Monte Carlo Methods

Quiz Study Guide — Chapter 3

Manuel D. Rossetti, PhD P.E.

How to Use This Deck

This deck is a **checklist**, not a summary. Every slide is phrased as something you should be able to *state*, *derive*, or *compute* without looking it up.

- **Question formats:** multiple choice, true/false, matching, fill-in-the-blank, multi-slot fill-in-the-blank, and numeric answers.
- Work the **Drills** slides by hand first; the answers are in the appendix at the end.
- The **Common Traps** slides are built from the actual wrong answers — read them last.

Scope

Monte Carlo methods here mean **static** simulation — time either ticks in a regular pattern or is not a factor at all — and the focus is on estimating expected values of random variables.

What This Chapter Covers

Statistics in the KSL

- The `CollectorIfc` foundation
- `Statistic`, `Histogram`, `IntegerFrequency`, `StateFrequency`
- The method of batch means

Monte Carlo estimation

- Confidence intervals and half-widths
- Sample-size determination
- Monte Carlo integration
- The four application examples
- `MCExperiment` and `MC1DIntegration`

Objectives — Statistics Framework

You should be able to:

- State what `CollectorIfc` defines and what the base `Collector` does with observations
- Name the assumption behind the `Statistic` class's confidence interval
- Explain how to estimate a probability such as $P(X \geq 20)$ in the KSL
- Distinguish `Histogram` from `CachedHistogram`, and `IntegerFrequency` from `StateFrequency`
- State the batch-size formula and the trade-off in choosing the number of batches
- Quote the `BatchStatistic` defaults

Objectives — Estimation

You should be able to:

- Interpret a confidence interval correctly, and say why the common misreading is wrong
- Write the half-width formula and the three sample-size methods
- Set up a Monte Carlo integration estimator for $\theta = \int_a^b g(x)\,dx$
- Describe the craps, news vendor, and stochastic activity network models
- Explain the macro/micro replication structure of `MCExperiment`

Where “Monte Carlo” Comes From

The term originates with the **Monte Carlo casino** in the Principality of Monaco, where games of chance are well known.

For this chapter, Monte Carlo methods are limited to **static simulation**: time either ticks in a regular pattern or is not a factor, and the focus is on estimating **expected values of random variables**.

The Collector Foundation

- **`CollectorIfc`** is the foundation of the KSL statistics framework. It defines `collect()` methods that **observe and tabulate values** presented to it, plus a `reset()` to clear state.
- By default the base **`Collector`** class **summarizes observations without storing them**, while remaining *observable* so other classes can connect to the observation process.
- The **`Statistic`** class uses **one-pass algorithms**, so no observed data need be stored. It implements **both** `CollectorIfc` and `StatisticIfc`.

Confidence Intervals in the KSL

The `Statistic` class computes a confidence interval assuming the observations are **normally distributed, or that the sample size is large enough to invoke the central limit theorem**.

$$h = t_{1-\alpha/2,\, n-1}\,\frac{s}{\sqrt{n}} \qquad \text{(the half-width)}$$

! Interpretation — asked as both MC and true/false

A confidence interval is an assurance about the **procedure**: if many such intervals were computed, approximately $(1-\alpha)100\%$ would contain the true parameter. A *specific computed* interval does **not** have probability $1-\alpha$ of containing μ.

Estimating a Probability

To estimate $P(X \geq 20)$ in the KSL, collect the **boolean expression** `x >= 20.0` with a `Statistic`.

- The boolean is recorded as **1.0 / 0.0**.
- Its **average** then estimates the probability.

A variable that takes the value 1 when an event occurs and 0 otherwise is called an **indicator** variable.

