The Template Textbook

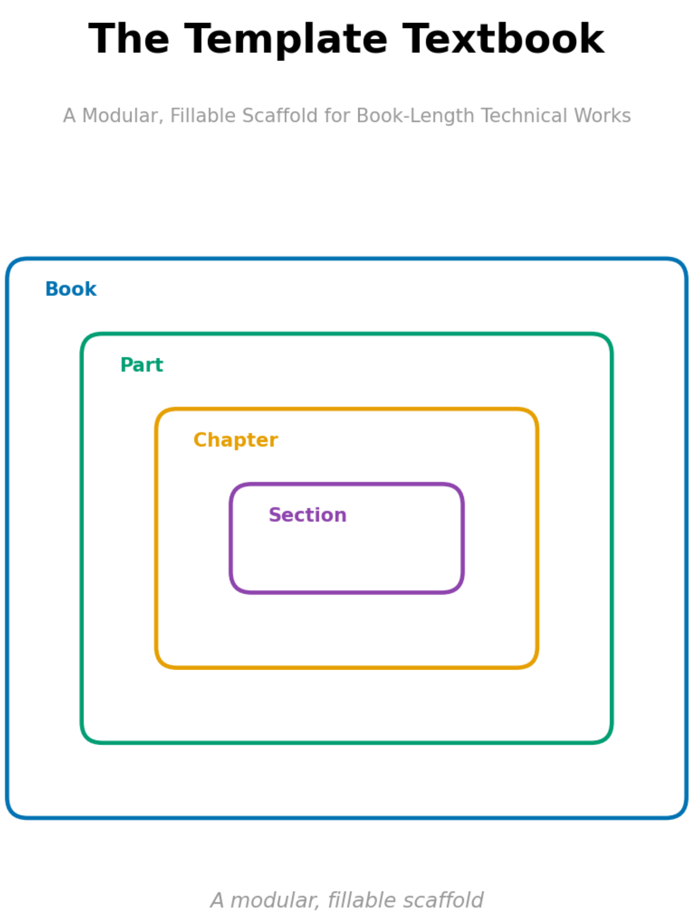
A Modular, Fillable Scaffold for Book-Length Technical Works

Daniel Ari Friedman

Active Inference Institute

daniel@activeinference.institute

ORCID: 0000-0001-6232-9096



Edition 0.1 – 2026 DOI: 10.5281/zenodo.20533125

Publishing Information

The Template Textbook

A Modular, Fillable Scaffold for Book-Length Technical Works

Daniel Ari Friedman

Active Inference Institute

daniel@activeinference.institute

ORCID: 0000-0001-6232-9096

Edition 0.1 – 2026

DOI: 10.5281/zenodo.20533125
Source repository: https://github.com/docxology/template_textbook

> *"Form follows function—that has been misunderstood. Form and function should be one, joined in a spiritual union."*
>
> — Frank Lloyd Wright

Acknowledgements

This template textbook is a structural scaffold. Replace this acknowledgement with your own once the chapters are filled.

Suggested citation: Daniel Ari Friedman (2026). *The Template Textbook: A Modular, Fillable Scaffold for Book-Length Technical Works* (Edition 0.1). Active Inference Institute. https://github.com/docxology/template_textbook. https://doi.org/10.5281/zenodo.20533125.

This open textbook is generated from version-controlled Markdown, tested Python modules, programmatic figures, and rendered Mermaid diagrams. Corrections and improvements may be submitted via the source repository linked above.

Accessibility note: the compact PDF is optimized for dense print. Reader-profile builds, HTML output, and source Markdown can be generated from the same manuscript materials.

Contents

1 Front Matter 10
1.1 Dedication . . . 10
1.2 About This Template . . . 10
1.3 How to Read This Book . . . 10
1.4 Methodology: How This Book Is Generated . . . 11

2 Preface 12
2.1 Why This Book Exists . . . 12
2.2 Who This Book Is For . . . 12
2.3 How to Use the Labs and Question Banks . . . 12
2.4 A Note on Reproducibility . . . 12

3 Part 0: Orientation and Methods 13

4 Orientation to the Field 14
4.1 Learning Objectives . . . 14
4.1.1 Study Blueprint . . . 14
4.2 Orientation . . . 15
4.3 A Worked Formalism . . . 15
4.4 Going Deeper . . . 15
4.5 Summary . . . 15
4.6 Key Terms . . . 15
4.7 Further Reading . . . 15
4.8 Practice . . . 16

5 Core Methods and Tools 17
5.1 Learning Objectives . . . 17
5.1.1 Study Blueprint . . . 17
5.2 Orientation . . . 18
5.3 A Worked Formalism . . . 18
5.4 Going Deeper . . . 18
5.5 Summary . . . 18
5.6 Key Terms . . . 18
5.7 Further Reading . . . 18
5.8 Practice . . . 19

6 Quantitative Foundations 20
6.1 Learning Objectives . . . 20
6.1.1 Study Blueprint . . . 20
6.2 Orientation . . . 21
6.3 A Worked Formalism . . . 21
6.4 Going Deeper . . . 21
6.5 Summary . . . 21
6.6 Key Terms . . . 21
6.7 Further Reading . . . 21
6.8 Practice . . . 22

7 Part I: Fundamentals **23**

8 First Principles **24**
8.1 Learning Objectives . . . 24
8.1.1 Study Blueprint . . . 24
8.2 From a system to a model . . . 25
8.3 A first quantitative law . . . 25
8.4 Worked example . . . 26
8.5 How the pieces connect . . . 26
8.6 Summary . . . 26
8.7 Key Terms . . . 26
8.8 Further Reading . . . 27
8.9 Practice . . . 27

9 Building Blocks **28**
9.1 Learning Objectives . . . 28
9.1.1 Study Blueprint . . . 28
9.2 Orientation . . . 29
9.3 A Worked Formalism . . . 29
9.4 Going Deeper . . . 29
9.5 Summary . . . 29
9.6 Key Terms . . . 29
9.7 Further Reading . . . 29
9.8 Practice . . . 30

10 Structure and Form **31**
10.1 Learning Objectives . . . 31
10.1.1 Study Blueprint . . . 31
10.2 Orientation . . . 32
10.3 A Worked Formalism . . . 32
10.4 Going Deeper . . . 32
10.5 Summary . . . 32
10.6 Key Terms . . . 32
10.7 Further Reading . . . 32
10.8 Practice . . . 33

11 Part II: Core Systems **34**

12 Systems Overview **35**
12.1 Learning Objectives . . . 35
12.1.1 Study Blueprint . . . 35
12.2 Orientation . . . 36
12.3 A Worked Formalism . . . 36
12.4 Going Deeper . . . 36
12.5 Summary . . . 36
12.6 Key Terms . . . 36
12.7 Further Reading . . . 36
12.8 Practice . . . 37

13 Dynamics and Change 38
13.1 Learning Objectives . . . 38
13.1.1 Study Blueprint . . . 38
13.2 Orientation . . . 39
13.3 A Worked Formalism . . . 39
13.4 Going Deeper . . . 39
13.5 Summary . . . 39
13.6 Key Terms . . . 39
13.7 Further Reading . . . 39
13.8 Practice . . . 40

14 Regulation and Control 41
14.1 Learning Objectives . . . 41
14.1.1 Study Blueprint . . . 41
14.2 Orientation . . . 42
14.3 A Worked Formalism . . . 42
14.4 Going Deeper . . . 42
14.5 Summary . . . 42
14.6 Key Terms . . . 42
14.7 Further Reading . . . 42
14.8 Practice . . . 43

15 Part III: Applications and Synthesis 44

16 Applied Models 45
16.1 Learning Objectives . . . 45
16.1.1 Study Blueprint . . . 45
16.2 Orientation . . . 46
16.3 A Worked Formalism . . . 46
16.4 Going Deeper . . . 46
16.5 Summary . . . 46
16.6 Key Terms . . . 46
16.7 Further Reading . . . 46
16.8 Practice . . . 47

17 Case Studies 48
17.1 Learning Objectives . . . 48
17.1.1 Study Blueprint . . . 48
17.2 The data . . . 49
17.3 Fitting a trend . . . 49
17.4 Interpolation versus extrapolation . . . 50
17.5 Summary . . . 50
17.6 Key Terms . . . 50
17.7 Further Reading . . . 50
17.8 Practice . . . 50

18 Frontiers and Open Problems 51
18.1 Learning Objectives . . . 51

18.1.1 Study Blueprint . . . 51
18.2 Orientation . . . 52
18.3 A Worked Formalism . . . 52
18.4 Going Deeper . . . 52
18.5 Summary . . . 52
18.6 Key Terms . . . 52
18.7 Further Reading . . . 52
18.8 Practice . . . 53

19 Lab — Orientation to the Field **54**
19.1 Objectives . . . 54
19.2 Background . . . 54
19.3 Procedure . . . 54
19.4 Analysis . . . 54
19.5 Computational Workflow . . . 54
19.6 Reflection . . . 54

20 Lab — Core Methods and Tools **55**
20.1 Objectives . . . 55
20.2 Background . . . 55
20.3 Procedure . . . 55
20.4 Analysis . . . 55
20.5 Computational Workflow . . . 55
20.6 Reflection . . . 55

21 Lab — Quantitative Foundations **56**
21.1 Objectives . . . 56
21.2 Background . . . 56
21.3 Procedure . . . 56
21.4 Analysis . . . 56
21.5 Computational Workflow . . . 56
21.6 Reflection . . . 56

22 Lab — First Principles **57**
22.1 Objectives . . . 57
22.2 Background . . . 57
22.3 Procedure . . . 57
22.4 Analysis . . . 57
22.5 Computational Workflow . . . 57
22.6 Reflection . . . 57

23 Lab — Building Blocks **58**
23.1 Objectives . . . 58
23.2 Background . . . 58
23.3 Procedure . . . 58
23.4 Analysis . . . 58
23.5 Computational Workflow . . . 58
23.6 Reflection . . . 58

24 Lab — Structure and Form **59**
24.1 Objectives . . . 59
24.2 Background . . . 59
24.3 Procedure . . . 59
24.4 Analysis . . . 59
24.5 Computational Workflow . . . 59
24.6 Reflection . . . 59

25 Lab — Systems Overview **60**
25.1 Objectives . . . 60
25.2 Background . . . 60
25.3 Procedure . . . 60
25.4 Analysis . . . 60
25.5 Computational Workflow . . . 60
25.6 Reflection . . . 60

26 Lab — Dynamics and Change **61**
26.1 Objectives . . . 61
26.2 Background . . . 61
26.3 Procedure . . . 61
26.4 Analysis . . . 61
26.5 Computational Workflow . . . 61
26.6 Reflection . . . 61

27 Lab — Regulation and Control **62**
27.1 Objectives . . . 62
27.2 Background . . . 62
27.3 Procedure . . . 62
27.4 Analysis . . . 62
27.5 Computational Workflow . . . 62
27.6 Reflection . . . 62

28 Lab — Applied Models **63**
28.1 Objectives . . . 63
28.2 Background . . . 63
28.3 Procedure . . . 63
28.4 Analysis . . . 63
28.5 Computational Workflow . . . 63
28.6 Reflection . . . 63

29 Lab — Case Studies **64**
29.1 Objectives . . . 64
29.2 Background . . . 64
29.3 Procedure . . . 64
29.4 Analysis . . . 64
29.5 Computational Workflow . . . 64
29.6 Reflection . . . 64

30 Lab — Frontiers and Open Problems **66**

30.1 Objectives . . . 66
30.2 Background . . . 66
30.3 Procedure . . . 66
30.4 Analysis . . . 66
30.5 Computational Workflow . . . 66
30.6 Reflection . . . 66

31 Question Bank — Orientation to the Field 67
31.1 Recall . . . 67
31.2 Application . . . 67
31.3 Synthesis . . . 67

32 Question Bank — Core Methods and Tools 68
32.1 Recall . . . 68
32.2 Application . . . 68
32.3 Synthesis . . . 68

33 Question Bank — Quantitative Foundations 69
33.1 Recall . . . 69
33.2 Application . . . 69
33.3 Synthesis . . . 69

34 Question Bank — First Principles 70
34.1 Recall . . . 70
34.2 Application . . . 70
34.3 Synthesis . . . 70

35 Question Bank — Building Blocks 71
35.1 Recall . . . 71
35.2 Application . . . 71
35.3 Synthesis . . . 71

36 Question Bank — Structure and Form 72
36.1 Recall . . . 72
36.2 Application . . . 72
36.3 Synthesis . . . 72

37 Question Bank — Systems Overview 73
37.1 Recall . . . 73
37.2 Application . . . 73
37.3 Synthesis . . . 73

38 Question Bank — Dynamics and Change 74
38.1 Recall . . . 74
38.2 Application . . . 74
38.3 Synthesis . . . 74

39 Question Bank — Regulation and Control 75
39.1 Recall . . . 75

39.2 Application . . . 75
39.3 Synthesis . . . 75

40 Question Bank — Applied Models **76**
40.1 Recall . . . 76
40.2 Application . . . 76
40.3 Synthesis . . . 76

41 Question Bank — Case Studies **77**
41.1 Recall . . . 77
41.2 Application . . . 77
41.3 Synthesis . . . 77

42 Question Bank — Frontiers and Open Problems **78**
42.1 Recall . . . 78
42.2 Application . . . 78
42.3 Synthesis . . . 78

43 Appendix A — Authoring Guide: Filling the Stubs **79**
43.1 The Big Idea: One Source of Truth . . . 79
43.2 Step 1 — Decide the Structure in `config.yaml` . . . 79
43.3 Step 2 — Materialise the Stubs . . . 79
43.4 Step 3 — Fill the Content Contract . . . 79
43.5 Step 4 — Add a Figure . . . 80
43.6 Step 5 — Add a Glossary Term . . . 80
43.7 Step 6 — Add a Reference . . . 80
43.8 Step 7 — Audit and Test . . . 80
43.9 Growing the Book . . . 80

