", linestyle="--", label="mu")
plt.axvline(S_star, color="g", linestyle="--", label="S*")
plt.ylim(0, 0.09)
plt.legend()
plt.show()
```



下では，Excel，Python, 標準正規分布表を用いて S^* を求める方法を示す。

5.5.1　標準正規分布表

標準正規分布表を調べると，$\Phi(z) = 0.8$ のとき，z は約 0.84 である。したがって，S^* は

$$S^* = z\sigma + \mu \approx 0.84 \times 5 + 100 = 104.2$$

となる。

5.5.2 Excel

Excel では，NORM.INV(確率, 平均, 標準偏差) 関数を用いて $F_D^{-1}(0.8)$ を求めることができる。

```
=NORM.INV(0.8, 100, 5)
```

5.5.3 Python

Python では，SciPy ライブラリの ppf() 関数を用いて, 逆関数を求めることができる。

```
from scipy.stats import norm

# 正規分布のパラメータ
mu = 100
sigma = 5

# 在庫超過費用と在庫不足費用
h = 10
p = 40

# 臨界率
critical_ratio = p / (h + p)

# 最適発注量
S_star = norm.ppf(critical_ratio, loc=mu, scale=sigma)
print(f"Optimal order quantity S*: {S_star:.2f}")
```

```
Optimal order quantity S*: 104.21
```

5.6 再定式化

新聞 1 部の仕入れ価格を c, 販売価格を r, 残存価額を v, 欠品費用を p, 保管費用を h とする。ここで, 以下の条件を満たすとする。

- $r > c$. 販売価格は仕入れ価格より高い。
- $r > v$. 販売価格は残存価値より高い。

新聞売り子が S 部の新聞を仕入れ, 需要 D が d であったとする。このとき, 新聞売り子の利益 $\pi(S, d)$ は以下のように表される。

5.6.1　コスト関数

$$\pi(S,d) = r\min\{d,S\} - cS + v\max\{0,S-d\} \tag{5.26}$$
$$- h\max\{0,S-d\} - p\max\{0,d-S\} \tag{5.27}$$

第一項は販売利益, 第二項は仕入れコスト, 第三項は残存価値, 第四項は保管コスト, 第五項は欠品コストである。

以下は，$\max\{0,x\} = x^+$ を用いて書き換えた形である。整理すると, 利益は以下のように表される。

$$\pi(S,d) = r\min\{d,S\} - cS + (v-h)(S-d)^+ - p(d-S)^+$$

第一項を以下のように書き換えることができる。

$$r\min\{d,S\} = rd - r(d-S)^+$$

> **ノート**
>
> - $d < S$ の場合，$r\min\{d,S\} = rd$ となる。
> - $d >= S$ の場合，$r\min\{d,S\} = rd - r(d-S) = rS$ となる。

したがって, 利益は以下のように書き換えられる。

$$\pi(S,d) = rd - cS + (v-h)(S-d)^+ - (p+r)(d-S)^+$$

利益の最大化は, コストの最小化に帰着される。したがって, コスト関数 $g(S,d) = -\pi(S,d)$ は以下のように表される。

$$g(S,d) = cS - rd + (h-v)(S-d)^+ + (p+r)(d-S)^+$$

D が確率変数であるため, コストの期待値 $g(S)$ は以下のように表される。

$$g(S) = \mathbb{E}[g(S, D)] \tag{5.28}$$
$$= \int_0^\infty g(S, d) f_D(d) dd \tag{5.29}$$
$$= cS - r\mathbb{E}[D] + (h - v)\mathbb{E}[(S - D)^+] + (p + r)\mathbb{E}[(D - S)^+] \tag{5.30}$$
$$= cS - r\mu + (h - v) \int_0^\infty (S - d)^+ f_D(d) dd + (p + r) \int_0^\infty (d - S)^+ f_D(d) dd \tag{5.31}$$
$$= cS - r\mu + (h - v) \int_0^S (S - d) f_D(d) dd + (p + r) \int_S^\infty (d - S) f_D(d) dd \tag{5.32}$$

5.6.2 最適化

$g(S)$ の 1 階微分は以下のように求める。

$$\frac{dg(S)}{dS} = c + (h - v)F_D(S) - (p + r)(1 - F_D(S)) \tag{5.33}$$
$$\tag{5.34}$$

よって，$dg(S)/dS = 0$ から，

$$c + (h - v)F_D(S) - (p + r)(1 - F_D(S)) = 0 \tag{5.35}$$
$$F_D(S) = \frac{p + r - c}{h + p + r - v} \tag{5.36}$$

になる。2 階微分は

$$\frac{d^2 g(S)}{dS^2} = (h - v) f_D(S) + (p + r) f_D(S) \tag{5.37}$$
$$= (h - v + p + r) f_D(S) \tag{5.38}$$

である。したがって, コスト関数 $g(S)$ は凸関数であり，1 階微分が 0 になる点は最小値を与える。

コスト関数 $g(S)$ を最小化するための最適発注量 S^* は

$$S^* = F_D^{-1}\left(\frac{p+r-c}{h+p+r-v}\right)$$

となる。ここで，F_D^{-1} は需要 D の累積分布関数の逆関数である。

新聞売り子問題において, より一般的に, 在庫超過費用(overage cost)と在庫不足費用(underage cost)を考慮する。

$$C_o = h + c - v, \quad C_u = p + r - c$$

5.6.3　臨界率

在庫超過費用 C_o は, 在庫が余ったときのコストである。1 部の在庫超過に対し, 保管コストと仕入れコストが発生するが, 残存価額が得られないため, $C_o = h + c - v$ となる。

在庫不足費用 C_u は, 在庫が不足したときのコストである。1 部の在庫不足に対し, 欠品コスト p と失われた販売機会の利益 $r - c$ が発生するため, $C_u = p + r - c$ となる。

従って,

$$\begin{aligned} S^* &= F_D^{-1}\left(\frac{p+r-c}{h+p+r-v}\right) \\ &= F_D^{-1}\left(\frac{C_u}{C_o + C_u}\right) \end{aligned}$$

が得られる。

5.7　初期在庫を考慮した新聞売り子問題*

新聞売り子の初期在庫を I とする。$I \leq S^*$ の場合, 最適発注量は $S^* - I$ となる。すなわち, 在庫量を S^* にすればよい。

また, $g(S)$ は凸関数であるため, $I > S^*$ の場合, 何も発注しないことが最適である。

したがって, 最適発注量は

$$Q = \begin{cases} S^* - I, & \text{if } I \leq S^*, \\ 0, & \text{if } I > S^*. \end{cases}$$

となる。

このような発注方式を **Base Stock Policy** (BSP) と呼ぶ。BSP は, 各期間の在庫量を観測し, 在庫量が S^* に引き上げられるように発注する方式である。新聞売り子問題において，BSP は最適な方策であると知られている。

5.8 文献案内

Arrow, Harris, と Marschak (1951) は, 新聞売り子問題を初めて定式化した。

Scarf (1959) の論文では，(s, S) 方策が発注費用を考慮した複数期間の新聞売り子問題において最適であることを示している。

実際には, 需要の分布が不明であることが多い。Huber ほか (2019) は, データ駆動新聞売り子問題(Data-Driven Newsvendor Problem)に関する研究が行われている。

Qin ほか (2011) は, 新聞売り子問題に関する研究をレビューした。

5.9 練習問題

練習 5.1. ある弁当屋では，1 個 500 円で弁当を仕入れ，1 個 800 円で販売している。売れ残った弁当を廃棄する場合，1 個当たり 10 円の廃棄費用がかかる。弁当の需要 D は, 正規分布 $N(50, 8^2)$ に従うとする。このとき, 最適な弁当の発注量を求めよ。

解答 5.1. 弁当 1 個当たりの在庫超過費用 h, 在庫不足費用 p は以下のように求められる。

$$h = 10 + 500 = 510, \quad p = 800 - 500 = 300$$

したがって，S^* は以下のように求められる。

$$S^* = F_D^{-1}\left(\frac{p}{h+p}\right) = F_D^{-1}\left(\frac{300}{510+300}\right) \approx F_D^{-1}\left(0.37\right)$$

標準正規分布表を調べると，$\Phi(z) = 0.37$ のとき，z は約 -0.33 である。したがって，S^* は

$$S^* = -0.33 \times \sigma + \mu = -0.33 \times 8 + 50 \approx 47.36$$

となる。

Python を用いて計算すると, 以下のようになる。

