Appendix C

AI Prompting Templates for Analytics

Dr. Jose Mendoza, Academic Director and Clinical Associate Professor
Version 1.0 · August 2026

Appendix Information

PURPOSE

This appendix collects the prompting templates the guide uses for recurring analytics tasks. Its purpose is not to help you obtain polished output quickly; it is to help you specify a task precisely enough that the output can be predicted, tested, explained, and documented. Each entry is a copy-and-adapt structure that makes the analyst state the specification, the grain, the constraints, and the expected output before an assistant produces anything, and that requires the assistant to expose missing information rather than fill it.

VERSION AND DATE

Version 1.0 · August 2026 · Language: English (United States)

SUGGESTED CITATION

Mendoza, J. (2026). AI prompting templates for analytics. In *Applied business analytics for marketing decision-making: Business analytics and data visualization* (Appendix C, Version 1.0) [Open educational resource]. CC BY 4.0.

LICENSE AND RIGHTS

GENERATIVE AI USE

Generative artificial intelligence and other AI-assisted tools were used in the research, writing, revision, and production of this appendix, including outlining, preliminary drafts, revision of prose, and document formatting. These tools were used under the author’s direction and are not credited as authors, researchers, or sources. The author determined the appendix’s scope, boundaries, and content, and reviewed and approved all AI-assisted material: factual claims and citations were checked against the underlying sources rather than accepted from AI-generated summaries, and every numerical output used in the companion worked examples was generated from the underlying code and checked against a stated

expectation rather than entered manually. Responsibility for the accuracy, originality, and final form of this appendix rests entirely with the author. A fuller statement appears in the front matter of the complete guide.

COMPANION FILES

An editable Markdown and Word template library, a plain-text version optimized for copying, a chapter-indexed prompt finder, and one completed worked example per major workflow are in the Applied Business Analytics companion repository.

How to Read a Template

The appendix contains twelve numbered sections. Sections C.1 through C.4 are the apparatus every template assumes — the division of labor, the prompt anatomy, the audit stack, and the privacy rule. Sections C.5 through C.10 hold the library. Sections C.11 and C.12 hold the anti-patterns and the index.

Every entry opens with an **Owned by** line naming the chapter and section it is drawn from, and then runs through the same six working parts, in the same order. The parts are labeled in bold rather than given headings, so that an entry stays on one or two pages and can be scanned while you work. Table C.1 states what the metadata line and each of the six parts give you.

Two labels carry numbers. **Template C.1** through **Template C.19** identify the entries themselves and are the labels a chapter cites. **Prompt C.1** onward identify the individual copy-and-adapt blocks, which run continuously through the appendix because several templates need more than one prompt. Sections are numbered separately, C.1 through C.12, so Section C.7 is where Template C.9 lives. Tables follow the same pattern: Table C.1, Table C.2, and so on.

Text inside square brackets is a placeholder you replace. Everything outside the brackets is meant to be sent as written, including the sentences that prohibit something — those sentences are load-bearing, and Section C.2 explains why. The bracketed convention is the chapters' own:

it first appears in Section 4.10's `[one cleaning step]` and Section 12.12's `[paste the schema]`.

Prompts are set in a monospaced block so that line breaks survive copying. Nothing in a prompt block is code; it is text to paste into an assistant.

Table C.1

The metadata line and the six working parts of a template entry

Part	What it gives you
Owned by (metadata)	The chapter and section the template is drawn from, and how closely its prompt follows the chapter's wording. It tells you which chapter to return to when a judgment call arises.
Use this template when	The task and the chapter that owns it. If your task is not the one described, the audit at the end will not fit it.
What you must decide first	The decisions that must exist before the prompt is sent. An assistant asked to make these decisions will make them, silently, and the template will have failed.
Copy-and-adapt template	The prompt, with bracketed placeholders. Send it as written apart from the placeholders.
Likely failure modes	What the assistant is most likely to get wrong on this task, drawn from the owning chapter's named failure modes. Read this before reading the response.
Audit after the response	The check that must run, and the table that governs it. Section C.3 maps every audit instrument in the guide.
What to record in Appendix D	What this template adds to the standard Appendix D record of Section C.3. Documentation is part of the task, not a step after it.

C.1 How This Library Works

The guide gives an AI assistant a specific and limited job. It may draft, propose, explain, debug, enumerate, or critique. It may not decide what the analysis is, what the data mean, which definition governs, which threshold applies, or what should be recommended. Every chapter from Chapter 2 onward states some version of that division, and every template in this appendix is built to enforce it at the point where it is most often lost, which is the moment the prompt is written.

The division matters because assistant failures in analytics are rarely obvious. A leaky pipeline does not usually look different from an honest one; it looks better. A summary computed on the wrong denominator arrives as a tidy table. A cluster solution fitted once and never refit reads as structure. Fluency is not reliable evidence of correctness, and the general term for confident, plausible, wrong output is *confabulation* (National Institute of Standards and Technology, 2024). Capability is also uneven in ways that are invisible from the outside, so

strong performance on one task does not guarantee strong performance on an adjacent one (Dell'Acqua et al., 2023). Table C.2 records the division of labor the guide applies.

Table C.2

What may be delegated and what may not

The assistant may	The analyst must	Stated in
Draft code from written derivation rules and window boundaries	Write the derivation rules and the window boundaries	Sections 4.10, 6.12, 8.11
Enumerate candidates — questions, confounders, features, chart forms, titles	Choose among them, and treat every proposal as a candidate for the specification rather than as analysis	Sections 5.10, 6.12, 7.11, 12.12, 13.11
Compute what is fully specified — power, break-even, sensitivity tables, reconciliations	Declare the inputs before the computation, and never after seeing a result	Sections 9.9, 11.7, 11.11
Explain an error, a method, or an inherited cell	Understand the explanation well enough to defend it in the meeting that funds the work	Sections 1.9, 10.11
Argue against a finished result	Decide which objections survive, and record why the rest were discarded	Sections 2.8, 11.11, 12.12, 13.11
Draft prose from verified numbers	Own every verb and every number in the sentence, and sign it	Sections 7.11, 12.12, 13.11
Draft a candidate specification, definition, registration, or view specification from written inputs	Choose among the candidates, supply every rule the draft could not know, and own the version that governs the work	Sections 2.8, 3.9, 11.11, 12.12
Nothing that has not been framed as a draft or a candidate for your approval	Define the metric, choose the threshold, define the label, name the segment, set the axis range, and take responsibility for the recommendation	Sections 3.9, 6.10, 9.9, 12.12

The last two rows are read together and are easy to misread apart. Several templates in this appendix do ask an assistant to draft a specification, a metric definition, an experiment registration, or a set of view specifications, and each of the owning chapters authorizes it. What the final row forbids is the same act performed without a candidate frame around it. A drafted metric definition is a proposal the analyst selects from and signs; a metric definition the assistant chose silently is the failure Section 3.9 names. The difference is not the output. It is whether the prompt made the choice visible and whether the analyst made it.

C.1.1 The Two-Prompt Pattern

The pattern emerges in Chapter 6 and becomes standard from Chapter 7 onward, and the templates in Sections C.7 and C.8 all follow it. The first prompt asks for mechanics only and ends by forbidding narration. The analyst then runs the audit. Only when the work survives does a second prompt ask for prose, and that prompt names the verbs the evidence licenses and forbids the ones it does not. Earlier templates use a single prompt, because the tasks they serve do not separate production from interpretation in the same way.

The order is the control. A response that arrives with its own interpretation attached is much harder to audit, because the interpretation tells you what to see before you have looked. Chapter 12 makes the same point about charts: a rendered chart is persuasive in a way its specification is not.