44 Appendix B — Notation and Symbols **82**
44.1 Conventions . . . 82
44.2 Symbol Table . . . 82

45 Appendix C — Mathematical Review **83**
45.1 Functions and Variables . . . 83
45.2 Growth and Decay . . . 83
45.3 Saturating Responses . . . 83
45.4 Linear Fits . . . 83
45.5 Basic Statistics . . . 83

46 Appendix — Formalisms **84**
46.1 1. Definitions . . . 84
46.2 2. A theorem with proof . . . 84
46.3 3. A lemma . . . 84
46.4 4. Algorithms (pseudocode) . . . 85
46.5 5. A step-by-step derivation . . . 85
46.6 6. A system of numbered equations . . . 85
46.7 7. Dimensioned quantities . . . 86
46.8 8. Notation summary . . . 86

47 Appendix — Format Gallery **87**
47.1 1. Text and inline formatting . . . 87
47.2 2. Lists . . . 87
47.3 3. Block quotes and callouts . . . 88
47.4 4. Tables . . . 88
47.5 5. Mathematics and units . . . 89
47.6 6. Figures . . . 89
47.7 7. Diagrams (Mermaid) . . . 89
47.8 8. Code . . . 92
47.9 9. Cross-references and citations . . . 92
47.1010. Media and data . . . 93
47.1111. Pedagogical blocks . . . 93
47.1212. Miscellany . . . 94

48 Master Glossary **95**
48.0.1 Boundary . . . 95
48.0.2 Dynamics . . . 95
48.0.3 Emergence . . . 95
48.0.4 Equilibrium . . . 95
48.0.5 Feedback . . . 95
48.0.6 Gradient . . . 95
48.0.7 Model . . . 95
48.0.8 Network . . . 95
48.0.9 Observable . . . 95
48.0.10 Parameter . . . 95
48.0.11 Regulation . . . 95
48.0.12 State . . . 95
48.0.13 System . . . 95
48.0.14 Threshold . . . 96
48.0.15 Variable . . . 96

49 Appendix E — Index of Key Terms **97**
49.1 Placeholder Entries . . . 97

1 Front Matter

1.1 Dedication

TKTK — dedicate this book to whom you choose. One or two lines is conventional. Delete this comment and the placeholder once written.

1.2 About This Template

This is **not a finished book**. It is a *scaffold* — a complete, internally consistent skeleton for a book-length technical work, with every structural element in place and every author-specific passage left as a marked **stub** for you to fill.

The whole book is **data-driven from a single source of truth**, `config.yaml`. That file declares the parts, the chapters inside each part, the front matter, the appendices, and the labs and question banks. The table of contents, chapter numbering, figure numbering, and the manuscript-integrity tests all read from it. You grow the book by editing `config.yaml` and then materialising any missing files — you never hand-number a chapter, figure, equation, or table.

What is already provided for you:

- **Twelve chapters** across four parts, each a valid chapter shell with a labelled heading, a figure, a metadata badge, a Study Blueprint, Learning Objectives, a worked formalism (equation + parameter table), an inline Mermaid diagram, and Summary / Key Terms / Further Reading / Practice sections.
- **A matching lab and question bank** for every chapter, under `labs/` and `questions/`.
- **A tested computational backbone** in `src/` — the worked equations are real, tested Python functions (`textbook.models`), and figures are generated deterministically. Chapter prose *calls* these functions rather than retyping the mathematics.
- **A test gate** (`tests/test_manuscript_integrity.py` plus `scripts/audit_textbook_quality.py`) that checks the structural contract holds as you write.

Everywhere author-specific content belongs, you will find a stub marker: `<!-- STUB -->`, `TODO:`, or `TKTK`. The quality audit counts these, so your progress toward a finished book is measurable. See Appendix A — Authoring Guide and `AGENTS.md` for the full filling workflow.

1.3 How to Read This Book

The book is organised into four parts that build on one another. A first-time reader should move through them in order; an instructor can assign parts independently.

- **Part 0 — Orientation and Methods.** Where the field sits, the core methods and tools, and the quantitative foundations the rest of the book assumes. Start here even if you are experienced; it fixes notation and conventions.
- **Part I — Fundamentals.** First principles, the building blocks, and how structure and form arise. The conceptual bedrock.
- **Part II — Core Systems.** A systems overview, dynamics and change, and regulation and control. The working theory.

- **Part III — Applications and Synthesis.** Applied models, case studies, and frontiers with open problems. Where the ideas meet practice and the edge of what is known.

Each chapter ends with a **Practice** section pointing to its **lab** (a guided, hands-on exercise) and its **question bank** (self-check questions). Work the lab after reading; use the question bank to confirm you can recall and apply the material. New terms are linked to the Master Glossary the first time they appear.

1.4 Methodology: How This Book Is Generated

This manuscript is rendered, not typeset by hand. The pipeline reads `config.yaml`, runs the analysis scripts that produce figures and diagrams, assembles the Markdown sections in declared order, and renders a PDF through Pandoc with `pandoc-crossref` resolving every cross-reference and citation.

To build the book from the repository root:

```
uv run python scripts/02_run_analysis.py --project templates/template_textbook
uv run python scripts/03_render_pdf.py    --project templates/template_textbook
```

or run the full pipeline with `./run.sh`. See `README.md` for the manuscript directory layout and `SYNTAX.md` for the exact authoring syntax.

2 Preface

2.1 Why This Book Exists

TKTK — state, in two or three paragraphs, the problem this book solves and why it is worth a reader's time. What gap in the existing literature does it fill? What will a reader be able to do after finishing it that they could not do before?

This template is domain-neutral by design. Wherever the sample chapters speak of "systems", "dynamics", or "regulation", substitute the concepts of your own field. The structure — orientation, fundamentals, core systems, applications — generalises across most technical subjects; the content is yours to supply.

2.2 Who This Book Is For

TKTK — describe the intended reader. Assumed background? A prerequisite course or a specific level of mathematical maturity? Whether the book suits self-study, a one-semester course, or a reference shelf.

The book assumes the quantitative foundations laid out in **Part 0** and nothing more. A reader comfortable with that material can follow every chapter.

2.3 How to Use the Labs and Question Banks

Each chapter is paired with two companion documents:

- **A lab** (under `labs/`) — a guided, hands-on exercise that puts the chapter's worked formalism to work. Labs are meant to be *done*, not just read: run the tested functions in `textbook.models`, vary the parameters, and observe how the predictions change. Work the lab immediately after reading the chapter, while the ideas are fresh.
- **A question bank** (under `questions/`) — self-check questions that confirm you can recall and apply the chapter's load-bearing claims. Use it as a diagnostic: a question you cannot answer points you back to a specific section.

Instructors can assign the lab as homework and draw exam or quiz items from the question bank. Because both are generated from the same `config.yaml` as the chapters, they stay in lockstep with the manuscript as it grows.

2.4 A Note on Reproducibility

Every equation in this book is implemented as a tested function and every figure is generated deterministically from code. Nothing in the prose is computed by hand. If you build the book yourself you will get byte-identical figures and numbers — see `README.md` for the build commands.

— The Author, TKTK (place), TKTK (date)

3 Part 0: Orientation and Methods

This part covers **orientation and methods**. It contains the following chapters:

- *Orientation to the Field* — sec. 4
- *Core Methods and Tools* — sec. 5
- *Quantitative Foundations* — sec. 6

How to use this part.

4 Orientation to the Field

Figure 1. Overview schematic for “Orientation to the Field”. Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

4.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

4.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **regulation**, **boundary**, **state**.
- **Quantitative lens:** the worked formalism in eq. 1.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 19.
- **Question bank:** sec. 31.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

4.2 Orientation

This section introduces the central ideas of *Orientation to the Field*. Foundational treatments include [Kim, 2020, Brown, 2017]. Key terms such as **regulation**, **boundary**, **state** are defined in the glossary.

4.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 1:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{1}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 1.

Table 1. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 2. Mermaid diagram

Note

4.4 Going Deeper

See the overview in fig. 1 and revisit the objectives above as you read. Further evidence: [Kim, 2020, Brown, 2017].

4.5 Summary

4.6 Key Terms

feedback, **gradient**, **threshold**, **network**.

4.7 Further Reading

- TODO: one-line annotation [Wilson, 2021].

4.8 Practice

- **Lab:** sec. 19
- **Question bank:** sec. 31

5 Core Methods and Tools

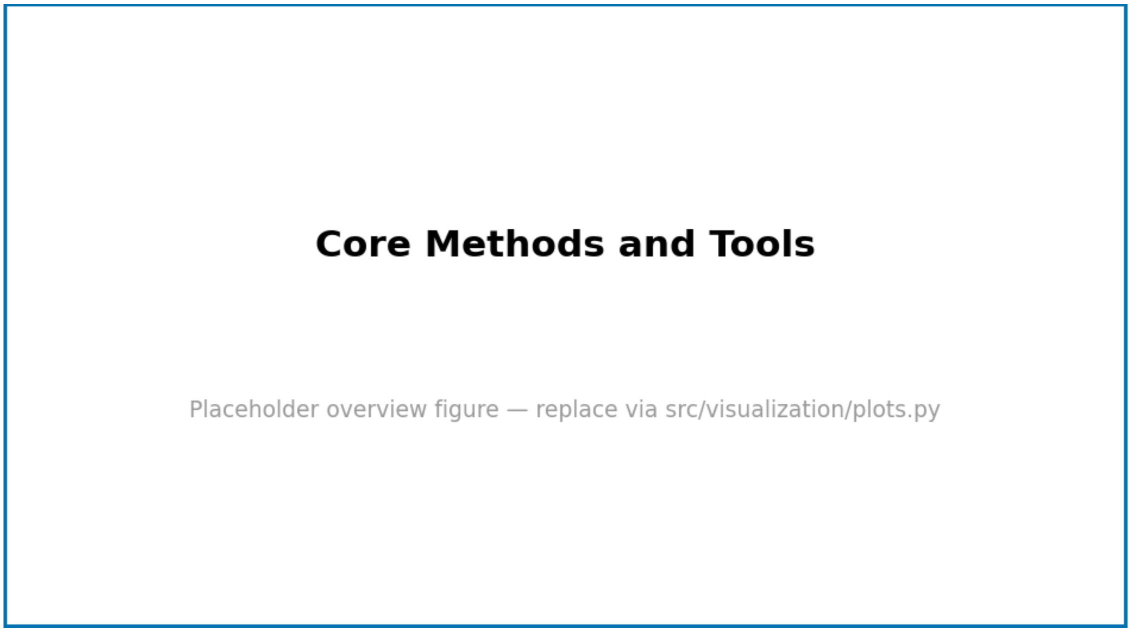


Figure 3. Overview schematic for "Core Methods and Tools". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

5.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

5.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **model**, **parameter**, **variable**.
- **Quantitative lens:** the worked formalism in eq. 2.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 20.
- **Question bank:** sec. 32.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

5.2 Orientation

This section introduces the central ideas of *Core Methods and Tools*. Foundational treatments include [Kim, 2020, Brown, 2017]. Key terms such as **model**, **parameter**, **variable** are defined in the glossary.

5.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 2:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{2}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 2.

Table 2. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 4. Mermaid diagram

Note

5.4 Going Deeper

See the overview in fig. 3 and revisit the objectives above as you read. Further evidence: [Kim, 2020, Brown, 2017].

5.5 Summary

5.6 Key Terms

emergence, **regulation**, **boundary**, **state**.

5.7 Further Reading

- TODO: one-line annotation [Wilson, 2021].

5.8 Practice

- **Lab:** sec. 20
- **Question bank:** sec. 32

6 Quantitative Foundations

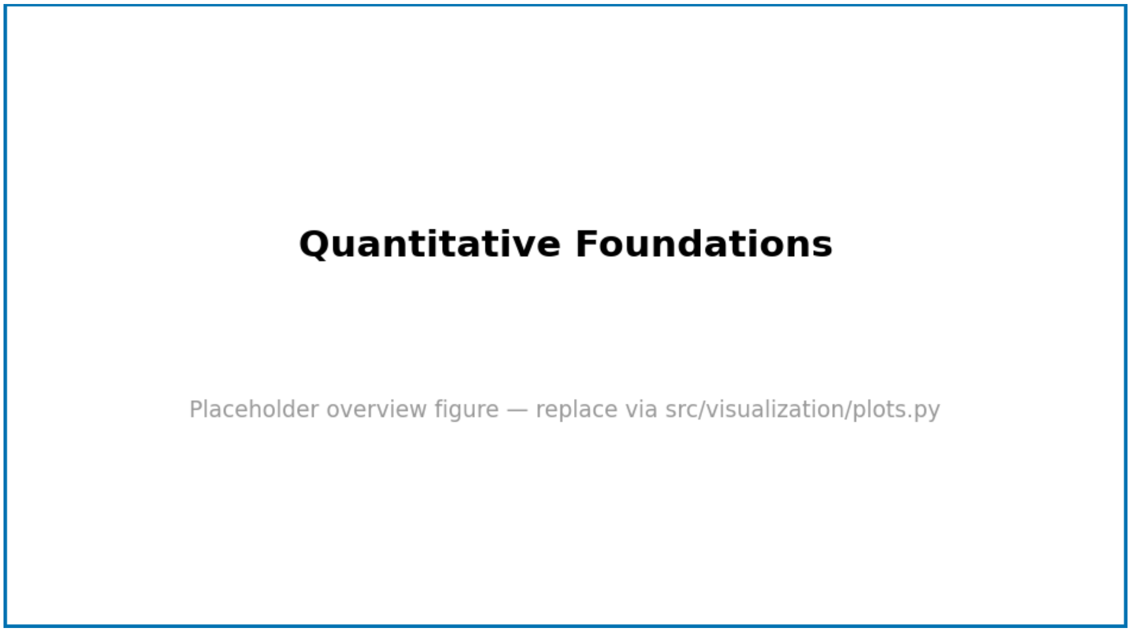


Figure 5. Overview schematic for "Quantitative Foundations". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

6.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

6.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **feedback**, **gradient**, **threshold**.
- **Quantitative lens:** the worked formalism in eq. 3.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 21.
- **Question bank:** sec. 33.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

6.2 Orientation

This section introduces the central ideas of *Quantitative Foundations*. Foundational treatments include [Smith, 2020, Doe, 2019]. Key terms such as **feedback**, **gradient**, **threshold** are defined in the glossary.