```
from scipy.stats import norm

# 正規分布のパラメータ
mu = 50
sigma = 8
# 在庫超過費用と在庫不足費用
h = 510
p = 300
# 臨界率
critical_ratio = p / (h + p)
# 最適発注量
S_star = norm.ppf(critical_ratio, loc=mu, scale=sigma)
print(f"Optimal order quantity S*: {S_star:.4f}")
```

```
Optimal order quantity S*: 47.3530
```

第6章　まとめ

- 在庫管理の目的は，**発注量**と**発注時期**を決定し, 顧客の需要を満たしつつ, コストを最小化することである.
- 科学的在庫管理では, 在庫モデルを定式化し, 発注量と発注時期を決定する.
- **在庫方策**は, 在庫の状況に応じて, 発注量と発注時期を決定するルールである．(s, S) 方策，(r, Q) 方策，BSP 方策などがある.
- 安全在庫の章では, コストを最小化するのではなく, サービスレベルを満たすための在庫方策を決定する方法を学んだ.

紹介した在庫モデルを分類すると, 次の表のようになる.

モデル名	分類
EOQ モデル	決定論的連続観測モデル
安全在庫	確率的連続観測モデル
Wagner-Whitin モデル	多期間決定論的周期観測モデル
新聞売り子問題	単一期間確率的周期観測モデル

第II部

待ち行列理論

第7章　待ち行列モデルの基礎

学習目標

- ケンドールの記号を理解する.
- 滞在時間, 待ち時間, 系内客数, 待ち行列長などの評価指標を理解する.
- リトルの法則を理解し, 適用できる.
- $G/G/1$，$G/G/c$ 待ち行列の基本的な関係式を理解する.

図 7.1 は, 典型的な待ち行列システムを示している. 待ち行列システムは待ち行列とサーバから構成される. これ以降は, 待ち行列システムを単に**システム**と呼ぶことにする. 図に示すように，**客**（customer）はシステムに**到着**（arrival）し，**待ち行列**（queue）に並び, サービスを受け, 最後にシステムを**退出**（departure）する.

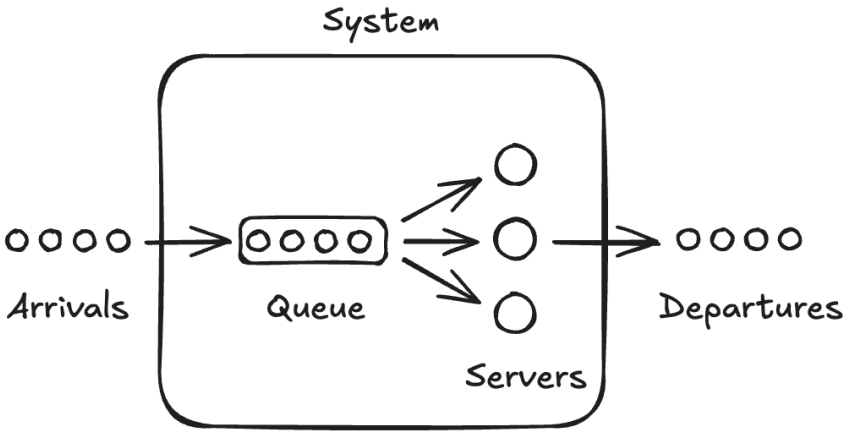


図 7.1: A typical queueing system

7.1　ケンドールの記号

待ち行列モデルは，**ケンドールの記号**（Kendall's notation）により表現する.

$$A/B/X$$

ここで，A は到着間隔分布の種類，B はサービス時間分布の種類，X はサーバ数を表す．A と B には以下の記号が用いられる．

記号	意味
M	指数分布
D	決定論的
G	一般分布
E_k	k 次アーラン分布
H_k	k 次超指数分布

例えば, 客の到着間隔が指数分布に従い, サービス時間が一定であり, サーバ数が 1 である場合, その待ち行列モデルは $M/D/1$ と表される. 本章の最後では，$G/G/1$ 待ち行列と $G/G/c$ 待ち行列について説明する. これらのモデルから得られた一般的な結論は, 他の多くの待ち行列モデルにも適用できる．

> **i** なぜ指数分布は M と書くのか
>
> 1. アーラン分布 E_k と間違えないようにするため．
> 2. 指数分布のマルコフ性（Markovian property）と無記憶性（memoryless property）に由来する．

7.2　評価指標

待ち行列理論の代表的な性能評価指標を紹介する．

滞在時間は客がシステムに到着してサービスを受け, システムを退出するまでにかかる時間である．**待ち時間**は, 客がシステムに到着してからサービスを受けるまでの時間である. 平均滞在時間は W, 平均待ち時間は W_q で表す．

系内客数は待ち行列システム内にいる客数である．**待ち行列長**は待ち行列にいる客数である. 平均系内客数は L, 平均待ち行列長は L_q で表す．

	システム	待ち行列
客数	系内客数	待ち行列長
時間	滞在時間	待ち時間

7.3 リトルの法則

リトルの法則（Little's Law）は, 待ち行列理論における基本的な関係式であり, 平均系内客数 L, 平均到着率 λ, 平均滞在時間 W の関係を示すものである. リトルの法則は $L = \lambda W$ と表される. すなわち, 平均系内客数 L は, 平均到着率 λ と平均滞在時間 W の積に等しい.

例 7.1 (大学の在籍学生数). ある大学の経営システム系では, 毎年 80 人の新入生が入学し, 卒業までに 4 年間在籍する. このとき, 経営システム系に在籍している学生の平均人数を求めよ.

平均到着率 λ は 80 人/年, 平均滞在時間 W は 4 年である. リトルの法則により, 平均系内客数 L は次のように求められる.

$$L = \lambda W = 80 \times 4 = 320 \text{ 人}$$

よって, 経営システム系には 320 人の学生が在籍していることになる.

例 7.2. ある銀行では, 1 時間に平均 20 人の客が来店し, 各客が銀行に滞在する平均時間は 12 分である. このとき, この銀行には平均何人の客がいるか.

平均到着率 λ は 20 人/時間, 平均滞在時間 $W = 12$ 分 $= 12/60$ 時間 である. リトルの法則から, 平均系内客数 L は

$$L = \lambda W = 20 \times \frac{12}{60} = 4 \text{ 人}$$

となる. すなわち, 銀行内には平均して 4 人の客が滞在する. 平均滞在時間 W や平均到着率 λ が増加すると, 平均系内客数 L も増加することがわかる.

時刻 t までに累積到着客数を $A(t)$ とする. 時刻 t における系内客数を $N(t)$ とする. 客 k の滞在時間を $W^(k)$ とする. $t \to \infty$ とすると, 極限が存在するとき, 平均到着率 λ, 平均系内客数 L, 平均滞在時間 W はそれぞれ次の式で定義される.

$$\lambda = \lim_{t\to\infty} \frac{A(t)}{t}, \quad L = \lim_{T\to\infty} \frac{1}{T}\int_0^T N(t)dt, \quad W = \lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} W^{(i)}$$

これまでは, システム全体に関するリトルの法則を紹介したが, 待ち行列を「システム」として考えた場合にも同様の関係が成り立つ. 平均待ち行列長を L_q, 平均待ち時間を W_q, 平均到着率を λ とすると,

$$L_q = \lambda W_q$$

が成り立つ.

7.4　$G/G/1$，$G/G/c$ 待ち行列

$G/G/1$ と $G/G/c$ 待ち行列における一般的な結論を説明する. 記号のまとめを以下に示す.

記号	意味
λ	平均到着率, 単位時間あたりの到着客数の平均
L	平均系内客数
W	平均滞在時間
L_q	平均待ち行列長
W_q	平均待ち時間
S	サービス時間を表す確率変数
μ	平均サービス率, 単位時間あたりに処理できる客数の平均
c	サーバ数
ρ	利用率

7.4.1　サービス率と利用率

S をサービス時間を表す確率変数とする．**平均サービス時間**を $E[S]$ と表す. 単位時間あたりに一つのサーバが処理できる客数の平均は

$$\mu = \frac{1}{E[S]}$$

で計算する．μ を**平均サービス率**（average service rate）と呼ぶ.

> **ノート**
>
> 平均サービス時間は客がサービスを受けるのにかかる時間の平均であり, 平均サービス率は単位時間あたりに処理できる客数の平均である.