The Statistics Classes

Class	Role
`CollectorIfc`	Defines `collect()` for observing values and `reset()` to clear state
`Statistic`	Mean, variance, CI, skewness, kurtosis on collected values
`Histogram`	Tabulates values into bins and summarizes the binned data
`CachedHistogram`	**Caches observations** until enough are seen to recommend reasonable break points

Frequency and Reporting Classes

Class	Role
`IntegerFrequency`	Counts and proportions of observed **integers**, with an optional range and over/under-flow counts
`StateFrequency`	Visitation and **transition** statistics — useful for Markov chain analysis
`StatisticReporter`	Takes a list of `StatisticIfc` objects and produces nicely formatted summary reports

The Method of Batch Means

With n observations and k batches, the batch size is

$$b = \left\lfloor \frac{n}{k} \right\rfloor$$

- **Larger batches are generally better:** batch means become more nearly independent, because data within a batch is conceptually farther in lag from data in other batches.
- **But** a very large batch size means **fewer batches**, which **increases the variance of the variance estimator**.
- The batch means are treated like a random sample; an approximate CI uses their mean and standard deviation with a t critical value on $k-1$ degrees of freedom.

`BatchStatistic` Defaults

Table 1: Memorize these three numbers

Property	Default
`minNumBatches`	**20**
`minNumObsPerBatch`	**16**
`maxBatchMultiple`	**2**

`reformBatches()` lets the user **rebatch** the data to a user-specified number of batches.

Monte Carlo Integration

To estimate $\theta = \int_a^b g(x)\,dx$, define

$$Y = (b-a)\,g(X), \qquad X \sim U(a,b)$$

Chapter example. For $\theta = \int_1^4 \sqrt{x}\,dx$, generate $X_i \sim U(1,4)$ and form

$$Y_i = 3\sqrt{X_i}$$

The $(b-a)$ factor is the piece students most often drop.

Sample-Size Determination

Table 2: The bound ϵ is called the **margin of error**

Method	Formula
Normal approximation	$n \geq \left(\frac{z_{1-\alpha/2}\, s}{\epsilon}\right)^2$
Student-t iterative	Solve $t_{1-\alpha/2,\, n-1}\, s/\sqrt{n} \leq \epsilon$ **iteratively** — the critical value depends on n
Half-width ratio	$n \geq n_0 \left(\frac{h_0}{h}\right)^2$ from a pilot half-width h_0 on n_0 replications
Proportion	$n = \left(\frac{z_{1-\alpha/2}}{\epsilon}\right)^2 \hat{p}(1-\hat{p})$, worst case $\hat{p} = 0.5$

Notes on Sample Size

- The Student-t bound is **not** a closed-form equation in n: the critical value itself depends on n, so it must be solved **iteratively**.
- The **half-width ratio** method tends to be **conservative —** it often recommends a larger sample size than the normal-approximation method.
- Sample size grows **quadratically** as the desired half-width shrinks: halving h quadruples n.
- For a proportion with no pilot estimate, use the worst case $\hat{p} = 0.5$, which maximizes $\hat{p}(1 - \hat{p})$.
- `Statistic.estimateSampleSize()` is the KSL companion-object function implementing the normal-approximation calculation.

Craps and the News Vendor

Craps. On the first roll the shooter immediately **wins on 7 or 11** and immediately **loses on 2, 3, or 12**. Two dice are modeled with `DUniformRV` over 1..6.

News vendor. Profit for a day given order quantity q and demand D:

$$G(D, q) = s\,\min(D, q) + u\,\max(0,\, q - D) - c\,q$$

with $s > c > u$: s is the selling price, c the unit cost, and u the **salvage** value. Demand is modeled with `DEmpiricalRV`.

The Stochastic Activity Network

- Activity durations are modeled with the **triangular** distribution, parameterized by minimum, mode, and maximum (`TriangularRV`, one per activity).
- The time to complete a single realization is the **maximum among the realized path lengths** — the longest path.
- The longest path through the network is the **critical** path.
- The chapter's network enumerates **four** unique paths.

`MCExperiment`

The KSL conceptualizes a Monte Carlo experiment as a **two-loop** process:

- **Macro replications** (outer loop) — their sample averages are tested against the half-width criterion.
- **Micro replications** (inner loop) — many are averaged to form one macro observation, so by the CLT the macro observations are approximately normal.
- Total observations $= n \times r$ (macro $\times$ micro).