6.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 3:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{3}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 3.

Table 3. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 6. Mermaid diagram

Note

6.4 Going Deeper

See the overview in fig. 5 and revisit the objectives above as you read. Further evidence: [Smith, 2020, Doe, 2019].

6.5 Summary

6.6 Key Terms

observable, **system**, **model**, **parameter**.

6.7 Further Reading

- TODO: one-line annotation [Lee, 2021].

6.8 Practice

- **Lab:** sec. 21
- **Question bank:** sec. 33

7 Part I: Fundamentals

This part covers **fundamentals**. It contains the following chapters:

- *First Principles* — sec. 8
- *Building Blocks* — sec. 9
- *Structure and Form* — sec. 10

How to use this part.

8 First Principles

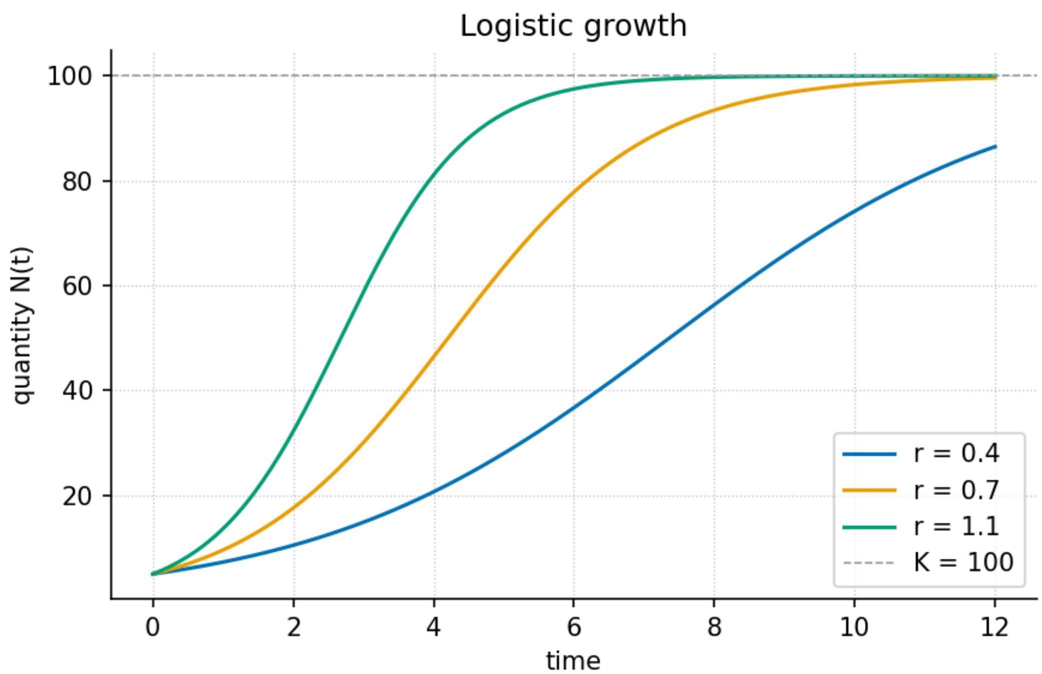


Figure 7. A logistic growth curve for three growth rates, approaching the carrying capacity $K = 100$. Produced deterministically by `visualization.plots.plot_logistic_growth`.

Level 1/3 · 25 min read · 40 min lecture · Prerequisites: none

This chapter is a worked reference. It is filled to completion to show what a finished chapter looks like. The other chapters in this template ship as structurally-complete stubs (marked with stub comments and TODO notes); fill them the same way.

8.1 Learning Objectives

By the end of this chapter you should be able to:

1. Explain what it means to model a **system** as a small set of **state** variables and a rule for how they change.
2. Distinguish a **parameter** from a **variable**, and identify each in a worked model.
3. Derive and interpret the logistic growth law and predict its long-run behaviour toward an **equilibrium**.
4. Compute model values with the tested function `textbook.models.logistic_growth` rather than re-deriving arithmetic by hand.

8.1.1 Study Blueprint

- **Big idea:** A great deal of change in the world is captured by a handful of variables and a single rule relating their rates — the *first principle* of quantitative modelling.
- **Core concepts:** **system**, **state**, **parameter**, **equilibrium**.
- **Quantitative lens:** the logistic law in eq. 5.
- **Data skill:** read an S-curve and estimate its carrying capacity by eye, then confirm numerically.

- **Common misconception to repair:** "exponential" and "logistic" are not the same — unbounded growth is an early-time approximation, not the rule.
- **Primary lab:** sec. 22.
- **Question bank:** sec. 34.
- **Bridge to computation:** `textbook.models.logistic_growth`.

Opening Vignette: Counting before you explain.

A population biologist watches a colony double, then double again, then — unexpectedly — slow. A start-up tracks users that grow the same way before the market saturates. A chemist measures a reaction that races, then crawls as substrate runs low. Three unrelated fields, one shape. The job of a first model is not to capture everything; it is to capture *that shape* with as few moving parts as possible.

8.2 From a system to a model

A **model** is a deliberate simplification. We choose a small number of **state** variables — here, a single quantity $N(t)$ — and write a rule for how the state changes. The art is leaving things out: a first model that fits on one line teaches more than a faithful one that fills a page.

The variables are what change; the **parameter**s are what we hold fixed while we reason. In the growth model below, N is the variable, while the rate r and the carrying capacity K are parameters. Confusing the two is the most common beginner error: if you find yourself "solving for K over time," you have mislabelled a parameter as a variable.

8.3 A first quantitative law

Unbounded (exponential) growth assumes nothing ever pushes back. Real systems saturate. The simplest law that grows fast when small and levels off when large is the **logistic** equation, written as a rate of change in eq. 4 and in closed form in eq. 5:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) \tag{4}$$

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{5}$$

Read eq. 4 aloud: the rate of change is proportional to how much there is (rN) *times* how much room remains ($1 - N/K$). When N is small the second factor is near 1 and growth is nearly exponential; as N approaches K the factor approaches 0 and growth stalls. The parameters are collected in tbl. 4.

Table 4. Parameters of the logistic model. Variables change over time; parameters are held fixed while reasoning.

Symbol	Name	Role	Example value
$N(t)$	quantity	variable	computed
r	intrinsic rate	parameter	$0.8\ \mathrm{s}^{-1}$

Symbol	Name	Role	Example value
K	carrying capacity	parameter	100 units
N_0	initial value	parameter	5 units

8.4 Worked example

Take $r = 0.8$, $K = 100$, and $N_0 = 5$. Rather than evaluate the exponential by hand, call the tested backbone:

```
import numpy as np
from textbook import models

t = np.array([0.0, 2.0, 5.0, 10.0])
N = models.logistic_growth(t, r=0.8, carrying_capacity=100.0, initial=5.0)
# N -> [ 5.00, 20.68, 74.18, 99.37 ]
```

The trajectory starts at $N(0) = 5.00$, reaches $N(2) = 20.68$, passes the steep middle near $t = 5$ with $N(5) = 74.18$, and by $t = 10$ has all but arrived at $N(10) = 99.37$ — within one part in a hundred of the carrying capacity. This is the S-curve plotted in fig. 7: an early near-exponential rise, an inflection, and a long approach to the asymptote.

The long-run behaviour is exact, not approximate. Because $r > 0$, the term $e^{-rt} \to 0$, so $N(t) \to K$. The proof is one line and is recorded as Theorem 1 in sec. 46; the same fact is asserted numerically by the test suite, so the prose and the code cannot silently disagree.

8.5 How the pieces connect

```
graph TD
  A[Choose state variables] --> B[Write a rate rule]
  B --> C[Solve or simulate]
  C --> D[Predict long-run behaviour]
  D -->|compare to data| A
```

Figure 8. Mermaid diagram

This loop — choose, write, solve, predict, compare — is the first principle the rest of the book elaborates. Foundational treatments of model-building include [Brown, 2017] and [Patel, 2018].

8.6 Summary

A model trades completeness for clarity: a few **state** variables and one rule. The logistic law adds a single idea — finite room — to exponential growth, and that idea changes the long-run behaviour from unbounded increase to a stable **equilibrium** at the carrying capacity K. Compute with the tested `logistic_growth` function so your worked numbers are reproducible and correct by construction.

8.7 Key Terms

system, **model**, **state**, **parameter**, **equilibrium**.

8.8 Further Reading

- Brown [2017] — a readable introduction to reasoning from first principles; start with its opening chapter on what a model is for.
- Patel [2018] — a broader survey of model families; use it to see where the logistic law sits among alternatives.

8.9 Practice

- **Lab:** sec. 22 — measure an S-curve and estimate its parameters.
- **Question bank:** sec. 34 — recall through synthesis.

9 Building Blocks

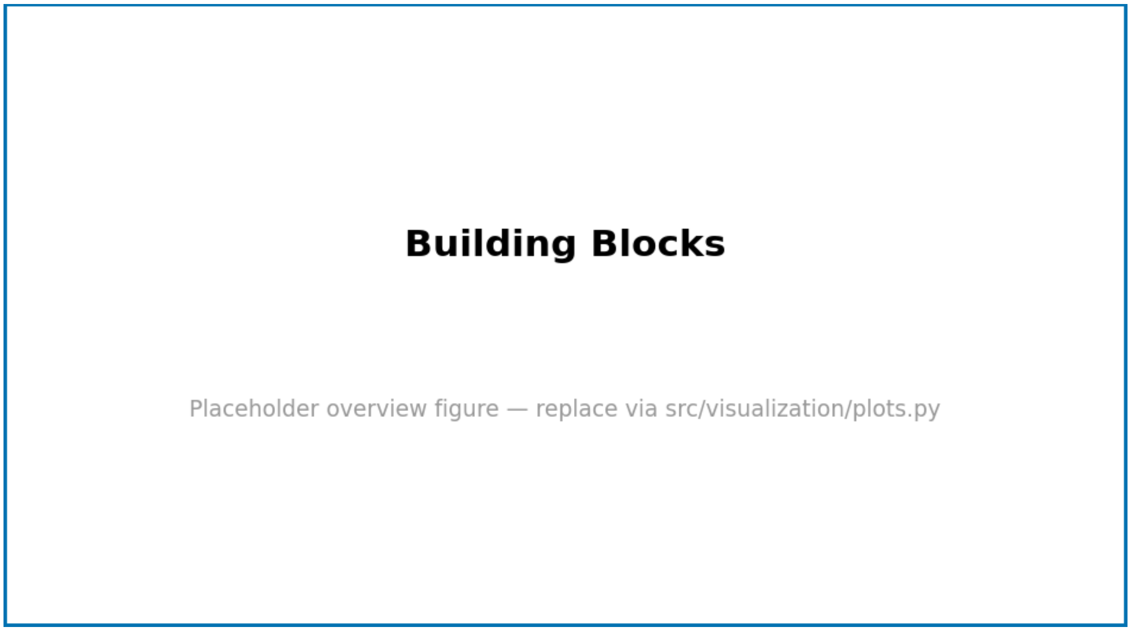


Figure 9. Overview schematic for "Building Blocks". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

9.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

9.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **equilibrium**, **feedback**, **gradient**.
- **Quantitative lens:** the worked formalism in eq. 6.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 23.
- **Question bank:** sec. 35.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

9.2 Orientation

This section introduces the central ideas of *Building Blocks*. Foundational treatments include [Taylor, 2019, Smith, 2020]. Key terms such as **equilibrium**, **feedback**, **gradient** are defined in the glossary.

9.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 6:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{6}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 5.

Table 5. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 10. Mermaid diagram

Note

9.4 Going Deeper

See the overview in fig. 9 and revisit the objectives above as you read. Further evidence: [Taylor, 2019, Smith, 2020].

9.5 Summary

9.6 Key Terms

state, **observable**, **system**, **model**.

9.7 Further Reading

- TODO: one-line annotation [Doe, 2019].

