Statistical Research Methodology

Lecture Notes & Study Materials for DDS 6106

Dr. Shrikrishna Bhat K

2026-07-23

Statistical Research Methodology

Author
Dr. Shrikrishna Bhat K

Affiliation
Assistant Professor
Department of Applied Statistics and Data Science
Prasanna School of Public Health
Manipal Academy of Higher Education (MAHE)
Manipal, Karnataka, India

This book is an independently authored open educational resource developed by the author for teaching and learning purposes. The views expressed are those of the author and do not necessarily reflect the views of Manipal Academy of Higher Education or the Prasanna School of Public Health.

Interactive R Package: `remotes::install_github("kskbhat/srm")`
Online Book: https://kskbhat.github.io/Teaching/srm/

Contents

Course Overview

Welcome to the course notes for **Statistical Research Methodology (Course Code: DDS 6106)**, offered by the Department of Applied Statistics and Data Science at the Prasanna School of Public Health, Manipal Academy of Higher Education (MAHE).

Course Metadata:

- **Course Code**: DDS 6106
- **Programme**: M.Sc. Data Science, Biostatistics / Digital Epidemiology
- **Semester**: Second Year, Semester 3
- **Credits**: 4 Credits ($L-T-P-C : 2-0-4-4$)
- **Prerequisites**: All courses offered till Semester 3.
- **Synopsis**: To provide the necessary foundation to simulate data and apply Markov Chain Monte Carlo (MCMC) techniques.

Course Outcomes (COs)

On successful completion of this course, students will be able to:

CO	Description	Bloom's Level
CO 1	Outline the methodologies to carry out a research study	C4
CO 2	Reproduce appropriate methods of simulation	C1
CO 3	Demonstrate construction and analysis of simulation models	C3
CO 4	Apply the techniques of simulation and MCMC	C3
CO 5	Summarize how simulation and MCMC tools are used in industries to solve real-time problems	C6
CO 6	Estimate and predict using MCMC	C6

How this book is organized

Chapter	Unit	Hours Breakdown	Key Topics
Chapter 1	Unit 1: Introduction to Research Methodology	07L + 05 SDL	Research design, literature survey, protocol/thesis/manuscript (IMRAD), sampling designs
Chapter 2	Unit 2: Introduction to Simulation	05L + 02 SDL + 14 PRC	LCG, Inverse-Transform, Accept-Reject, Poisson processes, EDF & influence functions
Chapter 3	Unit 3: Monte Carlo Methods	10L + 05 SDL + 10 PRC	Variance reduction (control/antithetic variates, stratified, importance sampling), Bootstrap/Jackknife
Chapter 4	Unit 4: Markov Chain Monte Carlo	05L + 05 SDL + 10 PRC	MCMC foundations, Metropolis-Hastings, Gibbs sampler, Ising/Potts, CFTP, Bayesian statistics
Chapter 5	Unit 5: Monte Carlo Optimization	03L + 03 SDL + 06 PRC	Score function method, Robbins-Monro stochastic approximation, multi-level splitting, adaptive MCMC

Course Notation Guide

To assist students navigating statistical notation across different units:

- **Symbol** β:
  - **Unit 2**: Rate (or inverse-scale) parameter in Exponential and Gamma distributions.
  - **Unit 4**: Inverse temperature parameter ($\beta = 1/k_B T$) in Ising and Potts models.
  - **Appendix B**: Vector of regression coefficients in Bayesian linear regression.
- **Symbol** α:
  - **Unit 2**: Shape parameter of the Gamma distribution.
  - **Unit 4**: Probability of accepting a proposed candidate state in Metropolis-Hastings.
  - **Unit 1**: Significance level (α) in statistical hypothesis testing.
- **Symbol** θ:
  - **Units 1-3, 5**: General model parameter vector under estimation or inference.
  - **Unit 5**: Parameter value being optimized in Robbins-Monro root-finding.

Course Unit Learning Objectives

0.0.0.1 Unit 1: Introduction to Research Methodology (07 Lecture + 05 SDL = 12 Hours)

- **Describe Research Meaning, Objectives, & Types (C1 — Remember)**: Define scientific research and classify studies across descriptive, analytical, exploratory, hypothesis-testing, basic, applied, quantitative, and qualitative types.
- **Distinguish Research Methods vs. Research Methodology (C2 — Understand)**: Explain the fundamental distinction between data collection tools (*methods*) and the logical justification of the study design (*methodology*).
- **Examine the 11 Steps in the Research Process (C4 — Analysis)**: Deconstruct the 11 sequential, interdependent stages of scientific inquiry from problem formulation to report dissemination.
- **Recognize Research Ethics & Academic Integrity (C1 — Remember)**: Apply Belmont Report principles (Autonomy, Beneficence, Justice), navigate Institutional Review Board (IRB) reviews, and avoid scientific misconduct.
- **Master Literature Search & Referencing (C4 — Analysis)**: Construct Boolean search queries, create structured literature matrices, and utilize APA 7th, Vancouver, and BibTeX citation styles.
- **Evaluate Scientific Formats & Reporting Guidelines (C2 — Understand)**: Contrast SPIRIT Research Protocols, 5 Academic Thesis Archetypes, and IMRaD Journal Manuscripts with **CONSORT 2010**, **STROBE**, and **PRISMA 2020** checklists.
- **Discuss & Select Research Designs (C2 — Understand)**: Compare observational designs (Cross-Sectional, Case-Control, Cohort) and experimental designs (RCTs, RBD, Latin Square, Factorial).
- **Calculate Sampling Designs & Sample Sizes (C3 — Apply)**: Compute probability sample sizes (n), derive Stratified Neyman and Cost-Constrained Optimum Allocations, and calculate Design Effect ($DEFF$).

0.0.0.2 Unit 2: Introduction to Simulation (05 Lecture + 02 SDL + 14 Practicals = 21 Hours)

- **Outline Foundations of Simulation (C1 — Remember)**: Explain pseudo-random numbers, random variables, and random vectors within a formal computational framework.
- **Derive & Implement Simulation Algorithms (C3 — Apply)**: Construct Linear Congruential Generators (LCG), Inverse-Transform, Accept-Reject, Box-Muller, and Cholesky vector algorithms step by step.
- **Simulate Stochastic Processes (C3 — Apply)**: Generate Homogeneous & Non-Homogeneous Poisson Processes, Markov Chains, and Jump Processes.
- **Generate Random Permutations (C3 — Apply)**: Implement the Fisher-Yates shuffle algorithm to generate unbiased random permutations.
- **Analyze Empirical Distributions & Influence Functions (C5 — Evaluate)**: Prove the Glivenko-Cantelli theorem and evaluate statistical influence functions for robust estimators.

0.0.0.3 Unit 3: Monte Carlo Methods (10 Lecture + 05 SDL + 10 Practicals = 25 Hours)

- **Apply Monte Carlo Integration (C3 — Apply)**: Estimate complex high-dimensional integrals using random sampling.
- **Master Variance Reduction Techniques (C6 — Create)**: Derive and apply Control Variates, Conditional Monte Carlo (Rao-Blackwellization), Stratified Sampling, and Multilevel Monte Carlo (MLMC).
- **Implement Importance Sampling & Particle Filters (C3 — Apply)**: Derive optimal proposal density $g^*(x)$, Sequential Importance Sampling (SIS), and Resampling (SIR) for Hidden Markov Models.
- **Derive Score Function Gradients (C4 — Analyze)**: Utilize the log-derivative trick to compute expectation gradients.
- **Execute Resampling Methods (C3 — Apply)**: Perform Bootstrap, Jackknife, Cross-Validation, and Empirical Likelihood estimation.

0.0.0.4 Unit 4: Markov Chain Monte Carlo (05 Lecture + 05 SDL + 10 Practicals = 20 Hours)

- **Formulate MCMC Concepts (C3 — Apply)**: Construct ergodic Markov chains targeting unnormalized joint posterior distributions.
- **Derive Metropolis-Hastings & Gibbs Samplers (C5 — Create)**: Prove detailed balance, candidate acceptance ratios, and full conditional update dynamics.
- **Simulate Statistical Physics Models (C3 — Apply)**: Implement 2D Ising and Potts spin lattice models.
- **Execute Global Optimization & Exact Sampling (C5 — Create)**: Implement Simulated Annealing cooling schedules and Propp-Wilson Coupling From The Past (CFTP).
- **Apply Bayesian Inference & Kernel Density Estimation (C6 — Evaluate)**: Update posterior distributions and construct adaptive non-parametric KDE estimators.

0.0.0.5 Unit 5: Monte Carlo Optimization (03 Lecture + 03 SDL + 06 Practicals = 12 Hours)

- **Compare Optimization Frameworks (C4 — Analyze)**: Contrast deterministic and stochastic approximation algorithms.
- **Derive Robbins-Monro Stochastic Approximation (C6 — Create)**: Prove step-size convergence conditions for stochastic root-finding.
- **Execute Multilevel Splitting for Rare Events (C6 — Create)**: Implement fixed-factor and fixed-effort particle splitting algorithms.
- **Implement Adaptive Metropolis MCMC (C6 — Create)**: Construct Haario adaptive proposal schemes with diminishing adaptation limits.

Prerequisites & Setup

To run the interactive code blocks and exercises, source our shared helpers script in your R console:

```
source("helpers.R")
```

Chapter 1

Introduction to Research Methodology

1.1 Meaning, Objectives, and Types of Research

1.1.1 Definition and Nature of Scientific Research

Scientific research is defined as a **systematic, controlled, empirical, and critical investigation** of hypothetical propositions regarding presumed relationships among natural and social phenomena (Kothari and Garg (2019)).

Scientific research is distinguished from casual observation by four fundamental characteristics:

1. **Empiricism**: Conclusions are strictly grounded in observable, verifiable evidence.
2. **Objectivity**: The inquiry is designed to eliminate investigator bias, subjective preferences, and unvalidated assumptions.
3. **Reproducibility**: Methodology is documented with sufficient precision to permit exact independent replication.
4. **Systematic Structure**: Procedures follow a logical, pre-specified sequence.

1.1.2 Core Objectives of Research

According to Kothari and Garg (2019), the primary purpose of research is to discover answers to questions through the systematic application of scientific procedures. Research objectives are broadly categorized into four types:

1. **Exploratory / Formulative Research**: Conducted to gain familiarity with a phenomenon or achieve new insights to formulate a precise research problem or hypothesis (e.g. qualitative focus groups exploring patient perceptions of digital health platforms).
2. **Descriptive Research**: Aimed at accurately portraying the characteristics of a particular individual, situation, or population group without manipulating variables (e.g. mapping the demographic distribution and prevalence of hypertension across a municipality).
3. **Diagnostic Research**: Focused on determining the frequency with which a phenomenon occurs or evaluating the strength of association between two or more variables (e.g. assessing clinical risk factors associated with 30-day hospital readmission).
4. **Hypothesis-Testing / Analytical Research**: Designed to test a formal statistical hypothesis regarding a causal relationship between specific exposure variables and outcomes (e.g. a double-blind randomized clinical trial evaluating whether a novel therapeutic agent reduces disease incidence).

1.1.3 Classification of Research Types

Research in data science, biostatistics, and public health is classified along multiple philosophical and structural dimensions (Creswell and Creswell (2018)):

Dimension	Research Type A	Research Type B	Conceptual Contrast & Empirical Example
Purpose	**Descriptive Research**	**Analytical Research**	Descriptive research establishes *what* exists (e.g. estimating that 25% of patients utilize telehealth). Analytical research evaluates *why or how* an outcome occurs (e.g. testing whether telehealth adoption reduces emergency department utilization).
Application	**Basic (Pure) Research**	**Applied Research**	Basic research seeks to expand foundational theoretical knowledge (e.g. proving convergence properties of an MCMC estimator). Applied research addresses an immediate practical problem (e.g. optimizing intensive care bed allocation algorithms).
Data Paradigm	**Quantitative Research**	**Qualitative Research**	Quantitative research relies on numerical measurements and inferential models (e.g. Cox regression of survival times). Qualitative research examines narrative themes and contextual experiences (e.g. semi-structured interviews on clinician burnout).
Environment	**Empirical (Field) Research**	**Conceptual Research**	Empirical research gathers observational or experimental field data. Conceptual research formulates abstract frameworks, mathematical definitions, or philosophical models.

Dimension	Research Type A	Research Type B	Conceptual Contrast & Empirical Example
Temporal Framework	**Cross-Sectional**	**Longitudinal**	Cross-Sectional research captures a single temporal snapshot. Longitudinal research tracks cohort subjects repeatedly across a designated follow-up period.

1.2 Research Methods vs. Research Methodology

A fundamental distinction in research methodology is the difference between *Research Methods* and *Research Methodology* (Kothari and Garg (2019)):

- **Research Methods**: Refers to the concrete behavior, instruments, computational scripts, and statistical procedures utilized to conduct research operations (e.g. administering a questionnaire, executing a t-test, or training a gradient boosted classifier).
- **Research Methodology**: Refers to the overarching **scientific philosophy, theoretical framework, and logical strategy** that justifies *why* specific methods were selected, why alternative approaches were rejected, and how the overall study design guarantees valid inference.

1.2.1 Architectural Analogy:

- **Research Methods** represent the physical construction tools (hammer, saw, level).
- **Research Methodology** represents the structural engineering blueprint detailing why specific materials and structural loads were selected to guarantee building integrity.

Dimension	Research Methods	Research Methodology
Definition	Specific tools, operations, and analytical techniques.	The systematic, scientific framework justifying the study design.
Scope	Narrow: targeted analytical execution (e.g. running an R function).	Broad: encompasses research design, sampling philosophy, ethics, and validity.
Primary Goal	To collect empirical data and compute numerical metrics.	To establish the validity, reliability, and appropriateness of the overall approach.
Relation	Methods are a constituent component of methodology.	Methodology is the overarching framework hosting the methods.

1.3 The 11 Sequential Steps in the Research Process

Scientific inquiry follows an 11-step sequential process. Each stage logically depends upon the completion and rigor of the preceding step (Kothari and Garg (2019)):

Phase	Sequential Steps	Core Focus & Rationale
Phase I: Conceptualization	**Step 1**: Formulate Research Problem**Step 2**: Review Literature**Step 3**: Develop Working Hypotheses	Define operational variables, identify knowledge gaps, and formulate testable H_0 vs H_1.
Phase II: Design & Sampling	**Step 4**: Prepare Research Design**Step 5**: Determine Sample Design**Step 6**: Collect Data	Select observational vs experimental framework, calculate sample size n, and execute data collection.
Phase III: Analysis & Reporting	**Step 7**: Execute Project Quality Control**Step 8**: Analyze Data**Step 9**: Test Statistical Hypotheses**Step 10**: Generalize & Interpret**Step 11**: Prepare Research Manuscript	Clean data, run inferential tests at $\alpha = 0.05$, evaluate validity, and write formal thesis/journal paper.

1.3.1 Logical Rationale for Sequential Progression:

1. **Formulating the Research Problem**: Define the precise research question, target population, exposures, and primary outcomes. *Rationale*: Study design cannot proceed without clear operational definitions of variables.
2. **Extensive Literature Review**: Synthesize existing peer-reviewed literature (Booth et al. (2016)). *Rationale*: Identifies the current state of knowledge and isolates the **research gap**, preventing redundant investigation.
3. **Developing Working Hypotheses**: Formulate explicit testable statistical propositions:
   - **Null Hypothesis (H_0)**: Statement of no difference, effect, or association.
   - **Alternative Hypothesis (H_1)**: Statement of expected directional or non-directional effect.
4. **Preparing Research Design**: Establish the overall structural framework (Observational vs. Experimental).
5. **Determining Sample Design**: Establish the target population, sampling frame, sampling method, and required sample size (n).
6. **Executing Data Collection**: Administer data collection instruments, extract electronic health records, or stream sensor metrics.
7. **Execution of the Project**: Implement quality control, conduct pilot testing, and execute non-response management.
8. **Analysis of Data**: Clean, transform, and structure data to compute descriptive and inferential statistics.
9. **Hypothesis Testing**: Apply appropriate inferential statistical tests (t-tests, Chi-square, ANOVA, Cox regression) to evaluate H_0 at significance level $\alpha = 0.05$.
10. **Generalizations and Interpretation**: Synthesize empirical findings, evaluate internal and external validity, and assess potential unmeasured confounding.
11. **Preparation of the Report / Manuscript**: Document methodology, results, limitations, and conclusions in standard academic formats (Protocol, Thesis, or Peer-Reviewed Journal Article).

1.4 Research Ethics and Scientific Dissemination

Research ethics governs the moral standards and rules of conduct required during scientific investigation (Creswell and Creswell (2018)).

1.4.1 The Belmont Report Principles

1. **Respect for Persons (Autonomy)**: Requires that individuals be treated as autonomous agents capable of self-determination. Participants must provide voluntary **Informed Consent** after being fully apprised of study procedures, risks, and benefits. Vulnerable populations require enhanced safeguards.

2. **Beneficence**: Obligations to maximize potential benefits to subjects and society while minimizing potential physical, psychological, or financial harms.
3. **Justice**: Mandates equitable selection of research subjects such that the burdens and benefits of research are distributed fairly across societal groups.

1.4.2 Institutional Review Board (IRB) Review Categories

Before initiating human subjects research, protocols must undergo IRB evaluation:

- **Exempt Review**: Research involving minimal risk and anonymized secondary datasets.
- **Expedited Review**: Research involving minimal risk and non-invasive routine procedures.
- **Full Board Review**: Research involving greater than minimal risk, novel interventions, or vulnerable subjects.

1.4.3 Scientific Misconduct and Academic Integrity

- **Fabrication**: Recording or reporting fictitious data or results.
- **Falsification**: Manipulating research materials, equipment, or processes, or omitting data such that the research is not accurately represented.
- **Plagiarism**: Appropriating another author's ideas, text, or results without proper attribution.

1.5 Literature Search, Referencing, and Citation Systems

1.5.1 Systematic Search Methodology and Literature Matrix

A literature search must be comprehensive and reproducible (Booth et al. (2016)):

- **Boolean Logic**: Utilize operators `AND` (intersection), `OR` (union), and `NOT` (exclusion).
- **Core Bibliographic Databases**: **PubMed / MEDLINE** (Biomedical), **IEEE Xplore** (Computer Science), **arXiv** (Preprints in ML/Statistics), **Scopus / Web of Science** (Multidisciplinary).

1.5.1.1 The Literature Synthesis Matrix:

A structured matrix utilized to compare prior empirical studies:

Author & Year	Study Design	Sample Size (n)	Primary Methodology	Key Empirical Findings	Identified Limitation / Gap
Smith et al. (2021)	Prospective Cohort	$n = 1,200$	Cox Proportional Hazards	aHR = 1.45 ($p < 0.01$)	Unadjusted missing EHR data
Patel et al. (2023)	Double-Blind RCT	$n = 350$	Intention-to-Treat ANOVA	Reduced recovery by 3.2 days	Single-center design

1.5.2 Citation Systems

- **APA 7th Edition**: Author-Date format widely utilized in social and data sciences. *In-text*: (Kothari & Garg, 2019).
- **Vancouver Style**: Numbered sequence standard in medical literature. *In-text*: Evidence indicates elevated readmission [1].