CONCEPT

Mechanics First, Audit Between, Narration Last

The two-prompt pattern separates two things an assistant will otherwise do in one breath: producing an artifact and telling you what it means. Prompt one ends with a phrase equivalent to *narrate nothing yet*. The audit runs on the artifact alone. Prompt two receives only the audited numbers and is told which verbs the evidence licenses.

Chapters 12 and 13 extend the pattern to five steps, because a chart and a sentence are produced by the same act that makes their claim. The five steps are: request, written prediction, build, audit and grade, then repair — or, in Chapter 13, the reader. Templates C.16 through C.18 carry that longer form.

Source: Course concept developed for this guide, consolidating Sections 7.11, 8.11, 9.11, 10.11, 11.11, 12.12, and 13.11.

C.1.2 Predict Before You Send

Every template assumes the prediction discipline of Section 1.7. Before the response arrives, write down what you expect: the row count, the total, the sign, the plausible band, the number of candidates that will overclaim. The prediction is what converts reading a response into auditing one, and the gap between predicted and actual is the finding.

Three chapters make the prediction an explicit step rather than an assumption. Section 10.11 requires the rolling-origin error to be predicted before the pipeline runs, because an analyst who has computed the baselines already knows the plausible band and a result outside it is a finding about the pipeline rather than about the future. Section 11.11 requires the interval width to be predicted from the arm sizes, because an interval far narrower than predicted means the analysis grain slipped. Section 13.11 requires a prediction of how many drafted titles will overclaim, and in which of three shapes.

C.1.3 The Templates Are Modular

No template is a script. Each is a structure to adapt, and adapting it well means deleting the parts that do not apply to your task and adding the constraints your task needs.

Three controls recur throughout the library, although their wording changes by task: **use only supplied information**, **expose rather than fill missing information**, and **stop at the boundary of the delegated work**. The second appears as the literal word `UNSPECIFIED` where a template asks the assistant to complete a form, and as task-specific language elsewhere — *say which field is missing and stop*, *name any feature you cannot trace*, *say what else you need rather than guessing*. Whatever the wording, do not delete the control; if a template's phrasing does not fit your task, replace it with phrasing that does.

C.2 The Common Prompt Anatomy

Every template in this appendix is assembled from the same twelve components. Not all twelve appear in every prompt — a debugging request does not need a decision, and a title request does not need a split protocol — but a prompt that omits a component it needs is a prompt that has delegated that component silently. Table C.3 lists them.

Table C.3

The twelve components of an analytics prompt

Component	What you supply
Decision	The decision or deliverable the analysis supports. Without it the assistant optimizes for an interesting answer rather than a usable one.
Task	One bounded action. Two actions in one prompt produce an artifact whose faults cannot be attributed.
Data context	Table names, schema, derivation rules, definitions, certified totals, and a safe sample. Everything the assistant is allowed to know.
Unit and grain	What one input row represents and what one output row must represent. The single most productive line in most prompts.
Method	The method already chosen, or the alternatives to be compared. Never left open unless comparison is the task.
Constraints	Columns allowed, exclusions, reference categories, seeds, privacy limits, and course rules.
Expected output	The required format: a table, a code cell, a numbered list, twelve sentences. Stated as a shape, not as a wish.
Prediction	What the assistant must state before producing the artifact — row-count effects, totals, output grain, expected cardinality, evaluation figures.

Component	What you supply
Verification	The checks the output must print or explain, and the totals it must reconcile to.
Uncertainty	The instruction to write UNSPECIFIED for anything the supplied information does not determine, and to ask rather than fill.
Terminal prohibition	The clause that stops the assistant at the boundary of the task: narrate nothing yet; do not interpret; do not render conclusions; choose no threshold yourself.
Documentation	The reminder that the exchange is recorded under Appendix D, including the output you rejected.

Two of the twelve deserve their own note, because they are the ones students omit and the ones that do most of the work.

The **uncertainty** clause is the defense against invention. An assistant asked to fill a template will fill it; asked to define a metric with no context, it will produce a definition and will not say that it chose one; asked to design an experiment from an incomplete brief, it will supply a duration, a unit of randomization, and a treatment dose, silently and plausibly, so that the result reads like a plan you made. The clause that defeats this is short, and every template carries a version of it: *use only the fields and facts supplied below; where the information is insufficient, write UNSPECIFIED and ask me, and do not fill it yourself.* Every template that asks the assistant to complete a structure from supplied information carries a version of it; the templates that ask for a critique or a diagnosis carry the task-specific equivalent instead.

The **terminal prohibition** is the defense against scope drift. It is the sentence that keeps prompt one from becoming the whole analysis. The chapters supply the wording, and the wording is task-specific: *change nothing in the specification; narrate nothing yet* in Sections 7.11 and 8.11; *do not touch the test set, output scores not verdicts, choose no threshold yourself* in Section 9.11; *do not interpret* in Section 11.11; *do not render or describe conclusions* in Section 12.12.

CONCEPT

UNSPECIFIED Is the Load-Bearing Clause

The count of UNSPECIFIED rows a response returns is a measurement of your own prompt, not of the assistant. A registration draft that comes back with eight unspecified rows has told you that your brief was missing eight decisions, all of which someone would otherwise have made for you.

A response with no unspecified rows against an incomplete brief has not answered your question. It has answered a question it invented, and the invention is invisible because it is written in the same voice as everything else.

Source: Course concept developed for this guide, informed by Section 11.11.

C.3 The Audit Stack

A prompt is only half of a delegation. The other half is the audit that runs on the response, and the guide builds its audits cumulatively rather than replacing one with the next. Chapter 8 establishes a five-point audit — frame, leakage, split hygiene, baseline, and error in decision units — and every later chapter adds a supplement that runs with it rather than instead of it. Two chapters happen to label their supplements S1 through S4 — Table 9.5 for classification and Table 10.5 for forecasting — so cite the table as well as the point. Table C.4 maps every task in this appendix to the instrument that governs it.

Two entries in the table need a caution. In Chapter 12 only three of the five points transfer: *frame* becomes the question the view was built to answer, *baseline* becomes the reference the reader is meant to read values against, and *error in decision units* becomes whether the quantities are in units a decision maker can act on, while *leakage* and *split hygiene* have no visual analogue and are marked not applicable rather than skipped. In Chapter 11, *baseline* reads as the control arm and the break-even line.

Table C.4

Which audit governs which task

Task	Audit instrument	Defined in
Any analytic specification	The ten components, each present and none reconstructed after the results	Table 2.5
A data dictionary or metric definition	Allowed values checked against business rules rather than observed values; measurement levels corrected; the four elements plus grain and unit	Sections 3.5, 3.9; Appendix B
A cleaning step	The verification log: predicted and actual row count, distinct keys, totals, and missing counts, with every mismatch explained	Section 4.2.1
A descriptive table	Three moves — recompute one cell by hand, check printed totals against the log, strike every causal verb	Section 5.10
A segmentation	Four moves — verify one centroid by hand, read the stability indices, check the holdout profile against written predictions, strike unredeemable nouns from every name	Section 6.12
A regression	The four-point audit — signs, magnitudes, units, diagnostics	Table 7.5
A predictive pipeline	The five-point audit — frame, leakage, split hygiene, baseline, error in decision units	Table 8.5

Task	Audit instrument	Defined in
A classifier	Table 8.5 plus Table 9.5's S1–S4 — base rate and floor, matrix arithmetic, threshold provenance, calibration and selection hygiene	Tables 8.5 and 9.5
A forecast	Table 8.5 plus Table 10.5's S1–S4 — time-aware evaluation, selection separated from the grade, calendar and breaks declared, metric and interval honesty	Tables 8.5 and 10.5
An experiment	Table 8.5 plus Table 11.7's E1–E3 — registration precedes analysis, assignment integrity, inference honesty	Tables 8.5 and 11.7
A chart or Tableau view	The three transferable points of Table 8.5 plus Table 12.5's V1–V3 — grain and aggregation, scale and baseline honesty, encoding fits the purpose	Tables 8.5 and 12.5; Appendix E
A dashboard, title, or memo	Tables 8.5 and 12.5 plus Table 13.6's C1–C3 — every sentence is earned, the reader's takeaway matches the intended one, what was removed is disclosed	Tables 8.5, 12.5, and 13.6; Appendix E

C.3.1 The Record Every Exchange Leaves

Documentation is the fourth discipline of Section 1.7, and it is part of the task rather than a step after it. Every exchange run from this appendix leaves the same standard record, stated here once so that nineteen entries need not repeat it.