9.8 Practice

- **Lab:** sec. 23
- **Question bank:** sec. 35

10 Structure and Form

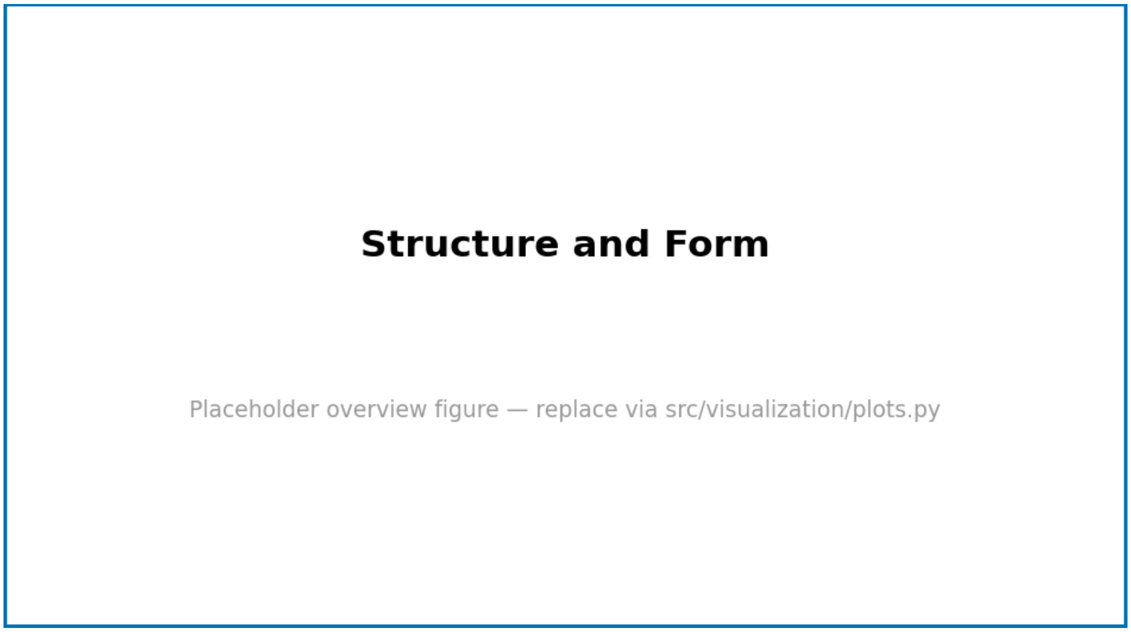


Figure 11. Overview schematic for "Structure and Form". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

10.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

10.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **threshold**, **network**, **dynamics**.
- **Quantitative lens:** the worked formalism in eq. 7.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 24.
- **Question bank:** sec. 36.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

10.2 Orientation

This section introduces the central ideas of *Structure and Form*. Foundational treatments include [Lee, 2021, Garcia, 2022]. Key terms such as **threshold**, **network**, **dynamics** are defined in the glossary.

10.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 7:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{7}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 6.

Table 6. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 12. Mermaid diagram

Note

10.4 Going Deeper

See the overview in fig. 11 and revisit the objectives above as you read. Further evidence: [Lee, 2021, Garcia, 2022].

10.5 Summary

10.6 Key Terms

model, **parameter**, **variable**, **equilibrium**.

10.7 Further Reading

- TODO: one-line annotation [Patel, 2018].

10.8 Practice

- **Lab:** sec. 24
- **Question bank:** sec. 36

11 Part II: Core Systems

This part covers **core systems**. It contains the following chapters:

- *Systems Overview* — sec. 12
- *Dynamics and Change* — sec. 13
- *Regulation and Control* — sec. 14

How to use this part.

12 Systems Overview

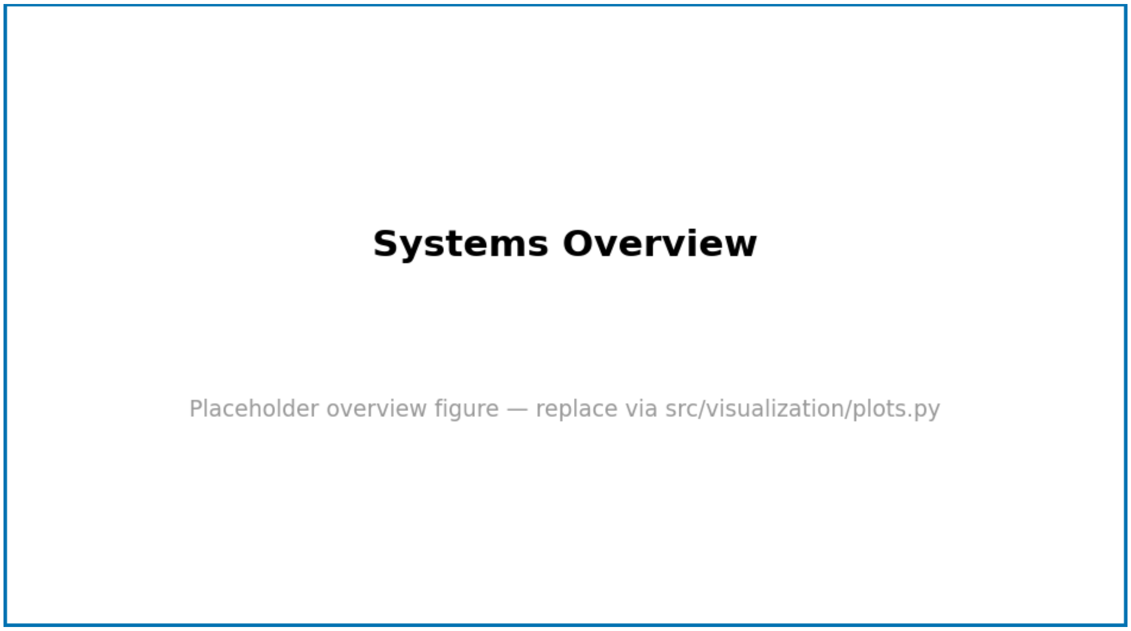


Figure 13. Overview schematic for "Systems Overview". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

12.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

12.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **equilibrium**, **feedback**, **gradient**.
- **Quantitative lens:** the worked formalism in eq. 8.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 25.
- **Question bank:** sec. 37.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

12.2 Orientation

This section introduces the central ideas of *Systems Overview*. Foundational treatments include [Patel, 2018, Nguyen, 2023]. Key terms such as **equilibrium**, **feedback**, **gradient** are defined in the glossary.

12.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 8:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{8}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 7.

Table 7. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 14. Mermaid diagram

Note

12.4 Going Deeper

See the overview in fig. 13 and revisit the objectives above as you read. Further evidence: [Patel, 2018, Nguyen, 2023].

12.5 Summary

12.6 Key Terms

state, **observable**, **system**, **model**.

12.7 Further Reading

- TODO: one-line annotation [Kim, 2020].

12.8 Practice

- **Lab:** sec. 25
- **Question bank:** sec. 37

13 Dynamics and Change

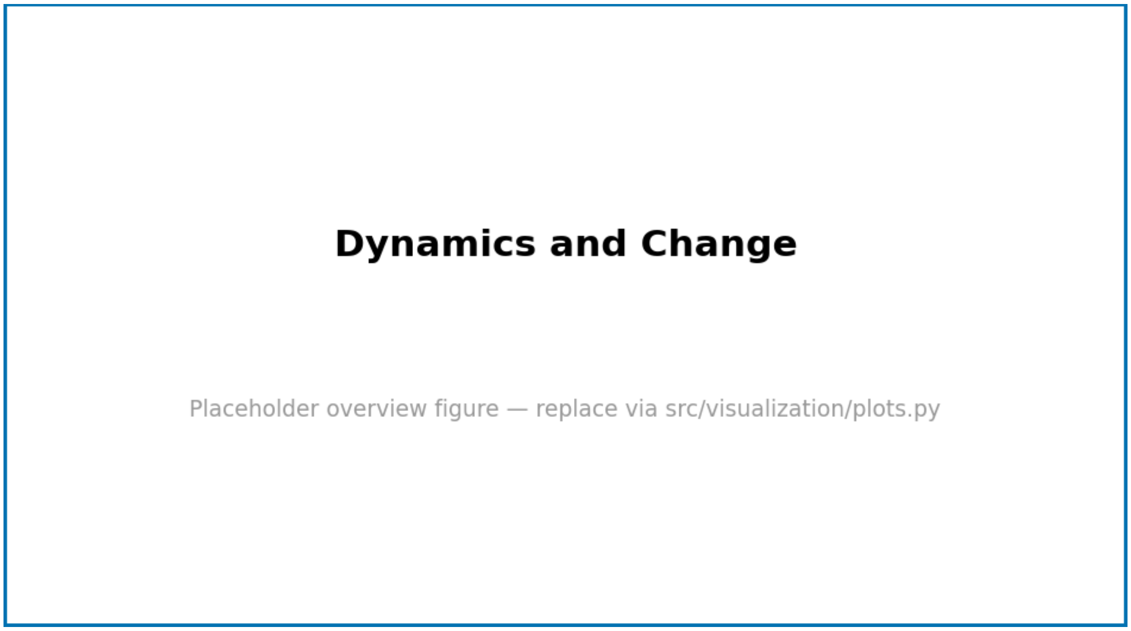


Figure 15. Overview schematic for "Dynamics and Change". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

13.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

13.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **parameter**, **variable**, **equilibrium**.
- **Quantitative lens:** the worked formalism in eq. 9.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 26.
- **Question bank:** sec. 38.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

13.2 Orientation

This section introduces the central ideas of *Dynamics and Change*. Foundational treatments include [Brown, 2017, Wilson, 2021]. Key terms such as **parameter**, **variable**, **equilibrium** are defined in the glossary.

13.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 9:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{9}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 8.

Table 8. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 16. Mermaid diagram

Note

13.4 Going Deeper

See the overview in fig. 15 and revisit the objectives above as you read. Further evidence: [Brown, 2017, Wilson, 2021].

13.5 Summary

13.6 Key Terms

regulation, **boundary**, **state**, **observable**.

13.7 Further Reading

- TODO: one-line annotation [Taylor, 2019].

13.8 Practice

- **Lab:** sec. 26
- **Question bank:** sec. 38

14 Regulation and Control

Figure 17. Overview schematic for "Regulation and Control". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

14.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

14.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **network**, **dynamics**, **emergence**.
- **Quantitative lens:** the worked formalism in eq. 10.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 27.
- **Question bank:** sec. 39.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

14.2 Orientation

This section introduces the central ideas of *Regulation and Control*. Foundational treatments include [Wilson, 2021, Taylor, 2019]. Key terms such as **network**, **dynamics**, **emergence** are defined in the glossary.

14.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 10:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{10}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 9.

Table 9. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 18. Mermaid diagram

Note

14.4 Going Deeper

See the overview in fig. 17 and revisit the objectives above as you read. Further evidence: [Wilson, 2021, Taylor, 2019].

14.5 Summary

14.6 Key Terms

parameter, **variable**, **equilibrium**, **feedback**.

14.7 Further Reading

- TODO: one-line annotation [Smith, 2020].

14.8 Practice

- **Lab:** sec. 27
- **Question bank:** sec. 39

15 Part III: Applications and Synthesis

This part covers **applications and synthesis**. It contains the following chapters:

- *Applied Models* — sec. 16
- *Case Studies* — sec. 17
- *Frontiers and Open Problems* — sec. 18

How to use this part.

16 Applied Models

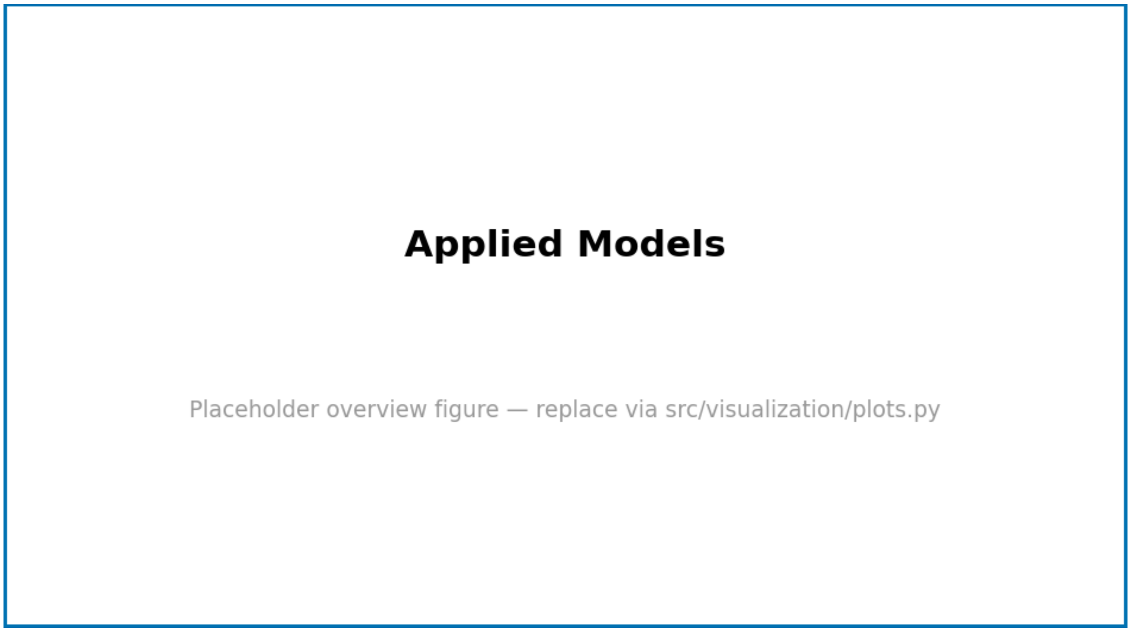


Figure 19. Overview schematic for "Applied Models". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

16.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

16.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **equilibrium**, **feedback**, **gradient**.
- **Quantitative lens:** the worked formalism in eq. 11.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 28.
- **Question bank:** sec. 40.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

16.2 Orientation

This section introduces the central ideas of *Applied Models*. Foundational treatments include [Patel, 2018, Nguyen, 2023]. Key terms such as **equilibrium**, **feedback**, **gradient** are defined in the glossary.

16.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 11:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{11}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 10.

Table 10. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 20. Mermaid diagram

Note

16.4 Going Deeper

See the overview in fig. 19 and revisit the objectives above as you read. Further evidence: [Patel, 2018, Nguyen, 2023].

16.5 Summary

16.6 Key Terms

state, **observable**, **system**, **model**.

16.7 Further Reading

- TODO: one-line annotation [Kim, 2020].