利用率 ρ は, 次の式で定義される.

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

ここで，c はサーバ数を表す. 利用率 ρ は, 到着率 λ がシステムの処理能力 $c\mu$ に対してどの程度であるかを示す指標である. システムの忙しさを表す.

システムの使用を開始するとき, システムの状態(系内客数など)は初期状態に大きく依存する. 時間が十分に経過すると, システムの状態は初期状態の影響を受けなくなり, 一定の条件下で**定常状態**（steady state)にある. 定常状態において, システムの状態は一定の確率分布に従い, 時間が経過しても変化しない．**定常状態**が存在するためには，$\rho < 1$ である必要がある.

λ と μ が与えられたとき，$\rho < 1$ という条件を用いて, システムが定常状態に達するためのサーバ数 c の最小値を求めることができる. すなわち,

$$\rho = \frac{\lambda}{c\mu} < 1$$

を満たす最小の整数 c を求めればよい.

例 7.3. あるコールセンターでは, オペレーター 10 人が勤務している．1 件の電話対応に平均して 10 分かかる．1 時間あたり 30 件の電話がかかってくるとする.

1. 社員一人 1 時間あたりに対応できる電話の平均件数を求めよ.
2. このコールセンターの利用率を求めよ.
3. 安定した運用のために必要な最小のサーバ数を求めよ.

平均サービス時間 $E[S] = 10$ 分 $= 1/6$ 時間 である. 平均サービス率 μ は

$$\mu = \frac{1}{E[S]} = \frac{1}{1/6} = 6 \text{ 件/時間}$$

となる. すなわち, 社員一人は 1 時間あたり平均 6 件の電話に対応できることになる.

平均到着率 λ は 30 件/時間, 平均サービス率 μ は 6 件/時間, サーバ数 c は 10 である. 利用率 ρ は次のように計算される.

$$\rho = \frac{30}{10 \times 6} = \frac{30}{60} = 0.5$$

よって, このコールセンターの利用率は 0.5 である.

安定した運用のためには，$\rho < 1$ を満たす必要がある. すなわち,

$$c > \frac{\lambda}{\mu}$$
$$c > 5$$

したがって, 必要な最小のサーバ数は 6 である.

7.4.2　L，W，L_q，W_q の関係

客がシステムに到着してから退出するまでの平均滞在時間 W は, 平均待ち時間 W_q と平均サービス時間 $E[S]$ の和で表される. すなわち,

$$W = W_q + E[S] = W_q + \frac{1}{\mu}$$

である. リトルの法則により,

$$L = \lambda W = \lambda \left(W_q + \frac{1}{\mu} \right) = \lambda W_q + \frac{\lambda}{\mu} = L_q + \frac{\lambda}{\mu}$$

が成り立つ. $L = L_q + \lambda/\mu$ が成り立つから, λ/μ はサービスを受けている客数の平均を表していると言える. また, $\lambda/\mu = L - L_q$ であるから, 稼働中のサーバ数の平均を表しているとも言える.

以上の関係式をまとめると, 次のようになる.

$$L = \lambda W$$
$$W = W_q + \frac{1}{\mu}$$
$$L_q = \lambda W_q$$
$$L = L_q + \frac{\lambda}{\mu}$$

これらの関係式は 4 つの指標 L, W, L_q, W_q のうち, 1 つがわかれば, 他の 3 つを計算できることを示している.

7.5　用語集

英語	日本語
service facility	サービス施設
server	サーバ
number of servers	サーバ数
interarrival time	到着間隔
service time	サービス時間
arrival rate	到着率
service rate	サービス率
offered load	

7.6 演習問題

練習 7.1 (アイスクリーム屋さん). あるアイスクリーム屋さんでは，1 名の店員が客の注文を受け付け, アイスクリームを提供している. 平均して，1 時間に 30 人の客が来店している. 店員は，1 時間に 60 人の客に対応できる. 客の平均待ち時間を 5 分とする.

1. このアイスクリーム屋さんの利用率を求めよ.
2. 客の平均滞在時間, 平均系内客数, 平均待ち行列長を求めよ.

解答 7.1. 平均到着率 λ は 30 人/時間, 平均サービス率 μ は 60 人/時間である. サーバ数 $c = 1$ であるため, 利用率は

$$\rho = \frac{30}{1 \times 60} = 0.5$$

となる.

平均サービス時間 $E[S] = 1$ 分, 平均待ち時間 $W_q = 5$ 分 であるから, 平均滞在時間 W は

$$W = W_q + E[S] = 5 + 1 = 6 \text{ 分}$$

となる. リトルの法則により, 平均系内客数 L は

$$L = \lambda W = 30 \times \frac{6}{60} = 3 \text{ 人}$$

となる. 同様に, 平均待ち行列長 L_q は

$$L_q = \lambda W_q = 30 \times \frac{5}{60} = 2.5 \text{ 人}$$

となる.

第8章　ポアソン過程

$M/M/c$ と $M/M/1$ 待ち行列は, 待ち行列理論において最も基本的なモデルである. これらのモデルでは, 到着間隔とサービス時間が指数分布に従う.

8.1　指数分布

連続型確率変数 X が**指数分布**（exponential distribution）に従うとき，X の確率密度関数は

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

で与えられる. ここで $\lambda > 0$ は分布のパラメータである.

指数分布に従う確率変数 X の平均と分散は

$$E[X] = \frac{1}{\lambda}, \quad \mathrm{Var}(X) = \frac{1}{\lambda^2}$$

である.

X の累積分布関数は

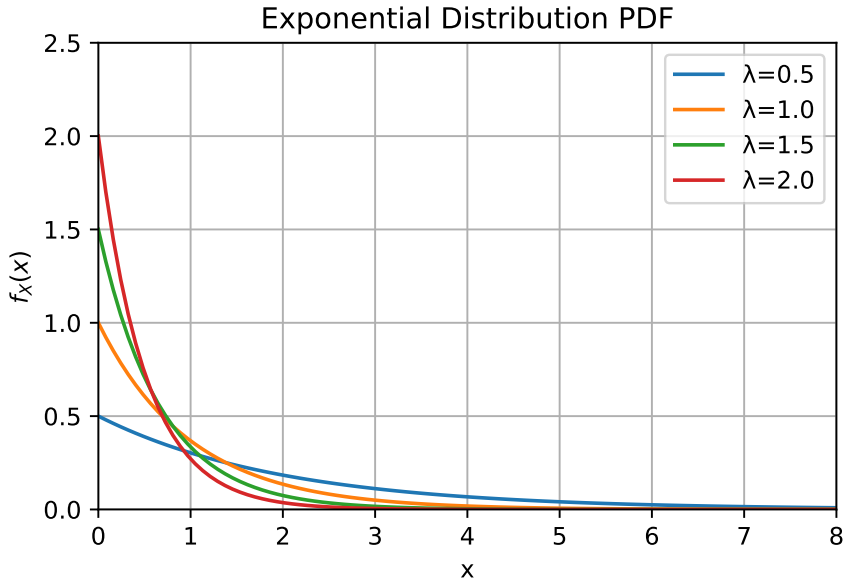
$$F_X(x) = P(X \leq x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

である.

次の図は, パラメータ $\lambda = 0.5, 1.0, 1.5, 2.0$ の場合における指数分布の確率密度関数を示している. 確率密度関数 $f_X(x)$ は x の増加に伴い単調減少する.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import expon

x = np.linspace(0, 8, 100)
```

```
lambda_params = [0.5, 1.0, 1.5, 2.0]
for lam in lambda_params:
    pdf = expon.pdf(x, scale=1 / lam)
    plt.plot(x, pdf, label=f"λ={lam}")
plt.title("Exponential Distribution PDF")
plt.xlabel("x")
plt.ylabel("$f_X(x)$")
plt.xlim(0, 8)
plt.ylim(0, 2.5)
plt.legend()
plt.grid()
plt.show()
```