The standard record. The prompt as sent; the response disposition, meaning accepted, modified, or rejected; the specific error or limitation you found; the prediction you wrote before the check and the verification you then performed; and the change the exchange produced in your final work. Appendix D is the form this record is submitted on, and Table 1.6 is its compact notebook version.

Each entry's closing line names only what that template adds to the standard record — the leaderboard, the feature trace, the counted UNSPECIFIED rows, the clause-by-clause diff. Where an entry adds nothing, the standard record is the whole obligation.

One general check runs on every response in this appendix and belongs to no chapter in particular. It is Section 1.9's four questions, asked in order: does the output answer the question I specified; does one row still represent what I said it represents; can I reproduce at least one number by hand; can I state in one sentence what this result does not show. If any answer is no, the work is not ready to leave your desk.

C.4 Privacy and Data Handling

Every template in this appendix assumes that what you paste into an assistant is safe to paste. Deciding that is a measurement-custody decision rather than a convenience, and it is the analyst's, not the tool's. Table C.5 states what may be supplied.

Table C.5

What may and may not be supplied to an external AI assistant

May be supplied	Must not be supplied
Schemas, column names, data types, and levels of measurement	Customer records, order records, or any row describing an identifiable person
Derivation rules, metric definitions, and window boundaries	Names, email addresses, phone numbers, addresses, or account identifiers
Certified aggregate totals — row counts, distinct keys, revenue sums	Proprietary, contractual, or otherwise restricted institutional data
Course-provided synthetic StyleCraft files and the small practice files	Credentials, API keys, tokens, connection strings, or file paths that contain any of these
Small synthetic samples you constructed yourself	Confidential employer data, in any volume, absent an approved enterprise environment
Error messages and tracebacks, with paths and identifiers removed	Anything you are not authorized to disclose outside the approved course or organizational environment

The synthetic StyleCraft files are safe by construction; that is what they are for. Where a template asks you to paste sample rows, paste them from the course files. In professional work, the equivalent move is a schema plus a synthetic sample, or an approved enterprise tool, and the choice is recorded in the data and privacy declaration of Appendix D. Section 1.13 owns this rule and Section 3.13 extends it to measurement custody; this appendix only enforces it.

One further habit is worth adopting early. Record, at the time of the exchange, exactly what you supplied — not a later reconstruction. The declaration is easy to write honestly on the day and impossible to write honestly a week afterward.

C.5 Specification and Measurement Templates

The four templates in this section serve the framing work of Chapters 2 and 3. They are the templates most often skipped and the ones whose omission is most expensive, because a specification error propagates through everything built on it and is not detectable in any later output.

Template C.1 — Turn a Business Request into an Analytic Specification

Owned by. Chapter 2, Section 2.5. The ten components are Table 2.5's.

Use this template when. A stakeholder has made a request in business language and you need a first draft of the specification that will govern the work. The assistant is a brainstorming partner here, widening your view of the problem; it is not the author of the specification.

What you must decide first.

- The request as it was actually made, quoted rather than paraphrased. A paraphrase has already done half the framing.
- Which facts about the situation are established and which are your assumptions. Supply only the first.
- Whether you are willing to accept the request's framing. An assistant will usually accept the framing it is given, and asked to outline an analysis showing that a decision is working, it will comply.

Prompt C.1. Draft an analytic specification from a business request

```
CONTEXT
Business request, verbatim: [PASTE THE REQUEST AS IT WAS MADE]
Who made it: [ROLE]  Deadline stated or implied: [DATE OR NONE]
Established facts: [TWO OR THREE SENTENCES OF FACT ONLY]
Tables available, with grain: [TABLE = ONE ROW EQUALS ONE ...]
Fields available: [PASTE THE SCHEMA]

TASK
Draft a first version of an analytic specification with exactly these ten
components, one short paragraph each, and nothing else:
  1  Decision                     what choice will this analysis inform?
  2  Decision-maker and deadline  who decides, and by when?
  3  Analytical questions         what must be measured or compared?
  4  Unit of analysis             what does one row represent?
  5  Data                         what data, from where, what period?
  6  Candidate methods            what fits the questions and the deadline?
  7  Deliverable                  what will the decision-maker receive?
  8  Verification plan            how will the output be checked?
  9  Limitations and risks        what can this analysis not establish?
 10  AI-use plan                  where will AI assist, and how checked?

CONSTRAINTS
Use only the fields and facts supplied above.
Where the request does not determine a component, write UNSPECIFIED and
state the question you would need answered; do not fill it yourself.
Identify ambiguity. Do not resolve it by assumption.
Do not propose findings. Do not state what you expect the answer to be.
End with a numbered list of open questions for the requester.
```

Likely failure modes.

- Generic fluency — questions that would fit any retailer rather than this one, and no organizational constraint that no dataset records.
- Framing amplification — the plan advocates for the answer the request implies rather than testing it. Template C.2 exists for this reason and should be run on the result.
- A verification plan written as a sentiment rather than as checks with numbers attached.
- An AI-use plan that says AI will be used without saying how its output will be checked.

Audit after the response.

- Every one of the ten components present, and none of them reconstructed after the fact.
- The unit of analysis stated as *one row equals one what*, not as a topic.

- The verification plan naming quantities that can fail: row counts against a known roster, rate bounds, totals reconciled to a certified source, one row hand-checked.
- Every UNSPECIFIED row treated as a question for the requester, not as a gap for you to close quietly.

What to record in Appendix D. Beyond the standard record of Section C.3: the request as pasted, the assistant's draft, which components you rewrote entirely, which UNSPECIFIED rows you resolved and with whose answer.

Template C.2 — Review an Existing Analytic Specification Adversarially

Owned by. Chapter 2, Section 2.8. The request sentence is that section's, verbatim; the two closing controls are this appendix's.

Use this template when. You have a specification you believe in and want it attacked before it becomes expensive. Run this on your own work, including work you drafted with Template C.1.

What you must decide first.

- That you want a critique and not a rewrite. A rewritten plan transfers ownership of the framing to the assistant, which is the thing this template exists to prevent.
- What you expect the analysis to show. The prompt asks for an alternative explanation of that expected result, which requires you to have stated it.

Prompt C.2. Adversarial review of a specification

```
Here is my analysis plan.

[PASTE THE COMPLETED SPECIFICATION]

Here is what I expect the analysis to show: [ONE SENTENCE]

Act as a skeptical reviewer. Identify the weakest analytical question, one
stakeholder I have ignored, one alternative explanation for the result I
expect, and one verification check I have not planned.

Answer as a numbered critique. Do not rewrite the plan.
For each point, state what would have to be true for the point to matter.
Do not raise an objection that would apply equally to any analysis plan.
```

Likely failure modes.

- Generic objections that would apply to any plan — add more data, consider seasonality — and that cost nothing to raise.
- A rewrite disguised as a critique, in which the assistant supplies an improved plan rather than naming defects in yours.
- Objections that are unfalsifiable, so that acting on them cannot be evaluated.