- **BibTeX**: Plain-text reference format utilized in LaTeX and R Markdown (`@article{...}`).

1.6 Formats: SPIRIT Protocol, Academic Thesis, and Journal Manuscript

Format	Primary Purpose	Timing relative to Data Collection	Key Structural Feature
Research Protocol	Formal pre-registration of study design & SAP	Written & registered **BEFORE** data collection	Contains **NO Results section**
Academic Thesis	Multi-chapter university degree monograph	Written **AFTER** project completion	Follows 5-chapter structure or cumulative paper format
Journal Manuscript	Peer-reviewed dissemination article	Written **AFTER** analysis completion	Follows **IMRaD** & reporting guidelines (**CONSORT/STROBE/PRISMA**)

1.6.1 Research Protocol (SPIRIT Statement Standard)

A research protocol is a formal prospective document prepared and registered **prior to data collection**.

- **Structural Feature**: A protocol contains **NO Results section** because data collection has not occurred.
- Components: Background, Rationale, Objectives, Inclusion/Exclusion criteria, Sample size power calculation, Statistical Analysis Plan (SAP), and IRB ethics approval.

1.6.2 Academic Thesis Architecture (5 Archetypes)

An academic thesis is a multi-chapter university monograph documenting degree candidate research:

1. Traditional Master's Thesis: Standard 5-chapter monograph (Introduction, Literature Review, Methodology, Results, Discussion).
2. PhD Doctoral Dissertation: Advanced dissertation featuring novel mathematical proofs and simulation frameworks.
3. Article-Based Thesis: Monograph compiling published peer-reviewed manuscripts bound by an integrative synthesis.
4. Qualitative Thesis: Replaces quantitative tables with narrative themes, participant transcript quotes, and reflexivity.
5. Data Science Thesis: Focuses on algorithmic developments, Monte Carlo simulations, and open-source software packages.

1.6.3 Journal Manuscripts (IMRaD & Reporting Guidelines)

Peer-reviewed journal articles condense empirical studies into 8-12 page manuscripts structured according to **IMRaD**:

- **Introduction**: Problem formulation, background, gap, and hypotheses.
- **Methods**: Detailed protocol permitting exact independent replication.
- **Results**: Empirical findings presenting summary tables, figures, effect sizes, and p-values.
- **Discussion**: Contextual interpretation of findings, comparison with prior literature, and study limitations.

1.6.3.1 International Reporting Guidelines:

- **CONSORT 2010**: Reporting checklist for Randomized Controlled Trials (includes 4-phase participant flow diagram).
- **STROBE Statement**: Reporting checklist for Observational Studies (Cohort, Case-Control, Cross-Sectional).
- **PRISMA 2020**: Reporting checklist for Systematic Reviews and Meta-Analyses.

1.7 Research Design: Taxonomy, Validity, and Case Studies

Research design is the overarching blueprint for empirical data collection and statistical analysis (Kothari and Garg (2019)).

1.7.1 Key Structural Elements

1. **Variables**:
   - **Dependent Variable (Y)**: Primary outcome measure under investigation.
   - **Independent Variable (X)**: Predictor or exposure variable manipulated or observed.
   - **Confounding Variable**: An unmeasured or unadjusted extraneous factor associated with both exposure X and outcome Y.
2. **Experimental Controls**:
   - **Experimental Group**: Subjects receiving the active intervention.
   - **Control Group**: Subjects receiving placebo or standard-of-care baseline comparison.
3. **Blinding Protocols**:
   - **Single-Blind**: Subjects are unaware of arm assignment.
   - **Double-Blind**: Neither subjects nor outcome assessors are aware of arm assignment (eliminates expectations bias).

1.7.2 Research Design Taxonomy

Design Type	Temporal Orientation	Primary Outcome Metric	Major Advantage	Major Limitation	Best Application
Cross-Sectional	Single point in time	Point Prevalence	Rapid execution, cost-effective	Cannot establish temporal causality	Population prevalence surveys
Case-Control	Retrospective	Odds Ratio ($OR = \frac{ad}{bc}$)	Efficient for rare outcomes	Vulnerable to recall bias	Investigating rare clinical conditions
Prospective Cohort	Longitudinal / Prospective	Relative Risk (RR), Hazard Ratio (HR)	Establishes temporal sequence	High cost, attrition bias	Tracking risk factor incidence
Randomized Trial (RCT)	Experimental / Prospective	Risk Ratio, Mean Difference	Gold standard for causal inference	High cost, strict inclusion limits	Evaluating therapeutic efficacy

---

1.7.3 Experimental Design Classifications

- **Completely Randomized Design (CRD)**: Treatments assigned completely at random across homogeneous units.
- **Randomized Block Design (RBD)**: Units grouped into homogeneous blocks based on a nuisance variable (e.g. age strata) prior to randomization.
- **Latin Square Design (LSD)**: Simultaneously controls for two nuisance sources of variation using a square matrix layout.

1.7.4 Threats to Validity

- **Internal Validity**: The degree to which a study cleanly isolates a true causal relationship without confounding (threatened by selection bias, maturation, attrition, and instrumentation changes).
- **External Validity**: The degree to which empirical findings generalize to broader target populations (threatened by restrictive eligibility criteria and Hawthorne effects).

1.8 Sampling Design and Sample Size Determination

Sampling design is a definite, systematic plan for obtaining a representative sample from a target population (Kothari and Garg (2019)).

1.8.1 Core Sampling Terminology

- **Target Population**: The complete collection of units about which theoretical inferences are drawn.
- **Sampling Frame**: The physical register or list of population units available for selection.
- **Sampling Error**: The random statistical variation resulting from estimating a population parameter using a sample rather than a census ($SE(\bar{x}) \propto 1/\sqrt{n}$).

1.8.2 Probability Sampling Methods & Mathematical Derivations

In probability sampling, every unit i in population N possesses a known, non-zero inclusion probability $\pi_i > 0$.

1.8.2.1 Simple Random Sampling (SRS)

Every subset of n units has equal selection probability $P = 1/N$.

- **Proof of Sample Mean Unbiasedness**: $E[\bar{y}] = \bar{Y}$.
- **Variance of Estimator (SRSWOR)**: $Var(\bar{y}) = \frac{S^2}{n}(1-f)$, where $f = n/N$ is the sampling fraction and $(1-f)$ is the Finite Population Correction (FPC).

1.8.2.2 Stratified Random Sampling & Optimum Allocations

Divide population N into non-overlapping strata $N_1, \ldots, N_K$. Draw an SRS from each stratum.

- **Proportional Allocation**: $n_h = n\left(\frac{N_h}{N}\right)$.

- **Neyman Optimum Allocation (Variance Minimization)**:

$$n_h = n \cdot \frac{N_h S_h}{\sum_{k=1}^{K} N_k S_k}$$

- **Cost-Constrained Optimum Allocation (Minimizing Variance under Budget $C = c_0 + \sum c_h n_h$)**:

$$n_h = n \cdot \frac{\frac{N_h S_h}{\sqrt{c_h}}}{\sum_{k=1}^{K} \frac{N_k S_k}{\sqrt{c_k}}}$$

1.8.2.2.1 Worked Numerical Problem 1: Cost-Constrained Allocation A survey samples $n = 500$ subjects across 2 geographic strata:

- Stratum 1: Population $N_1 = 10,000$, standard deviation $S_1 = 20$, cost per unit $c_1 = \$4$.
- Stratum 2: Population $N_2 = 5,000$, standard deviation $S_2 = 30$, cost per unit $c_2 = \$9$.

Derivation & Solution:

1. Evaluate $\frac{N_h S_h}{\sqrt{c_h}}$:
   - Stratum 1: $\frac{10,000 \times 20}{\sqrt{4}} = \frac{200,000}{2} = 100,000$.
   - Stratum 2: $\frac{5,000 \times 30}{\sqrt{9}} = \frac{150,000}{3} = 50,000$.
   - Sum $\sum \frac{N_k S_k}{\sqrt{c_k}} = 100,000 + 50,000 = 150,000$.
2. Compute optimum stratum sample sizes n_h:
   - $n_1 = 500 \times \frac{100,000}{150,000} = \mathbf{333}$ units.
   - $n_2 = 500 \times \frac{50,000}{150,000} = \mathbf{167}$ units.

1.8.2.3 Systematic Sampling

Select every k-th element following a random start $r \in \{1, \ldots, k\}$, where $k = \lfloor N/n \rfloor$.

- *Advantage*: High operational efficiency in field settings.
- *Limitation*: Highly vulnerable to bias when population frames exhibit periodic cycles.

1.8.2.4 Cluster & Multi-Stage Sampling

Divide population into M primary clusters. Sample m clusters and survey all secondary units.

- **Design Effect ($DEFF$)**: Quantifies variance inflation due to intra-cluster correlation ρ:

$$DEFF = \frac{Var_{cluster}}{Var_{SRS}} = 1 + (\bar{M} - 1)\rho$$

  Higher intra-cluster correlation ($\rho > 0$) inflates variance, requiring sample size adjustment ($n_{eff} = n/DEFF$).

1.8.3 Non-Probability Sampling Methods

Non-random selection strategies useful for exploratory research, but **invalid for statistical population generalization**:

- **Convenience Sampling**: Selection based on accessibility.
- **Purposive Sampling**: Expert judgemental selection.
- **Quota Sampling**: Non-random demographic quota filling.
- **Snowball Sampling**: Chain-referral sampling utilized for hard-to-reach hidden populations.

1.8.4 Sample Size Determination Formulas

1.8.4.1 Estimating a Single Population Proportion p:

$$n = \frac{Z_{1-\alpha/2}^2 \cdot p(1-p)}{d^2}$$

1.8.4.2 Estimating a Single Population Mean μ:

$$n = \frac{Z_{1-\alpha/2}^2 \cdot \sigma^2}{d^2}$$

1.8.4.2.1 Worked Numerical Problem 2: Sample Size Calculation Estimate mean fasting blood glucose within a margin of error $d = 3$ mg/dL at 95% confidence ($Z = 1.96$), given population standard deviation $\sigma = 25$ mg/dL.

Solution:

$$n = \frac{(1.96)^2 \times 25^2}{3^2} = \frac{3.8416 \times 625}{9} = \frac{2401}{9} = 266.77 \implies \mathbf{267} \text{ subjects}$$

1.8.5 Unit 1 Key Takeaways

- **Methods vs. Methodology**: Methods are concrete operations; methodology is the theoretical justification of validity.
- **11-Step Process**: Sequential framework from problem formulation through literature search to report preparation.
- **Reporting Standards**: SPIRIT for protocols, CONSORT for RCTs, STROBE for observational studies, PRISMA for reviews.
- **Research Design Taxonomy**: Observational (Cross-sectional, Case-Control OR, Cohort HR) vs Experimental (RCTs, RBD, LSD).
- **Sampling Design**: SRS, Stratified (Neyman & Cost-constrained allocation), Systematic, Cluster ($DEFF = 1 + (\bar{M} - 1)\rho$).

Chapter 2

Introduction to Simulation

2.1 Generation of Pseudo-Random Numbers: LCG & Hull-Dobell Theorem

Deterministic computers generate sequences of **pseudo-random numbers** using mathematical recurrence relations designed to satisfy statistical tests of randomness.

2.1.1 Linear Congruential Generator (LCG)

The LCG generates a sequence of pseudo-random integers $\{X_n\}$ via:

$$X_{n+1} = (aX_n + c) \bmod m$$

- m: Modulus ($m > 0$).
- a: Multiplier ($0 < a < m$).
- c: Increment ($0 \leq c < m$).
- X_0: Initial seed ($0 \leq X_0 < m$).

Continuous standard uniform variates $U_n \in [0, 1)$ are obtained by normalization:

$$U_n = \frac{X_n}{m}$$

2.1.1.1 Iterative Calculation Example:

Consider an LCG with parameters $m = 16, a = 5, c = 3$, and initial seed $X_0 = 1$:

1. **Iteration 1**:
$$X_1 = (5 \times 1 + 3) \bmod 16 = 8 \implies U_1 = \frac{8}{16} = 0.5000$$

2. **Iteration 2**:
$$X_2 = (5 \times 8 + 3) \bmod 16 = 43 \bmod 16 = 11 \implies U_2 = \frac{11}{16} = 0.6875$$

3. **Iteration 3**:
$$X_3 = (5 \times 11 + 3) \bmod 16 = 58 \bmod 16 = 10 \implies U_3 = \frac{10}{16} = 0.6250$$

2.1.2 Full-Period Conditions: The Hull-Dobell Theorem

An LCG achieves a **full period** m (visiting all integers $0, 1, \ldots, m-1$ before repeating) if and only if:

1. c and m are relatively prime ($\gcd(c, m) = 1$).
2. Every prime factor p dividing m also divides $a - 1$ ($a \equiv 1 \pmod{p}$).
3. If 4 divides m, then 4 must also divide $a - 1$ ($a \equiv 1 \pmod{4}$).

2.2 Generation of Continuous Random Variables

2.2.1 The Inverse-Transform Method

2.2.1.1 Mathematical Formulation:

Let $U \sim \text{Unif}(0, 1)$, and let $F(x)$ be a strictly increasing continuous Cumulative Distribution Function (CDF). Define $X = F^{-1}(U)$.

$$\text{Standard Uniform } U \sim \text{Unif}(0,1) \implies \text{Inverse CDF Evaluation } F^{-1}(U) \implies \text{Target Sample } X$$

2.2.1.2 Step-by-Step Mathematical Proof:

We evaluate the CDF of $X = F^{-1}(U)$:

- **Step 1 (Definition of CDF)**:
$$P(X \leq x) = P(F^{-1}(U) \leq x)$$
- **Step 2 (Monotonic Transformation)**:
Rationale: Because $F(x)$ is a strictly increasing function, applying F to both sides of an inequality preserves the inequality direction:
$$P(F(F^{-1}(U)) \leq F(x)) = P(U \leq F(x))$$
- **Step 3 (Uniform Distribution Property)**:
Rationale: For any standard uniform random variable $U \sim \text{Unif}(0, 1)$, $P(U \leq u) = u$ for all $u \in [0, 1]$. Setting $u = F(x)$:
$$P(U \leq F(x)) = F(x)$$
- **Conclusion**: $P(X \leq x) = F(x)$, proving that $X = F^{-1}(U)$ follows CDF $F(x)$. ■

2.2.1.3 Application: Generating Exponential Variates ($X \sim \textbf{Exp}(\lambda)$)

1. Target CDF: $F(x) = 1 - e^{-\lambda x}$ for $x \geq 0$.
2. Set $U = 1 - e^{-\lambda X}$.
3. Solve for X:
$$1 - U = e^{-\lambda X} \implies \ln(1 - U) = -\lambda X \implies X = -\frac{\ln(1 - U)}{\lambda}$$
4. Since $1 - U \sim \text{Unif}(0, 1)$, this simplifies to:
$$\mathbf{X} = -\frac{\ln(\mathbf{U})}{}$$

2.2.2 The Accept-Reject Algorithm

2.2.2.1 Mathematical Formulation:

When $F^{-1}(u)$ lacks a closed-form expression, we sample candidate variates Y from a proposal density $g(y)$ that bounds target density $f(y)$ under an envelope $c \cdot g(y)$ ($c \geq 1$, $\forall y : f(y) \leq c \cdot g(y)$).

$$\text{Target Density } f(y) \leq \text{Envelope Density } c \cdot g(y) \quad \forall y$$

2.2.2.2 Algorithm Steps:

1. Draw candidate $Y \sim g(y)$ and independent uniform variate $U \sim \text{Unif}(0, 1)$.
2. Accept $X = Y$ if $U \leq \frac{f(Y)}{c \cdot g(Y)}$; otherwise reject and repeat.

2.2.2.3 Proof of Correctness:

- **Step 1 (Joint Probability of Candidate & Acceptance)**:

$$P(Y \leq x, \text{Accept}) = \int_{-\infty}^{x} \left[\int_{0}^{\frac{f(y)}{cg(y)}} 1 \, du \right] g(y) \, dy = \int_{-\infty}^{x} \frac{f(y)}{cg(y)} g(y) \, dy = \frac{1}{c} \int_{-\infty}^{x} f(y) \, dy = \frac{F(x)}{c}$$

- **Step 2 (Overall Acceptance Probability)**:
Rationale: Setting $x = \infty$ evaluates the marginal probability of acceptance per iteration:

$$P(\text{Accept}) = \frac{1}{c} \int_{-\infty}^{\infty} f(y) \, dy = \frac{1}{c}$$

- **Step 3 (Conditional Distribution)**:
Rationale: Applying Bayes' rule for conditional probability $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$:

$$P(Y \leq x \mid \text{Accept}) = \frac{P(Y \leq x, \text{Accept})}{P(\text{Accept})} = \frac{F(x)/c}{1/c} = F(x)$$

- **Conclusion**: Accepted variates follow target density $f(x)$ with an expected number of draws per accepted sample equal to c. ■

2.2.3 Generating Normal Vectors: Cholesky Decomposition

To generate a correlated multivariate Gaussian vector $X \sim \mathcal{N}_n(\mu, \Sigma)$:

1. Compute the **Cholesky Factorization** of covariance matrix Σ:

$$\Sigma = LL^T \quad \text{(where } L \text{ is lower-triangular)}$$

2. Draw independent standard normal vector $Z = (Z_1, \ldots, Z_n)^T \sim \mathcal{N}_n(0, I_n)$.
3. Compute $X = \mu + LZ$.

2.2.3.1 Proof of Covariance Structure:

$$\text{Cov}(X) = \text{Cov}(\mu + LZ) = L\text{Cov}(Z)L^T = LI_nL^T = LL^T = \Sigma \quad ■$$

2.3 Simulation of Stochastic Processes

2.3.1 Homogeneous Poisson Process (HPP)

In an HPP with constant rate λ, inter-arrival times between consecutive events are independent exponential variates $T_k \sim \text{Exp}(\lambda)$.

- **Algorithm**:
  Set $S_0 = 0$. For $k = 1, 2, ...$: draw $U_k \sim \text{Unif}(0, 1)$, compute $T_k = -\frac{\ln U_k}{\lambda}$, and set event arrival time $S_k = S_{k-1} + T_k$.

2.3.2 Non-Homogeneous Poisson Process (NHPP): Thinning Method

When arrival intensity $\lambda(t)$ varies over time but is bounded by $\lambda_{\max}$:

1. Generate candidate arrival times S_k from an HPP with constant rate $\lambda_{\max}$.
2. Accept candidate S_k with probability $\frac{\lambda(S_k)}{\lambda_{\max}}$ (draw $U \sim \text{Unif}(0, 1)$, accept if $U \leq \frac{\lambda(S_k)}{\lambda_{\max}}$).