指数分布に従う確率変数は, 時間を表すことが多い. パラメータ λ は, 単位時間あたりの事象の発生回数の平均を表す.

8.1.1　到着間隔とサービス時間

$M/M/1$ と $M/M/c$ 待ち行列では, 到着間隔とサービス時間が指数分布に従う. 確率変数 T を到着間隔を表すとする. このとき, $P(T \leq t)$ は, 時間 t 以内に次の客が到着する確率を表す. $E(T)$ は, 到着間隔の平均を表す. $\lambda = 1/E(T)$ は, 単位時間あたりの到着する客数の平均を表す. この λ は, 前の章で説明した**平均到着率**である.

例 8.1. ある店舗では，1 時間あたり平均 12 人の客が来店する．到着間隔 T が指数分布に従うと仮定する．次の問いに答えよ．

1. 平均到着間隔 $E[T]$ を求めよ．
2. 2 分以内に次の客が到着する確率を求めよ．

平均到着率は $\lambda = 12$ 人/時間である．平均到着間隔は

$$E[T] = 1/\lambda = 1/12 \text{ 時間} = 5 \text{ 分}$$

である．

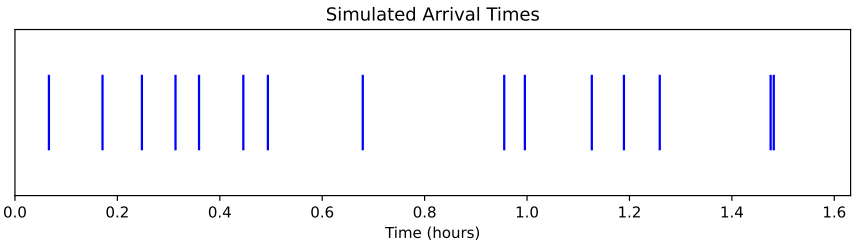
単位を時間に揃え，$P(T \leq 1/30)$ を計算する．

$$P(T \leq 1/30) = F_T(1/30) = 1 - e^{-12\times(1/30)} \approx 0.3297$$

よって，2 分以内に次の客が到着する確率は約 0.3297 である．

次の図は，例 8.1 で説明した到着間隔のシミュレーション例を示している．平均到着率 $\lambda = 12$ 人/時間であるとき，15 人の到着時間をシミュレーションした．横軸は時間を表し，青い線は各客の到着時間を示している．

```
from scipy.stats import expon
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)
lambda_param = 12  # 平均到着率(人/時間)
num_arrivals = 15  # シミュレーションする到着数
inter_arrival_times = expon.rvs(scale=1 / lambda_param, size=num_arrivals)
arrival_times = np.cumsum(inter_arrival_times)
plt.figure(figsize=(10, 2))
plt.eventplot(arrival_times, orientation="horizontal", colors="blue")
plt.title("Simulated Arrival Times")
plt.xlabel("Time (hours)")
plt.yticks([])
plt.xlim(0, max(arrival_times) + 0.15)
plt.show()
```