16.8 Practice

- **Lab:** sec. 28
- **Question bank:** sec. 40

17 Case Studies

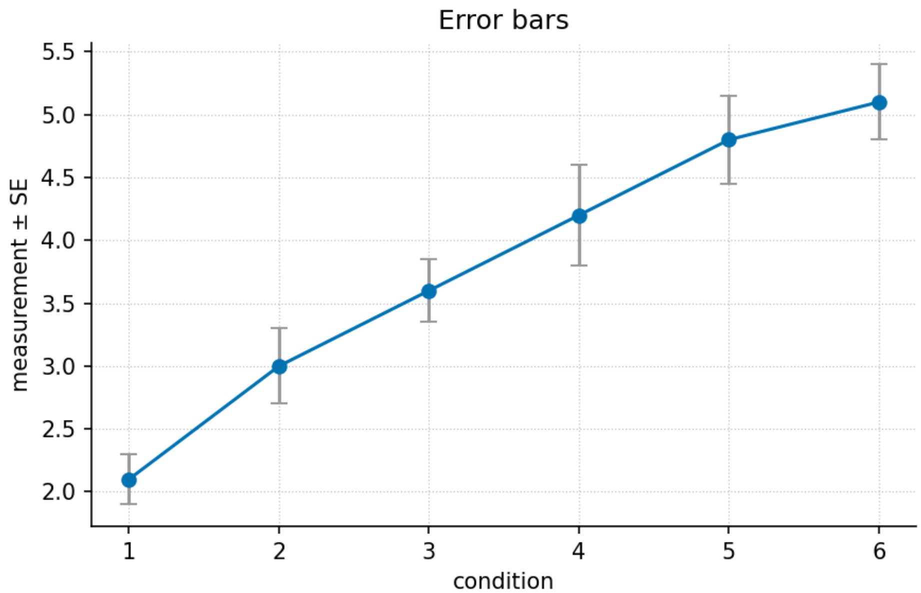


Figure 21. Group means with standard-error bars across six synthetic conditions, generated by `visualization.gallery.errorbar_plot`. In a real case study you would replace this gallery placeholder with your own measured group means.

Level 2/3 · 30 min read · 45 min lecture · Prerequisites: First Principles

This chapter is a worked reference in a different style from sec. 8: where that chapter *derives* a model, this one *applies* one to a small dataset. The remaining chapters ship as stubs.

17.1 Learning Objectives

By the end of this chapter you should be able to:

1. Frame a real question as a comparison between a **system** under control and treatment conditions.
2. Summarise grouped **observable** data with means and a measure of spread, and read an error-bar figure.
3. Fit a simple dose–response trend with `textbook.models.linear_fit` and state what its slope and R^2 mean.
4. Distinguish a defensible interpolation from an unjustified extrapolation.

17.1.1 Study Blueprint

- **Big idea:** A case study turns a general **model** into a decision about a specific dataset — and into an honest statement of its limits.
- **Core concepts:** **observable**, **gradient**, **threshold**.
- **Quantitative lens:** a linear dose–response fit, eq. 12.
- **Data skill:** group, average, and fit; then sanity-check the fit against the raw numbers in tbl. 11.
- **Common misconception to repair:** a high R^2 on three points is not strong evidence — it is an invitation to collect more.
- **Primary lab:** sec. 29.

- **Question bank:** sec. 41.
- **Bridge to computation:** `textbook.models.linear_fit`, `descriptive_statistics`.

Opening Vignette: From a spreadsheet to a decision.

A team runs a small pilot: a control condition and two treatment levels, two replicates each. The numbers land in a spreadsheet. The question is not "are they different?" — they obviously are — but "by how much, how confidently, and what should we predict next?" A case study is the discipline of answering those three questions without overreaching.

17.2 The data

The pilot produced six measurements across three conditions, recorded in `assets/data/sample_dataset.csv` and reproduced in tbl. 11.

Table 11. The pilot dataset: two replicates per condition.

Condition	Replicate	Measurement	Standard error
control	1	2.10	0.20
control	2	2.30	0.18
treatment (low)	1	3.60	0.25
treatment (low)	2	3.40	0.22
treatment (high)	1	4.80	0.35
treatment (high)	2	5.10	0.30

Grouping and averaging — the first move in almost every case study — gives means of 2.20 (control), 3.50 (low), and 4.95 (high). Across all six points the mean is 3.55 with a standard deviation of 1.13 (`textbook.models.descriptive_statistics`). The error-bar view is fig. 21.

17.3 Fitting a trend

Encode the dose as $0, 1, 2$ for control, low, and high, and fit a line with `textbook.models.linear_fit`:

```
import numpy as np
from textbook import models

dose = np.array([0.0, 1.0, 2.0])
response = np.array([2.20, 3.50, 4.95])   # condition means
fit = models.linear_fit(dose, response)
# fit.slope   1.375, fit.intercept   2.175, fit.r_squared   0.999
```

The fitted relationship in eq. 12 is

$$\hat{y}(d) = 1.375\, d + 2.175 \tag{12}$$

with $R^2 = 0.999$. The slope says each dose step adds about 1.4 units of response; the intercept, 2.175, is close to the measured control mean of 2.20, a reassuring internal check.

17.4 Interpolation versus extrapolation

Using eq. 12 to estimate the response at an *intermediate* dose is reasonable: the model interpolates within the range it was fit on. Using it to predict a *much higher* dose — say $d = 3$, giving $\hat{y} = 6.3$ — is an **extrapolation** the data cannot support. Real dose–response curves usually saturate (recall the **threshold** behaviour of the saturating response in sec. 8); a straight line will overpredict once the system approaches its ceiling.

```
graph LR
  A[Raw measurements] --> B[Group + average]
  B --> C[Fit a trend]
  C --> D{Within data range?}
  D -->|yes| E[Interpolate: defensible]
  D -->|no| F[Extrapolate: flag the risk]
```

Figure 22. Mermaid diagram

> **Warning.** A linear fit through three averaged points reports $R^2 = 0.999$, but that number describes how well the line passes through three dots — not how well it predicts the next experiment. Treat it as a reason to collect more data, not as a conclusion. Foundational guidance on this trap appears in [Kim, 2020] and [Wilson, 2021].

17.5 Summary

A case study moves from raw **observable**s to a summary, a fitted trend, and — crucially — an honest boundary around what the trend can claim. Here, averaging and a `linear_fit` recovered a clean dose–response slope of about 1.4 units per step with an intercept that matched the control, but the same fit must not be pushed beyond the measured range. The reusable move is: summarise, fit, then state the limits.

17.6 Key Terms

observable, **gradient**, **threshold**, **model**.

17.7 Further Reading

- Kim [2020] — practical guidance on summarising and visualising small datasets before fitting anything.
- Wilson [2021] — on the difference between a good fit and a good prediction.

17.8 Practice

- **Lab:** sec. 29 — re-run the grouping and fit, then add a fourth condition and watch the fit change.
- **Question bank:** sec. 41 — recall through synthesis.

18 Frontiers and Open Problems

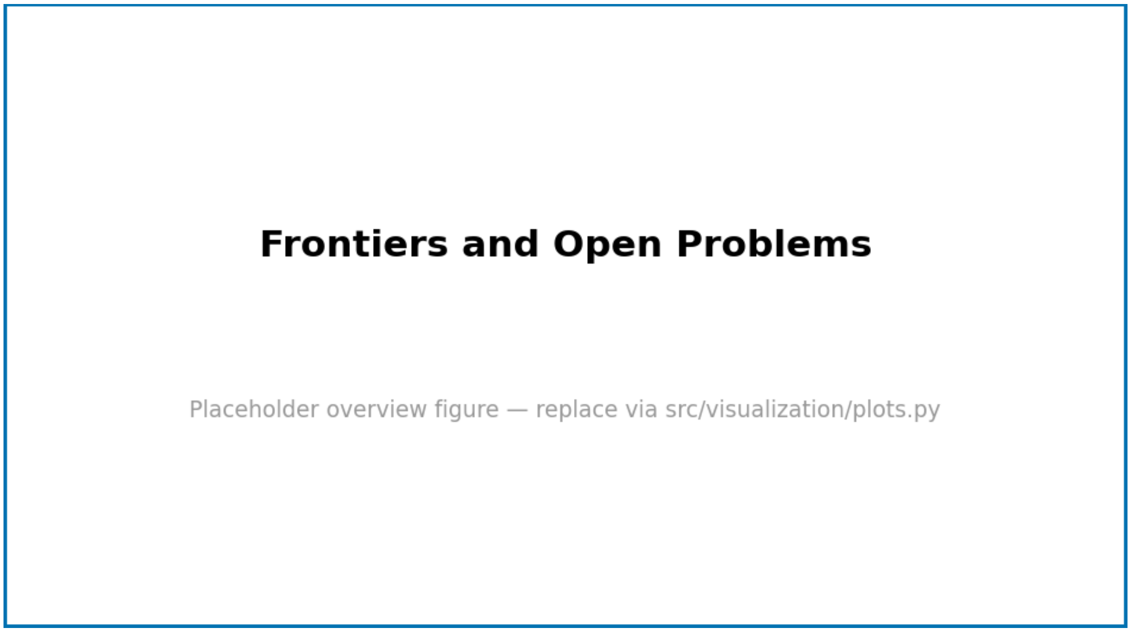


Figure 23. Overview schematic for "Frontiers and Open Problems". Replace this generated placeholder with a real figure produced by `src/visualization/plots.py`.

Level 1/3 · 30 min read · 45 min lecture · Prerequisites: none

18.1 Learning Objectives

By the end of this chapter you should be able to:

1. TODO: state the first measurable learning objective.
2. TODO: state the second learning objective.
3. TODO: state the third learning objective.

18.1.1 Study Blueprint

- **Big idea:**
- **Core concepts:** **state**, **observable**, **system**.
- **Quantitative lens:** the worked formalism in eq. 13.
- **Data skill:**
- **Common misconception to repair:**
- **Primary lab:** sec. 30.
- **Question bank:** sec. 42.
- **Bridge to computation:** `textbook.models`.

Opening Vignette: TKTK — a motivating story

18.2 Orientation

This section introduces the central ideas of *Frontiers and Open Problems*. Foundational treatments include [Wilson, 2021, Taylor, 2019]. Key terms such as **state**, **observable**, **system** are defined in the glossary.

18.3 A Worked Formalism

The recurring quantitative model for this chapter is shown in eq. 13:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{13}$$

It is implemented and tested in `textbook.models.logistic_growth`; never retype the maths in prose or scripts — call the tested function. The parameters appear in tbl. 12.

Table 12. Parameters of the worked model for this chapter.

Symbol	Meaning	Units
r	intrinsic rate	1/time
K	carrying capacity	quantity
N_0	initial value	quantity

A concept map of how the pieces fit together:

```
graph TD
  A[Inputs / assumptions] --> B[Model]
  B --> C[Predictions]
  C --> D[Comparison with data]
  D -->|revise| A
```

Figure 24. Mermaid diagram

Note

18.4 Going Deeper

See the overview in fig. 23 and revisit the objectives above as you read. Further evidence: [Wilson, 2021, Taylor, 2019].

18.5 Summary

18.6 Key Terms

threshold, **network**, **dynamics**, **emergence**.

18.7 Further Reading

- TODO: one-line annotation [Smith, 2020].

18.8 Practice

- **Lab:** sec. 30
- **Question bank:** sec. 42

19 Lab — Orientation to the Field

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

19.1 Objectives

19.2 Background

Linked chapter: sec. 4.

19.3 Procedure

1. TODO: first step.
2. TODO: second step.

19.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

19.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

19.6 Reflection

20 Lab — Core Methods and Tools

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

20.1 Objectives

20.2 Background

Linked chapter: sec. 5.

20.3 Procedure

1. TODO: first step.
2. TODO: second step.

20.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

20.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

20.6 Reflection

21 Lab — Quantitative Foundations

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

21.1 Objectives

21.2 Background

Linked chapter: sec. 6.

21.3 Procedure

1. TODO: first step.
2. TODO: second step.

21.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

21.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

21.6 Reflection

22 Lab — First Principles

Lab · 60 min · Materials: a computer with the project installed (`uv sync`), or graph paper and a calculator

22.1 Objectives

After this lab you will be able to (1) generate a logistic trajectory for chosen parameters, (2) read its carrying capacity and half-rise time from a plot, and (3) recover the growth rate from data with a simple fit.

22.2 Background

This lab makes the chapter concrete: see sec. 8 for the logistic law and the meaning of r, K, and N_0. You will produce the same S-curve shown in fig. 7 and then work backwards from numbers to parameters.

22.3 Procedure

1. Install the project: `uv sync`. Confirm the tests pass: `uv run --extra dev python -m pytest tests/test_models.py -q`.
2. Generate a trajectory for $r = 0.8$, $K = 100$, $N_0 = 5$ at $t = 0, 1, 2, \ldots, 12$ using the computational workflow below.
3. Plot N against t. Mark the value the curve approaches and the time at which N first exceeds $K/2 = 50$.
4. Repeat for $r = 0.4$ and $r = 1.2$, keeping K and N_0 fixed. Overlay the three curves.

22.4 Analysis

Summarise each trajectory with `textbook.models.descriptive_statistics(N)` and tabulate the maximum value and the time-to-half-capacity for each r. Describe, in one sentence, how increasing r changes the curve without changing where it ends.

22.5 Computational Workflow

```python
import numpy as np
from textbook import models

t = np.arange(0, 13, 1.0)
for r in (0.4, 0.8, 1.2):
    N = models.logistic_growth(t, r=r, carrying_capacity=100.0, initial=5.0)
    t_half = t[np.argmax(N > 50)]
    print(f"r={r}: N(12)={N[-1]:.1f}, first t with N>50 is t={t_half:.0f}")
    print("  stats:", models.descriptive_statistics(N))
```

22.6 Reflection

1. Two of your three curves end at almost the same value. Which parameter sets that value, and why is it independent of r?
2. A colleague claims the population “will keep growing forever.” Using eq. 5, explain in one sentence why it will not.
3. If your measured data rose past K and then fell back, what would that tell you about the adequacy of the logistic model for that system?

23 Lab — Building Blocks

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

23.1 Objectives

23.2 Background

Linked chapter: sec. 9.

23.3 Procedure

1. TODO: first step.
2. TODO: second step.

23.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

23.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

23.6 Reflection

24 Lab — Structure and Form

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

24.1 Objectives

24.2 Background

Linked chapter: sec. 10.