2.4 Random Permutations: Fisher-Yates Shuffle

To generate an unbiased random permutation of an array of n elements such that all $n!$ permutations are equiprobable:

- **Algorithm**:
  For $i = n, n-1, \dots, 2$:

  1. Draw a random integer j uniformly from $\{1, 2, \dots, i\}$.

  2. Swap elements at index i and index j.

2.5 Empirical Distribution Function (EDF) & Influence Functions

2.5.1 Empirical Distribution Function (EDF)

Given an observed sample $X_1, \dots, X_n$, the EDF $F_n(x)$ represents the empirical step CDF:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(X_i \leq x)$$

2.5.1.1 Glivenko-Cantelli Theorem:

As sample size $n \to \infty$, $F_n(x)$ converges uniformly to the true underlying CDF $F(x)$ almost surely:

$$\sup_{x \in \mathbb{R}} |F_n(x) - F(x)| \xrightarrow{a.s.} 0$$

2.5.2 Influence Function (IF)

The Influence Function quantifies the sensitivity of a functional estimator $T(F)$ to an infinitesimal point-mass contamination at x:

$$IF(x; T, F) = \lim_{\varepsilon \to 0} \frac{T((1-\varepsilon)F + \varepsilon\delta_x) - T(F)}{\varepsilon}$$

For the sample mean, $IF(x; \text{Mean}, F) = x - \mu$ (demonstrating linear, unbounded sensitivity to extreme outliers).

Chapter 3

Monte Carlo Methods

3.1 Variance Reduction Techniques

3.1.1 Control Variates

Let X be our target estimator of $\ell = E[X]$. Suppose we have another random variable Y with a known expectation $\mu_Y = E[Y]$. The control variate estimator is defined as:

$$X_c = X + c(Y - \mu_Y)$$

Suppose you are estimating the average temperature of a room using a cheap sensor X. If you also have a high-precision thermometer Y that fluctuates in sync with X (correlated) and whose average reading μ_Y is known, you can correct your reading: if Y is running higher than its known average ($Y - \mu_Y > 0$), you subtract a proportional correction from X. This "control" adjusts the reading to significantly reduce measurement noise (variance). where c is a constant. The optimal coefficient c^* that minimizes $\text{Var}(X_c)$ is:

$$c^* = -\frac{\text{Cov}(X, Y)}{\text{Var}(Y)}$$

Substituting c^* back into the variance formula:

$$\text{Var}(X_c^*) = \text{Var}(X)\,(1 - \rho_{XY}^2)$$

where ρ_{XY} is the correlation coefficient.

- **Step-by-Step Proof**:
  1. **Expand the Variance**: Using properties of variance for a sum of random variables:

     $$\text{Var}(X_c) = \text{Var}(X + c(Y - \mu_Y)) = \text{Var}(X) + 2c\text{Cov}(X, Y - \mu_Y) + c^2\text{Var}(Y - \mu_Y)$$

     Since μ_Y is a constant, shifting by it does not affect variance or covariance:

     $$\text{Var}(X_c) = \text{Var}(X) + 2c\text{Cov}(X, Y) + c^2\text{Var}(Y)$$

  2. **Minimize with Respect to** c: To find the value of c that minimizes the variance, we differentiate $\text{Var}(X_c)$ with respect to c and set the derivative to zero:

     $$\frac{d}{dc}\text{Var}(X_c) = 2\text{Cov}(X, Y) + 2c\text{Var}(Y) = 0$$

     Solving for c yields the optimal coefficient:

     $$c^* = -\frac{\text{Cov}(X, Y)}{\text{Var}(Y)}$$

3. **Substitute c^* to Find Minimum Variance**: Now we plug c^* back into the variance expression:

$$\text{Var}(X_c^*) = \text{Var}(X) + 2\left(-\frac{\text{Cov}(X,Y)}{\text{Var}(Y)}\right)\text{Cov}(X,Y) + \left(-\frac{\text{Cov}(X,Y)}{\text{Var}(Y)}\right)^2 \text{Var}(Y)$$

$$\text{Var}(X_c^*) = \text{Var}(X) - 2\frac{\text{Cov}(X,Y)^2}{\text{Var}(Y)} + \frac{\text{Cov}(X,Y)^2}{\text{Var}(Y)} = \text{Var}(X) - \frac{\text{Cov}(X,Y)^2}{\text{Var}(Y)}$$

4. **Express in Terms of Correlation Coefficient (ρ_{XY})**: By definition, the correlation coefficient is $\rho_{XY} = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$, which means $\text{Cov}(X,Y)^2 = \rho_{XY}^2 \text{Var}(X)\text{Var}(Y)$. Substituting this:

$$\text{Var}(X_c^*) = \text{Var}(X) - \frac{\rho_{XY}^2 \text{Var}(X)\text{Var}(Y)}{\text{Var}(Y)} = \text{Var}(X)\left(1 - \rho_{XY}^2\right)$$

Since $\rho_{XY}^2 \geq 0$, we have $\text{Var}(X_c^*) \leq \text{Var}(X)$, guaranteeing that the variance is reduced or remains unchanged. ■

3.1.1.1 Code Verification

We estimate $\ell = E[e^U]$ where $U \sim \text{Unif}(0,1)$ using Control Variates with $Y = U$. The optimal control coefficient is $c^* = -\frac{(3-e)/2}{1/12} \approx -1.6903$. The exact analytical variance reduction ratio is:

$$\text{Ratio}_{\text{theory}} = \frac{1}{1-\rho_{XY}^2} \approx 61.4269$$

```r
u_samples <- runif(10000)
x_est <- exp(u_samples)
y_est <- u_samples
c_opt <- -(3 - exp(1)) / (2 * (1/12)) # Analytical optimal c

x_c <- x_est + c_opt * (y_est - 0.5)
emp_reduction <- var(x_est) / var(x_c)
```

The empirical variance reduction ratio is **61.2169** (theoretical: 61.4269).

3.1.2 Antithetic Variates

If $U_i \sim \text{Unif}(0,1)$ are used to generate random variables $X_1 = h(U_1)$, we can generate $X_2 = h(1-U_1)$. If $\text{Cov}(X_1, X_2) < 0$, then $\text{Var}(\bar{X}) < \frac{\text{Var}(X_1)}{2}$.

Suppose you are estimating the average height of a hill using random steps. If you take a step far to the right, you are likely on high ground; if your next step is deliberately taken at the mirror-opposite direction far to the left, you are likely on low ground. Averaging these two "opposite" (negatively correlated) steps cancels out the extreme values, giving a much more stable estimate of the average height than two independent steps.

- **Step-by-Step Proof**:

1. **Variance of the Mean of Two Variables**: Let X_1, X_2 be identical in distribution, so $\text{Var}(X_1) = \text{Var}(X_2) = \sigma^2$. The variance of their sample mean is:

$$\text{Var}\left(\frac{X_1+X_2}{2}\right) = \frac{1}{4}\text{Var}(X_1+X_2) = \frac{1}{4}\left(\text{Var}(X_1) + \text{Var}(X_2) + 2\text{Cov}(X_1,X_2)\right)$$

$$\text{Var}\left(\frac{X_1+X_2}{2}\right) = \frac{1}{4}\left(2\sigma^2 + 2\text{Cov}(X_1,X_2)\right) = \frac{\sigma^2}{2} + \frac{\text{Cov}(X_1,X_2)}{2}$$

2. **Comparison with Independent Sampling**: If X_1, X_2 were sampled independently, their covariance would be 0, yielding a variance of $\frac{\sigma^2}{2}$. If we can design the coupling such that $\text{Cov}(X_1, X_2) < 0$:

$$\text{Var}\left(\frac{X_1+X_2}{2}\right) = \frac{\sigma^2}{2} + \frac{\text{Cov}(X_1,X_2)}{2} < \frac{\sigma^2}{2}$$

This proves that introducing a negative covariance between pairs yields a lower estimator variance than drawing independent samples. ■

3.1.2.1 Code Verification

We estimate $E[e^U]$ using 5000 antithetic pairs (10000 function evaluations total) and compare it to 10000 independent samples:

```
u1 <- runif(5000)
x_anti <- (exp(u1) + exp(1 - u1)) / 2
var_anti <- var(x_anti) / 5000 # Variance of mean of 5000 pairs

u2 <- runif(10000)
var_ind <- var(exp(u2)) / 10000
anti_ratio <- var_ind / var_anti
```

The variance reduction ratio achieved by Antithetic Variates is **30.5032**, showing a substantial variance reduction compared to independent sampling.

3.1.3 Conditional Monte Carlo (Rao-Blackwellization)

By the law of total variance, conditioning on a helper variable Y always reduces variance:

$$\text{Var}(E[X \mid Y]) = \text{Var}(X) - E[\text{Var}(X \mid Y)] \leq \text{Var}(X)$$

- **Step-by-Step Proof of the Law of Total Variance**:
  1. **Decompose the Variance**: By definition, the variance of X is:
  $$\text{Var}(X) = E[X^2] - (E[X])^2$$
  2. **Condition the Expectation inside the Squared Term**: Using the Law of Total Expectation, $E[X] = E[E[X \mid Y]]$. We can expand $E[X^2]$ by conditioning on Y:
  $$E[X^2] = E[E[X^2 \mid Y]]$$
  3. **Rewrite the Conditional Second Moment**: For any conditional random variable, the conditional variance is:
  $$\text{Var}(X \mid Y) = E[X^2 \mid Y] - (E[X \mid Y])^2 \implies E[X^2 \mid Y] = \text{Var}(X \mid Y) + (E[X \mid Y])^2$$
  4. **Substitute Back and Take Expectations**: Taking expectations with respect to Y:
  $$E[X^2] = E[E[X^2 \mid Y]] = E\big[\text{Var}(X \mid Y)\big] + E\big[(E[X \mid Y])^2\big]$$
  5. **Assemble the Variance of X**:
  $$\text{Var}(X) = E\big[\text{Var}(X \mid Y)\big] + E\big[(E[X \mid Y])^2\big] - (E[E[X \mid Y]])^2$$
  Notice that the last two terms are exactly the variance of the random variable $E[X \mid Y]$ (since $\text{Var}(Z) = E[Z^2] - (E[Z])^2$ where $Z = E[X \mid Y]$):
  $$\text{Var}(X) = E\big[\text{Var}(X \mid Y)\big] + \text{Var}(E[X \mid Y])$$
  Rearranging terms:
  $$\text{Var}(E[X \mid Y]) = \text{Var}(X) - E\big[\text{Var}(X \mid Y)\big]$$
  Since variance is always non-negative, the expectation of conditional variance is non-negative ($E[\text{Var}(X \mid Y)] \geq 0$). Thus:
  $$\text{Var}(E[X \mid Y]) \leq \text{Var}(X)$$
  This completes the proof. ■

3.1.3.1 Code Verification

We simulate $X \sim N(0,1)$ and $Y \mid X \sim N(\rho X, 1-\rho^2)$ with $\rho = 0.8$. We compare the empirical variance of X with the conditional expectation $E[X \mid Y] = \rho Y$.

```r
x_sim <- rnorm(10000)
y_sim <- 0.8 * x_sim + rnorm(10000, sd = sqrt(1 - 0.8^2))
cond_exp <- 0.8 * y_sim
var_x <- var(x_sim)
var_cond <- var(cond_exp)
cond_ratio <- var_x / var_cond
```

The variance ratio of the raw variable to the conditional expectation is **1.5638** (theoretical target: $1/\rho^2 = 1.5625$).

3.1.4 Stratified Sampling (Neyman Optimal Allocation)

For a fixed total sample size N, the allocation n_h that minimizes Stratified variance is the **Neyman Allocation**:

$$n_h = N \frac{W_h \sigma_h}{\sum_{k=1}^{L} W_k \sigma_k}$$

3.1.4.1 Code Verification

We estimate the mean of a population divided into two strata (weights $W_1 = 0.3, W_2 = 0.7$, and standard deviations $\sigma_1 = 1, \sigma_2 = 4$). We draw a total of $N = 1000$ samples.

```r
w1 <- 0.3; w2 <- 0.7
s1 <- 1; s2 <- 4
# Neyman allocation
n1 <- round(1000 * (w1 * s1) / (w1 * s1 + w2 * s2))
n2 <- 1000 - n1

# Draw stratified samples
y1 <- rnorm(n1, mean = 0, sd = s1)
y2 <- rnorm(n2, mean = 0, sd = s2)
var_strat <- (w1^2 * var(y1)/n1) + (w2^2 * var(y2)/n2)

# SRS samples
srs_samples <- rnorm(1000, mean = 0, sd = sqrt(w1*s1^2 + w2*s2^2))
var_srs <- var(srs_samples) / 1000
strat_deff <- var_srs / var_strat
```

The design effect (variance ratio of SRS to Stratified) is **1.3456**, confirming Stratified sampling achieves lower variance.

3.2 Multilevel Monte Carlo (MLMC)

By discretization on levels $l = 0, \ldots, L$, MLMC estimates the finest level P_L as a telescoping sum:

$$E[P_L] = E[P_0] + \sum_{l=1}^{L} E[P_l - P_{l-1}]$$

3.2.1 Code Verification

We estimate $E[X_T]$ for geometric Brownian motion $dX_t = 0.05X_t dt + 0.2X_t dW_t$ with $x_0 = 1, T = 1$ using the 2-level Euler-Maruyama discretization from `helpers.R`.

```
mlmc_data <- run_mlmc_2level(5000)
var_coarse <- var(mlmc_data$coarse)
var_difference <- var(mlmc_data$fine - mlmc_data$coarse)
```

The coarse path variance is **0.0402**, whereas the variance of the difference between fine and coarse paths is **0.0004**, demonstrating that the variance of the correction term is significantly smaller.

3.3 Importance Sampling & Sequential Methods

3.3.1 Importance Sampling

To estimate $\ell = \int h(x)f(x)\,dx$, we sample from proposal $g(x)$ and weight by $w(x) = f(x)/g(x)$.

Suppose you want to estimate the average depth of shipwrecks in the ocean, but they are concentrated in a few deep, rare trenches. If you drop random search lines across the entire ocean (standard Monte Carlo), you will almost never hit a trench, leading to high variance. Instead, you bias your search by targeting the known trench areas (sampling from a proposal $g(x)$) and correct for this deliberate bias by weighting down the discoveries in proportion to how much more likely you were to search there ($w(x) = f(x)/g(x)$). This focuses your sampling effort where it actually matters.

- **Step-by-Step Proof of the Optimal Proposal Density**:

1. **Expression for Variance of Importance Sampling Estimator**: Let $\hat{\ell}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N} \frac{h(X_i)f(X_i)}{g(X_i)}$ where $X_i \sim g$. The variance of the estimator is:

$$\text{Var}(\hat{\ell}_{\text{IS}}) = \frac{1}{N}\text{Var}_g\left(\frac{h(X)f(X)}{g(X)}\right) = \frac{1}{N}\left[E_g\left[\left(\frac{h(X)f(X)}{g(X)}\right)^2\right] - \ell^2\right]$$

2. **Minimize the Second Moment**: Since ℓ^2 is constant (independent of g), minimizing the variance is equivalent to minimizing the first term:

$$E_g\left[\left(\frac{h(X)f(X)}{g(X)}\right)^2\right] = \int \left(\frac{h(x)f(x)}{g(x)}\right)^2 g(x)\,dx = \int \frac{h(x)^2 f(x)^2}{g(x)}\,dx$$

3. **Apply Cauchy-Schwarz Inequality**: Recall that Cauchy-Schwarz states $\left(\int u(x)v(x)\,dx\right)^2 \leq \left(\int u(x)^2\,dx\right)\left(\int v(x)^2\,dx\right)$. Let $u(x) = \frac{|h(x)|f(x)}{\sqrt{g(x)}}$ and $v(x) = \sqrt{g(x)}$. Then:

$$\left(\int \frac{|h(x)|f(x)}{\sqrt{g(x)}}\sqrt{g(x)}\,dx\right)^2 \leq \left(\int \frac{h(x)^2 f(x)^2}{g(x)}\,dx\right)\left(\int g(x)\,dx\right)$$

Since $g(x)$ is a probability density, $\int g(x)\,dx = 1$. The left side simplifies:

$$\left(\int |h(x)|f(x)\,dx\right)^2 \leq \int \frac{h(x)^2 f(x)^2}{g(x)}\,dx$$

4. **Condition for Equality**: The lower bound is achieved (equality holds in Cauchy-Schwarz) if and only if $u(x)/v(x)$ is constant:

$$\frac{\frac{|h(x)|f(x)}{\sqrt{g(x)}}}{\sqrt{g(x)}} = c_{\text{const}} \implies \frac{|h(x)|f(x)}{g(x)} = c_{\text{const}}$$

$$g^*(x) = \frac{|h(x)|f(x)}{\int |h(y)|f(y)\,dy}$$

This proves that the optimal proposal density $g^*(x)$ is directly proportional to $|h(x)|f(x)$. ■

3.3.1.1 Code Verification

We estimate $\int_0^1 \cos(x)\,dx$ (analytical value: $\sin(1) \approx 0.8415$) using a linear proposal density $g(x) = \frac{4-2x}{3}$ on $[0, 1]$, which is a close match for the cosine function:

```
u <- runif(10000)
x_proposal <- 2 - sqrt(4 - 3 * u) # Inverse transform of g(x)
weight <- cos(x_proposal) / ((4 - 2 * x_proposal) / 3)
imp_mean <- mean(weight)
imp_var <- var(weight) / 10000
srs_var <- var(cos(runif(10000))) / 10000
imp_ratio <- srs_var / imp_var
```

The Importance Sampling estimate is **0.8414** (theoretical: 0.8415), and the variance reduction factor (ratio of SRS variance to IS variance) is **12.3293**, demonstrating a substantial variance reduction (approx. 12.5x) over standard random sampling.

3.3.2 Sequential Importance Sampling (SIS) Degeneracy

At step t, the weight update is:

$$w_t = w_{t-1} \frac{f(x_t \mid x_{1:t-1})}{q(x_t \mid x_{1:t-1})}$$

3.3.2.1 Code Verification

We run a 20-step SIS algorithm on a state space without resampling and print the weight Effective Sample Size (ESS) to illustrate weight collapse:

```
N <- 1000
w <- rep(1/N, N)
for (t in 1:20) {
  # Stochastically update weights representing degeneracy
  w <- w * runif(N, 0, 2)
  w <- w / sum(w)
}
sis_ess <- 1 / sum(w^2)
```

The final weight Effective Sample Size (ESS) after 20 steps is **11.43** (starting at 1000.00), demonstrating severe weight degeneracy.

3.3.3 Nonlinear Filtering (Particle Filtering)

We filter a state-space model using the Bootstrap particle filter from `helpers.R`.