例 8.2. あるコールセンターでは，1 時間あたり平均 6 人の客を処理できる．サービス時間 T が指数分布に従うと仮定する．次の問いに答えよ．

1. 平均サービス時間 $E[T]$ を求めよ．
2. 3 分以内にサービスが完了する確率を求めよ．

平均サービス率は $\mu = 6$ である．平均サービス時間は次のように計算される．

$$E[T] = \frac{1}{\mu} = \frac{1}{6} \text{ 時間} = 10 \text{ 分}$$

$P(T \leq 0.05)$ を計算する．

$$P(T \leq 0.05) = F_T(0.05) = 1 - e^{-6\times 0.05} \approx 0.2592$$

よって，3 分以内にサービスが完了する確率は約 0.2592 である．

8.1.2　無記憶性

定義 8.1. 確率変数 T が**無記憶性**（memoryless）を持つとは，任意の非負実数 s, t に対して

$$P(T > s + t \mid T > s) = P(T > t), \quad (s, t \geq 0)$$

が成り立つことをいう．

確率変数 T　を到着間隔とする．無記憶性とは，すでに時間 s が経過している場合（$T > s$），さらに時間 t 経過した後に到着が発生していない確率（$T > s + t$）が，最初から時間 t 経過した後に到着が発生していない確率と等しいことを意味する．

例 8.3. 仮に，バスの到着間隔が指数分布に従うのであれば，「すでに 10 分待っていたので，そろそろバスが来るだろう」という考え方は成り立たない．

定理 8.1. 指数分布に従う確率変数は無記憶性を持つ．

証明.

$$\begin{aligned} P(T > s+t \mid T > s) &= \frac{P(T > s+t, T > s)}{P(T > s)} \\ &= \frac{P(T > s+t)}{P(T > s)} \\ &= \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} \\ &= e^{-\lambda t} \end{aligned}$$

□

指数分布が無記憶性を持つ唯一の連続分布である.

8.2 ポアソン過程

確率過程 $\{N(t), t \geq 0\}$ とは, 時間 t とともに変化する確率変数 $N(t)$ の集まりである.

計数過程（counting process）とは, 次の条件を満たす確率過程 $\{N(t), t \geq 0\}$ である.

1. $N(t) \geq 0$
2. $N(t) \in \mathbb{Z}$
3. $N(t)$ は単調増加である

$N(t)$ は, 時刻 t までに発生した事象の回数を表す. 例として, 客の到着数, 機械の故障回数などが挙げられる.

定義 8.2. 確率過程 $\{N(t), t \geq 0\}$ が**独立増分**（independent increments）を持つとは, 任意の $0 \leq t_1 < t_2 < \cdots < t_n$ に対して,

$$N(t_2) - N(t_1), N(t_3) - N(t_2), \ldots, N(t_n) - N(t_{n-1})$$

が互いに独立であることをいう.

計数過程では, $N(t_i) - N(t_{i-1})$ は時間区間 $(t_{i-1}, t_i]$ に発生した事象の回数を表す. 計数過程が独立増分を持つとは, 重なりのない時間区間における事象の発生回数が互いに独立であることを意味する.

例 8.4. ある店舗での客の到着数を表す計数過程 $\{N(t), t \geq 0\}$ を考える. この計数過程が独立増分を持つとは, 例えば, 時間区間 $(0, 1]$ に到着した客数と時間区間 $(3, 6]$ に到着した客数が独立であることを意味する.

定義 8.3. 確率過程 $\{N(t), t \geq 0\}$ が**定常増分** (stationary increments)を持つとは, 任意の $t > s \geq 0$ と $h \geq 0$ に対して, $N(t)-N(s)$ と $N(t+h)-N(s+h)$ が同一の分布に従うことをいう.

計数過程が定常増分を持つとは, 事象の発生回数は時間区間の長さのみに依存し, 時間の位置には依存しないことを意味する.

定義 8.4. 以下の性質を満たす計数過程 $\{N(t), t \geq 0\}$ を パラメータ $\lambda > 0$ の**ポアソン過程** (Poisson process)という.

1. $N(0) = 0$
2. $P(N(t + \Delta t) - N(t) = 1) = \lambda \Delta t + o(\Delta t)$
3. $P(N(t + \Delta t) - N(t) \geq 2) = o(\Delta t)$
4. 独立増分を持つ

ここで, $o(\Delta t)$ は

$$\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$$

を満たす関数である.

$N(0) = 0$ は, 時刻 0 では事象が発生していないことを表す.

$P(N(t + \Delta t) - N(t) = 1) = \lambda \Delta t + o(\Delta t)$ は, 時間区間 $[t, t + \Delta t]$ に事象が 1 回発生する確率を表す. Δt が十分小さいとき, 事象が 1 回発生する確率は $\lambda \Delta t$ に近似できる.

$P(N(t + \Delta t) - N(t) \geq 2) = o(\Delta t)$ は, 時間区間 $[t, t + \Delta t]$ に事象が 2 回以上発生する確率を表す. Δt が十分小さいとき, 事象が 2 回以上発生する確率は無視できるほど小さい.

定義 8.5. 離散型確率変数 X が**ポアソン分布** (Poisson distribution)に従うとは, 次の確率質量関数を持つことをいう.

$$p_X(k) = P(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k = 0, 1, 2, ...$$

ここで, $\lambda > 0$ は分布のパラメータである.

ポアソン分布の平均と分散は

$$E[X] = \lambda, \quad \mathrm{Var}(X) = \lambda$$

である.

次の定理は, ポアソン過程 $\{N(t), t \geq 0\}$ における $N(t)$ はポアソン分布に従うことを示している.

定理 8.2. パラメータ $\lambda > 0$ のポアソン過程 $\{N(t), t \geq 0\}$ に対して，$N(t)$ はパラメータ λt のポアソン分布に従う. すなわち,

$$P(N(t) = k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}, \quad k = 0, 1, 2, \ldots .$$

証明. 省略する. □

例 8.5. ある店舗では，1 分間あたり平均 5 人の客が来店する．$\{N(t), t \geq 0\}$ を時刻 t までに到着した客数を表すポアソン過程とする. 次の問いに答えよ.

1. 時刻 $t = 3$ 分までに 2 人の客が到着する確率を求めよ.
2. 時間区間 $[2, 5]$ 分に 2 人の客が到着する確率を求めよ.

平均到着率は $\lambda = 5$ 人/分である. 時刻 $t = 3$ 分までに到着した客数 $N(3)$ は, パラメータ $\lambda t = 5 \times 3 = 15$ のポアソン分布に従う．$P(N(3) = 2)$ を計算する.

$$P(N(3) = 2) = e^{-15} \frac{15^2}{2!}$$

独立増分と定常増分の性質から, 時間区間 $[2, 5]$ 分に 2 人の客が到着する確率は, 時刻 $t = 3$ 分までに 2 人の客が到着する確率と等しい.