Audit after the response.

- Take the critique point by point. Revise on at most two points, and reject at least one with a one-sentence justification. The justification is the graded artifact in Exercise 2.7 and the useful one in practice.
- For each point you accepted, state what would have to be true for it to matter, and whether it is true here.

What to record in Appendix D. Beyond the standard record of Section C.3: the specification version you submitted, the critique in full, which two points you acted on, which point you rejected and why.

Template C.3 — Draft or Audit a Data Dictionary Entry

Owned by. Chapter 3, Section 3.9. The instruction sentence is that section's, verbatim; the framing and the closing controls are generalized from its surrounding prose.

Use this template when. You need the tedious first draft of documentation for a table you are about to work with. This is close to the ideal delegation: the assistant is fast at the draft and structurally unable to supply the part that matters.

What you must decide first.

- Which sample rows you will paste, and that they are safe to paste. Use the course synthetic files; see Section C.4.
- Who owns the business rules for these columns. The draft has to be audited against them, and if nobody owns them, that is the finding.
- The table's grain, in your own words, before you ask.

Prompt C.3. Draft a data dictionary from a sample

```
Below is a small sample of rows from [TABLE NAME], which is course-provided
synthetic data. The table's grain is [ONE ROW EQUALS ONE ...].

[PASTE 5 TO 10 ROWS]

Draft a data dictionary entry for each column: definition, type, level of
measurement, allowed values, and one plausible known issue. Mark any column
where you are guessing.

For each column also state the table and grain it belongs to, the source
system or derivation rule if one is inferable, and what a missing value
would mean.

Where the sample cannot tell you the answer, write UNSPECIFIED rather than
inferring an allowed range from the values you can see. Do not treat the
observed values as the permitted values.
```

Likely failure modes.

- Inference from a sample presented as knowledge of a system. An assistant seeing discount values between 0 and 0.4 will document the allowed range as 0 to 0.4 when the business rule permits 0.8.

- Inherited silences. Nothing in a file reveals that a null identifier comes in two kinds, so no assistant will document it unprompted.
- Stored data type reported as level of measurement, which converts an identifier into a quantity and is among the most common silent errors in AI-generated analysis code.
- Generic known-issue guesses that read as diligence and contain nothing.

Audit after the response.

- Verify allowed values against business rules rather than observed values.
- Correct the measurement levels; the stored type is not the level.
- Add the issues only humans know, and mark the entry incomplete until a source owner has seen it.
- Predict the type and level of every column before you read the draft, and grade the draft against your predictions rather than the reverse.

What to record in Appendix D. Beyond the standard record of Section C.3: the sample you pasted and its source, the draft, at least one allowed-values range the assistant inferred from the sample rather than from a rule, at least one known issue it could not have known.

Template C.4 — Draft Candidate Metric Definitions for Analyst Selection

Owned by. Chapter 3, Section 3.5, using the entry schema of Appendix B. This entry implements the blueprint's *Define a Metric Completely* with the selection step made explicit: the assistant drafts candidates, the analyst defines the metric.

Use this template when. A metric is being used in your organization or your project and you cannot find a written definition of it. This template produces *candidate* definitions in the guide's canonical fields, which you select among and then take to whoever owns the number. Section 3.9 owns the reason the plural matters: an assistant asked to define a metric will produce a definition — some definition — and a definition that arrives fluently is easily mistaken for the organization's.

What you must decide first.

- Which decision the metric supports. A metric with no decision behind it cannot be evaluated, only computed.
- Whether Appendix B already carries the metric. If it does, use that entry; do not generate a competing definition.
- The window boundary convention you intend to use, inclusive or exclusive on each end. The assistant will pick one silently otherwise.

Prompt C.4. Define a metric in the canonical fields

```
Metric to define: [NAME AS IT IS USED IN THE ORGANIZATION]
Decision it supports: [DECISION]
Tables and fields available, with grain: [PASTE SCHEMA]
Window boundary convention: [INCLUSIVE / EXCLUSIVE ON EACH END]

Return the definition in exactly these fields, one line each, and nothing
else:
  Metric name          Business question     Formula in plain language
  Numerator            Denominator           Window, with its boundary rule
  Filters              Grain                 Unit
  Required fields      Interpretation        Common variants
  Verification

RULES
Use only the fields listed above. If the metric cannot be computed from
them, say which field is missing and stop.
Where more than one numerator, denominator, filter, window boundary, or
eligibility rule is defensible, return the alternatives separately under
Common variants. Do not select among them, and do not present one as the
definition.
Mark any element not determined by the business rules I supplied as
UNSPECIFIED rather than inferring it.
State whether the result is a proportion, a percentage, a ratio, a count, or
a currency amount.
Give a verification an analyst can perform by hand on ten rows.
Do not compute the metric. Do not interpret a value.
```

Likely failure modes.

- Fluent completion of ambiguity. Asked to define a metric with no context, an assistant produces a definition — some definition — and the fluency of its arrival is easily mistaken for authority.
- A denominator that conceals a grain: a rate that looks defined but is computed over order lines when the claim is about orders.
- A single definition offered where practice legitimately varies, with the choice unmentioned. This is the failure the template exists to prevent, and the reason it asks for candidates rather than an answer.
- A verification that restates the formula rather than testing it.

Audit after the response.

- Select among the variants yourself, in writing, and record the reason. A template that returns one variant where two are defensible has already failed.
- Check each of the five elements — numerator, denominator, window, filters, grain — against the decision the metric serves.
- Ask whether the assistant silently changed eligibility or mixed grains between the numerator and the denominator.
- Compute the metric by hand on ten rows and confirm the stated verification catches a deliberately wrong denominator.

- Reconcile the result against Appendix B if an entry exists, and record any divergence as a definitional finding rather than resolving it privately.

What to record in Appendix D. Beyond the standard record of Section C.3: the metric name as it was used in the request, the returned definition, the element you had to correct, whether the assistant disclosed choosing among variants.

C.6 Preparation and Description Templates

The four templates in this section serve Chapters 4 and 5, where the assistant is a competent junior pair of hands and the analyst is accountable for what the code did to the data. The division is sharp: in Chapter 4 the assistant drafts cleaning code and the analyst audits the counts; in Chapter 5 the assistant drafts summaries and the analyst audits the claims.

Template C.5 — Draft One Data-Cleaning Step

Owned by. Chapter 4, Section 4.10. The instruction block is that section's, verbatim, with placeholders in place of its example; the constraints below it are generalized from the same section.

Use this template when. You have decided what a cleaning step should do and want the code written. One step at a time; a prompt that asks for a pipeline gets a pipeline whose faults cannot be attributed to any step.

What you must decide first.

- Which remedy the step applies, by name. Drop, impute, or flag is an analytic decision with a bias attached, and Chapter 4 owns it. An assistant asked to "handle" missing values will default to one and not say so.
- The baseline quantities from your verification log: row count, distinct orders and customers, revenue total, missing counts in the affected columns.
- Your own prediction of what the step will change, written before the response arrives.

Prompt C.5. Draft one cleaning step with its predicted effects

```
DATA DICTIONARY ENTRIES for the affected columns:
[PASTE]

CERTIFIED BASELINE from my verification log:
  rows [N]   distinct orders [N]   distinct customers [N]
  revenue [$N]   missing values in the affected columns [N]

SAMPLE ROWS (course-provided synthetic data):
[PASTE 5 TO 10 ROWS]

Write pandas code to [ONE CLEANING STEP]. Before the code, state which
quantities the step should change — columns, rows affected, row count,
revenue total — and what the code deliberately leaves alone. Where the
baseline above determines an exact expected value, state it. Where it does
not, write UNSPECIFIED and include code that computes the pre-step figure
before modifying anything. If the step is a join, state the expected
cardinality and enforce it in the code. List any rows it would drop and why.

CONSTRAINTS
Do not perform any other cleaning step.
Do not modify the raw frame in place; assign to a new name.
Do not choose a missing-value remedy. I have chosen [DROP | IMPUTE | FLAG]
and you are implementing it.
Print the row count, the revenue total, and the affected missing-value
counts before and after.
Use only the columns described in the dictionary entries above.
```

Likely failure modes.