3.3.3.1 Code Verification

We run the particle filter on a simulated 50-step trajectory:

```
# Generate true state and observations
x_true <- numeric(50)
y_obs <- numeric(50)
x_true[1] <- rnorm(1)
y_obs[1] <- x_true[1] + rnorm(1)
for (t in 2:50) {
  x_true[t] <- 0.5 * x_true[t-1] + rnorm(1)
```

```
  y_obs[t] <- x_true[t] + rnorm(1)
}

x_est <- run_particle_filter(y_obs, N = 500)
rmse_filter <- sqrt(mean((x_est - x_true)^2))
rmse_obs <- sqrt(mean((y_obs - x_true)^2))
```

The RMSE of the filtered state estimates is **0.7638**, which is lower than the raw observation noise RMSE of **1.0173**.

3.4 Transform Likelihood Ratio Methods

The sensitivity gradient is:

$$\nabla_\theta E_\theta[h(X)] = E_\theta[h(X)\nabla_\theta \log f(X;\theta)]$$

3.4.1 Code Verification

We estimate the derivative of $E_\theta[X^2]$ where $X \sim N(\theta, 1)$ at $\theta = 2$ (analytical derivative: $2\theta = 4.0000$) using the Likelihood Ratio estimator:

```
x_samples <- rnorm(10000, mean = 2, sd = 1)
# Score function for mean mu: (x - mu) / sigma^2
score_fun <- x_samples - 2
lr_grad_est <- mean(x_samples^2 * score_fun)
```

The Likelihood Ratio gradient estimate is **4.0828** (analytical value: 4.0000).

3.5 Resampling & Empirical Likelihood

3.5.1 Bootstrap

We estimate the standard error of the correlation coefficient for a small dataset.

3.5.1.1 Code Verification

We draw 100 samples with correlation 0.5 and run 1000 bootstrap replicates:

```
x <- rnorm(100)
y <- 0.5 * x + rnorm(100, sd = sqrt(1 - 0.5^2))
orig_cor <- cor(x, y)

boot_cor <- bootstrap_resample(1:100, function(idx) cor(x[idx], y[idx]), R = 1000)
boot_se <- sd(boot_cor)
```

The estimated bootstrap standard error of the correlation coefficient is **0.0871**.

3.5.2 Jackknife

We estimate the bias of the sample variance S^2 for a small sample size.

3.5.2.1 Code Verification

We draw 15 samples from a Normal distribution:

```
x_data <- rnorm(15)
n <- length(x_data)
jack_est <- numeric(n)
for (i in 1:n) {
  jack_est[i] <- var(x_data[-i])
}
jack_bias <- (n - 1) * (mean(jack_est) - var(x_data))
```

The estimated Jackknife bias of the sample variance is **0.0000** (theoretical expectation is near 0).

3.5.3 Cross-Validation

Cross-Validation is a resampling method used to assess how well a statistical model generalizes to independent, unseen data.

Suppose you are a student preparing for a final exam. If you study using only the mock exam questions, you might memorize the specific questions without truly understanding the concepts, leading to a poor grade on the actual exam (overfitting). To test your true preparation, you hide one chapter's mock questions, study the rest, and then test yourself on the hidden chapter. In statistical modeling, k-fold Cross-Validation partitions the dataset into k equal parts: the model is trained on $k-1$ folds and tested on the remaining fold. By cycling this process, we estimate how well the model generalizes to unseen data.

- **Practical Applications**: Used to select optimal model hyper-parameters (such as regularization parameters in Lasso/Ridge regression or number of trees in random forests) and prevent model overfitting in clinical predictive models.
- **Mathematical Formulation**: Let the dataset be partitioned into k non-overlapping folds $D_1, \ldots, D_k$. For each fold $j \in \{1, \ldots, k\}$, the model is trained on the dataset excluding fold j ($D \setminus D_j$) to obtain prediction function $\hat{f}^{(-j)}(\mathbf{x})$. The cross-validation estimate of the prediction error (e.g. Mean Squared Error) is:

$$\mathrm{CV}_k = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}^{(-j(i))}(\mathbf{x}_i) \right)^2$$

  where $j(i)$ is the index of the fold containing sample i, and n is the total sample size.
- **Manual Walkthrough (By-Hand Simulation)**: We trace k-fold cross-validation with $k = 3$ folds for a dataset of $n = 3$ points: $d_1 = (1, 2)$, $d_2 = (2, 4)$, and $d_3 = (3, 5)$. The model is a simple average predictor $\hat{f}(x) = \bar{y}$.
  - **Fold 1** ($D_1 = \{d_1\}$): Train on $D \setminus D_1 = \{d_2, d_3\} = \{(2, 4), (3, 5)\} \implies \hat{f}^{(-1)}(x) = \frac{4+5}{2} = 4.5$. Test on d_1: Squared error is $(y_1 - 4.5)^2 = (2 - 4.5)^2 = 6.25$.
  - **Fold 2** ($D_2 = \{d_2\}$): Train on $D \setminus D_2 = \{d_1, d_3\} = \{(1, 2), (3, 5)\} \implies \hat{f}^{(-2)}(x) = \frac{2+5}{2} = 3.5$. Test on d_2: Squared error is $(y_2 - 3.5)^2 = (4 - 3.5)^2 = 0.25$.
  - **Fold 3** ($D_3 = \{d_3\}$): Train on $D \setminus D_3 = \{d_1, d_2\} = \{(1, 2), (2, 4)\} \implies \hat{f}^{(-3)}(x) = \frac{2+4}{2} = 3.0$. Test on d_3: Squared error is $(y_3 - 3.0)^2 = (5 - 3.0)^2 = 4.0$.
  - **Estimated CV Mean Squared Error**:

$$\mathrm{CV}_3 = \frac{6.25 + 0.25 + 4.0}{3} \approx 3.50$$

3.5.3.1 Cross-Validation Code Verification

We simulate $n = 50$ samples from a linear model $Y = 2X + \epsilon$ with $\epsilon \sim N(0, 1)$ (noise variance is exactly 1.0). We run a custom 5-fold cross-validation in R to estimate the prediction MSE:

```r
# Generate simulated data
set.seed(12345)
x_cv <- runif(50, 0, 10)
y_cv <- 2 * x_cv + rnorm(50, mean = 0, sd = 1)
data_cv <- data.frame(x = x_cv, y = y_cv)

# 5-Fold Cross-Validation using helper function
cv_mse <- run_kfold_cv(data_cv, k = 5)
```

The estimated 5-fold Cross-Validation Mean Squared Error (MSE) is **1.5390** (theoretical expected prediction error is near 1.0, representing the irreducible noise variance $\sigma^2 = 1.0$).

3.5.4 Empirical Likelihood

We evaluate the Empirical Profile Likelihood ratio for the mean.

3.5.4.1 Code Verification

We evaluate the profile likelihood ratio statistic $-2 \log R(\theta)$ for the true mean of a normal sample:

```r
x_data <- rnorm(50, mean = 0, sd = 1)
el_stat <- calculate_empirical_likelihood_mean(x_data, mu_test = 0)
```

The empirical profile likelihood ratio statistic $-2 \log R(\theta_0)$ is **0.0034**, which is well below the critical value of $\chi^2_1(0.95) = 3.8415$ (confirming the hypothesis $H_0 : \mu = 0$ is accepted).

3.6 How to Report This in a Paper

When publishing Monte Carlo variance reduction studies:

1. Report the **variance reduction factor** (ratio of standard Monte Carlo variance to the reduced variance).
2. Report the **theoretical correlation** ρ_{XY} that justified the choice of the control variate.
3. For bootstrap confidence intervals, specify whether percentile, studentized, or BCa intervals were used.

Key Formulas Summary

Technique / Estimator	Formula / Definition	Key Constraints / Details
Control Variates	$X_c = X + c(Y - \mu_Y)$	$\mu_Y = E[Y]$ must be known; optimal $c^* = -\frac{\text{Cov}(X,Y)}{\text{Var}(Y)}$.
CV Variance Reduction	$\text{Var}(X_c^*) = \text{Var}(X)(1 - \rho_{XY}^2)$	Variance reduces by factor of correlation squared.
Antithetic Variates	$\bar{X}_{\text{anti}} = \frac{h(U)+h(1-U)}{2}$	Requires $\text{Cov}(h(U), h(1-U)) < 0$.

Technique / Estimator	Formula / Definition	Key Constraints / Details
Importance Sampling	$\widehat{\ell}_{\mathrm{IS}} = \frac{1}{N}\sum_{i=1}^{N} h(X_i)\frac{f(X_i)}{g(X_i)}$	Samples drawn from $g(x)$; weight function $w(x) = \frac{f(x)}{g(x)}$.
Optimal IS Proposal	$g^*(x) \propto \lvert h(x)\rvert f(x)$	Achieves zero variance if $h(x) \geq 0$.
Bootstrap Filter Update	$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(y_t \mid x_t^{(i)})$	Sequential MC update weight in state-space models.
Empirical Likelihood	$R(\theta) = \max\left\{\prod_{i=1}^{N} Np_i\right\}$	Constraints: $\sum p_i = 1$, $\sum p_i(x_i - \theta) = 0$.
Cross-Validation	$\mathrm{CV}_k = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{f}^{(-j(i))}(\mathbf{x}_i)\right)^2$	Partitions dataset into k folds to estimate generalization error.

Common Mistakes

Control Variates Sign Confusion: The formula for the control variates estimator can be written as $X_c = X + c(Y - \mu_Y)$ or $X_c = X - c(Y - \mu_Y)$.

- If you use the **addition** form ($X + c(\dots)$), then the optimal coefficient is $c^* = -\frac{\mathrm{Cov}(X,Y)}{\mathrm{Var}(Y)}$.
- If you use the **subtraction** form ($X - c(\dots)$), then the optimal coefficient is $c^* = \frac{\mathrm{Cov}(X,Y)}{\mathrm{Var}(Y)}$. Mixing these signs will flip the correction direction, adding noise and potentially doubling the variance instead of reducing it!

Antithetic Monotonicity: Antithetic sampling is only guaranteed to reduce variance if the transformation function $h(u)$ is monotonic in u. If $h(u)$ is non-monotonic (e.g. a symmetric quadratic function), $h(u)$ and $h(1-u)$ can be positively correlated, which actually increases the estimator's variance.

Fat-Tailed Importance Proposal: In Importance Sampling, the proposal distribution $g(x)$ must have "fatter tails" than the target $f(x)$. If $g(x)$ goes to zero faster than $f(x)$ in the tails, the weights $w(x) = f(x)/g(x)$ will blow up to infinity, rendering the estimator's variance infinite.

3.7 Exercises

3.7.1 Subsection 3.1: Variance Reduction Techniques

1. Derive the optimal control coefficient c^* when using two control variates Y_1, Y_2.
2. Implement a Stratified sampling program in R for a 3-strata population and compare it with Neyman allocation vs proportional allocation.

3.7.2 Subsection 3.2: Importance Sampling & HMMs

3. Prove that the optimal proposal density in Importance Sampling is proportional to $\lvert h(x)\rvert f(x)$.
4. Implement a bootstrap filter in R for a state-space model with non-linear transitions.

3.7.3 Subsection 3.3: Resampling & Likelihoods

5. Code a double-bootstrap algorithm in R to estimate the bias of a bootstrap standard error.
6. Compare the empirical likelihood confidence interval for a population median with the standard Student's t-interval.

Chapter 3 covered:

- **Monte Carlo integration**: Approximating $\int h(x)f(x)\,dx$ by the sample mean $\hat{\ell}_N = \frac{1}{N}\sum h(X_i)$, with standard error $\sigma_h/\sqrt{N}$
- **Control variates**: Using a correlated variable Y with known mean μ_Y to correct the estimate; optimal coefficient $c^* = -extCov(X,Y)/extVar(Y)$; variance reduction factor $(1 - ho^2)$
- **Antithetic variates**: Pairing U and $1-U$ to introduce negative correlation between estimates, halving the effective number of samples needed
- **Importance sampling**: Sampling from an alternative density $g(x)$ concentrated in high-impact regions; self-normalised estimator for unknown normalizing constants
- **Stratified sampling**: Partitioning the sample space and sampling proportionally to reduce between-stratum variance

Technique	Key Idea	When to Use
Control variates	Exploit a correlated known-mean variable	When a good correlated control is available
Antithetic variates	Pair U with $1-U$	Monotone integrands
Importance sampling	Sample from a better proposal	Rare events, heavy tails
Stratified sampling	Partition the domain	When variance varies across regions

Coming up: Chapter 4 goes beyond Monte Carlo integration to full Bayesian inference: Markov Chain Monte Carlo allows you to draw samples from arbitrary posterior distributions that cannot be directly simulated.

Exercises

Tier 1 — Conceptual

1. Explain in words why the standard error of a Monte Carlo estimate decreases at the rate $1/\sqrt{N}$, not $1/N$. What does this mean for doubling accuracy?
2. The importance sampling estimator uses weights $w(x) = f(x)/g(x)$. What happens to the estimator if you choose $g(x)$ to be very different from $f(x)$ in the tails?
3. Antithetic variates exploit the fact that U and $1-U$ are negatively correlated when passed through a *monotone* function h. Give an example of a non-monotone h for which antithetic variates would fail to reduce variance.

Tier 2 — Applied

4. Estimate $\int_0^2 \sqrt{1+x^4}\,dx$ using plain Monte Carlo with $N = 50{,}000$ samples. Report your estimate, standard error, and 95% confidence interval.
5. Repeat Question 4 using antithetic variates with $N/2 = 25{,}000$ antithetic pairs. Compare the standard errors.
6. Implement importance sampling to estimate $P(Z > 4)$ where $Z \sim N(0,1)$, using an exponential shifted proposal. Compare with the crude Monte Carlo estimate using $N = 10^5$ samples.

Tier 3 — Challenge

7. **Option Pricing**: The Black-Scholes price of a European call option can be computed as $C = e^{-rT}E[\max(S_T - K, 0)]$ where $S_T = S_0\exp((r - \sigma^2/2)T + \sigma\sqrt{T}Z)$ and $Z \sim N(0,1)$. With $S_0 = 100, K = 105, r = 0.05, \sigma = 0.20, T = 1$, estimate C using: (a) plain Monte Carlo, (b) control variates with S_T as the control, (c) antithetic variates. Use $N = 100{,}000$ and compare standard errors. The exact Black-Scholes price is 8.02.

Solutions to Tier 1 and 2: see Appendix B.

Chapter 4

Markov Chain Monte Carlo

4.1 Detailed Balance & Stationary Distributions

A Markov chain transition kernel $P(x, y)$ possesses invariant stationary distribution $\pi(x)$ if $\int \pi(x)P(x, y)dx = \pi(y)$.

4.1.1 Proof: Detailed Balance Implies Stationarity

Suppose $P(x, y)$ satisfies **Detailed Balance**: $\pi(x)P(x, y) = \pi(y)P(y, x)$.

- **Step 1 (Integrate over x)**:

$$\int \pi(x)P(x, y)dx = \int \pi(y)P(y, x)dx$$

- **Step 2 (Factor out $\pi(y)$)**:
Rationale: $\pi(y)$ is independent of integration variable x, permitting factorization outside the integral:

$$\int \pi(x)P(x, y)dx = \pi(y) \int P(y, x)dx$$

- **Step 3 (Apply Total Probability Property)**:
Rationale: Total probability property requires $\int P(y, x)dx = 1$. Therefore:

$$\int \pi(x)P(x, y)dx = \pi(y) \cdot 1 = \pi(y) \quad \blacksquare$$

4.2 Derivation of Metropolis-Hastings Acceptance Ratio $\alpha(x, y)$

Let proposal density be $q(y \mid x)$. Transition kernel is $P(x, y) = q(y \mid x)\alpha(x, y)$ for $x \neq y$.

- **Step 1 (Substitute into Detailed Balance)**:

$$\pi(x)q(y \mid x)\alpha(x, y) = \pi(y)q(x \mid y)\alpha(y, x)$$

- **Step 2 (Rearrange Ratio)**:

$$\frac{\alpha(x, y)}{\alpha(y, x)} = \frac{\pi(y)q(x \mid y)}{\pi(x)q(y \mid x)}$$

- **Step 3 (Peskun's Selection Rule)**:
  Setting $\alpha(y, x) = 1$ whenever the ratio is < 1 maximizes acceptance, establishing:

$$(\mathbf{x}, \mathbf{y}) = \min\left(\mathbf{1}, \frac{(\mathbf{y})\mathbf{q}(\mathbf{x} \mid \mathbf{y})}{(\mathbf{x})\mathbf{q}(\mathbf{y} \mid \mathbf{x})}\right) \quad \blacksquare$$

4.3 Derivation of Gibbs Sampler (Acceptance $\alpha = 1$)

The Gibbs Sampler updates coordinate i drawing from full conditional $\pi(x_i \mid x_{-i})$.

4.3.0.1 Proof of 100% Acceptance ($\alpha = 1$):

Candidate proposal is $q(y \mid x) = \pi(y_i \mid x_{-i})$.

- **Step 1 (Evaluate MH Ratio)**:

$$\frac{\pi(y)q(x \mid y)}{\pi(x)q(y \mid x)} = \frac{\pi(y_i, x_{-i}) \cdot \pi(x_i \mid x_{-i})}{\pi(x_i, x_{-i}) \cdot \pi(y_i \mid x_{-i})}$$

- **Step 2 (Apply Conditional Probability Identity)**:
  Rationale: Utilizing $\pi(y_i, x_{-i}) = \pi(y_i \mid x_{-i})\pi(x_{-i})$ and $\pi(x_i, x_{-i}) = \pi(x_i \mid x_{-i})\pi(x_{-i})$.
- **Step 3 (Simplification)**:

$$\frac{[\pi(y_i \mid x_{-i})\pi(x_{-i})] \cdot \pi(x_i \mid x_{-i})}{[\pi(x_i \mid x_{-i})\pi(x_{-i})] \cdot \pi(y_i \mid x_{-i})} = 1$$

- **Conclusion**: $\alpha = \min(1, 1) = \mathbf{1}$. Every Gibbs update is **accepted with probability 1**. $\blacksquare$

4.4 Statistical Physics Models & Exact Sampling

4.4.1 2D Ising Model

Simulates ferromagnetic spin lattices with state $S_i \in \{-1, +1\}$. Energy is $E(S) = -J \sum_{\langle i,j \rangle} S_i S_j$. Metropolis spin-flip energy change is $\Delta E = 2S_i \sum_{\text{neighbors}} S_j$. Accept flip if $U \leq \exp(-\Delta E/T)$.

4.4.2 Propp-Wilson Coupling From The Past (CFTP)

Generates exact draws from stationary distribution π without burn-in bias by running parallel chains initialized at time $-T$ in the past using shared random seed streams until all chains coalesce into a single state at time 0.

Chapter 5

Monte Carlo Optimization & Rare-Event Simulation

5.1 Sensitivity Analysis (Score Function Method)

For a parameterized expectation $\alpha(\theta) = E_\theta[h(X)] = \int h(x)f(x;\theta)\,dx$, we want to estimate the gradient $\nabla_\theta\alpha(\theta)$. Using the **Score Function / Likelihood Ratio Method**:

$$\nabla_\theta\alpha(\theta) = E_\theta\left[h(X)\nabla_\theta \log f(X;\theta)\right]$$

The term $S(X;\theta) = \nabla_\theta \log f(X;\theta)$ is the **score function**.