8.3 ポアソン過程の到着間隔

Y_k を k 回目の事象が発生するまでの時間とし，T_k は $(k-1)$ 回目から k 回目の事象が起こるまでの時間とする．$k = 1, 2, \ldots$ に対して,

$$T_1 = Y_1, \quad T_k = Y_k - Y_{k-1} \quad (k = 2, 3, \ldots)$$

と定義する. また,

$$Y_k = \sum_{i=1}^{k} T_i = T_1 + T_2 + \cdots + T_k$$

である.

定理 8.3. $\{N(t), t \geq 0\}$ をパラメータ $\lambda > 0$ のポアソン過程とする. このとき，$k = 1, 2, \ldots$ に対して, 時間間隔 $T_k \sim Exp(\lambda)$ に従い，$T_1, T_2, \ldots$ は互いに独立である.

証明. 省略する. □

第III部

OR キャリア

第9章　キャリア

9.1　リンク

- Informs
  - Consider a Career in Operations Research and Analytics!
- 日本オペレーションズ・リサーチ学会
  - OR キャリアセッション
  - 企業事例交流会
  - OR 実施事例
- 日本経営工学会
  - 産学連携研究交流会
- キヤノン
  - 日本オペレーションズ・リサーチ(OR)学会レポート活動の舞台裏を大公開！

9.2　関連学会

- 日本オペレーションズ・リサーチ学会
- 日本経営工学会

参考文献

Arrow, Kenneth J, Theodore Harris, と Jacob Marschak. 1951.「Optimal Inventory Policy」. *Econometrica* 19 (3): 250.

Camm, Jeffrey, James Cochran, Michael Fry, Jeffrey Ohlmann, David Anderson, Dennis Sweeney, と Thomas Williams. 2022. *An introduction to management science: Quantitative approaches to decision making*. 16th 版. Florence, AL: South-Western College Publishing.

Harris, Ford W. 1990.「How many parts to make at once」. *Oper. Res.* 38 (6): 947–50.

Hillier, Frederick, と Gerald Lieberman. 2025. *ISE introduction to operations research*. 11th 版. Columbus, OH: McGraw-Hill Education.

Huber, Jakob, Sebastian Müller, Moritz Fleischmann, と Heiner Stuckenschmidt. 2019.「A data-driven newsvendor problem: From data to decision」. *Eur. J. Oper. Res.* 278 (3): 904–15.

Qin, Yan, Ruoxuan Wang, Asoo J Vakharia, Yuwen Chen, と Michelle M H Seref. 2011.「The newsvendor problem: Review and directions for future research」. *Eur. J. Oper. Res.* 213 (2): 361–74.

Scarf, Herbert. 1959.「The optimality of (S, s) policies in the dynamic inventory problem」.

Snyder, Lawrence V, と Zuo-Jun Max Shen. 2019. *Fundamentals of supply chain theory*. 2nd 版. Nashville, TN: John Wiley & Sons.

付録 A　微分積分

A.1　極値（Extremum）

1 変数関数 $f(x)$ が点 $x = a$ で極値をとるとき、

$$f'(a) = 0$$

が成り立つ。$f''(a) > 0$ のとき、$f(a)$ は極小値をとる。$f''(a) < 0$ のとき、$f(a)$ は極大値をとる。

A.2　凸関数（Convex Function）

1 変数 2 階微分可能な関数 $f(x)$ が凸関数であることの必要十分条件は、すべての x について

$$f''(x) \geq 0$$

が成り立つことである。

凸関数 $f(x)$ の極小値は、最小値である。

付録B　確率

B.1　確率変数

離散型確率変数 X が特定の値 x をとる確率を

$$P(X = x) = p_X(x)$$

と表すとき、$p_X(x)$ を X の**確率質量関数** (PMF) という。

連続型確率変数 X がある区間 $[a, b]$ にある値をとる確率を

$$P(a \leq X \leq b) = \int_a^b f_X(x)dx$$

と表す。$f_X(x)$ を X の**確率密度関数** (PDF) という。

確率変数 X の**累積分布関数** (CDF) を

$$F_X(x) = P(X \leq x) = \begin{cases} \sum_{k \leq x} p_X(k) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{x} f_X(t)dt & \text{if } X \text{ is continuous} \end{cases}$$

と表す。確率密度関数 $f_X(x)$ は累積分布関数 $F_X(x)$ の微分である。

$$f_X(x) = \frac{d}{dx}F_X(x)$$

B.2　確率分布

B.2.1　正規分布

連続型確率変数 X は**正規分布**（normal distribution）に従うとき、$X \sim N(\mu, \sigma^2)$ と表す。ここで μ は平均、σ^2 は分散である。X の確率密度関数は

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

と表す。平均は $E[X] = \mu$、分散は $\mathrm{Var}(X) = \sigma^2$ である。

X が $N(\mu, \sigma^2)$ に従うとき、$Y = aX + b$ は、$N(a\mu + b, a^2\sigma^2)$ に従う。特に、$Z = \frac{X-\mu}{\sigma}$ は標準正規分布(standard normal distribution)に従う。すなわち、$Z \sim N(0, 1)$ である。

連続型確率変数 Y が標準正規分布に従うとき、Y の累積分布関数は

$$\Phi(y) = P(Y \leq y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{t^2}{2}} dt$$

と表す。**標準正規分布表**から、y の値に対する $\Phi(y)$ を調べることができる。

Python では、以下のように $\Phi(y)$ を計算できる。

```
from scipy.stats import norm
def phi(y):
    return norm.cdf(y)

phi(0)  # 0.5
```

また、$\Phi(y) = 0.95$ のときの y の値を求めるには、以下のようにする。

```
from scipy.stats import norm
def phi_inverse(p):
    return norm.ppf(p)

phi_inverse(0.95)  # 約 1.64485
```

正規分布は**再生性**（reproductive property）を持つ。すなわち、$X_1, X_2, \ldots, X_n$ が独立に $N(\mu_i, \sigma_i^2)$ に従うとき、$Y = \sum_{i=1}^{n} a_i X_i$ は $N\left(\sum_{i=1}^{n} a_i \mu_i, \sum_{i=1}^{n} a_i^2 \sigma_i^2\right)$ に従う。

付録C　標準正規分布表

下表は、標準正規分布 $N(0,1)$ の累積分布関数

$$\phi(z) = P(Z \leq z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{t^2}{2}} dt$$