- Silent row drops. Coercion followed by a drop, a filter written wider than intended, or a grouping key containing nulls will shrink the file, and nothing announces it.
- Wrong join keys, wrong key types, or a dimension table that is not unique on the key — producing a result that runs, returns rows, and is quietly wrong.
- Unstated imputation defaults, usually mean or mode imputation, importing a bias without naming the choice.
- Editing the raw frame in place, which destroys the ability to re-run the log from its baseline.

Audit after the response.

- Compare every predicted quantity against the actual and explain every mismatch before continuing.
- If the assistant neither states the expected row-count effect nor explains why it is not determinable from the information you supplied, the audit has failed and the code should not be run. An honest `UNSPECIFIED` with a pre-step count in the code is a pass; silence is not.
- For a join, confirm the cardinality was both stated and enforced, and read the matched and unmatched counts.
- Record the step in the verification log whether it passed or not. A step that behaved unexpectedly and was explained is a log entry, not a failure.

What to record in Appendix D. Beyond the standard record of Section C.3: the step in one sentence, the predicted and actual quantities, the mismatch you investigated, whether the assistant disclosed its own remedy choice.

Template C.6 — Diagnose a Python Error

Owned by. This appendix. No chapter prints a debugging prompt; Section 1.9 names debugging as a legitimate use and Appendix A, Section A.14, catalogs the recurring errors by symptom.

Use this template when. A cell has failed and you want the error explained rather than the cell rewritten. The distinction matters: an assistant handed a broken cell will often return a working cell that does something else.

What you must decide first.

- What you expected the cell to produce. Without it, any code that runs looks like a fix.
- The smallest cell that still reproduces the error. Pasting a whole notebook invites a rewrite of the whole notebook.
- Whether the traceback or any path in it contains an identifier you should remove before pasting.

Prompt C.6. Diagnose an error without rewriting the cell

```
I am working in Google Colab with pandas.

EXACT ERROR MESSAGE (full traceback, identifiers removed):
[PASTE]

THE SMALLEST CELL THAT REPRODUCES IT:
[PASTE THE CELL]

RELEVANT DATA TYPES AND TWO SAMPLE ROWS:
[PASTE df.dtypes AND df.head(2)]

What I expected: [ONE SENTENCE]
What actually happened: [ONE SENTENCE]

Return, in this order and nothing else:
  1  a plain-language explanation of what the error means
  2  the single most likely cause, given what I pasted
  3  the smallest correction that would fix it
  4  one check confirming the correction did not change any output I did
     not intend to change

Do not rewrite the cell to do something different from what it does now.
Do not add cleaning, filtering, or aggregation I did not ask for.
If the traceback is insufficient to identify the cause, say what else you
need rather than guessing.
```

Likely failure modes.

- A rewrite that changes behavior. The cell runs; it no longer does what you wanted; nothing says so.

- Helpful additions — a dropna, a fillna, a filter — inserted to make the error go away rather than to explain it.
- A confident diagnosis of a cause the traceback does not support, when the real cause is upstream.
- A correction that fixes this cell and silently breaks a later one, which the assistant cannot see.

Audit after the response.

- Run the corrected cell and then re-run the cells that depend on it, checking that no unrelated output changed.
- Read the traceback yourself from the final line upward and confirm the assistant's reading of it, per Appendix A, Section A.3.
- If the correction added a step you did not ask for, remove it and see whether the error returns. If it does not, the added step was not the fix.

What to record in Appendix D. Beyond the standard record of Section C.3: the exact error, the minimal cell, the correction you accepted, the check confirming nothing else changed.

Template C.7 — Produce a Descriptive Table

Owned by. Chapter 5, Section 5.10. The instruction block is that section's, verbatim, with placeholders in place of its example; the constraints below it are generalized from the same section.

Use this template when. You know exactly which table you want and want the groupby-and-aggregate scaffold written. Descriptive work is where assistance is most fluent and, for that reason, most quietly dangerous: the failures are analytic decisions hiding inside the computation, and they arrive as a tidy table that looks finished.

What you must decide first.

- The exact table, stated as a shape: what one output row represents and which columns it carries.
- The denominator of every rate in it. If you cannot state the denominator, the table is not specified yet.
- The certified totals the table must reconcile to, from the Chapter 4 verification log.

Prompt C.7. Produce a descriptive table that must reconcile

```
DATA DICTIONARY ENTRIES:
[PASTE]

CERTIFIED TOTALS from my verification log:
  rows [N]   distinct orders [N]   distinct customers [N]
  total revenue [$N]

Write pandas code to produce [ONE SPECIFIC TABLE]. Before the code, state:
the unit of analysis of each output row; the exact denominator of every rate
or proportion; which measure of center you will use and why, given that
order value is right-skewed; and the reconciliation I should expect — which
totals of your table must equal the certified totals above. After the code,
print those reconciliation figures.

CONSTRAINTS
Do not narrate the table.
Do not use a causal verb anywhere in the response.
Do not average group means to produce an overall figure; weight them, and
state what you weighted by.
Report every rate with its denominator in the column name or a printed
note.
Use only the fields in the dictionary entries above. If the table requires
a field I have not supplied, say so and stop.
```

Likely failure modes.

- The silent mean, reached for on data known to be skewed, reporting a center the median contradicts.
- The invented denominator, chosen without disclosure, which retires the four-element discipline into a code comment.
- The averaged average — group means averaged to produce an overall figure that reconciles to nothing.
- The unreconciled table, whose filtered totals no longer match the certified file and whose filter is mentioned nowhere.
- Narrative overreach: asked to explain a table, an assistant supplies causes, converting description into speculation at exactly the step stakeholders quote.

Audit after the response.

- Recompute one cell by hand from the certified table.
- Check the printed reconciliation figures against the verification log.
- Strike every causal verb from any narration before it travels; descriptive language includes *is associated with*, *accounts for*, and *coincides with*.
- Confirm that any overall figure was weighted rather than averaged, and that the weight is named.

What to record in Appendix D. Beyond the standard record of Section C.3: the table requested, the stated denominator and center with the assistant's reason, the reconciliation figures, and any denominator or center you had to correct.

Template C.8 — Propose an Exploratory Analysis Plan

Owned by. Chapters 4 through 7 and 12, which each name candidate enumeration as a legitimate delegation. Section 5.10 sets the governing condition: proposals are candidates for the specification, not analysis.

Use this template when. You are before the modeling stage and want a wider slate of views than you would generate unaided. Enumerating rivals is a breadth task, and breadth is what an assistant has.

What you must decide first.

- What has already been established, and with what evidence. Supply it, so the proposals build on it rather than rediscovering it.
- That you are asking for candidates and will choose among them yourself.
- A ceiling on the number of proposals. An unbounded request returns an unbounded list that nobody triages.

Prompt C.8. Enumerate candidate views before modeling

```
Decision this feeds: [DECISION]
Tables and fields available, with grain: [PASTE SCHEMA]
What has already been established, with its source:
[CERTIFIED FINDINGS, EACH WITH THE SECTION OR LOG ENTRY IT CAME FROM]

Propose at most [SIX] tables or views worth building before any model is
fitted. For each, state:
  the question it answers, in one sentence
  the fields required
  the output grain — one row equals one what
  the pattern you would expect if the established findings hold
  one anomaly that would be worth investigating
  one verification check the output must pass

CONSTRAINTS
Treat every proposal as a candidate for my specification, not as analysis.
Do not state findings. Do not rank the proposals by how interesting you
expect the answer to be.
Name any proposal that requires a field I have not listed, and mark it
UNSPECIFIED rather than assuming the field exists.
Make the proposals differ from one another in question, not only in form.
```

Likely failure modes.