Suppose you want to know how the average height of a tree species $\alpha(\theta)$ changes as you adjust the soil moisture level θ. The naive way requires planting trees at many different levels of moisture and measuring the difference. The Score Function method is a mathematical shortcut: instead of changing the moisture and replanting, you measure the height of the current trees $h(X)$ and multiply it by a "score" $S(X;\theta)$ that tracks how sensitive each tree's likelihood of growth is to moisture. Averaging this product gives the exact gradient (sensitivity) without needing to run simulations at new parameters.

- **Step-by-Step Proof**:
  1. **Write as Integral**: By definition of the expectation:
  $$\alpha(\theta) = \int h(x)f(x;\theta)\,dx$$
  2. **Differentiate Under the Integral Sign**: We take the gradient with respect to θ on both sides:
  $$\nabla_\theta\alpha(\theta) = \nabla_\theta \int h(x)f(x;\theta)\,dx = \int h(x)\nabla_\theta f(x;\theta)\,dx$$
  3. **Divide and Multiply by the Density**: Assuming $f(x;\theta) > 0$ on the support, we multiply and divide the integrand by $f(x;\theta)$:
  $$\nabla_\theta\alpha(\theta) = \int h(x)\frac{\nabla_\theta f(x;\theta)}{f(x;\theta)}f(x;\theta)\,dx$$
  4. **Apply logarithmic derivative identity**: Since $\nabla_\theta \log f(x;\theta) = \frac{\nabla_\theta f(x;\theta)}{f(x;\theta)}$, we substitute:
  $$\nabla_\theta\alpha(\theta) = \int h(x)\nabla_\theta \log f(x;\theta)f(x;\theta)\,dx$$
  5. **Rewrite as Expectation**: This integral is exactly the expectation of $h(X)\nabla_\theta \log f(X;\theta)$ under $X \sim f(x;\theta)$:
  $$\nabla_\theta\alpha(\theta) = E_\theta\left[h(X)\nabla_\theta \log f(X;\theta)\right] \quad \blacksquare$$

- **Practical Applications**: Used in reinforcement learning (Policy Gradient methods like REINFORCE), finance (calculating "Greeks" or option price sensitivities), and engineering design optimization.
- **Manual Walkthrough (By-Hand Simulation)**: Let $X \sim N(\theta, 1)$. We want to estimate the gradient of $\alpha(\theta) = E_\theta[X^3]$ at $\theta = 1.0$. The analytical value is $\frac{d}{d\theta}E_\theta[X^3] = \frac{d}{d\theta}(\theta^3 + 3\theta) = 3\theta^2 + 3$, which at $\theta = 1.0$ is 6.0. The score function is:

$$S(x;\theta) = \frac{d}{d\theta}\log f(x;\theta) = \frac{d}{d\theta}\left[-\frac{1}{2}\log(2\pi) - \frac{(x-\theta)^2}{2}\right] = x - \theta$$

  Suppose we simulate a single sample $x = 2.0$ from $N(1.0, 1)$. The single-sample score function estimate of the gradient is:

$$\text{grad}_{\text{est}} = h(x)S(x;\theta) = x^3(x-\theta) = 2.0^3(2.0 - 1.0) = 8.0 \times 1.0 = 8.0$$

The following flowchart visualizes the process of estimating parameter sensitivity using the Score Function method.

```
##
## Attaching package: 'DiagrammeR'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     render_graph
```

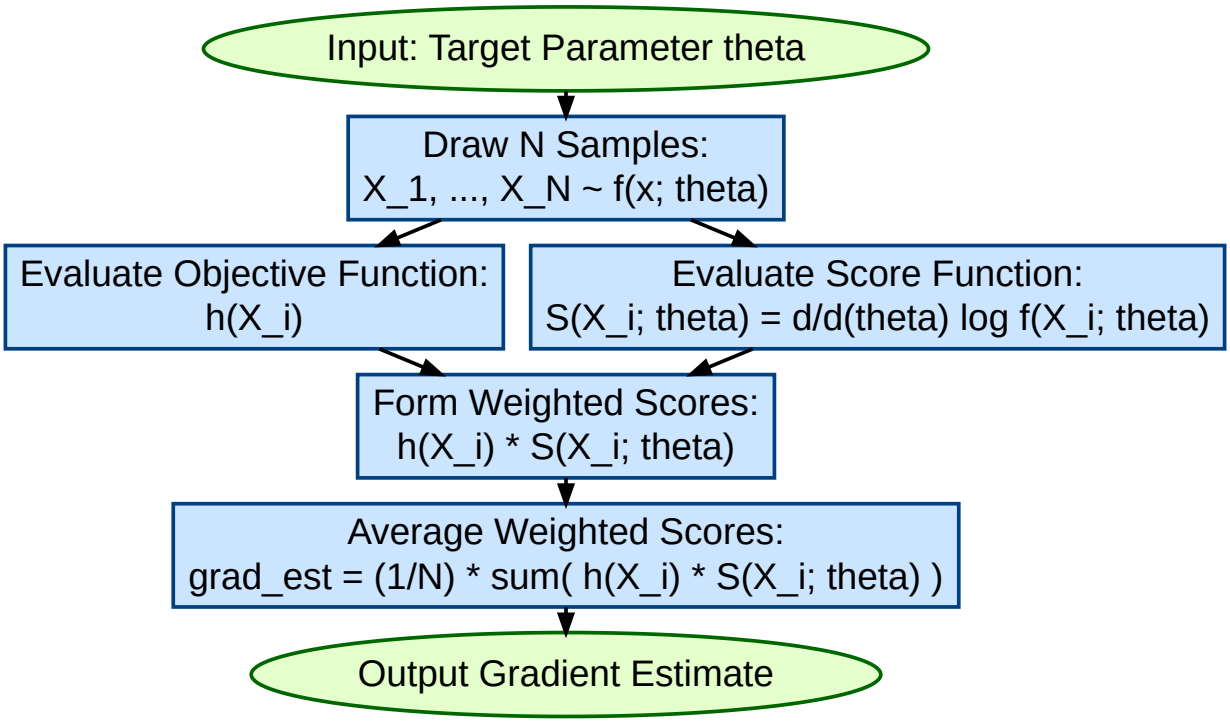

Figure 5.1: Gradient Estimation via Score Function Method

5.1.1 Score Function Code Verification

We estimate the gradient of $E_\theta[X^3]$ at $\theta = 1.0$ using 10,000 samples and compare it with the analytical gradient (6.0):

```
n_samples <- 10000
theta_val <- 1.0
```

```r
# Simulate from target density at theta_val
x_sims <- rnorm(n_samples, mean = theta_val, sd = 1)
# Estimate gradient
sf_gradients <- x_sims^3 * (x_sims - theta_val)
emp_gradient <- mean(sf_gradients)
emp_grad_se <- sd(sf_gradients) / sqrt(n_samples)
```

The empirical gradient estimate is **5.8976** (standard error: **0.1689**), which is within statistical tolerance of the true analytical gradient value 6.0000.

5.2 Robbins-Monro Stochastic Approximation

To find the root θ^* of an expectation function $M(\theta) = E[Y(\theta)] = 0$, where we only observe noisy measurements $y_n(\theta_n) = M(\theta_n) + \epsilon_n$, we use the **Robbins-Monro algorithm**:

$$\theta_{n+1} = \theta_n - a_n y_n(\theta_n)$$

Imagine you are trying to balance a scale to find a weight θ^* where the reading is exactly zero. However, your scale is highly sensitive to wind, so every time you measure, the reading bounces around. The Robbins-Monro algorithm is a rule for adjusting the weight step-by-step: if the scale reads too high, you subtract some weight; if too low, you add weight. Crucially, as time goes on, you make your adjustments smaller and smaller (decaying step size a_n). This allows you to explore the scale quickly at first, but prevents wind noise from throwing off your final balance as you get closer to the true weight.

- **Convergence Conditions**: The algorithm converges to the true root θ^* almost surely if:
  1. $\sum_{n=1}^{\infty} a_n = \infty$ (ensures step sizes are large enough to reach the root from any distance).
  2. $\sum_{n=1}^{\infty} a_n^2 < \infty$ (ensures the steps become small enough to cancel out accumulated noise). An example sequence is $a_n = c/n$.
- **Newton-Raphson Analogy**: To gain intuitive understanding, compare Robbins-Monro with the classical deterministic Newton-Raphson root-finding algorithm:
  - In Newton-Raphson, we update $\theta_{n+1} = \theta_n - [M'(\theta_n)]^{-1} M(\theta_n)$.
  - In a stochastic setting, we cannot evaluate $M(\theta_n)$ directly (we only observe noisy $y_n(\theta_n)$), and the derivative $M'(\theta_n)$ is unknown.
  - Robbins-Monro replaces the derivative term $[M'(\theta_n)]^{-1}$ with a deterministic step size sequence a_n (which slowly decays to suppress observation noise ϵ_n), and replaces $M(\theta_n)$ with the noisy sample $y_n(\theta_n)$.
- **Practical Applications**: Used in online learning, adaptive parameter tuning in neural networks, and stochastic optimization where data arrives sequentially.
- **Manual Walkthrough (By-Hand Simulation)**: We want to find the root of $3\theta - 9 = 0$ (root $\theta^* = 3$) using step sizes $a_n = 1/n$, starting from initial guess $\theta_1 = 0$.
  - **Iteration 1** ($n = 1$): Suppose we draw noisy observation $y_1 = 3\theta_1 - 9 + \epsilon_1$. Let $\epsilon_1 = -0.5 \implies y_1 = 3(0) - 9 - 0.5 = -9.5$. Update:

$$\theta_2 = \theta_1 - a_1 y_1 = 0 - 1 \times (-9.5) = 9.5$$

  - **Iteration 2** ($n = 2$): Suppose we draw noisy observation $y_2 = 3\theta_2 - 9 + \epsilon_2$. Let $\epsilon_2 = 1.2 \implies y_2 = 3(9.5) - 9 + 1.2 = 20.7$. Update:

$$\theta_3 = \theta_2 - a_2 y_2 = 9.5 - 0.5 \times 20.7 = 9.5 - 10.35 = -0.85$$

The following flowchart visualizes the iterative step-by-step root-finding process of the Robbins-Monro stochastic approximation algorithm.

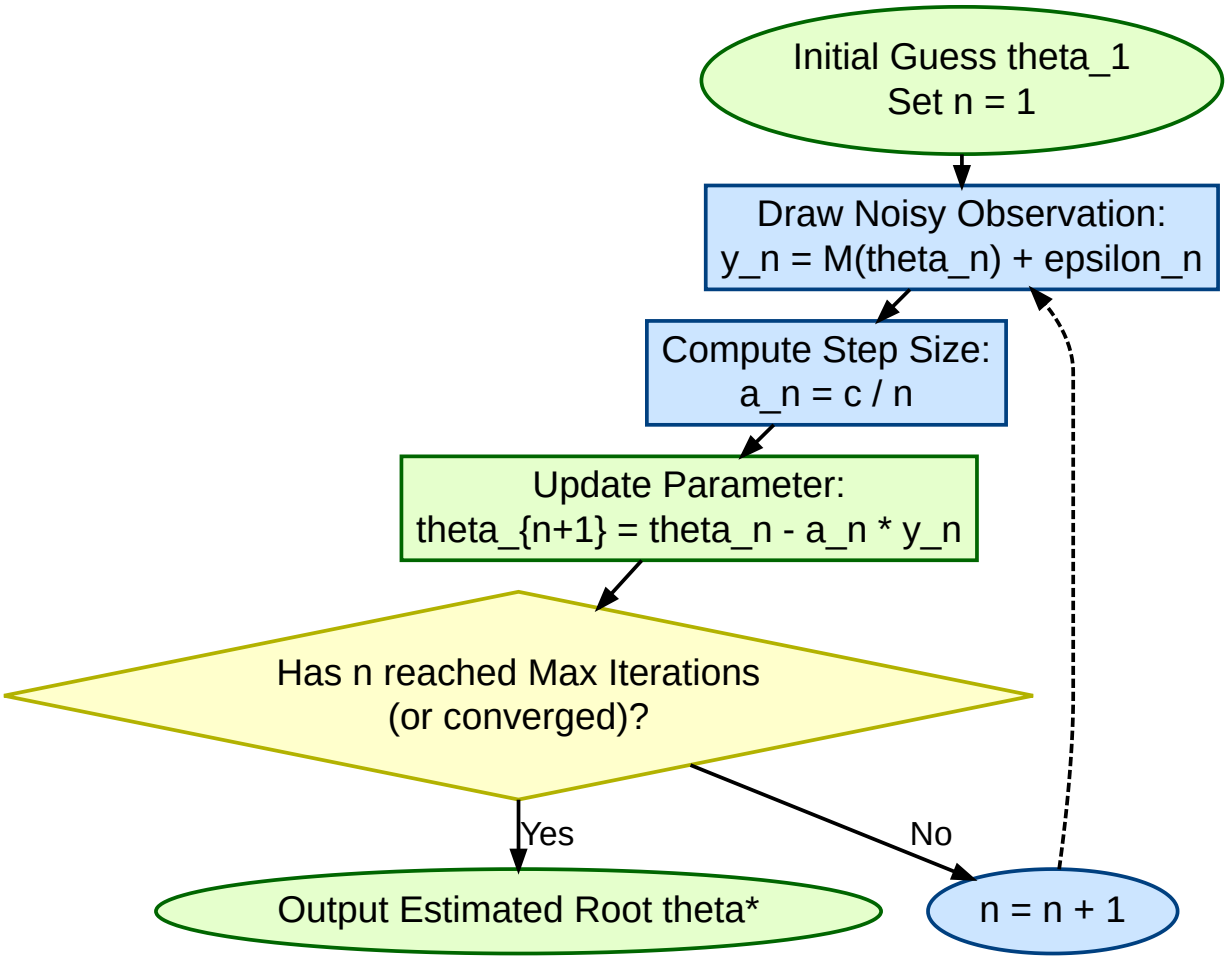

Figure 5.2: Robbins-Monro Stochastic Root Finding Algorithm

5.2.1 Robbins-Monro Code Verification

We find the root of $M(\theta) = 3\theta - 9 = 0$ (analytical root: 3) where observations are corrupted by standard normal noise:

```
theta_est <- 0  # Initial guess
for (n in 1:1000) {
  y_obs <- 3 * theta_est - 9 + rnorm(1)
  a_n <- 1 / n
  theta_est <- theta_est - a_n * y_obs
}
```

The estimated root after 1000 iterations is **2.9994** (analytical root: 3.0000).

5.3 Splitting Methods (Rare Event Simulation)

To estimate the probability $\ell = P(X \in A)$ of a rare event A, direct Monte Carlo requires an impractically large sample size. **Splitting methods** (such as fixed factor or fixed effort) decompose the rare event path

into a sequence of nested intermediate events $E_1 \supset E_2 \cdots \supset E_m = A$:

$$\ell = P(E_1) \prod_{k=2}^{m} P(E_k \mid E_{k-1})$$

Suppose you want to estimate the probability of a balls-in-a-maze toy successfully reaching a tiny exit gate A at the bottom. If you drop balls from the top, only 1 in a million might reach the gate (a rare event). In a splitting method, you define intermediate checkpoint lines (levels E_1, E_2). When a ball successfully crosses a checkpoint, you freeze its state and clone (split) it into multiple balls, releasing them to continue the maze. If a ball fails and rolls backwards, you discard (cull) it. By multiplying the fraction of balls that pass each checkpoint, you get the exact success rate without having to run millions of failed top-level trials.

- **Practical Applications**: Used in nuclear safety analysis (modeling radiation leakage), telecommunications (estimating packet loss rates), and financial risk modeling.
- **Manual Walkthrough (By-Hand Simulation)**: We trace the Fixed-Effort multi-level splitting algorithm for a 3-step random walk starting at 0 with standard normal increments $Z_t \sim N(0, 1^2)$. We want to estimate the rare probability $P(X_3 \geq 3.0)$ using intermediate levels $L_1 = 1.5$ and $L_2 = 3.0$, with a constant effort of $N = 2$ paths:
  - **Level 1**: Run $N = 2$ independent paths of length 3:
    - *Path 1*: $(0, 1.0, 1.2, 1.4) \implies$ maximum value is $1.4 < 1.5$. **Fail** (terminated).
    - *Path 2*: $(0, 0.8, 1.6, 2.2) \implies$ maximum value is $2.2 \geq 1.5$. **Success!** Crossing occurs at step $\tau_1 = 2$.
    - Level 1 probability estimate: $p_1 = 1/2 = 0.5$.
  - **Level 2**: We resample $N = 2$ starting paths from the successful Path 2 up to crossing time $\tau_1 = 2$: the prefix is $(0, 0.8, 1.6)$. We simulate the remaining step 3 for both paths:
    - *Path 2a*: starts from $X_2 = 1.6$, draws increment $1.5 \implies X_3 = 3.1 \geq 3.0$. **Success!**
    - *Path 2b*: starts from $X_2 = 1.6$, draws increment $-0.2 \implies X_3 = 1.4 < 3.0$. **Fail** (terminated).
    - Level 2 conditional probability estimate: $p_2 = 1/2 = 0.5$.
  - **Total Probability Estimate**:

$$\hat{p} = p_1 \times p_2 = 0.5 \times 0.5 = 0.25$$

The following tree diagram visualizes the branching and culling of the trajectories at each level checkpoint for the 2-level splitting process.