の値を示すものである。

例 C.1. $z = 1.23$ のとき、$\phi(1.23)$ の値を求めよ。

表の行から 1.2、列から $+0.03$ を選ぶ。交差する値は 0.89065 である。したがって、$\phi(1.23) = 0.89065$ である。

例 C.2. $\phi(z) = 0.89065$ のとき、z の値を求めよ。

表の中から 0.89065 を探す。行から 1.2、列から $+0.03$ を選ぶ。したがって、$z = 1.23$ である。

例 C.3. $X \sim N(100, 15^2)$ のとき、$P(X \leq 120)$ の値を求めよ。

$$P(X \leq 120) = \phi\left(\frac{120 - 100}{15}\right) \tag{C.1}$$

$$= \phi\left(\frac{20}{15}\right) \tag{C.2}$$

$$\approx \phi(1.33) \tag{C.3}$$

$$\tag{C.4}$$

表から $\phi(1.33) \approx 0.90824$ である。

例 C.4. $X \sim N(100, 15^2)$ のとき、$P(X \leq x) = 0.9$ のとき、x の値を求めよ。

$$P(X \leq x) = 0.9 \tag{C.5}$$

$$\phi\left(\frac{x-100}{15}\right) = 0.9 \tag{C.6}$$

$$\frac{x-100}{15} \approx 1.28 \tag{C.7}$$

$$x - 100 \approx 19.2 \tag{C.8}$$

$$x \approx 119.2 \tag{C.9}$$

$$\tag{C.10}$$

したがって、$x \approx 119.2$ である。

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-3.9	0.00005	0.00005	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004	0.00003	0.00003
-3.8	0.00007	0.00007	0.00007	0.00006	0.00006	0.00006	0.00006	0.00005	0.00005	0.00005
-3.7	0.00011	0.00010	0.00010	0.00010	0.00009	0.00009	0.00008	0.00008	0.00008	0.00008
-3.6	0.00016	0.00015	0.00015	0.00014	0.00014	0.00013	0.00013	0.00012	0.00012	0.00011
-3.5	0.00023	0.00022	0.00022	0.00021	0.00020	0.00019	0.00019	0.00018	0.00017	0.00017
-3.4	0.00034	0.00032	0.00031	0.00030	0.00029	0.00028	0.00027	0.00026	0.00025	0.00024
-3.3	0.00048	0.00047	0.00045	0.00043	0.00042	0.00040	0.00039	0.00038	0.00036	0.00035
-3.2	0.00069	0.00066	0.00064	0.00062	0.00060	0.00058	0.00056	0.00054	0.00052	0.00050
-3.1	0.00097	0.00094	0.00090	0.00087	0.00084	0.00082	0.00079	0.00076	0.00074	0.00071
-3.0	0.00135	0.00131	0.00126	0.00122	0.00118	0.00114	0.00111	0.00107	0.00104	0.00100
-2.9	0.00187	0.00181	0.00175	0.00169	0.00164	0.00159	0.00154	0.00149	0.00144	0.00139
-2.8	0.00256	0.00248	0.00240	0.00233	0.00226	0.00219	0.00212	0.00205	0.00199	0.00193
-2.7	0.00347	0.00336	0.00326	0.00317	0.00307	0.00298	0.00289	0.00280	0.00272	0.00264
-2.6	0.00466	0.00453	0.00440	0.00427	0.00415	0.00402	0.00391	0.00379	0.00368	0.00357
-2.5	0.00621	0.00604	0.00587	0.00570	0.00554	0.00539	0.00523	0.00508	0.00494	0.00480
-2.4	0.00820	0.00798	0.00776	0.00755	0.00734	0.00714	0.00695	0.00676	0.00657	0.00639
-2.3	0.01072	0.01044	0.01017	0.00990	0.00964	0.00939	0.00914	0.00889	0.00866	0.00842
-2.2	0.01390	0.01355	0.01321	0.01287	0.01255	0.01222	0.01191	0.01160	0.01130	0.01101
-2.1	0.01786	0.01743	0.01700	0.01659	0.01618	0.01578	0.01539	0.01500	0.01463	0.01426
-2.0	0.02275	0.02222	0.02169	0.02118	0.02068	0.02018	0.01970	0.01923	0.01876	0.01831
-1.9	0.02872	0.02807	0.02743	0.02680	0.02619	0.02559	0.02500	0.02442	0.02385	0.02330
-1.8	0.03593	0.03515	0.03438	0.03362	0.03288	0.03216	0.03144	0.03074	0.03005	0.02938
-1.7	0.04457	0.04363	0.04272	0.04182	0.04093	0.04006	0.03920	0.03836	0.03754	0.03673
-1.6	0.05480	0.05370	0.05262	0.05155	0.05050	0.04947	0.04846	0.04746	0.04648	0.04551
-1.5	0.06681	0.06552	0.06426	0.06301	0.06178	0.06057	0.05938	0.05821	0.05705	0.05592

z	−0.00	−0.01	−0.02	−0.03	−0.04	−0.05	−0.06	−0.07	−0.08	−0.09
-1.4	0.08076	0.07927	0.07780	0.07636	0.07493	0.07353	0.07215	0.07078	0.06944	0.06811
-1.3	0.09680	0.09510	0.09342	0.09176	0.09012	0.08851	0.08692	0.08534	0.08379	0.08226
-1.2	0.11507	0.11314	0.11123	0.10935	0.10749	0.10565	0.10383	0.10204	0.10027	0.09853
-1.1	0.13567	0.13350	0.13136	0.12924	0.12714	0.12507	0.12302	0.12100	0.11900	0.11702
-1.0	0.15866	0.15625	0.15386	0.15151	0.14917	0.14686	0.14457	0.14231	0.14007	0.13786
-0.9	0.18406	0.18141	0.17879	0.17619	0.17361	0.17106	0.16853	0.16602	0.16354	0.16109
-0.8	0.21186	0.20897	0.20611	0.20327	0.20045	0.19766	0.19489	0.19215	0.18943	0.18673
-0.7	0.24196	0.23885	0.23576	0.23270	0.22965	0.22663	0.22363	0.22065	0.21770	0.21476
-0.6	0.27425	0.27093	0.26763	0.26435	0.26109	0.25785	0.25463	0.25143	0.24825	0.24510
-0.5	0.30854	0.30503	0.30153	0.29806	0.29460	0.29116	0.28774	0.28434	0.28096	0.27760
-0.4	0.34458	0.34090	0.33724	0.33360	0.32997	0.32636	0.32276	0.31918	0.31561	0.31207
-0.3	0.38209	0.37828	0.37448	0.37070	0.36693	0.36317	0.35942	0.35569	0.35197	0.34827
-0.2	0.42074	0.41683	0.41294	0.40905	0.40517	0.40129	0.39743	0.39358	0.38974	0.38591
-0.1	0.46017	0.45620	0.45224	0.44828	0.44433	0.44038	0.43644	0.43251	0.42858	0.42465
-0.0	0.50000	0.49601	0.49202	0.48803	0.48405	0.48006	0.47608	0.47210	0.46812	0.46414