Defaults: initial sample size $k = 30$, max macro replications $M = 10{,}000$, micro replications $r = 100$, desired half-width $= 0.001$. So the convention is a **large** number of macro replications, not a small one.

Two More KSL Names

- **`MCReplicationIfc`** — the Kotlin functional interface whose single abstract method `replication(j: Int): Double` defines **one observation** of a Monte Carlo experiment.
- **`MC1DIntegration`** — a subclass of `MCExperiment` that evaluates one-dimensional integrals, accepting a function $g(x)$ and a sampling distribution $f(x)$.

Drills

1. Normal approximation: a pilot run yields $s = 6$ and you want a 99% CI with $\epsilon = 0.1$. With $z_{0.995} \approx 2.5758$, find n (round up).
2. News vendor with $q = 30$, $s = 0.25$, $c = 0.15$, $u = 0.02$. Compute $G(50, 30)$ and $G(10, 30)$ in dollars.
3. Compute the true value of $\int_1^4 \sqrt{x}\, dx$ to three decimals.
4. Half-width ratio: $n_0 = 20$ replications give $h_0 = 2.2248$. Find n for a desired $h = 0.5$.
5. Half-width ratio: $n_0 = 10$ gives $h_0 = 9.39$ minutes. Find n for $h = 2$ minutes.
6. Proportion sample size with $\hat{p} = 0.8$, $z = 1.96$, $\epsilon = 0.05$. Then repeat with the worst-case $\hat{p}$.
7. Probability of rolling a 7 with two fair dice, to four decimals.

Common Traps

- Batch size is $b = \lfloor n/k \rfloor$ — **floor** of n over k.
- Larger batches improve **independence** but shrink k, which **inflates the variance of the variance estimator**. Both halves of the trade-off get asked.
- A *specific* computed CI does **not** have probability $1 - \alpha$ of covering μ.
- The Student-t half-width bound is **not** closed form in n — it must be solved iteratively.
- The half-width ratio method is **conservative**, not anti-conservative.
- Monte Carlo integration needs the $(b - a)$ multiplier: $Y = (b - a)g(X)$, not $g(X)$.
- `MCExperiment` defaults to **many macro** replications ($M = 10{,}000$), not few.
- The SAN completion time is the **maximum** path length, not the sum of all activities.

Appendix — Drill Answers (1 of 2)

1. $n \geq (2.5758 \times 6/0.1)^2 = (154.548)^2 = 23885.1 \Rightarrow \mathbf{23886}$
2. $G(50, 30) = 0.25(30) + 0.02(0) - 0.15(30) = 7.5 - 4.5 = \$\mathbf{3.00}$.
   $G(10, 30) = 0.25(10) + 0.02(20) - 0.15(30) = 2.5 + 0.4 - 4.5 = -\$\mathbf{1.60}$
3. $\int_1^4 \sqrt{x}\, dx = \frac{2}{3} x^{3/2} \Big|_1^4 = \frac{2}{3}(8 - 1) = 14/3 = \mathbf{4.667}$
4. $n \geq 20\,(2.2248/0.5)^2 = 20(19.799) = 395.98 \Rightarrow \mathbf{396}$

Appendix — Drill Answers (2 of 2)

5. $n \geq 10\,(9.39/2)^2 = 10(22.03) = 220.3 \Rightarrow \mathbf{221}$
6. $\hat{p} = 0.8$: $n = (1.96/0.05)^2(0.8)(0.2) = 1536.64(0.16) = 245.9 \Rightarrow \mathbf{246}$. Worst case $\hat{p} = 0.5$: $n = 1536.64(0.25) = 384.2 \Rightarrow \mathbf{385}$
7. Six of the 36 equally likely outcomes sum to 7, so $6/36 = \mathbf{0.1667}$