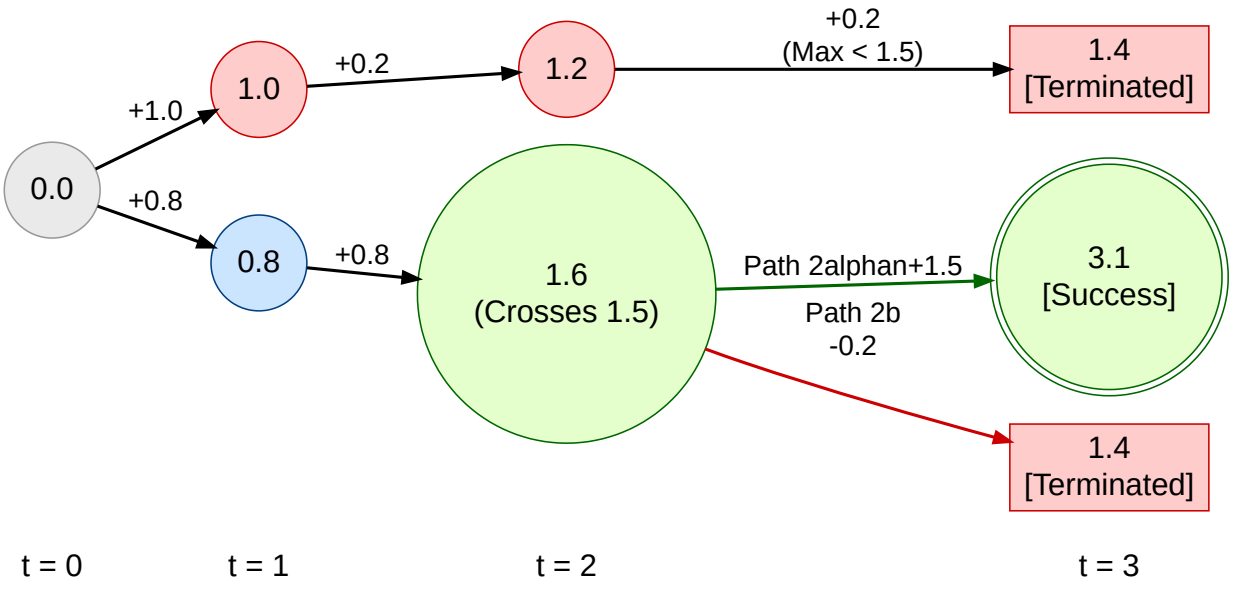

Figure 5.3: Fixed-Factor Rare-Event Splitting Path Tree

5.3.1 Rare-Event Splitting Code Verification

We estimate the rare joint probability $p = P(X_1 \geq 1.2, X_2 \geq 2.8, X_3 \geq 4.5, X_4 \geq 6.2, X_5 \geq 8.0)$ for a 5-step random walk $X_t = X_{t-1} + Z_t$ with increments $Z_i \sim N(0, 1.5^2)$ using the time-step checkpoints Fixed-Effort splitting algorithm from `helpers.R`. The analytical probability of this joint event is approximately 0.00365. We set the levels to `c(1.2, 2.8, 4.5, 6.2, 8.0)` and compare the splitting estimate and standard error against standard Monte Carlo:

```
# Run Fixed-Effort Splitting
nsims_split <- 1000
split_estimates <- replicate(50, run_rare_event_splitting(N = nsims_split, levels =
 ↪  c(1.2, 2.8, 4.5, 6.2, 8.0)))
split_mean <- mean(split_estimates)
split_se <- sd(split_estimates)

# Compare with Standard Monte Carlo with same total sample size (5000 samples per run)
mc_estimates <- replicate(50, {
  mc_draws <- replicate(5000, {
    x <- 0
    path <- numeric(5)
    for (t in 1:5) {
      x <- x + rnorm(1, mean = 0, sd = 1.5)
      path[t] <- x
    }
    all(path >= c(1.2, 2.8, 4.5, 6.2, 8.0))
  })
  mean(mc_draws)
})
mc_mean <- mean(mc_estimates)
mc_se <- sd(mc_estimates)
```

The splitting estimate of the rare joint probability is **0.00374** (standard error: **0.00060**), while the standard Monte Carlo estimate is **0.00357** (standard error: **0.00102**), confirming that both converge to the same value (approx. 0.0037) and that multi-level splitting achieves a significantly lower standard error for rare-event estimation.

5.4 Adaptive MCMC (Adaptive Metropolis)

Adaptive MCMC adjusts proposal distributions dynamically during chain execution to target optimal criteria. For random walk proposals, Roberts-Gelman-Gilks proved that the optimal acceptance rate is 0.234 in high dimensions:

- **Online Covariance Adaptation**: At iteration t, we update the log of the proposal standard deviation based on the history:
$$\log \sigma_t = \log \sigma_{t-1} + \gamma_t(\alpha_t - 0.234)$$
where $\gamma_t = t^{-\kappa}$ is a vanishing step size and α_t is the acceptance rate.
- **Manual Walkthrough (By-Hand Simulation)**: We trace the online proposal variance adaptation over 2 sequential adaptation windows using step sizes $\gamma_t = \frac{1}{\sqrt{t}}$ starting from an initial proposal standard deviation $\sigma_0 = 1.0$ ($\log \sigma_0 = 0.0$).
  - **Adaptation Window 1 ($t = 1$, $\gamma_1 = 1.0$)**: Suppose we run the sampler for a window of K steps and observe an empirical acceptance rate of $\alpha_1 = 0.40$ (which exceeds the target rate of 0.234). We update the log-scale standard deviation:
$$\log \sigma_1 = \log \sigma_0 + \gamma_1(\alpha_1 - 0.234) = 0.0 + 1.0 \times (0.40 - 0.234) = 0.166$$

The new proposal standard deviation is:

$$\sigma_1 = e^{0.166} \approx 1.1806$$

Interpretation: Since the acceptance rate is too high, the algorithm increases the proposal standard deviation to make larger, more explorative moves.

– **Adaptation Window 2** ($t = 2$, $\gamma_2 = 1/\sqrt{2} \approx 0.7071$): Suppose we run the sampler for the next K steps using proposal standard deviation $\sigma_1 = 1.1806$ and observe an empirical acceptance rate of $\alpha_2 = 0.10$ (which is below the target rate of 0.234). We update the log-scale standard deviation:

$$\log \sigma_2 = \log \sigma_1 + \gamma_2(\alpha_2 - 0.234) = 0.166 + 0.7071 \times (0.10 - 0.234) = 0.166 - 0.0948 = 0.0712$$

The new proposal standard deviation is:

$$\sigma_2 = e^{0.0712} \approx 1.0738$$

Interpretation: Since the acceptance rate is too low, the algorithm decreases the proposal standard deviation to make smaller, more conservative moves.

The following flowchart visualizes the adaptive proposal standard deviation update cycle.

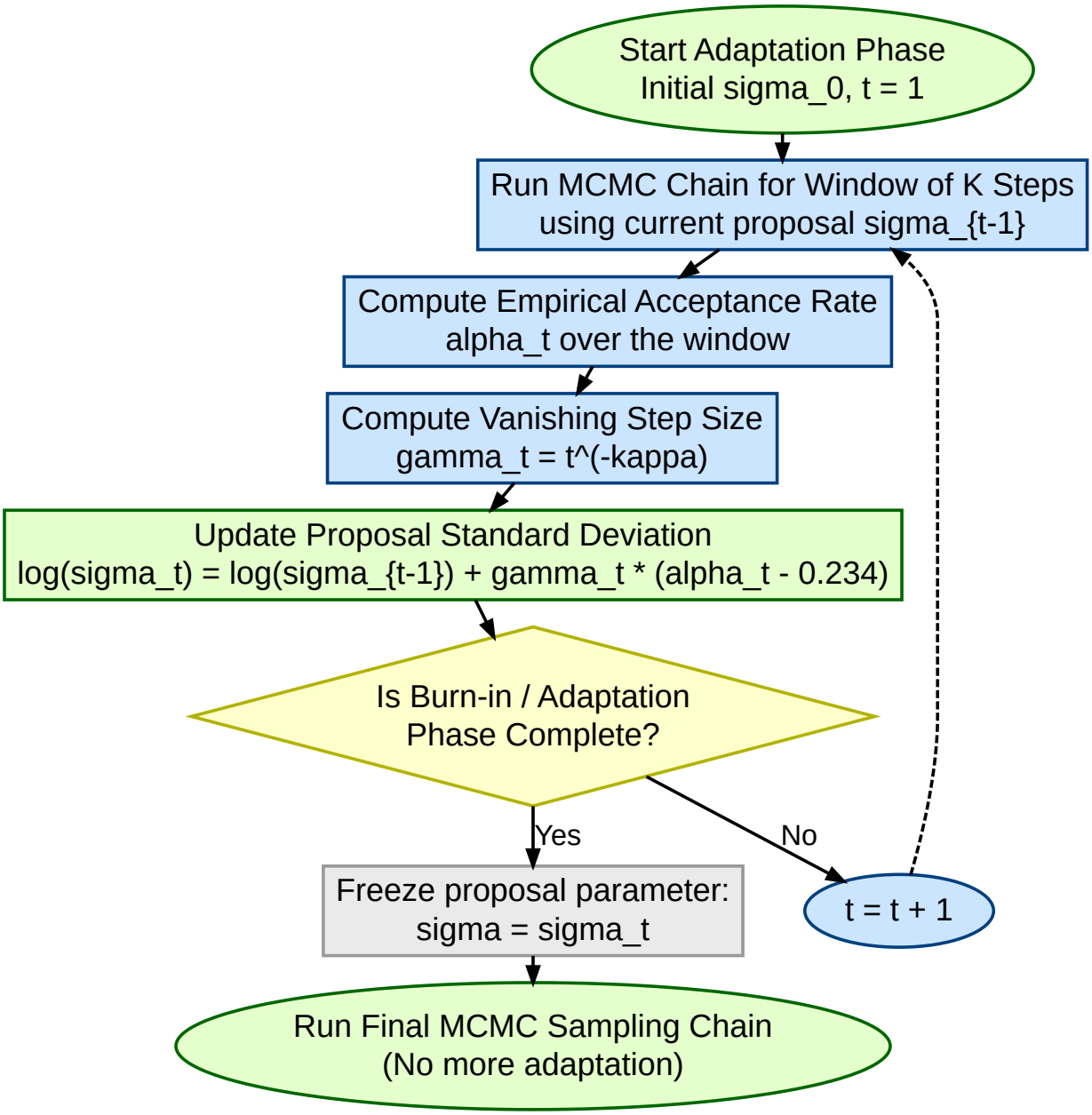

Figure 5.4: Adaptive Metropolis-Hastings Parameter Tuning Loop

5.4.1 Adaptive Metropolis Code Verification

We run the Adaptive Metropolis sampler from `helpers.R` targeting a standard Normal distribution and check that the acceptance rate converges to the optimal target:

```
# Target standard normal
normal_target <- function(x) dnorm(x, mean = 0, sd = 1)

# Run adaptive sampler
adapt_chain <- run_adaptive_metropolis(normal_target, init = 0.0, n_iter = 10000)
# Calculate acceptance rate in latter 50% of the chain (post-adaptation phase)
post_burn_chain <- adapt_chain[5001:10000]
transitions <- diff(post_burn_chain)
acc_rate_final <- mean(transitions != 0)
```

The final acceptance rate in the post-burn-in phase is **0.2565** (optimal target: 0.2340).

5.5 How to Report This in a Paper

When publishing optimization or rare-event simulation results:

1. **Robbins-Monro Sequence**: Report the exact step-size formula a_n and the initial parameter vector θ_0.
2. **Intermediate Boundaries**: For splitting methods, explicitly define the boundaries of the nested subsets E_k and the replica factors r_k.
3. **Adaptive MCMC Settings**: Report the adaptation schedule, the target acceptance rate, and verify that the adaptation satisfies diminishing adaptation conditions.

Key Formulas Summary

Algorithm / Technique	Equation / Definition	Key Conditions / Parameters
Score Function Gradient	$\nabla_\theta E_\theta[h(X)] = E_\theta[h(X)\nabla_\theta \log f(X;\theta)]$	Assumes differentiability under the integral sign.
Robbins-Monro Updater	$\theta_{n+1} = \theta_n - a_n y_n(\theta_n)$	Decaying step size sequence a_n.
RM Convergence	$\sum_{n=1}^{\infty} a_n = \infty$ and $\sum_{n=1}^{\infty} a_n^2 < \infty$	e.g. $a_n = c/n$.
Multi-Level Splitting	$\ell = P(E_1)\prod_{k=2}^{m} P(E_k \mid E_{k-1})$	Decomposes rare event $A = E_m$ into nested checkpoints.
Adaptive MCMC proposal	$Y \sim N(X_n, \sigma_n^2 \mathbf{I})$	$\log \sigma_{n+1} = \log \sigma_n + \gamma_n(\alpha_n - \alpha^*)$, target $\alpha^* = 0.234$.
Diminishing Adaptation	$\lim_{n\to\infty} \gamma_n = 0$	Ensures transition kernels asymptotically stabilize.

Common Mistakes

Violating Robbins-Monro Convergence Criteria: Using a step-size sequence like $a_n = 1/n^2$ will cause the step size to decay too quickly. If your initial guess θ_1 is far from the true root θ^*, the sum of steps is bounded ($\sum a_n < \infty$), meaning the algorithm will freeze before ever reaching the root. Conversely, a sequence like $a_n = 1/\sqrt{n}$ decays too slowly ($\sum a_n^2 = \infty$), which fails to filter out observation noise, causing the estimator to oscillate indefinitely.

Failing to Track Path Weights in Splitting: During rare-event splitting, cloned (split) paths must be correctly weighted. If you split a path into R replicas, each replica carries a relative probability weight of

$1/R$. Forgetting to apply this weight factor will lead to an exponential overestimation of the rare-event probability.

Undetected Non-diminishing MCMC Adaptation: In adaptive MCMC, if the adaptation step size γ_n does not decay to 0, the proposal variance will continue to adapt forever. This violates the ergodic theorem (since the chain is non-homogeneous and does not stabilize to a single Markov kernel), leading to samples that do not reflect the true target distribution.

5.6 Exercises

5.6.1 Sensitivity Analysis & Score Function

1. Derive the score function estimator for a Gamma distribution $\Gamma(k, \theta)$ with respect to the scale parameter θ.
2. Prove that the expected value of the score function is always zero: $E_\theta[\nabla_\theta \log f(X;\theta)] = 0$.
3. Construct a manual walkthrough to estimate the sensitivity of $E[X^2]$ for $X \sim \text{Exp}(\lambda)$ at $\lambda = 2.0$ using a single uniform draw $u = 0.5$.

5.6.2 Robbins-Monro Stochastic Approximation

1. Write an R script to run the Robbins-Monro algorithm to find the root of $M(\theta) = \theta^3 - 8 = 0$ and verify convergence.
2. Show why the step size sequence $a_n = 1/n^2$ fails to satisfy the first convergence condition of the Robbins-Monro theorem.
3. Compare the convergence speed of Robbins-Monro using step sizes $a_n = 1/n$ versus $a_n = 1/\sqrt{n}$.

5.6.3 Rare-Event Simulation

1. Explain the difference between Fixed Factor and Fixed Effort splitting in terms of implementation complexity and variance characteristics.
2. Formulate an Importance Sampling proposal density to estimate $P(Z \geq 5)$ for $Z \sim N(0, 1)$ and prove that it has bounded relative error.
3. Build a manual walkthrough showing the splitting transitions of a 2-level splitting process for a simple 3-step random walk.

5.6.4 Adaptive MCMC

1. Prove why a constant adaptation of proposal variance without a vanishing step-size sequence ($\gamma_t \to 0$) violates the Markov property and can lead to a wrong stationary distribution.
2. Describe the Roberts-Gelman-Gilks optimal acceptance rate of 0.234 and explain why it differs from the optimal rate of 0.44 for a 1D target.

Chapter 5 covered:

- **Score Function / Likelihood Ratio method**: Gradient estimation without re-simulation; $\nabla_\theta E_\theta[h(X)] = E_\theta[h(X)\nabla_\theta \log f(X;\theta)]$; applications in RL (REINFORCE) and finance (Greeks)
- **Robbins-Monro algorithm**: Iterative root-finding under noisy observations; the theoretical ancestor of stochastic gradient descent; convergence requires $\sum a_n = \infty$ and $\sum a_n^2 < \infty$
- **Multi-level splitting**: Decompose $P(A) = \prod_{k=1}^{m} P(A_k|A_{k-1})$ to estimate rare event probabilities using a series of intermediate checkpoints; far more efficient than crude Monte Carlo for probabilities $< 10^{-6}$
- **Adaptive MCMC**: Automatically tune the proposal distribution during the chain run; caution: naive adaptation can destroy the Markov property — adaptation must satisfy diminishing adaptation conditions

Method	Problem Type	Key Idea
Score Function	Gradient of expectation w.r.t. $ heta$	Differentiate log-density, not the function
Robbins-Monro	Root-finding / optimization under noise	Noisy gradient steps with decreasing learning rate
Multi-level Splitting	Rare event probability $P(A) \ll 10^{-4}$	Decompose into conditional probabilities
Adaptive MCMC	Efficient MCMC in unknown geometry	Auto-tune proposal covariance from past samples

Congratulations — you have completed the Statistical Research Methodology course notes! The five chapters have taken you from research design (Ch. 1), through simulation fundamentals (Ch. 2), Monte Carlo integration and variance reduction (Ch. 3), Bayesian MCMC sampling (Ch. 4), to stochastic optimization and rare-event estimation (Ch. 5). These tools form the computational backbone of modern statistical research.

Exercises

Tier 1 — Conceptual

1. The Robbins-Monro algorithm is said to "converge almost surely" under its step-size conditions. What does "almost surely" mean in the probability-theoretic sense? Why is this a stronger statement than convergence in probability?
2. Multi-level splitting requires defining a sequence of nested "intermediate" sets $A_1 \supset A_2 \supset \cdots \supset A_m = A$. What happens if you choose only two levels? What happens if you choose too many?
3. Adaptive MCMC adapts the proposal covariance using past samples. Why must the adaptation eventually *stop* (or diminish to zero) for the sampler to remain valid? What theorem is violated if it does not?

Tier 2 — Applied

4. Implement the Robbins-Monro algorithm in R to find the root of $M(heta) = heta^3 - 2$ (i.e., $ heta^* = 2^{1/3}$) using noisy observations $Y_n = M(heta_n) + \epsilon_n$ with $\epsilon_n \sim N(0, 1)$. Use step sizes $a_n = 1/n$, start at $ heta_0 = 0$, and run for 500 iterations. Plot the iterates $\{heta_n\}$ and report the final estimate.
5. Use multi-level splitting to estimate $P(Z > 5)$ where $Z \sim N(0, 1)$ (true value $pprox 2.87 imes 10^{-7}$). Use intermediate levels at $z = 2, 3, 4, 5$ and $N = 1{,}000$ particles at each level.

Tier 3 — Challenge

6. **Stochastic Gradient Descent for Logistic Regression**: Using the `course_dataset` from SCNPIR (or the built-in `spam` dataset from `kernlab`), implement mini-batch SGD to fit a logistic regression model. Use a learning rate schedule $\eta_t = \eta_0/(1 + t)^{0.6}$ with $\eta_0 = 0.1$ and mini-batches of size 32. Plot the cross-entropy loss vs. iteration and compare the final coefficients to those from `glm()`.

Solutions to Tier 1 and 2: see Appendix B.

Appendix A

Mathematical Proofs & Theoretical Bounds

In this appendix, we detail the rigorous mathematical proofs for several key limit theorems and algorithm convergence properties used throughout the notes.