z	+0.00	+0.01	+0.02	+0.03	+0.04	+0.05	+0.06	+0.07	+0.08	+0.09
0.0	0.50000	0.50399	0.50798	0.51197	0.51595	0.51994	0.52392	0.52790	0.53188	0.53586
0.1	0.53983	0.54380	0.54776	0.55172	0.55567	0.55962	0.56360	0.56749	0.57142	0.57535
0.2	0.57926	0.58317	0.58706	0.59095	0.59483	0.59871	0.60257	0.60642	0.61026	0.61409
0.3	0.61791	0.62172	0.62552	0.62930	0.63307	0.63683	0.64058	0.64431	0.64803	0.65173
0.4	0.65542	0.65910	0.66276	0.66640	0.67003	0.67364	0.67724	0.68082	0.68439	0.68793
0.5	0.69146	0.69497	0.69847	0.70194	0.70540	0.70884	0.71226	0.71566	0.71904	0.72240
0.6	0.72575	0.72907	0.73237	0.73565	0.73891	0.74215	0.74537	0.74857	0.75175	0.75490
0.7	0.75804	0.76115	0.76424	0.76730	0.77035	0.77337	0.77637	0.77935	0.78230	0.78524
0.8	0.78814	0.79103	0.79389	0.79673	0.79955	0.80234	0.80511	0.80785	0.81057	0.81327
0.9	0.81594	0.81859	0.82121	0.82381	0.82639	0.82894	0.83147	0.83398	0.83646	0.83891
1.0	0.84134	0.84375	0.84614	0.84849	0.85083	0.85314	0.85543	0.85769	0.85993	0.86214
1.1	0.86433	0.86650	0.86864	0.87076	0.87286	0.87493	0.87698	0.87900	0.88100	0.88298
1.2	0.88493	0.88686	0.88877	0.89065	0.89251	0.89435	0.89617	0.89796	0.89973	0.90147
1.3	0.90320	0.90490	0.90658	0.90824	0.90988	0.91149	0.91308	0.91466	0.91621	0.91774
1.4	0.91924	0.92073	0.92220	0.92364	0.92507	0.92647	0.92785	0.92922	0.93056	0.93189
1.5	0.93319	0.93448	0.93574	0.93699	0.93822	0.93943	0.94062	0.94179	0.94295	0.94408
1.6	0.94520	0.94630	0.94738	0.94845	0.94950	0.95053	0.95154	0.95254	0.95352	0.95449
1.7	0.95543	0.95637	0.95728	0.95818	0.95907	0.95994	0.96080	0.96164	0.96246	0.96327
1.8	0.96407	0.96485	0.96562	0.96638	0.96712	0.96784	0.96856	0.96926	0.96995	0.97062

z	+0.00	+0.01	+0.02	+0.03	+0.04	+0.05	+0.06	+0.07	+0.08	+0.09
1.9	0.97128	0.97193	0.97257	0.97320	0.97381	0.97441	0.97500	0.97558	0.97615	0.97670
2.0	0.97725	0.97778	0.97831	0.97882	0.97932	0.97982	0.98030	0.98077	0.98124	0.98169
2.1	0.98214	0.98257	0.98300	0.98341	0.98382	0.98422	0.98461	0.98500	0.98537	0.98574
2.2	0.98610	0.98645	0.98679	0.98713	0.98745	0.98778	0.98809	0.98840	0.98870	0.98899
2.3	0.98928	0.98956	0.98983	0.99010	0.99036	0.99061	0.99086	0.99111	0.99134	0.99158
2.4	0.99180	0.99202	0.99224	0.99245	0.99266	0.99286	0.99305	0.99324	0.99343	0.99361
2.5	0.99379	0.99396	0.99413	0.99430	0.99446	0.99461	0.99477	0.99492	0.99506	0.99520
2.6	0.99534	0.99547	0.99560	0.99573	0.99585	0.99598	0.99609	0.99621	0.99632	0.99643
2.7	0.99653	0.99664	0.99674	0.99683	0.99693	0.99702	0.99711	0.99720	0.99728	0.99736
2.8	0.99744	0.99752	0.99760	0.99767	0.99774	0.99781	0.99788	0.99795	0.99801	0.99807
2.9	0.99813	0.99819	0.99825	0.99831	0.99836	0.99841	0.99846	0.99851	0.99856	0.99861
3.0	0.99865	0.99869	0.99874	0.99878	0.99882	0.99886	0.99889	0.99893	0.99896	0.99900
3.1	0.99903	0.99906	0.99910	0.99913	0.99916	0.99918	0.99921	0.99924	0.99926	0.99929
3.2	0.99931	0.99934	0.99936	0.99938	0.99940	0.99942	0.99944	0.99946	0.99948	0.99950
3.3	0.99952	0.99953	0.99955	0.99957	0.99958	0.99960	0.99961	0.99962	0.99964	0.99965
3.4	0.99966	0.99968	0.99969	0.99970	0.99971	0.99972	0.99973	0.99974	0.99975	0.99976
3.5	0.99977	0.99978	0.99978	0.99979	0.99980	0.99981	0.99981	0.99982	0.99983	0.99983
3.6	0.99984	0.99985	0.99985	0.99986	0.99986	0.99987	0.99987	0.99988	0.99988	0.99989
3.7	0.99989	0.99990	0.99990	0.99990	0.99991	0.99991	0.99992	0.99992	0.99992	0.99992
3.8	0.99993	0.99993	0.99993	0.99994	0.99994	0.99994	0.99994	0.99995	0.99995	0.99995
3.9	0.99995	0.99995	0.99996	0.99996	0.99996	0.99996	0.99996	0.99996	0.99997	0.99997

付録D　線形代数

D.1　固有値と固有ベクトル

$n \times n$ 行列 $\mathbf{A}$ に対し, ベクトル $\mathbf{x} \neq \mathbf{0}$ とスカラー λ が存在して,

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

が成り立つとき，$\mathbf{x}$ を $\mathbf{A}$ の**固有ベクトル** (eigenvector)，λ を $\mathbf{A}$ の**固有値** (eigenvalue) と呼ぶ.

例 D.1. 次のような $\mathbf{A}, \mathbf{x}, \lambda$ を考える.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1/2 & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \lambda = 2$$

このとき,

$$\begin{bmatrix} 1 & 2 \\ 1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

が成り立つため，$\mathbf{x}$ は $\mathbf{A}$ の固有ベクトル，λ は $\mathbf{A}$ の固有値である.

$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ は単位行列 $\mathbf{I}$ を用いて $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$ と書ける. このとき，$\mathbf{x} \neq \mathbf{0}$ であるため，$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ が成り立つ.

例 D.2. 次のような行列 $\mathbf{A}$ を考える.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1/2 & 1 \end{bmatrix}$$

このとき，$\det(\mathbf{A} - \lambda\mathbf{I})$ は以下のようになる.

$$\begin{aligned}\det(\mathbf{A}-\lambda\mathbf{I}) &= \det\begin{bmatrix}1-\lambda & 2\\ 1/2 & 1-\lambda\end{bmatrix}\\ &= (1-\lambda)^2-1\\ &= \lambda^2-2\lambda\\ &= \lambda(\lambda-2)\end{aligned}$$

$\lambda(\lambda-2)=0$ より，$\lambda=0,2$ が得られる．

付録E　最適化問題