- Proposals that require fields the schema does not contain, offered without comment.
- Proposals ranked by how interesting the assistant expects the answer to be, which is a finding stated before any evidence exists.
- Findings smuggled into the framing — "a view showing that the new stores are outperforming" rather than a view that would answer whether they are.
- Six proposals that are one proposal cut six ways, because nothing in the prompt asked for them to differ.

Audit after the response.

- Delete every proposal whose output grain you cannot state.
- Delete every proposal that requires a field the schema does not carry, or obtain the field first.
- Add each surviving proposal to the specification as a question, not as a planned finding.
- Check that the expected patterns are consistent with the certified findings you supplied; where they are not, that is a question worth asking before building anything.

What to record in Appendix D. Beyond the standard record of Section C.3: the schema and established findings you supplied, the proposals returned, which you kept, which you discarded and why.

C.7 Segmentation, Drivers, and Prediction Templates

From Chapter 6 onward the assistant can draft the entire analytic act, not merely its mechanics. Prompted with a customer file, a current assistant will build features, scale them, choose a number of clusters, fit the model, name the segments, and write the strategy narrative in one response. The four templates in this section are all two-prompt templates for that reason, and the audit between the prompts is the part that cannot be delegated.

Template C.9 — Build and Validate a Segmentation

Owned by. Chapter 6, Section 6.12. The first instruction block is that section's, verbatim; the second renders its prose requirements as a prompt.

Use this template when. You are building a customer-grain feature table and fitting a clustering, and you want the mechanics drafted without the conclusions attached.

What you must decide first.

- The feature list with derivation rules, in writing, before the first prompt. Feature derivation is safe to delegate only when the rules come from your definitions rather than the assistant's defaults.
- The population rule, the analysis date, and the window.
- The number of clusters — after the sweep, and by the braid of elbow, silhouette, and business judgment. Not in the first prompt.
- The withheld attributes you will profile on. They must be withheld from the fit, which means naming them before it.

Prompt C.9. Build the feature table and run the evaluation sweep

```
FEATURE LIST WITH DERIVATION RULES:
[PASTE]
POPULATION RULE: [E.G. CUSTOMERS WITH AT LEAST ONE ORDER IN THE WINDOW]
ANALYSIS DATE: [DATE]        WINDOW: [N] MONTHS
CERTIFIED TOTALS: revenue [$N], distinct orders [N], distinct purchasers [N]
TREATMENT CAPACITY OF THE PLATFORM: [N DISTINCT TREATMENTS]

Build [FEATURE TABLE NAME] per these rules; print the three reconciliation
figures; fit k-means for k = 2 through 8 on standardized inputs with a fixed
random seed; report inertias and mean silhouettes; recommend nothing yet.

Use only the derivation rules above. Do not add, drop, or transform a
feature I have not specified. Do not name or describe anything.
```

Prompt C.10. Fit the chosen solution and return the validation artifacts

```
I have chosen k = [K], by elbow, silhouette, and business judgment.
Do not revisit that choice.

Return, in this order and nothing else:
  the fit on standardized inputs with seed [SEED]
  the centroid table in standardized units and in native units
  adjusted Rand indices across five independently seeded fits
  adjusted Rand indices across two 80 percent subsamples
  a profile of each cluster on these withheld attributes: [LIST]

Write code that leaves the fitted scaler and the fitted model in clearly
named objects, and print the object names. Do not return the fit as prose
thresholds; an eyeballed threshold approximates a model without
reproducing it.
Use only the features and withheld attributes listed above.
Do not name the clusters. Do not describe them. Do not interpret the
profile. Do not refit at assignment time.
```

Likely failure modes.

- The confident k, presented as found rather than chosen, with none of the braid.
- Skipped scaling, which produces spend bands wearing segment costume and is invisible in the output.
- The invented persona — incomes, life stages, attitudes — asserted from a schema that contains none of them.
- The single-run solution, in which initialization luck ships as structure because nothing refit, perturbed, or compared.
- The narrated geometry, attaching motives to what is so far a partition of a feature table.
- The overfit taxonomy: asked for detailed segments, an assistant supplies nine.

Audit after the response.

- Verify one centroid by hand: filter the feature table to one cluster, average one feature, compare.

- Read the stability indices and reject the solution if the partitions do not reproduce across seeds and subsamples.
- Check the holdout profile against the predictions you wrote down before running it.
- Strike every noun in every segment name that no input feature or holdout column can redeem.
- Confirm the code leaves the fitted scaler and model in named objects you can reuse at assignment time. Assignment applies the fitted objects; it must never refit, and it must never approximate them with thresholds.

What to record in Appendix D. Beyond the standard record of Section C.3: both prompts, the reconciliation figures, the sweep, the stability indices, the holdout profile against your written predictions, every name you rejected and the evidence that would have been needed to keep it.

Template C.10 — Fit and Audit a Regression

Owned by. Chapter 7, Section 7.11. Both instruction blocks are that section's, verbatim; the frames around them are generalized with placeholders.

Use this template when. You have a written model specification — variables, reference category, formula — and want it fitted and reported without narration.

What you must decide first.

- The variable list, from your specification rather than the assistant's initiative.
- The reference category, explicitly. It determines what every dummy coefficient means.
- The predicted sign of every coefficient, written before the fit. The four-point audit compares against these.
- The covariance estimator. HC3 is the guide's default and the prompt states it so the assistant cannot inherit a nonrobust one.

Prompt C.11. Fit the model exactly as specified

```
VARIABLE LIST WITH DERIVATION RULES:
[PASTE]
DECLARED RESPONSE: [VARIABLE, IN ITS UNITS]
REFERENCE CATEGORY: [LEVEL]
COVARIANCE ESTIMATOR: HC3
MODEL FORMULA, exactly as written in my specification:
[FORMULA]

Fit this model as specified with HC3 standard errors; report the coefficient
table with 95 percent intervals in native units, the residual-versus-fitted
plot, variance inflation factors, and R squared; change nothing in the
specification; narrate nothing yet.
```

Prompt C.12. Draft the reporting sentences after the model survives

```
Here is the audited coefficient table:
[PASTE ONLY THE AUDITED NUMBERS]
Held constant in this model: [LIST]

Draft three sentences reporting these estimates as associations, using
"is associated with" or "predicts," naming units and the held-constant
list, and making no causal claim.

Do not add a recommendation. Do not use a number that is not above.
```

Likely failure modes.

- The confident narration — drivers returned as causes, because fluent causal prose saturates the training data. The arithmetic may be flawless while the verb is unearned.
- The silent specification: variables entered, dropped, reference category set, interaction included or not, none of it reported as a choice.
- The plausible-but-impossible coefficient, delivered with the same serene formatting as a correct one.
- The units slip — a coefficient narrated without units, or coefficients on different scales compared as though they were comparable.
- The diagnostics skip: no residual plot, because the default workflow fits and reports but does not examine.
- A main effect quoted unconditionally while its interaction sits in the same model.

Audit after the response.

- Run the four points of Table 7.5 in order: signs against your written predictions and the certified descriptives; magnitudes against hand-computable anchors; every coefficient restated aloud in its units, with conditional coefficients read conditionally; residuals inspected and the covariance estimator matched to what they show.
- Audit the second prompt's verbs as strictly as the numbers. Assistants reintroduce causal language at the narration step even when the numbers are clean.