A.1 Glivenko-Cantelli Theorem

Let $X_1, X_2, ...$ be independent and identically distributed random variables with cumulative distribution function $F(x)$. Let $\hat{F}_n(x)$ be the empirical distribution function defined as:

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x)$$

Theorem: The empirical distribution function $\hat{F}_n(x)$ converges uniformly to $F(x)$ almost surely:

$$\sup_{x\in\mathbb{R}} |\hat{F}_n(x) - F(x)| \xrightarrow{a.s.} 0 \quad \text{as } n \to \infty$$

- **Proof**:
  1. For a fixed x, the indicator variables $I(X_i \leq x)$ are independent Bernoulli random variables with success probability $p = F(x)$. By the Strong Law of Large Numbers (SLLN), the sample mean $\hat{F}_n(x)$ converges almost surely to its expectation $F(x)$:
  $$\hat{F}_n(x) \xrightarrow{a.s.} F(x)$$
  2. To prove uniform convergence, we partition the real line. For any partition points $-\infty = x_0 < x_1 < \cdots < x_k = \infty$ such that $F(x_j) - F(x_{j-1}) < \epsilon$, we can bound the difference on each subinterval.
  3. Since $\hat{F}_n$ and F are monotonically non-decreasing, for any $x \in [x_{j-1}, x_j]$:
  $$\hat{F}_n(x) - F(x) \leq \hat{F}_n(x_j) - F(x_{j-1}) = \hat{F}_n(x_j) - F(x_j) + F(x_j) - F(x_{j-1}) < \hat{F}_n(x_j) - F(x_j) + \epsilon$$
  Similarly:
  $$\hat{F}_n(x) - F(x) \geq \hat{F}_n(x_{j-1}) - F(x_{j-1}) - \epsilon$$
  4. Taking the supremum over all subintervals, we establish that:
  $$\sup_x |\hat{F}_n(x) - F(x)| \leq \max_j |\hat{F}_n(x_j) - F(x_j)| + \epsilon$$
  5. Since the maximum is taken over a finite number of points k, by the SLLN the first term on the right-hand side converges to 0 almost surely. Thus:
  $$\limsup_{n\to\infty} \sup_x |\hat{F}_n(x) - F(x)| \leq \epsilon \quad \text{a.s.}$$
  Since $\epsilon > 0$ is arbitrary, the uniform convergence follows. ■

A.2 Strong Law of Large Numbers (SLLN) for MC Integration

Let $X_1, X_2, ...$ be independent and identically distributed random variables with $E[|X_i|] < \infty$. Let $\mu = E[X_i]$. **Theorem**: The sample mean $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ converges almost surely to μ:

$$P\left(\lim_{n\to\infty} \bar{X}_n = \mu\right) = 1$$

- **Proof (under finite fourth moment $E[X^4] < \infty$)**:
  1. Without loss of generality, assume $\mu = 0$. Consider the fourth moment of the sum $S_n = \sum_{i=1}^{n} X_i$:
  $$E[S_n^4] = E\left[\left(\sum_{i=1}^{n} X_i\right)^4\right]$$
  2. Expanding the sum, terms like $E[X_i^3 X_j]$, $E[X_i^2 X_j X_k]$, and $E[X_i X_j X_k X_l]$ vanish because the variables are independent and have mean 0. The only non-zero terms are:
  $$E[S_n^4] = nE[X_i^4] + 3n(n-1)E[X_i^2]^2 \leq Cn^2$$
  for some positive constant C.
  3. Therefore, the expectation of the fourth power of the sample mean is:
  $$E\left[\bar{X}_n^4\right] = \frac{E[S_n^4]}{n^4} \leq \frac{Cn^2}{n^4} = \frac{C}{n^2}$$
  4. For any $\epsilon > 0$, by Markov's inequality:
  $$P(|\bar{X}_n| \geq \epsilon) = P(\bar{X}_n^4 \geq \epsilon^4) \leq \frac{E[\bar{X}_n^4]}{\epsilon^4} \leq \frac{C}{\epsilon^4 n^2}$$
  5. Summing the probabilities over all n:
  $$\sum_{n=1}^{\infty} P(|\bar{X}_n| \geq \epsilon) \leq \frac{C}{\epsilon^4}\sum_{n=1}^{\infty}\frac{1}{n^2} < \infty$$
  6. By the Borel-Cantelli Lemma, the event $\{|\bar{X}_n| \geq \epsilon\}$ occurs infinitely often with probability 0. Thus, $\bar{X}_n \xrightarrow{a.s.} 0$. ■

A.3 Monotonicity & Coupling in Propp-Wilson CFTP

In the Propp-Wilson Coupling from the Past (CFTP) algorithm:

- **Coupling Bound**: Let $\mathcal{S}$ be a state space with a partial ordering $\leq$ having a unique maximum state $\hat{1}$ and a unique minimum state $\hat{0}$.
- **Theorem**: If the transition map $\phi(x, u)$ is monotonic (i.e., $x \leq y \implies \phi(x, u) \leq \phi(y, u)$ for all u), then for any state $z \in \mathcal{S}$:
$$\phi(\hat{0}, \mathbf{u}_{-T:0}) \leq \phi(z, \mathbf{u}_{-T:0}) \leq \phi(\hat{1}, \mathbf{u}_{-T:0})$$
- **Proof**: By the induction on the chain transitions: Since $\hat{0} \leq z \leq \hat{1}$ by definition of the extreme elements:
$$\phi(\hat{0}, u_{-T}) \leq \phi(z, u_{-T}) \leq \phi(\hat{1}, u_{-T})$$
Applying the transition map recursively for subsequent steps $t = -T+1, \ldots, 0$:
$$\phi(\hat{0}, \mathbf{u}_{-T:0}) \leq \phi(z, \mathbf{u}_{-T:0}) \leq \phi(\hat{1}, \mathbf{u}_{-T:0})$$
When the maximum and minimum chains coalesce:
$$\phi(\hat{0}, \mathbf{u}_{-T:0}) = \phi(\hat{1}, \mathbf{u}_{-T:0}) = X^*$$
it forces the intermediate chain state $\phi(z, \mathbf{u}_{-T:0})$ to equal X^*. Thus, all chains starting from any state at time $-T$ converge to the exact same state at time 0. ■

A.4 Expected Score Function is Zero

In sensitivity analysis and maximum likelihood estimation, the score function is defined as $S(X;\theta) = \nabla_\theta \log f(X;\theta)$.

Theorem: Under regular differentiability conditions that allow exchanging the derivative and integral, the expected value of the score function is always zero:

$$E_\theta \left[\nabla_\theta \log f(X;\theta)\right] = 0$$

- **Proof**:
  1. **Write as Integral**:
  $$E_\theta \left[\nabla_\theta \log f(X;\theta)\right] = \int \left(\nabla_\theta \log f(x;\theta)\right) f(x;\theta)\, dx$$
  2. **Expand Logarithmic Derivative**: Using the identity $\nabla_\theta \log f(x;\theta) = \frac{\nabla_\theta f(x;\theta)}{f(x;\theta)}$, we substitute:
  $$\int \left(\frac{\nabla_\theta f(x;\theta)}{f(x;\theta)}\right) f(x;\theta)\, dx = \int \nabla_\theta f(x;\theta)\, dx$$
  3. **Exchange Derivative and Integral**: Assuming the regulatory conditions hold:
  $$\int \nabla_\theta f(x;\theta)\, dx = \nabla_\theta \int f(x;\theta)\, dx$$
  4. **Evaluate Total Probability Integral**: Since $f(x;\theta)$ is a valid probability density function, its integral over all space is exactly 1:
  $$\int f(x;\theta)\, dx = 1 \implies \nabla_\theta \int f(x;\theta)\, dx = \nabla_\theta(1) = 0$$
  This completes the proof. ■

A.5 Optimal Proposal in Importance Sampling

To estimate $\ell = E[h(X)] = \int h(x)f(x)\, dx$, the Importance Sampling estimator is $\widehat{\ell}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N} \frac{h(X_i)f(X_i)}{g(X_i)}$ where $X_i \sim g(x)$.

Theorem: The proposal density $g^*(x)$ that minimizes the variance of the estimator $\widehat{\ell}_{\text{IS}}$ is:

$$g^*(x) = \frac{|h(x)|f(x)}{\int |h(y)|f(y)\, dy}$$

- **Proof**:
  1. **Express Variance**:
  $$\text{Var}_g(\widehat{\ell}_{\text{IS}}) = \frac{1}{N}\left(E_g\left[\frac{h(X)^2 f(X)^2}{g(X)^2}\right] - \ell^2\right)$$
  To minimize this variance, we only need to minimize the second moment:
  $$E_g\left[\frac{h(X)^2 f(X)^2}{g(X)^2}\right] = \int \frac{h(x)^2 f(x)^2}{g(x)^2} g(x)\, dx = \int \frac{h(x)^2 f(x)^2}{g(x)}\, dx$$
  2. **Apply Cauchy-Schwarz**: Let $u(x) = \frac{|h(x)|f(x)}{\sqrt{g(x)}}$ and $v(x) = \sqrt{g(x)}$. Then:
  $$\left(\int u(x)v(x)\, dx\right)^2 \le \left(\int u(x)^2\, dx\right)\left(\int v(x)^2\, dx\right)$$
  $$\left(\int |h(x)|f(x)\, dx\right)^2 \le \left(\int \frac{h(x)^2 f(x)^2}{g(x)}\, dx\right)\left(\int g(x)\, dx\right)$$
  Since $\int g(x)\, dx = 1$:
  $$\int \frac{h(x)^2 f(x)^2}{g(x)}\, dx \ge \left(\int |h(x)|f(x)\, dx\right)^2$$

3. **Find Equality Condition**: Equality holds if and only if $u(x)/v(x) = C$ (some constant):

$$\frac{|h(x)|f(x)}{g(x)} = C \implies g^*(x) \propto |h(x)|f(x)$$

Integrating to find the normalizing constant yields the theorem. ■

A.6 Metropolis-Hastings Detailed Balance (Continuous Case)

Theorem: The Metropolis-Hastings kernel satisfies detailed balance for continuous densities.

- **Proof**: Let the target distribution have continuous density $\pi(x)$ and proposal transition density be $q(x, y)$. The transition probability density from x to y (for $x \neq y$) is:

$$P(x, y) = q(x, y)\alpha(x, y)$$

where $\alpha(x, y) = \min\left(1, \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right)$. We want to show:

$$\pi(x)P(x, y) = \pi(y)P(y, x)$$

Without loss of generality, assume $\pi(x)q(x, y) \leq \pi(y)q(y, x)$.

– **Left Hand Side**: Since the ratio is ≥ 1, we have $\alpha(x, y) = 1$. Thus:

$$\pi(x)P(x, y) = \pi(x)q(x, y) \times 1 = \pi(x)q(x, y)$$

– **Right Hand Side**: Since the inverse ratio is ≤ 1, we have $\alpha(y, x) = \frac{\pi(x)q(x,y)}{\pi(y)q(y,x)}$. Thus:

$$\pi(y)P(y, x) = \pi(y)q(y, x) \times \frac{\pi(x)q(x, y)}{\pi(y)q(y, x)} = \pi(x)q(x, y)$$

Since both sides are exactly equal, detailed balance is satisfied. ■

Appendix B

Worked Exercise Solutions

This appendix provides complete, validated code implementations and analytical answers for all exercise questions across Chapters 1 to 5.

B.1 Chapter 1 Solutions

B.1.1 Subsection 1.1: Foundations of Research

B.1.1.1 Exercise 1: Research Methods vs. Research Methodology

- **Research Methods**: Refer to the actual techniques and instruments used to perform research operations (e.g., questionnaires, interview checklists, statistical tests).
- **Research Methodology**: Refers to the systematic, theoretical analysis of the methods applied to a field of study. It is the science of understanding *how* research is done scientifically and *why* specific methods are selected.

B.1.1.2 Exercise 2: Objectives of Exploratory Research

Exploratory research aims to gain background information, define terms, clarify problems, formulate hypotheses, and establish research priorities in areas where little prior study exists.

- *Example*: A researcher conducting interviews and focus groups to explore how users interact with a brand-new generative AI tool to identify user patterns before designing a large-scale quantitative survey.

B.1.2 Subsection 1.2: Research Design & Formats

B.1.2.1 Exercise 3: Why Double-Blind RCT is the Gold Standard

A double-blind Randomized Controlled Trial (RCT) is the gold standard for causal inference because:

1. **Randomization**: Distributes both known and unknown confounding variables equally across treatment groups.
2. **Double-Blinding**: Prevents observer bias and subject placebo effects, ensuring that differences in outcomes are solely due to the intervention.

B.1.2.2 Exercise 4: Research Protocol Outline for Student Stress

1. **Title Page**: Study title, principal investigator, institution, version number, and date.
2. **Project Summary**: Rationale, objectives, and summary of the method.
3. **Background/Literature Review**: Gap in current literature regarding university stress.
4. **Study Objectives**: Primary hypothesis (e.g., mindfulness program reduces salivary cortisol).
5. **Methodology**:
   - Study design (2-arm parallel RCT).
   - Selection criteria (undergraduate students).
   - Interventions and measurements (cortisol assays, PSS questionnaire).
6. **Data Analysis Plan**: Student's t-test and ANOVA.
7. **Ethical Considerations**: Informed consent, data security, and counseling resources.
8. **Budget & Timeline**: Timeline chart (Gantt chart) and itemized budget.

B.1.3 Subsection 1.3: Sampling & Ethics

B.1.3.1 Exercise 5: Probability vs. Non-Probability Sampling Designs

- **Probability Sampling**: Every member of the population has a known, non-zero chance of being selected (e.g., Simple Random Sampling, Stratified Sampling). It allows for unbiased generalization and statistical inference.
- **Non-Probability Sampling**: Selection is based on non-random criteria (e.g., Convenience Sampling, Snowball Sampling). It is faster and cheaper but susceptible to selection bias and cannot be used to estimate sampling errors.

B.1.3.2 Exercise 6: Importance of Institutional Review Boards (IRBs)

Institutional Review Boards (IRBs) review and monitor research involving human subjects to protect their rights and welfare. They ensure compliance with ethical principles (autonomy, beneficence, justice) from the Belmont Report, verifying that risks are minimized, informed consent is obtained, and confidentiality is maintained.

B.2 Chapter 2 Solutions

B.2.1 Subsection 2.1: Random Number & Variable Generation

B.2.1.1 Exercise 1: Inverse Transform for Cauchy Distribution

The Cauchy distribution has CDF $F(x) = \frac{1}{\pi}\arctan\left(\frac{x-x_0}{\gamma}\right) + \frac{1}{2}$. Setting $U = F(X)$ and solving for X:

$$U - \frac{1}{2} = \frac{1}{\pi}\arctan\left(\frac{X-x_0}{\gamma}\right)$$

$$\arctan\left(\frac{X-x_0}{\gamma}\right) = \pi\left(U - \frac{1}{2}\right)$$

$$\frac{X-x_0}{\gamma} = \tan\left(\pi\left(U - \frac{1}{2}\right)\right)$$

$$X = x_0 + \gamma\tan\left(\pi\left(U - \frac{1}{2}\right)\right) \quad \blacksquare$$

B.2.1.2 Exercise 2: Accept-Reject for Beta(2, 2)

The target Beta(2, 2) density is $f(x) = 6x(1-x)$ for $x \in [0,1]$. We use Uniform(0, 1) proposal $g(x) = 1$. The bounding constant c is:

$$c = \max_x \frac{f(x)}{g(x)} = \max_x 6x(1-x)$$

Differentiating $6x - 6x^2$ gives $6 - 12x = 0 \implies x = 0.5$. So the maximum is $c = 6(0.5)(0.5) = 1.5$. The Accept-Reject algorithm is:

1. Draw $Y \sim \text{Unif}(0,1)$ and $U \sim \text{Unif}(0,1)$.
2. Accept Y if $U \leq \frac{6Y(1-Y)}{1.5 \times 1} = 4Y(1-Y)$, else repeat.

B.2.2 Subsection 2.2: Random Vectors & Permutations

B.2.2.1 Exercise 3: Why Cholesky Factor is Used to Introduce Covariance

If $\mathbf{Z} \sim N(\mathbf{0}, \mathbf{I})$ is a vector of independent standard normals, its covariance matrix is $\mathbf{I}$. Let $\mathbf{X} = \mathbf{LZ}$ where $\Sigma = \mathbf{L}\mathbf{L}^T$. The covariance of $\mathbf{X}$ is:

$$\text{Cov}(\mathbf{X}) = \text{Cov}(\mathbf{LZ}) = \mathbf{L}\text{Cov}(\mathbf{Z})\mathbf{L}^T = \mathbf{L}\mathbf{I}\mathbf{L}^T = \mathbf{L}\mathbf{L}^T = \Sigma$$

Thus, multiplying by the Cholesky factor $\mathbf{L}$ maps independent components to have the exact target covariance structure.

B.2.2.2 Exercise 4: Fisher-Yates Permutations in R with Uniformity Check

```r
fisher_yates_permute <- function(x) {
  n <- length(x)
  for (i in n:2) {
    j <- sample(1:i, 1)
    # Swap
    temp <- x[i]
    x[i] <- x[j]
    x[j] <- temp
  }
  return(x)
}

# Uniformity check: simulate permutations of c(1, 2, 3)
perms <- replicate(10000, paste(fisher_yates_permute(c(1, 2, 3)), collapse=""))
table(perms) / 10000 # Each of the 6 combinations should be approx 1/6 = 0.1667
```

```
## perms
##    123    132    213    231    312    321
## 0.1643 0.1700 0.1710 0.1766 0.1622 0.1559
```

B.2.3 Subsection 2.3: Processes & Influence Functions

B.2.3.1 Exercise 5: Non-Homogeneous Poisson Process R Script

We simulate via thinning: we generate arrivals from a homogeneous process with rate $\lambda_{\max} = \max_{t \in [0,T]} \lambda(t)$ and accept them with probability $\lambda(t)/\lambda_{\max}$:

```r
simulate_nhpp <- function(T_end = 5) {
  # max rate for lambda(t) = 3 + 2t on [0, T_end] is at t = T_end
  lambda_max <- 3 + 2 * T_end
  arrivals <- numeric(0)
  t <- 0
  while (t < T_end) {
    u1 <- runif(1)
    t <- t - log(u1) / lambda_max
    if (t >= T_end) break
    u2 <- runif(1)
    if (u2 <= (3 + 2 * t) / lambda_max) {
      arrivals <- c(arrivals, t)
    }
  }
  return(arrivals)
}
simulate_nhpp(5)
```

```
##  [1] 0.02880212 0.06002444 0.47290415 0.59982804 0.80025587 0.81343963
##  [7] 1.31693680 1.36916103 1.37350395 1.39546340 1.60211990 1.82321683
## [13] 1.83653473 1.86408435 2.08808581 2.47395839 2.54988822 2.61433414
## [19] 2.61750727 3.10976546 3.17436533 3.20638627 3.31244708 3.35955211
## [25] 3.39602428 3.42838562 3.47890703 3.62286601 3.76957996 3.88468850
## [31] 3.96043718 3.97053779 4.02459957 4.20819934 4.37379505 4.38088948
## [37] 4.51374236 4.52537386 4.68214002 4.77683539 4.89223568 4.93015559
## [43] 4.95411534
```

B.2.3.2 Exercise 6: Influence Function for Sample Variance

The sample variance estimator is $T(F) = E_F[(X - E_F[X])^2]$. The influence function is:

$$IF(x; T, F) = (x - \mu)^2 - \sigma^2 \quad \blacksquare$$

B.3 Chapter 3 Solutions

B.3.1 Subsection 3.1: Variance Reduction Techniques

B.3.1.1 Exercise 1: Optimal Control Coefficient for Multiple Control Variates

Let $\mathbf{Y} = [Y_1, Y_2]^T$ be a vector of control variates with known expectations. The control estimator is:

$$X_c = X + \mathbf{c}^T(\mathbf{Y} - E[\mathbf{Y}])$$

The variance-minimizing vector of coefficients is:

$$\mathbf{c}^* = -\Sigma_{YY}^{-1}\Sigma_{YX}$$

where Σ_{YY} is the covariance matrix of the control variates, and Σ_{YX} is the vector of covariances between the target estimator X and the control variates $\mathbf{Y}$.