24.3 Procedure

1. TODO: first step.
2. TODO: second step.

24.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

24.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

24.6 Reflection

25 Lab — Systems Overview

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

25.1 Objectives

25.2 Background

Linked chapter: sec. 12.

25.3 Procedure

1. TODO: first step.
2. TODO: second step.

25.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

25.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

25.6 Reflection

26 Lab — Dynamics and Change

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

26.1 Objectives

26.2 Background

Linked chapter: sec. 13.

26.3 Procedure

1. TODO: first step.
2. TODO: second step.

26.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

26.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

26.6 Reflection

27 Lab — Regulation and Control

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

27.1 Objectives

27.2 Background

Linked chapter: sec. 14.

27.3 Procedure

1. TODO: first step.
2. TODO: second step.

27.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

27.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

27.6 Reflection

28 Lab — Applied Models

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

28.1 Objectives

28.2 Background

Linked chapter: sec. 16.

28.3 Procedure

1. TODO: first step.
2. TODO: second step.

28.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

28.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

28.6 Reflection

29 Lab — Case Studies

Lab · 75 min · Materials: the project installed (`uv sync`), `assets/data/sample_dataset.csv`

29.1 Objectives

After this lab you will be able to (1) load grouped data, (2) compute per-group means and an overall summary, (3) fit a dose–response trend, and (4) state where the fit may and may not be trusted.

29.2 Background

This lab operationalises sec. 17. You will reproduce the chapter's numbers (group means 2.20, 3.50, 4.95; fitted slope ≈ 1.375) and then probe the fit's limits by adding a condition.

29.3 Procedure

1. Read `assets/data/sample_dataset.csv` into three groups (control, low, high).
2. Compute each group's mean and the overall summary with `textbook.models.descriptive_statistics`.
3. Encode dose as $0, 1, 2$ and fit a line with `textbook.models.linear_fit`. Confirm slope ≈ 1.375, intercept ≈ 2.175, $R^2 \approx 0.999$.
4. Add a hypothetical fourth condition (dose 3) with a *saturating* response — say a mean of 5.3 rather than the linear prediction of 6.3 — refit, and compare the new slope and R^2.

29.4 Analysis

Tabulate the slope, intercept, and R^2 for the three-point and four-point fits. In two sentences, explain why the four-point fit's slope falls and what that implies about extrapolating the original line.

29.5 Computational Workflow

```
import csv
import numpy as np
from textbook import models

rows = list(csv.DictReader(open("manuscript/assets/data/sample_dataset.csv")))
groups: dict[str, list[float]] = {}
for r in rows:
    groups.setdefault(r["condition"], []).append(float(r["measurement"]))

means = {k: float(np.mean(v)) for k, v in groups.items()}
dose = np.array([0.0, 1.0, 2.0])
response = np.array([means["control"], means["treatment_low"], means["treatment_high"]])
print(models.linear_fit(dose, response))
```

29.6 Reflection

1. The intercept of the fit was close to the control mean. Why is that a useful internal consistency check?
2. Your four-point fit changed the slope. Which is the more honest summary of the system, and what would you measure next to decide?

3. When is a straight-line dose–response model adequate, and when does the saturating model from sec. 8 become necessary?

30 Lab — Frontiers and Open Problems

Lab · 60 min · Materials: notebook, calculator (or `textbook.models`)

30.1 Objectives

30.2 Background

Linked chapter: sec. 18.

30.3 Procedure

1. TODO: first step.
2. TODO: second step.

30.4 Analysis

Summarise results with `textbook.models.descriptive_statistics`.

30.5 Computational Workflow

```
from textbook.models import logistic_growth
# TODO: parameterise and plot the model for this lab's scenario.
```

30.6 Reflection

31 Question Bank — Orientation to the Field

Linked chapter: sec. 4.

31.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

31.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

31.3 Synthesis

5. TODO. *(Answer:)*

32 Question Bank — Core Methods and Tools

Linked chapter: sec. 5.

32.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

32.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

32.3 Synthesis

5. TODO. *(Answer:)*

33 Question Bank — Quantitative Foundations

Linked chapter: sec. 6.

33.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

33.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

33.3 Synthesis

5. TODO. *(Answer:)*

34 Question Bank — First Principles

Linked chapter: sec. 8. Answers follow each question in italics; in a print build, move them to an answer key if you prefer.

34.1 Recall

1. In the logistic model $N(t)$, which symbols are parameters and which is the variable? *(Answer: r, K, and N_0 are parameters held fixed; $N(t)$ is the variable that changes with time t.)*
2. What value does a logistic trajectory approach as $t \to \infty$ when $r > 0$? *(Answer: the carrying capacity K.)*

34.2 Application

3. With $r = 0.8$, $K = 100$, $N_0 = 5$, the chapter reports $N(5) = 74.18$ and $N(10) = 99.37$. Roughly what fraction of the carrying capacity remains unfilled at $t = 10$? *(Answer: about 0.63% — $N(10)$ is within one part in a hundred of K.)*
4. Doubling r from 0.4 to 0.8 changes the curve in what way, and what does it leave unchanged? *(Answer: the curve rises and reaches the neighbourhood of K sooner — a steeper S — but the final level K is unchanged because K is a separate parameter.)*

34.3 Synthesis

5. A dataset grows nearly exponentially at first, then levels off, but the plateau sits well below the resources you believe are available. Propose one modification to the logistic model that could explain a lower-than-expected plateau, and name the parameter it would change. *(Answer: any mechanism that effectively lowers the carrying capacity — competition, a toxin, or a resource cap — would reduce K; the plateau tracks K, not the nominal resource supply. Accept reasoned alternatives.)*

35 Question Bank — Building Blocks

Linked chapter: sec. 9.

35.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

35.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

35.3 Synthesis

5. TODO. *(Answer:)*

36 Question Bank — Structure and Form

Linked chapter: sec. 10.

36.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

36.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

36.3 Synthesis

5. TODO. *(Answer:)*

37 Question Bank — Systems Overview

Linked chapter: sec. 12.

37.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

37.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

37.3 Synthesis

5. TODO. *(Answer:)*

38 Question Bank — Dynamics and Change

Linked chapter: sec. 13.

38.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

38.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

38.3 Synthesis

5. TODO. *(Answer:)*

39 Question Bank — Regulation and Control

Linked chapter: sec. 14.

39.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

39.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

39.3 Synthesis

5. TODO. *(Answer:)*

40 Question Bank — Applied Models

Linked chapter: sec. 16.

40.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

40.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

40.3 Synthesis

5. TODO. *(Answer:)*

41 Question Bank — Case Studies

Linked chapter: sec. 17. Answers follow each question in italics.

41.1 Recall

1. What two summary numbers does the chapter report for each condition group, and which function computes them? *(Answer: the per-group mean — e.g.* 2.20, 3.50, 4.95 *— plus an overall spread from* `textbook.models.descriptive_statistics`*, which returns mean, std, min, max, and count.)*
2. What do the slope and intercept of the fitted line eq. 12 represent? *(Answer: the slope (*≈ 1.375*) is the response added per dose step; the intercept (*≈ 2.175*) is the predicted response at dose* 0*, which should match the control mean.)*

41.2 Application

3. The intercept (2.175) is close to the control mean (2.20). Why is this a useful check rather than a coincidence to ignore? *(Answer: a fit whose dose-*0 *prediction matches the measured control is internally consistent; a large mismatch would signal a coding error or a poor model.)*
4. The line predicts $\hat{y}(3) = 6.3$. Why should this prediction be flagged? *(Answer: dose* 3 *is outside the measured range* $[0, 2]$ *— it is an extrapolation, and real dose–response curves usually saturate, so a line will overpredict.)*

41.3 Synthesis

5. With only three averaged points the fit reports $R^2 = 0.999$. Explain why this is weak evidence and describe one experiment that would strengthen — or overturn — the linear conclusion. *(Answer:* R^2 *near* 1 *on three points mostly reflects that a line can pass through three dots; it says little about prediction. Collecting more doses, especially higher ones, would reveal curvature/saturation and test whether the linear model generalises. Accept reasoned alternatives.)*

42 Question Bank — Frontiers and Open Problems

Linked chapter: sec. 18.

42.1 Recall

1. TODO. *(Answer:)*
2. TODO. *(Answer:)*

42.2 Application

3. TODO. *(Answer:)*
4. TODO. *(Answer:)*

42.3 Synthesis

5. TODO. *(Answer:)*

43 Appendix A — Authoring Guide: Filling the Stubs

Reference appendix · For authors and contributors · Read this first.

This is the most important file in the template. It explains how to turn the empty scaffold into a finished book by **filling stubs** and **growing** structure from a single source of truth. Read it once end-to-end, then keep it open while you work.

43.1 The Big Idea: One Source of Truth

The book is **data-driven from `config.yaml`**. The list of parts, chapters, labs, question banks, and reference appendices lives there and nowhere else. The Python engine in `src/textbook/` reads that file and the scaffolding scripts in `scripts/` materialise the matching markdown files. You never hand-number a chapter, figure, equation, or section — pandoc-crossref does that at render time from the labels you write.

The flow is always the same:

```
edit config.yaml  ->  scaffold_chapter.py  ->  fill <!-- STUB --> blocks
      ->  add figures / glossary terms / references  ->  audit + tests
```

43.2 Step 1 — Decide the Structure in `config.yaml`

Open `config.yaml`. To add or rename a chapter, edit the `parts:` tree (each chapter has a `stem` and a `title`); to add a lab or question bank, add an entry under `appendices.labs` / `appendices.questions` for the matching part. The stem drives every downstream name:

- chapter file → `manuscript/<part>/<NN>_<stem>.md`, label `{#sec:<part>_<stem>}`
- lab file → `manuscript/labs/<part>/lab_<stem>.md`, label `{#sec:lab_<part>_<stem>}`
- question bank → `manuscript/questions/<part>/q_<stem>.md`, label `{#sec:q_<part>_<stem>}`

Keep stems short, lowercase, and `snake_case`. Do not invent a numbering scheme; the table of contents and labels are derived in `src/textbook/toc.py`.

43.3 Step 2 — Materialise the Stubs

Run the scaffolder. It only **creates files that are missing**, so it is safe to re-run after every config change:

```
uv run python scripts/scaffold_chapter.py
```

This writes a chapter (or lab, or question bank) pre-populated with every required structural element as `<!-- STUB -->` markers. Your job is to replace the stubs with real prose, never to delete the structure.

43.4 Step 3 — Fill the Content Contract

Every chapter must carry these elements (enforced by `validate_chapter` in `src/textbook/content.py` and by the integrity tests). Fill each, keep the markers' surrounding structure:

- a labelled H1: `# Title {#sec:<part>_<stem>}`
- a metadata badge line: `<!-- chapter-metadata-badge -->`
- a Study Blueprint: `<!-- curriculum-scaffold-start -->`
- **Learning Objectives**
- one figure with alt text and a crossref label `{#fig:...}`
- a worked formalism: an equation `{#eq:...}` plus a parameter table `{#tbl:...}`

- an inline ```mermaid diagram
- **Summary**, **Key Terms**, **Further Reading**, **Practice** sections

Cross-reference syntax is pandoc-crossref: [@fig:..], [@tbl:..], [@eq:..], [@sec:..]. Stub markers the audit counts are <!-- STUB -->, TODO:, and TKTK — drive these to zero as the chapter matures.

43.5 Step 4 — Add a Figure

Figures are deterministic matplotlib output. Each chapter expects a placeholder named <part_id>_<stem>.png. To make it real, add or edit a function in src/visualization/plots.py (the four worked figures show the pattern), then regenerate:

```
uv run python scripts/generate_figures.py
```

For diagrams, edit src/mermaid/diagram_specs.yaml and run scripts/generate_diagrams.py (PNG, or .mmd fallback). Reference the figure in prose with [@fig:<part>_<stem>] and give it alt text.

43.6 Step 5 — Add a Glossary Term

Glossary anchors are a **closed contract**. To add a term you must update **both**:

1. GLOSSARY_ANCHORS in src/textbook/constants.py
2. the matching entry in glossary.md

Link a term in prose as [**term**](#gl:<anchor>). The current anchors are system, model, parameter, variable, equilibrium, feedback, gradient, threshold, network, dynamics, emergence, regulation, boundary, state, observable.

43.7 Step 6 — Add a Reference

Citations are [@key] and must resolve in references.bib. Add a BibTeX entry, then cite it. Keep CITATION_KEYS in constants.py in sync if you add a key that the structural contract should track.

43.8 Step 7 — Audit and Test

Two gates decide whether the book is healthy:

```
# Quality gate: counts stubs, checks the content contract per file
uv run python scripts/audit_textbook_quality.py

# Full test suite (engine + manuscript integrity)
uv run --extra dev python -m pytest
```

tests/test_manuscript_integrity.py verifies that every chapter satisfies the content contract, that labels are unique, that citations resolve, and that glossary links point at real anchors. Green tests with remaining stubs mean the **structure** is correct but the **content** is unfinished — that is the normal state of a freshly scaffolded chapter.

43.9 Growing the Book

To add a whole new chapter end-to-end:

1. add its stem/title under the right part in config.yaml;

2. add matching lab and question-bank entries under `appendices`;
3. run `scripts/scaffold_chapter.py`;
4. fill the stubs (Steps 3–6);
5. run the audit and tests until the contract passes.

See also: Appendix B — Notation, Appendix C — Mathematical Review, and the appendices README.

44 Appendix B — Notation and Symbols

Reference appendix · Symbol glossary for the worked models.

This appendix lists the symbols used by the worked formalisms in `src/textbook/models.py`. Keep it in sync with the parameter tables (`{#tbl:...}`) in the chapters: every symbol that appears in an equation `{#eq:...}` should have a row here.