What to record in Appendix D. Beyond the standard record of Section C.3: both prompts, the coefficient table, your predicted signs beside the fitted ones, which of the four points caught the most serious problem, every verb you changed in the drafted sentences.

Template C.11 — Draft a Predictive-Model Pipeline

Owned by. Chapter 8, Section 8.11. The instruction blocks preserve that section's wording; the frame is Table 8.1's seven declarations, with its *calibration metrics* row renamed **error metrics** — MAE and RMSE measure prediction error, and Chapter 9 uses *calibration* in its probabilistic sense.

Use this template when. You have declared the frame and the label and want the pipeline built. This is the delegation with the highest ratio of mechanical work to judgment, and also the

one where a failure is least visible: a leaky pipeline does not look different from an honest one in a chat window — it looks better.

What you must decide first.

- All seven declarations of the frame, in writing, before the prompt: unit of prediction, target, features, horizon, decision metric, error metrics, baselines.
- The label with its four dates — feature-window start, snapshot, outcome-window start and end — and its eligibility rule.
- The split and cross-validation protocol with its seed.
- The range check that would convict a leak, computed from your snapshot before you see any result.

Prompt C.13. Build the pipeline exactly as framed

```
THE FRAME
  Unit of prediction   [ONE PREDICTION PER ...]
  Target               [QUANTITY, UNITS, OUTCOME WINDOW]
  Features             [FEATURE LIST WITH DERIVATION RULES]
  Horizon              [N MONTHS]
  Decision metric      [CAPTURE AT CAPACITY OR OTHER], against [INCUMBENT]
  Error metrics        MAE leading, RMSE beside it, both in [UNITS]
  Baselines            [MEAN], and [LAST-VALUE OR THE INCUMBENT LIST]

THE LABEL
  Snapshot date [DATE]; feature window [START] to [SNAPSHOT];
  outcome window [START] to [END]; eligibility rule [RULE]

THE PROTOCOL
  Split at the unit of prediction with seed [N]; [K]-fold cross-validation
  on the training rows only; every learned step inside a pipeline.

Build this pipeline exactly as specified; select among the candidates using
[K]-fold cross-validation on the training [UNITS] only; report one
leaderboard with [DECISION METRIC], MAE, and RMSE for every candidate and
both baselines; do not touch the test set; change nothing in the
specification; narrate nothing yet.

Before the code, list every feature and the window it is computed from, and
name any feature you cannot trace to the declared feature window.
```

Prompt C.14. Open the sealed test set once

```
I have selected [FINALIST] from the training-only leaderboard, using the
audit above. Do not revisit that choice.

Fit that exact pipeline on all training rows, open the sealed test set
once, and report [DECISION METRIC], MAE, and RMSE for the finalist and for
both baselines, on the same test rows.

Change no feature, transformation, setting, or evaluation rule. Do not tune
anything against what you see. Use only the specification above.
Narrate nothing.
```

Prompt C.15. Draft the comparison paragraph after the pipeline survives

```
Here are the audited results:
[PASTE ONLY THE AUDITED NUMBERS, INCLUDING BOTH BASELINES]

Draft the comparison paragraph for a marketing VP: the model's capture at
the program's capacity against the incumbent list, its test MAE against
both baselines in dollars, and no claim that any predicted difference is
caused by anything.

Use only the numbers above. Do not add a recommendation.
```

Likely failure modes.

- The convenient-table leak: features built from whatever table was supplied, including columns computed across the outcome window, because nothing in the prompt drew the snapshot wall. This is the single most common defect in AI-drafted pipelines (Kaufman et al., 2012).
- The celebration of the impossible — a leaky pipeline reporting a superb fit statistic, narrated as excellent.
- The missing opponent: metrics quoted with no baseline anywhere.
- The metric mismatch — distance metrics reported for a ranked-list decision, with no capture figure and no head-to-head against the incumbent.
- The vocabulary shuffle, in which classification words are applied to a numeric target.
- Hygiene slips: preprocessing fitted on pooled data, the test set consulted in a tuning loop, complexity chosen by watching test error.
- The deletion helpfulness — zero-value labels cleaned away as missing data, deleting the rows the model most needs.

Audit after the response.

- The test set is opened once, after the finalist is named and the training-only audit has passed. That is what the second prompt is for, and it is the step students most often collapse into the first.
- Run the five points of Table 8.5 in order: confirm the frame came back unedited; trace three features to their windows and run the range check; confirm every learned step sits inside a pipeline and the test set is untouched; confirm both baselines are present on every metric and the margin is computed; read the result for plausibility.
- If anything looks too good, stop and trace. In predictive work suspicion scales with success.
- Audit the second prompt's verbs: prediction language earns *expected* and *predicted*, never *will drive*.

What to record in Appendix D. Beyond the standard record of Section C.3: all three prompts, the training-only leaderboard, the feature-window trace, the range check with its result, the single sealed-test result with both baselines and the margin, and which audit point caught the most serious problem.

Template C.12 — Build and Audit a Classification or Churn Model

Owned by. Chapter 9, Section 9.11. The instruction blocks preserve that section's wording and controls, with the leaderboard separated into a ranking exhibit and a policy comparison so that the incumbent rule is judged at its operating point rather than ranked by AUC. The supplement is Table 9.5.

Use this template when. You are building or reviewing a classifier and need it graded on ranking evidence before any dollar figure is attached. Classification is where an assistant's unexamined defaults stop being statistical choices and start being business policy: a churn definition, a threshold, and a headline metric, all chosen silently in one fluent response.

What you must decide first.

- The churn or outcome label, verbatim: snapshot, both windows, activity rule, both eligibility conditions. The definition is supplied, never delegated.
- Which baselines are ranked and which are policies. A constant-score floor and a single-feature logistic model emit scores and belong on the ranking leaderboard; an incumbent yes/no rule is an operating point and belongs in the policy comparison, judged on its matrix and its economics.
- The threshold, derived from your program's own false-positive and false-negative economics. Not by the assistant, and not 0.5 by inheritance.
- The baselines, including the incumbent rule evaluated as a fixed policy rather than scored.
- The base rate, so that accuracy can be read against the do-nothing rule.

Prompt C.16. Build the classifier and grade it on ranking evidence only

```
THE LABEL, verbatim from my specification:
[SNAPSHOT, FEATURE WINDOW, OUTCOME WINDOW, ACTIVITY RULE, BOTH ELIGIBILITY
CONDITIONS]

FEATURES with derivation rules, computed from the feature window only:
[PASTE]

PROTOCOL: split with seed [N], stratified; stratified [K]-fold
cross-validation inside the training [UNITS].
BASELINES: a constant-score floor and a [SINGLE-FEATURE] logistic baseline
for the ranking comparison; [THE INCUMBENT RULE] evaluated as a fixed policy
at its own operating point, not scored.
COST ASSUMPTIONS, declared by me before any result: treatment cost [ ];
save rate [ ]; value of a save [ ].
THRESHOLD, derived by me from those costs: [P]

Build exactly this pipeline, and return two separate exhibits.

1  RANKING LEADERBOARD. The score-producing candidates and the
   continuous-score baselines only, selected and compared using stratified
   [K]-fold cross-validation on the training [UNITS], graded on
   cross-validated AUC alone. No dollar figures beside raw scores. Do not
   place [THE INCUMBENT RULE] on this leaderboard: it emits a yes/no
   decision, not a score, and AUC is not the instrument for it.

2  POLICY COMPARISON, after calibrating only the selected model inside the
   training data. Report its out-of-fold reliability table, then compare the
   calibrated model at the [P] threshold against [THE INCUMBENT RULE] as a
   fixed operating point, using the four confusion-matrix counts,
   precision, recall, treated-list size, and cross-validated expected net
   value under the cost assumptions above. Report precision as undefined,
   not zero, for any policy that treats nobody. Add a decile table and
   cumulative gains through the top two deciles for the selected calibrated
   model and for [THE INCUMBENT RULE].

Do not touch the test set. Do not output individual-level verdicts.
Choose no threshold yourself. Narrate nothing yet.
Use only the features listed above. Name any feature you cannot trace to
the declared feature window.
```

Prompt C.17. Draft the recommendation paragraph after the classifier survives

```
Here are the audited results:
[PASTE ONLY THE AUDITED NUMBERS]
Cost assumptions: [TREATMENT COST], [SAVE RATE], [VALUE OF A SAVE]

Draft the recommendation paragraph for a marketing VP: the model's capture
in the top two deciles against [THE INCUMBENT RULE]'s, the treated-list
size at the [P] threshold with the cost assumptions stated, and no sentence
that asserts an individual customer will churn.

Use only the numbers and assumptions above.
```

Likely failure modes.