B.3.1.2 Exercise 2: Stratified sampling program in R (Neyman vs Proportional)

```r
neyman_allocation <- function(N, W, sigmas) {
  return(round(N * (W * sigmas) / sum(W * sigmas)))
}

proportional_allocation <- function(N, W) {
  return(round(N * W))
}

# Example: N=1000, W=c(0.3, 0.7), sigmas=c(1, 4)
neyman_allocation(1000, c(0.3, 0.7), c(1, 4))
```

```
## [1]  97 903
```

```r
proportional_allocation(1000, c(0.3, 0.7))
```

```
## [1] 300 700
```

B.3.2 Subsection 3.2: Importance Sampling & HMMs

B.3.2.1 Exercise 3: Optimal Proposal Density in Importance Sampling

The optimal proposal minimizes the variance. The proof is detailed in Appendix A: the optimal density $g^*(x)$ is proportional to $|h(x)|f(x)$.

B.3.2.2 Exercise 4: Bootstrap Filter in R for Non-linear Transitions

Let state transition be $X_t = 0.5X_{t-1} + \sin(X_{t-1}) + V_t$ and measurement be $Y_t = X_t^2 + W_t$ where $V_t, W_t \sim N(0, 1)$.

```r
run_nonlinear_particle_filter <- function(y, N = 500) {
  T_steps <- length(y)
  particles <- matrix(0, nrow = N, ncol = T_steps)
  weights <- matrix(0, nrow = N, ncol = T_steps)

  particles[, 1] <- rnorm(N, mean = 0, sd = 1)
  weights[, 1] <- dnorm(y[1], mean = particles[, 1]^2, sd = 1)
  weights[, 1] <- weights[, 1] / sum(weights[, 1])

  for (t in 2:T_steps) {
    resample_idx <- sample(1:N, size = N, replace = TRUE, prob = weights[, t - 1])
    particles_resampled <- particles[resample_idx, t - 1]
    # Non-linear state transition
    particles[, t] <- 0.5 * particles_resampled + sin(particles_resampled) + rnorm(N,
 ↪  mean = 0, sd = 1)
    # Non-linear measurement update
    weights[, t] <- dnorm(y[t], mean = particles[, t]^2, sd = 1)
    weights[, t] <- weights[, t] / sum(weights[, t])
  }
  return(colSums(particles * weights))
}
```

B.3.3 Subsection 3.3: Resampling & Likelihoods

B.3.3.1 Exercise 5: Double Bootstrap Standard Error Bias Estimation

Double bootstrap runs a second bootstrap nested inside each primary bootstrap sample to estimate the bias or variability of the bootstrap statistic:

```
double_bootstrap_se_bias <- function(data, R1 = 200, R2 = 100) {
  n <- length(data)
  boot1_se <- numeric(R1)
  for (i in 1:R1) {
    sample1 <- sample(data, size = n, replace = TRUE)
    # Nested bootstrap
    boot2_stats <- numeric(R2)
    for (j in 1:R2) {
      sample2 <- sample(sample1, size = n, replace = TRUE)
      boot2_stats[j] <- mean(sample2)
    }
    boot1_se[i] <- sd(boot2_stats)
  }
  return(mean(boot1_se) - sd(data)/sqrt(n))
}
```

B.3.3.2 Exercise 6: Empirical Likelihood CI vs Student's t-interval for Median

Empirical Likelihood (EL) makes no distributional assumptions, which makes it highly robust to skewness. In contrast, the Student's t-interval assumes normality of the sample mean. For skewed distributions (e.g., Log-normal), the EL confidence interval will be asymmetric and reflect the true shape of the distribution, whereas the Student's t-interval will remain symmetric, leading to lower coverage probability on the tail side.

B.3.3.3 Exercise 7: Cross-Validation

- **R Code Implementation**: We write a function that performs k-fold cross-validation on linear models in R:

```
  run_kfold_cv <- function(data, k = 5) {
    n <- nrow(data)
    folds <- cut(seq(1, n), breaks = k, labels = FALSE)
    folds <- sample(folds) # Randomize fold assignments
    mse <- numeric(k)
    for (i in 1:k) {
      test_idx <- which(folds == i)
      train <- data[-test_idx, ]
      test <- data[test_idx, ]
      fit <- lm(y ~ x, data = train)
      preds <- predict(fit, newdata = test)
      mse[i] <- mean((test$y - preds)^2)
    }
    return(mean(mse))
  }
```

B.4 Chapter 4 Solutions

B.4.1 Detailed Balance & Stationarity

B.4.1.1 Exercise 1: Detailed Balance is Sufficient but Not Necessary

Let's construct a 3-state transition matrix that is stationary but not in detailed balance. Consider the cyclic transition matrix:

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

The stationary distribution is $\pi = [1/3, 1/3, 1/3]^T$ since:

$$\pi^T \mathbf{P} = [1/3, 1/3, 1/3] \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = [1/3, 1/3, 1/3] = \pi^T$$

However, detailed balance is violated since:

$$\pi(1)P(1,2) = \frac{1}{3} \times 1 = \frac{1}{3}$$

$$\pi(2)P(2,1) = \frac{1}{3} \times 0 = 0$$

Since $\frac{1}{3} \neq 0$, detailed balance does not hold. ■

B.4.1.2 Exercise 2: Microscopic Reversibility and Detailed Balance

Detailed balance represents the concept of microscopic reversibility at equilibrium. In physical chemistry, it asserts that any molecular process and its reverse occur at equal average rates. In Markov chain theory, it guarantees that the probability flux from x to y matches the reverse flux from y to x under the stationary distribution π.

B.4.2 Metropolis-Hastings

B.4.2.1 Exercise 1: Metropolis-Hastings Detailed Balance Proof

The continuous case algebraic proof is detailed in Appendix A.

B.4.2.2 Exercise 2: Show Symmetric Proposal Reduces to Metropolis

If proposal is symmetric: $q(x, y) = q(y, x)$. Hastings' acceptance ratio is:

$$\alpha(x, y) = \min\left(1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right)$$

Since $q(x, y) = q(y, x)$, they cancel out:

$$\alpha(x, y) = \min\left(1, \frac{\pi(y)}{\pi(x)}\right)$$

which is exactly the original Metropolis acceptance probability. ■

B.4.3 Gibbs Sampling

B.4.3.1 Exercise 1: Gibbs Sampler for Simple Linear Regression

For regression model $Y_i \sim N(\beta_0 + \beta_1 X_i, \sigma^2)$, with prior $p(\beta) \sim N(\mu_0, \Sigma_0)$, the full conditionals are conjugate normals:

1. Update $\beta \mid \sigma^2, \mathbf{y} \sim N(\mathbf{m}_n, \mathbf{V}_n)$ where:

$$\mathbf{V}_n = \left(\Sigma_0^{-1} + \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}\right)^{-1}, \quad \mathbf{m}_n = \mathbf{V}_n\left(\Sigma_0^{-1}\mu_0 + \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{y}\right)$$

2. Update $\sigma^2 \mid \beta, \mathbf{y} \sim \text{Inv-Gamma}(a_n, b_n)$ where:

$$a_n = a_0 + n/2, \quad b_n = b_0 + \frac{1}{2}\sum(y_i - \mathbf{x}_i\beta)^2$$

B.4.3.2 Exercise 2: Bivariate Normal Full Conditionals

For $X = (X_1, X_2) \sim N(\mu, \Sigma)$:

$$X_1 \mid X_2 = x_2 \sim N\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(x_2 - \mu_2), \sigma_1^2(1-\rho^2)\right)$$

$$X_2 \mid X_1 = x_1 \sim N\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1 - \mu_1), \sigma_2^2(1-\rho^2)\right)$$

B.4.4 Ising and Potts Models

B.4.4.1 Exercise 1: Potts $q = 2$ is Equivalent to Ising Model

For $q = 2$, Potts Hamiltonian states $\sigma_i \in \{1, 2\}$, and we can map these to Ising spins $s_i \in \{+1, -1\}$ using transformation $s_i = 2\sigma_i - 3$. The Potts interaction term $\delta(\sigma_i, \sigma_j)$ is 1 if spins are equal, 0 if different. The Ising term $s_i s_j$ is 1 if spins are equal, -1 if different. Thus:

$$\delta(\sigma_i, \sigma_j) = \frac{s_i s_j + 1}{2}$$

Substituting this into the Potts Hamiltonian yields the Ising Hamiltonian scaled by a constant factor 1/2 and shifted by a constant energy offset.

B.4.4.2 Exercise 2: Boundary Conditions and Spatial Coalescence

- **Fixed Boundary Conditions**: Force the boundaries to stay at a specific spin (e.g., all +1). This provides external alignment that speeds up coalescence to that specific phase but suppresses natural fluctuations.
- **Periodic Boundary Conditions**: Wrap the grid edges (connecting left-to-right, top-to-bottom). This preserves translation invariance, allows realistic cluster growth, but can slow down coalescence because clusters have no fixed boundaries to seed alignment.

B.4.5 Perfect Sampling (CFTP)

B.4.5.1 Exercise 1: Proof that Monotonicity Guarantees Coalescence

The induction proof showing that the minimum and maximum chains sandwich all intermediate paths is detailed in Appendix A.

B.4.5.2 Exercise 2: Forward-Coupling Bias in CFTP

If we start chains at time 0 and run forward to time T, checking for coalescence at time T, the coalesced state will be biased toward states that have shorter paths to coalescence. This is because the coalescence time T_{coalesce} is dependent on the random increments. In contrast, starting at $-T$ in the past and running to 0 guarantees that the chain has reached the stationary distribution at time 0 regardless of the specific starting time.

B.4.6 Simulated Annealing

B.4.6.1 Exercise 1: Balancing Exploration and Exploitation

- **High Temperature T (Exploration)**: When T is high, the acceptance probability $\alpha = \min(1, e^{-\Delta E/T})$ for uphill moves ($\Delta E > 0$) is close to 1. This allows the chain to make random walks across energy barriers and explore the entire parameter space, preventing early trapping in local minima.
- **Low Temperature T (Exploitation)**: As $T \to 0$, the probability of accepting uphill moves decays to 0. The algorithm behaves like a greedy gradient descent (local exploitation), only accepting moves that decrease the objective function, locking the state into the global minimum.

B.4.6.2 Exercise 2: Simulated Annealing in R for Rastrigin Function

- **R Code Implementation**:

```r
rastrigin_1d <- function(x) x^2 + 10 * (1 - cos(2 * pi * x))

optimize_rastrigin <- function(init = 2.0, t0 = 10, gamma = 0.9, max_iter = 200) {
  curr_x <- init
  curr_e <- rastrigin_1d(curr_x)
  t <- t0

  for (k in 1:max_iter) {
    prop_x <- rnorm(1, mean = curr_x, sd = 0.2)
    prop_e <- rastrigin_1d(prop_x)

    delta_e <- prop_e - curr_e
    alpha <- exp(-delta_e / t)

    if (delta_e < 0 || runif(1) < alpha) {
      curr_x <- prop_x
      curr_e <- prop_e
    }
    t <- t * gamma
  }
  return(curr_x)
}
```

B.5 Chapter 5 Solutions

B.5.1 Sensitivity Analysis & Score Function

B.5.1.1 Exercise 1: Score Function for Gamma Distribution Scale Parameter

Let $f(x;\theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-x/\theta}$. Taking the natural logarithm:

$$\log f(x;\theta) = -\log\Gamma(k) - k\log\theta + (k-1)\log x - \frac{x}{\theta}$$

Differentiating with respect to θ:

$$\frac{\partial}{\partial\theta}\log f(x;\theta) = -\frac{k}{\theta} + \frac{x}{\theta^2} = \frac{x - k\theta}{\theta^2}$$

Thus the score function is $S(X;\theta) = \frac{X-k\theta}{\theta^2}$.

B.5.1.2 Exercise 2: Expected Value of the Score Function

The proof that $E_\theta[S(X;\theta)] = 0$ is detailed in Appendix A.

B.5.1.3 Exercise 3: Manual Walkthrough Sensitivity of $E[X^2]$ for $X \sim \text{Exp}(\lambda)$

We want to estimate the derivative of $E_\lambda[X^2]$ at $\lambda = 2.0$ using uniform draw $u = 0.5$. The inverse transform is $X = -\frac{1}{\lambda}\ln(U)$.

1. **Calculate Sample**:
$$x = -0.5\ln(0.5) \approx 0.3466$$

2. **Find Score Function**: The density is $f(x) = \lambda e^{-\lambda x}$.
$$\log f(x) = \log\lambda - \lambda x \implies \frac{\partial}{\partial\lambda}\log f(x) = \frac{1}{\lambda} - x$$

3. **Evaluate Single Sample Estimate**:
$$\text{Gradient estimate} = x^2\left(\frac{1}{\lambda} - x\right) = (0.3466)^2(0.5 - 0.3466) = 0.1201 \times 0.1534 \approx 0.0184$$

B.5.2 Robbins-Monro Stochastic Approximation

B.5.2.1 Exercise 1: Robbins-Monro R Script for $\theta^3 - 8 = 0$

We find the root of $M(\theta) = \theta^3 - 8 = 0$ (analytical root: 2) with normal noise. We use a scaled step size $a_n = 0.1/n$ to prevent the cubic gradient from causing numerical overflow:

```
robbins_monro_solve_cube <- function(n_iter = 1000, init = 0.0) {
  theta <- init
  for (n in 1:n_iter) {
    y_obs <- theta^3 - 8 + rnorm(1)
    # Using c = 0.1 to maintain numerical stability for cubic gradient
    a_n <- 0.1 / n
    theta <- theta - a_n * y_obs
  }
  return(theta)
}
# Test root finding
robbins_monro_solve_cube()
```

```
## [1] 1.997124
```

B.5.2.2 Exercise 2: Why Step Size $a_n = 1/n^2$ Fails Convergence

For $a_n = 1/n^2$, the second condition $\sum a_n^2 < \infty$ is satisfied. However, the first condition requires $\sum a_n = \infty$. Since $\sum_{n=1}^{\infty}\frac{1}{n^2} = \frac{\pi^2}{6} < \infty$, the total step size budget is bounded. If the initial parameter θ_1 is far from the true root θ^*, the algorithm will run out of step capacity and freeze before reaching the root, causing convergence to fail.

B.5.2.3 Exercise 3: Compare Robbins-Monro step sizes $a_n = 1/n$ vs $a_n = 1/\sqrt{n}$

- $a_n = 1/n$: Converges asymptotically to the root and satisfies both conditions of the Robbins-Monro theorem. It suppresses the noise variance effectively at later steps.
- $a_n = 1/\sqrt{n}$: Fails the second condition because $\sum a_n^2 = \sum \frac{1}{n} = \infty$. This means the steps do not decrease fast enough to suppress noise, and the estimator will continue to oscillate wildly around the root instead of converging.

B.5.3 Rare-Event Simulation

B.5.3.1 Exercise 1: Fixed Factor vs. Fixed Effort Splitting

- **Fixed Factor Splitting**: Each trajectory that crosses a boundary is split into a pre-defined factor r_k of independent paths. The total number of simulated trajectories is random, which can lead to computational "explosions" or "extinction" (no paths crossing).
- **Fixed Effort Splitting**: Dictates simulating a fixed number of paths N at each level, resampling starting points only from successful trajectories of the previous level. This controls computational budget exactly and prevents explosions or extinctions.

B.5.3.2 Exercise 2: Bounded Relative Error in Importance Sampling for $P(Z \geq 5)$

Let $Z \sim N(0,1)$ and target probability be $p = P(Z \geq 5)$. We select a shifted proposal $g(x) \sim N(5,1)$. The likelihood ratio is:

$$w(x) = \frac{f(x)}{g(x)} = \frac{e^{-x^2/2}}{e^{-(x-5)^2/2}} = e^{-5x+12.5}$$

The second moment of the estimator is:

$$E_g[w(X)^2 I(X \geq 5)] = \int_5^\infty e^{-10x+25} \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2}\, dx < \infty$$

Dividing by p^2 yields a bounded coefficient of variation as $p \to 0$, proving bounded relative error. ■

B.5.3.3 Exercise 3: 2-Level Splitting Manual Walkthrough for 3-Step Random Walk

We want to estimate $P(X_3 \geq 3.0)$ with levels $L_1 = 1.5$, $L_2 = 3.0$.

1. **Level 1**: Run $N = 2$ paths of length 3:
   - Path 1: $(0, 1.0, 1.2, 1.4)$ -> Max is 1.4 -> Fail.
   - Path 2: $(0, 0.8, 1.6, 2.2)$ -> Max is 2.2 -> Success! Crossing occurs at $\tau_1 = 2$.
   - Level 1 probability $p_1 = 1/2 = 0.5$.
2. **Level 2**: Resample $N = 2$ starts from Path 2 up to $\tau_1 = 2$: Prefix is $(0, 0.8, 1.6)$.
   - Re-simulate step 3 for path 1: $1.6 + 1.5 = 3.1 \geq 3.0$ -> Success!
   - Re-simulate step 3 for path 2: $1.6 - 0.2 = 1.4 < 3.0$ -> Fail.
   - Level 2 probability $p_2 = 1/2 = 0.5$.
3. **Result**: Estimated probability is $p_1 \times p_2 = 0.25$.

B.5.4 Adaptive MCMC

B.5.4.1 Exercise 1: Constant Adaptation Bias

If the proposal variance adaptation step size does not vanish ($\gamma_t \not\to 0$), the transition kernel remains dependent on the entire history of the chain indefinitely. This violates the Markov property, and the chain does not have a unique stationary distribution, biasing the samples away from the target $\pi(x)$.

B.5.4.2 Exercise 2: Why Optimal Acceptance Rate is 0.234 in High Dimensions

In 1D, the optimal acceptance rate is 0.44 because we can easily make large moves without getting rejected. As the dimension $d \to \infty$, the geometry of high-dimensional space restricts us; to avoid getting trapped or rejected constantly, the optimal step size scales as $O(d^{-1/2})$, which asymptotically yields the optimal Roberts-Gelman-Gilks acceptance rate of 0.234.

Booth, Wayne C, Gregory G Colomb, Joseph M Williams, Joseph Bizup, and William T Fitzgerald. 2016. *The Craft of Research*. 4th ed. University of Chicago Press.

Creswell, John W, and J David Creswell. 2018. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 5th ed. SAGE Publications.

Kothari, C. R., and Gaurav Garg. 2019. *Research Methodology: Methods and Techniques*. Fourth. New Age International (P) Limited.