44.1 Conventions

- Scalars are italic lowercase (e.g. r, t); sets and spaces are uppercase.
- Subscript `0` denotes an initial value (e.g. N_0 at $t = 0$).
- Units are stated in SI / metric unless a chapter declares otherwise.

44.2 Symbol Table

Symbol	Name	Appears in	Units	Notes
t	time / independent variable	all dynamic models	s (or chapter unit)	
N	state quantity / population	`logistic_growth`	dimensionless	
N_0	initial state	`logistic_growth`, `exponential_decay`	dimensionless	
r	intrinsic growth rate	`logistic_growth`	1/time	
K	carrying capacity	`logistic_growth`	dimensionless	
λ (lambda)	decay constant	`exponential_decay`, `half_life`	1/time	
t½	half-life	`half_life`	time	
Vmax	maximum response	`saturating_response`	response unit	
K_{m}	half-saturation constant	`saturating_response`	input unit	
x, y	paired observations	`linear_fit`	data unit	
m, b	slope, intercept	`linear_fit`	derived	
μ, σ	mean, standard deviation	`descriptive_statistics`	data unit	

See also: Appendix C — Mathematical Review.

45 Appendix C — Mathematical Review

Reference appendix · Just-enough mathematics for the worked models.

A brief refresher on the mathematics the book relies on. Each section ties directly to a function in `src/textbook/models.py` so you can move from formula to tested code without a gap. Replace the stubs with the depth your audience needs.

45.1 Functions and Variables

A function maps an input to an output; here state quantities depend on a **variable** such as time, tuned by **parameters**. See `normalize_unit_interval` for rescaling inputs to [0, 1].

45.2 Growth and Decay

Logistic growth (`logistic_growth`) rises then saturates at the carrying capacity K; exponential decay (`exponential_decay`) falls by a fixed fraction per unit time, summarised by the half-life $t½$ (`half_life`). These are the canonical examples of **dynamics**.

45.3 Saturating Responses

The saturating (Michaelis–Menten-style) response (`saturating_response`) climbs toward Vmax with half-maximum at K_{m} — a recurring shape whenever a resource or signal becomes limiting near a **threshold**.

45.4 Linear Fits

Least-squares fitting (`linear_fit`) returns a slope m and intercept b for paired data, the simplest way to summarise a trend before reaching for a richer **model**.

45.5 Basic Statistics

Descriptive statistics (`descriptive_statistics`) report the mean μ and standard deviation σ — the first numbers to compute on any **observable** before inference.

See also: Appendix B — Notation and the Authoring Guide.

46 Appendix — Formalisms

A worked example of every **formal** element a technical book uses: definitions, theorems with proofs, lemmas, algorithms in pseudocode, step-by-step derivations, systems of numbered equations, and dimensioned quantities. Each maps to a tested function in `src/textbook/models.py`, so the prose and the code stay in agreement.

> **Convention.** Theorem-like environments below use a portable bold-label block-quote form that renders in every target. If your render profile loads `amsthm` (the preamble does), you may instead use native LaTeX `theorem`, `lemma`, `definition`, and `proof` environments.

46.1 1. Definitions

> **Definition 1 (Equilibrium).** A state N^* of a dynamical system $\dot{N} = f(N)$ is an *equilibrium* if $f(N^*) = 0$. See **equilibrium**.

> **Definition 2 (Carrying capacity).** For logistic dynamics, the *carrying capacity* K is the non-zero equilibrium toward which trajectories converge.

46.2 2. A theorem with proof

> **Theorem 1 (Logistic limit).** For the logistic model in eq. 14 with $r > 0$ and $0 < N_0 \le K$, the trajectory satisfies $\lim_{t\to\infty} N(t) = K$.

$$N(t) = \frac{K}{1 + Ae^{-rt}}, \qquad A = \frac{K - N_0}{N_0}. \tag{14}$$

Proof. Because $r > 0$, the term $e^{-rt} \to 0$ as $t \to \infty$. Hence the denominator $1 + Ae^{-rt} \to 1$, and therefore $N(t) \to K/1 = K$. The constant $A \ge 0$ follows from $0 < N_0 \le K$, so $N(t)$ is increasing and the limit is approached from below. ■

This is exactly the asymptotic behaviour the test `tests/test_models.py::test_logistic_growth_starts_at_initial_and_approaches_capacity` verifies numerically — the proof and the test assert the same fact.

46.3 3. A lemma

> **Lemma 1 (Half-life).** For exponential decay $y(t) = y_0 e^{-\lambda t}$ with $\lambda > 0$, the time at which y falls to half its initial value is $t_{1/2} = \ln 2/\lambda$, independent of y_0.

Proof. Set $y(t_{1/2}) = y_0/2$. Then $e^{-\lambda t_{1/2}} = 1/2$, so $-\lambda t_{1/2} = -\ln 2$, giving $t_{1/2} = \ln 2/\lambda$. The result does not depend on y_0. ■

Implemented as `textbook.models.half_life`.

46.4 4. Algorithms (pseudocode)

Where the preamble provides an algorithm package, use it; otherwise this fenced form renders everywhere.

```
Algorithm 1: Ordinary least-squares line fit
Input : points (x_i, y_i), i = 1..n,  n >= 2
Output: slope m, intercept b, coefficient of determination R^2
1  x_bar <- mean(x);  y_bar <- mean(y)
2  m <- sum((x_i - x_bar)(y_i - y_bar)) / sum((x_i - x_bar)^2)
3  b <- y_bar - m * x_bar
4  SS_res <- sum((y_i - (m x_i + b))^2)
5  SS_tot <- sum((y_i - y_bar)^2)
6  R^2 <- 1 - SS_res / SS_tot      (define R^2 = 1 when SS_tot = 0)
7  return (m, b, R^2)
```

This is `textbook.models.linear_fit`; Step 6's degenerate case is covered by `tests/test_models.py::test_linear_fit_constant_y_gives_r_squared_one`.

46.5 5. A step-by-step derivation

Starting from the logistic differential equation and separating variables:

$$\begin{aligned}
\frac{dN}{dt} &= rN\left(1 - \frac{N}{K}\right) \\
\int \frac{dN}{N(1 - N/K)} &= \int r\,dt \\
\ln\left(\frac{N}{K - N}\right) &= rt + C \\
N(t) &= \frac{K}{1 + Ae^{-rt}}, \qquad A = e^{-C},
\end{aligned}$$

which recovers eq. 14. Each line is one algebraic move; show your work at this granularity so readers can follow without gaps.

46.6 6. A system of numbered equations

A simple predator–prey system, with each equation individually referenceable (eq. 15, eq. 16):

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{15}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{16}$$

The vector-field figure style for such systems is demonstrated by the quiver plot in the format gallery (sec. 47).

46.7 7. Dimensioned quantities

State parameters with units explicitly, in math mode so they render in every target:

Table 13. Worked-model parameters with representative dimensioned values.

Symbol	Quantity	Example value
r	intrinsic rate	$0.8\ \mathrm{s}^{-1}$
λ	decay constant	$0.5\ \mathrm{s}^{-1}$
$t_{1/2}$	half-life	$1.386\ \mathrm{s}$
K	carrying capacity	100 individuals

46.8 8. Notation summary

Symbols used throughout are collected in the notation appendix (sec. 44); glossary definitions for narrative terms such as **gradient** and **threshold** are in the master glossary.

47 Appendix — Format Gallery

This appendix is a **kitchen-sink demonstration**: a working example of every content primitive this template supports. Copy any block into a chapter and adapt it. Each example is real and renders through the standard pipeline; figures are produced deterministically by `src/visualization/` and embedded from `../figures/`.

> **How to read this appendix.** Headings group primitives by kind: text, lists, callouts, tables, math, figures, diagrams, code, cross-references, media, and pedagogy blocks. The Markdown source is the example — view it next to the rendered output.

47.1 1. Text and inline formatting

Plain paragraph text wraps and flows normally. Inline styles: **bold**, *italic*, ***bold italic***, `inline code`, ~~strikethrough~~, H_2O with a subscript, $E = mc^2$ with a superscript, and a footnote.[1]

You can hard-break a line
with two trailing spaces, or separate paragraphs with a blank line. Escape literal Markdown with a backslash: *not italic*.

47.2 2. Lists

Unordered, with nesting:

- First item
- Second item
  - Nested item
  - Another nested item
    - Third level
- Third item

Ordered:

1. Step one
2. Step two
   1. Sub-step a
   2. Sub-step b
3. Step three

Task list (renders as checkboxes in many targets):

- [x] Scaffold the chapter
- [x] Generate figures
- [ ] Fill the prose
- [ ] Write the assessment answers

Definition list:

Parameter A fixed quantity that configures a model (see **parameter**).

[1] Footnotes collect at the end of the document (or page, in PDF). Use them for asides that would interrupt the sentence.

Variable A quantity that changes across states (see **variable**).

47.3 3. Block quotes and callouts

A plain block quote:

> "Form follows function." Use quotes for epigraphs and primary-source extracts.

Portable callouts (a bold label inside a block quote — renders in every target):

> **Note.** A neutral aside that adds context.

> **Tip.** A practical suggestion the reader can act on.

> **Warning.** A caveat, common error, or safety note.

> **Example.** A short worked illustration inline in the text.

> **Definition.** A precise statement of a term, often paired with a glossary entry such as **equilibrium**.

Pandoc fenced-div callout (richer styling where supported; falls back gracefully):

This is a Pandoc fenced `div`. If your render profile styles `.callout-note`, it appears as a boxed admonition; otherwise it renders as a normal block.

47.4 4. Tables

A simple table with column alignment and a cross-referencable caption (tbl. 14):

Table 14. Column alignment — left, centre, right.

Left	Centre	Right
alpha	1	10.0
beta	22	2.5
gamma	333	0.125

A multi-line / grid table (cells may contain longer wrapped text):

Symbol	Meaning	Typical range
r	intrinsic rate of change	0.1 – 2.0
K	carrying capacity / saturation level the system approaches	problem- dependent

---

47.5 5. Mathematics and units

Inline math: the half-life is $t_{1/2} = \ln 2/\lambda$.

A numbered display equation, cross-referenced as eq. 17:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{17}$$

Multi-line aligned derivation:

$$\begin{aligned} \frac{dN}{dt} &= rN\left(1 - \frac{N}{K}\right) \\ &= rN - \frac{r}{K}N^2. \end{aligned}$$

A matrix and a piecewise definition:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \qquad f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0. \end{cases}$$

Physical quantities with units, written in math mode so they render in **every** target (PDF, HTML, slides): a rate of $0.5\ \mathrm{s}^{-1}$, a length of $2.0\ \mathrm{m}$, and a concentration of $1.5\ \mathrm{mol\,L^{-1}}$. (For PDF-only builds you may instead use `siunitx` macros such as `\SI{0.5}{\per\second}`, which the preamble loads — but math-mode units are the portable choice.)

47.6 6. Figures

A single figure with caption, label, and alt text, cross-referenced as fig. 25:

Two figures side by side (Pandoc fenced div; falls back to stacked):

A multi-panel composite (fig. 28):

The full plot-type gallery lives in `../figures/gallery/` and includes: line, scatter-with-fit, bar, grouped bar, horizontal bar, histogram, box, violin, heatmap, contour, quiver field, step, stacked area, error bars, log-log, pie, annotated, and multi-panel.

47.7 7. Diagrams (Mermaid)

The pipeline renders fenced `mermaid` blocks to figures (and falls back to the `.mmd` source if the Mermaid CLI is absent). One worked example of each kind the builders in `src/mermaid/diagrams.py` support:

Flowchart:

Sequence:

State:

Class:

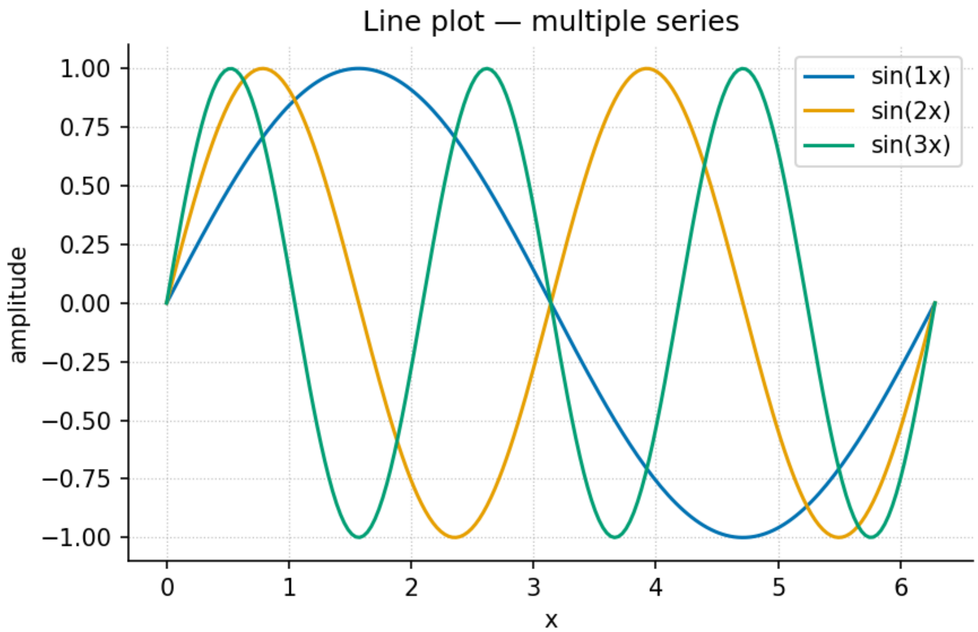


Figure 25. A multi-series line plot of three sine waves, produced by `visualization.gallery.line_plot`.

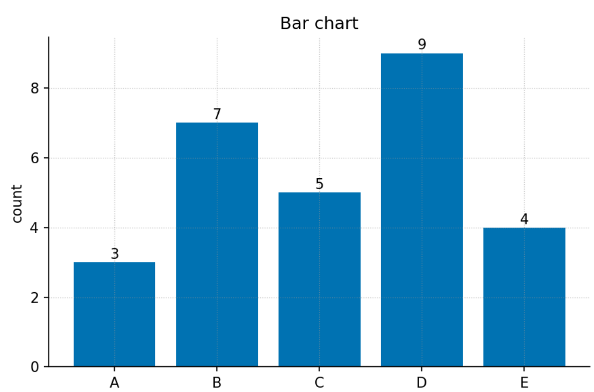


Figure 26. Bar chart.

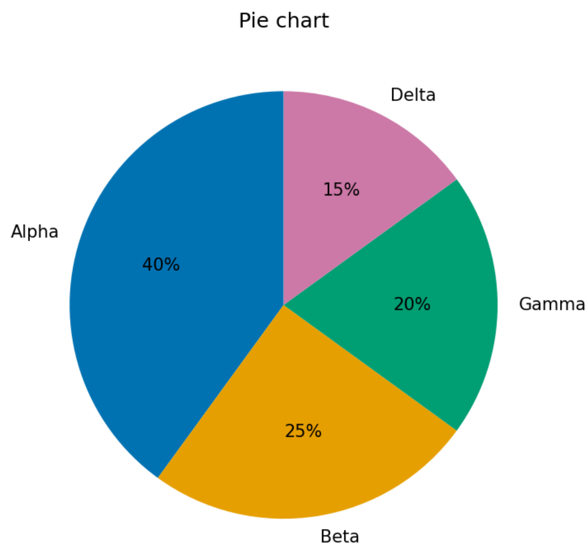


Figure 27. Pie chart.

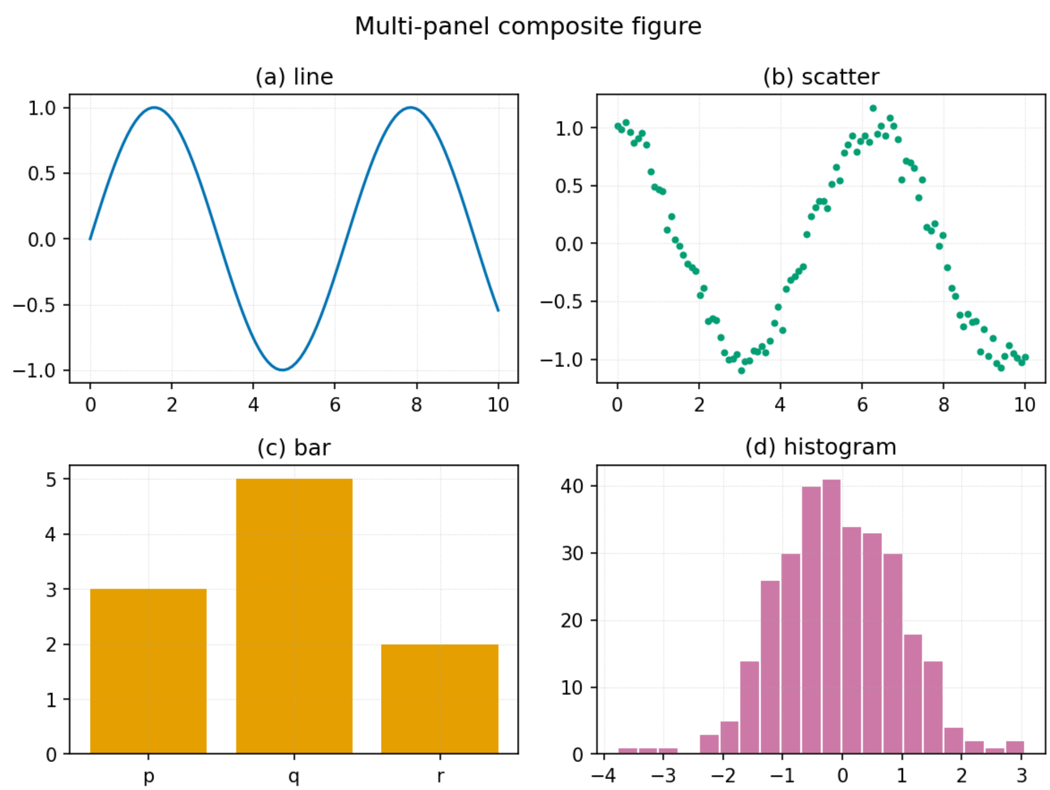


Figure 28. A 2×2 composite: line, scatter, bar, and histogram panels.

```
graph TD
  A[Inputs] --> B[Model]
  B --> C[Predictions]
  C -->|revise| A
```

Figure 29. Mermaid diagram

```
sequenceDiagram
  participant Author
  participant Engine
  Author ->> Engine: edit config.yaml
  Engine -->> Author: rendered PDF
```

Figure 30. Mermaid diagram

```
stateDiagram-v2
  [*] --> Stub
  Stub --> Drafted: fill
  Drafted --> Reviewed: review
  Reviewed --> [*]: publish
```

Figure 31. Mermaid diagram

Entity-relationship:

Pie, Gantt, mindmap, timeline, quadrant, and user-journey diagrams are also supported — see `src/mermaid/diagram_specs.yaml` for a worked spec of each.

47.8 8. Code

Inline code: call `textbook.models.logistic_growth(t, r=..., ...)`.

A fenced code block with a language (syntax-highlighted) and a caption (lst. 1):

A shell example:

```
uv run python scripts/generate_figures.py
uv run --extra dev python -m pytest tests/ --cov=src
```

47.9 9. Cross-references and citations

Cross-references resolve by label: figure fig. 25, table tbl. 14, equation eq. 17, and section sec. 46. Never hand-number — Pandoc fills these in.

Citations resolve against `references.bib`: a single source [Smith, 2020], multiple sources [Doe, 2019, Lee, 2021], and an in-text form — Garcia [2022] showed the effect first. A locator narrows the reference [Patel, 2018, pp. 12–14].

```
classDiagram
  class ChapterRef {
    +str part_id
    +str file
    +stem() str
  }
  ChapterRef --> TocEntry : numbered as
```

Figure 32. Mermaid diagram

```
erDiagram
  PART ||--o{ CHAPTER : contains
  CHAPTER ||--|| LAB : has
```

Figure 33. Mermaid diagram

Listing 1 Calling the tested computational backbone.

```python
from textbook import models
import numpy as np

t = np.linspace(0, 10, 100)
n = models.logistic_growth(t, r=0.8, carrying_capacity=100.0, initial=5.0)
print(n[-1])  # -> approaches the carrying capacity
```

47.10 10. Media and data

Embedded raster image (any PNG/JPG works the same way as a figure):

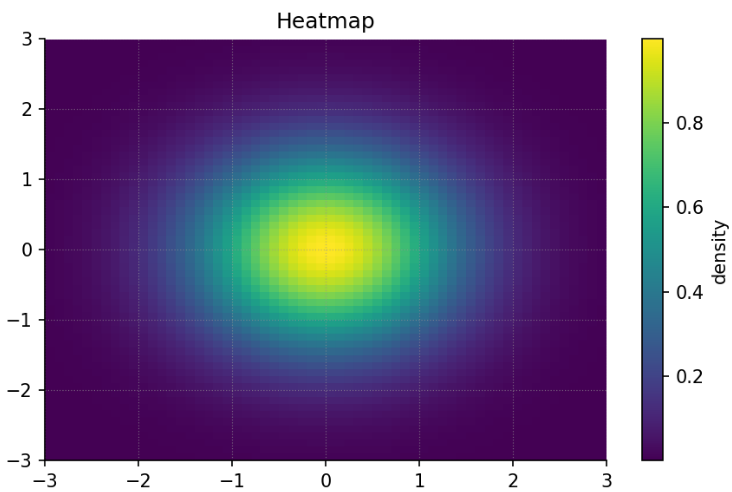


Figure 34. A generated heatmap embedded as a raster image.

Audio and video embed in HTML targets (PDF shows the caption + link). Syntax:

```
![Caption for an audio clip.](../assets/media/clip.mp3)
![Caption for a video.](../assets/media/demo.mp4){width=70%}
```

A downloadable data file lives at `assets/data/sample_dataset.csv`; its contents as a table:

Table 15. Sample dataset (mirrors `assets/data/sample_dataset.csv`).

condition	replicate	measurement	standard_error
control	1	2.10	0.20
control	2	2.30	0.18
treatment_low	1	3.60	0.25
treatment_high	1	4.80	0.35

The error-bar figure fig. 35 visualises this kind of data:

47.11 11. Pedagogical blocks

These are the reusable teaching elements chapters draw on.

Learning objective. After this section a reader can identify which Markdown primitive to use

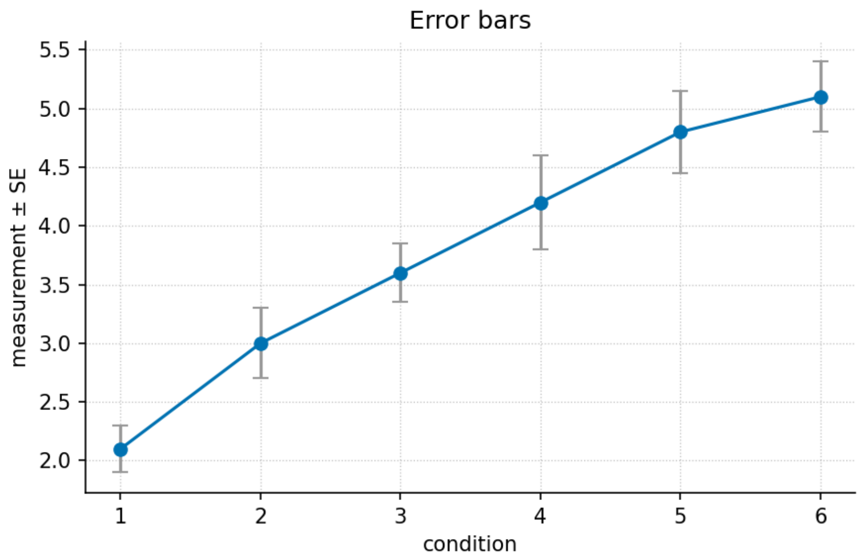


Figure 35. Means with standard-error bars.

Key terms. model, parameter, state.

Summary. This appendix demonstrated text, lists, callouts, tables, math and units, figures, diagrams, code, cross-references, media, and pedagogy blocks — the complete primitive set.

47.12 12. Miscellany

A horizontal rule separates major shifts in topic (three or more dashes):

Raw inline HTML is supported only inside <details>, <aside>, or <callout> per project style; everything else uses Markdown. Unicode renders directly: α, β, γ, Δ, , ∞, $\approx$, $\rightarrow$. For PDF math, prefer LaTeX (`$\alpha$`) over raw Unicode in equations.

48 Master Glossary

48.0.1 Boundary

The interface separating a system from its environment.

48.0.2 Dynamics

How the state of a system changes over time.

48.0.3 Emergence

System-level behaviour not present in the parts taken alone.

48.0.4 Equilibrium

A state in which opposing influences balance and net change is zero.

48.0.5 Feedback

A loop in which a system's output influences its own input.

48.0.6 Gradient

A spatial or quantitative difference that drives flow or change.

48.0.7 Model

A simplified, often quantitative, representation of a system.

48.0.8 Network

A set of elements (nodes) connected by relationships (edges).

48.0.9 Observable

A quantity that can be measured or recorded.

48.0.10 Parameter

A fixed quantity that configures a model's behaviour.

48.0.11 Regulation

The control of a system variable toward a target range.

48.0.12 State

The configuration of a system at a moment in time.

48.0.13 System

A set of interacting parts forming an integrated whole.

48.0.14 Threshold

A critical value at which a qualitative change occurs.

48.0.15 Variable

A quantity that can take different values across states or observations.

49 Appendix E — Index of Key Terms

Reference appendix · Generated index — do not hand-maintain.

This index is intended to be **generated** at build time, not edited by hand. A future indexing pass will scan the chapters for glossary links (`[**term**](#gl:<anchor>)`) and crossref labels (`{#sec:...}`, `{#fig:...}`, `{#tbl:...}`, `{#eq:...}`) and collate page/section references from the rendered PDF. Until that pass runs, the entries below are placeholders.

The authoritative term definitions live in Appendix D — Glossary; the closed list of anchors is `GLOSSARY_ANCHORS` in `src/textbook/constants.py`.

49.1 Placeholder Entries

- **dynamics** — see sec. 13
- **equilibrium** — see sec. 14
- **feedback** — see sec. 14
- **model** — see sec. 6
- **network** — see sec. 10
- **system** — see sec. 12

See also: Appendix D — Glossary.

References

Casey Brown. *First Principles: A Placeholder Reference*. Placeholder Press, 2017.

Jordan Doe. Core methods: A placeholder reference. *Journal of Placeholders*, 1:1–20, 2019.

Robin Garcia. Dynamics and change: A placeholder reference. *Journal of Placeholders*, 4:88–110, 2022.

Dana Kim. Working with data: A placeholder reference. *Journal of Placeholders*, 2:200–225, 2020.

Sam Lee. Systems thinking: A placeholder reference. *Journal of Placeholders*, 3:44–67, 2021.

Minh Nguyen. Synthesis across scales: A placeholder reference. *Journal of Placeholders*, 5:12–34, 2023.

Riya Patel. *Modelling Practice: A Placeholder Reference*. Placeholder Press, 2018.

Alex Smith. *Foundations of the Field: A Placeholder Reference*. Placeholder Press, 2020. STUB entry — replace with a real source.

Quinn Taylor. Toward a theory: A placeholder reference. *Journal of Placeholders*, 1:300–330, 2019.

Drew Wilson. Methods of analysis: A placeholder reference. *Journal of Placeholders*, 3:130–158, 2021.