- The accuracy celebration — a headline accuracy quoted without the base rate or the majority-class baseline, leaving the margin over doing nothing uncomputed.
- The silent threshold: a confusion matrix cut at 0.5 with no costs discussed, and every downstream number inheriting the default.
- The label improvisation — a churn definition invented and computed at the analysis date, building the label out of the feature window and manufacturing outcome-window contamination in one step. The resulting ranking is spectacular and spectacularly meaningless.
- The test-set race, in which a depth or a neighbor count is chosen by watching test numbers move.
- The mixed leaderboard, in which a fixed yes/no rule is given an AUC and ranked beside score-producing models. The number is computable and answers nothing: a rule with one operating point has no ranking to grade.
- The rebalancing reflex, which distorts the probability scale the threshold arithmetic consumes.
- The uncalibrated economics — expected-value tables computed from raw scores with no reliability exhibit anywhere.
- The verdict language, in which scores are narrated as facts about named individuals.

Audit after the response.

- Confirm the two exhibits came back separate, and that no dollar figure sits beside a raw score.
- Confirm the decile and gains exhibits cover both the model and the incumbent rule, so the reporting claim about the top two deciles has something behind it.
- Run Table 8.5's five points, then Table 9.5's four: base rate and floor stated with the margin computed; the confusion matrix's four counts re-added by hand and reconciled to the evaluation set, with precision and recall recomputed from raw counts; the threshold identified, its cost assumption stated, and the cut re-derived from the program's economics; calibration and selection hygiene confirmed, including that the reliability exhibit describes the model that was actually selected.
- Open the test set once, apply the threshold you derived, and re-add the matrix by hand.
- Audit the second prompt's verbs: propensity language earns *is likely to*, never *will*.

What to record in Appendix D. Beyond the standard record of Section C.3: both prompts, the ranking leaderboard and the policy comparison as two exhibits, the reliability table, the decile and gains exhibits, the hand-recomputed matrix, the threshold with its cost derivation and sensitivity, and which supplement point caught the most serious problem.

C.8 Forecasting and Experiment Templates

Two tasks in this guide have a property the others do not: the standard tooling's defaults are wrong for them. A forecasting pipeline built without instruction will shuffle ordered days into a

random split, and an experiment analysis built without instruction will apply trial machinery to observational data. Neither error announces itself, and both produce excellent numbers. The templates in this section spend most of their length closing those two routes.

Template C.13 — Draft or Audit a Forecasting Workflow

Owned by. Chapter 10, Section 10.11. Both instruction blocks preserve that section's wording, generalized with placeholders. The supplement is Table 10.5.

Use this template when. You are building a forecast that a plan will consume. The prompt is long because each of its lines closes a specific default, and a shorter prompt gets the default.

What you must decide first.

- The three windows and their dates: rolling-origin selection, sealed holdout, and planning horizon. The holdout must resemble the decision, not merely be recent.
- The seasonal structure to fit, and the structure not to fit. Annual seasonality is never fitted from two repetitions.
- The known calendar — openings, promotions, and holidays in every window, with dates. No period-7 method can find an event nobody mentioned.
- The predicted rolling-origin error for every row of the leaderboard, computed by hand for the baselines, written before the pipeline runs.

Prompt C.18. Build the forecast with the escape routes closed

```
SERIES      [DAILY / WEEKLY] [QUANTITY] from [CERTIFIED TABLE], reconciled
            to [TOTAL]
WINDOWS     rolling-origin selection through [DATE];
            sealed holdout [START] to [END];
            planning horizon [START] to [END]
METRIC      MAPE with denominators asserted and the minimum printed,
            MAE reported beside it
OPPONENTS   seasonal-naive at period [P]; naive as the floor
SEASONALITY fit [WEEKLY ONLY]; no annual fitting, under any circumstances
CALENDAR    [OPENINGS, PROMOTIONS, AND HOLIDAYS IN EVERY WINDOW, WITH DATES]

Compare candidates on rolling origins inside the selection window only; do
not touch the sealed holdout until I name the method; report every candidate
and both baselines in one table, by horizon week, with the number of errors
behind each figure; produce [80] percent intervals and aggregate paths
before quantiling for any weekly or total figure; narrate nothing yet.

Do not shuffle. Do not use a centered window anywhere in the pipeline.
Do not substitute a method I have not named.
```

Prompt C.19. Open the sealed holdout once, then produce the planning forecast

```
I have selected [METHOD] from the rolling-origin comparison. Do not revisit
that choice.

Fit it on the complete selection window, open the sealed holdout once, and
report holdout MAPE and MAE beside both baselines on the same dates.

Then refit through [FINAL OBSERVED DATE] and produce the planning forecast
and its [80] percent intervals under the named scenarios, aggregating paths
before quantiling for any weekly or total figure.

Use only the specification and calendar above. Do not reconsider the
method. Do not report the planning forecast as a graded one; it has no
actuals. Narrate nothing.
```

Prompt C.20. Draft the plan paragraph after the forecast survives

```
Here are the audited results and the override log:
[PASTE ONLY THE AUDITED NUMBERS AND THE SIX-ELEMENT OVERRIDE ENTRIES]

Draft the plan paragraph: the [MONTH] range under each named scenario, the
margin over seasonal-naive in dollars, the overrides listed with owners and
review dates.

Every figure must appear above. Do not state a point without its range.
```

Likely failure modes.

- The shuffled split: random or k-fold evaluation applied to ordered data, which is leakage by design and is caught in one line by the date-order check.
- The reused holdout, in which four methods are compared on the block later reported as the grade.
- The mismatched season — a holdout chosen for convenience, so a holiday plan is validated on late spring.
- The invented season: a period guessed at 12 on daily data, or annual seasonality fitted from two cycles.
- The silent extrapolation, straight through an announced opening and straight through December.
- The benched-metric revival — MAPE quoted on a series with near-zero days.
- The summed interval, in which weekly and season-total bounds are produced by adding daily bounds together. The numbers look plausible, which is why this one survives review.
- The naked point: a forecast delivered without an interval.
- The method upgrade, in which the assistant reaches for machinery the analyst cannot explain to the meeting that funds it.

Audit after the response.

- The holdout is opened once, after the method is named, and the planning forecast is refitted separately and never called a graded one. The second prompt exists to keep those three acts visibly distinct.
- Run Table 8.5's five points, then Table 10.5's four: the date-order check at every origin, with no centered window anywhere; selection separated from the grade; the calendar and the breaks declared and matched to overrides or stated assumptions; metric and interval honesty, including asserted MAPE denominators and path-first aggregation for any weekly or total figure.
- Compare the headline against the band you predicted before it ran, and classify any excess as pipeline rather than prophecy until it is traced.
- Audit the second prompt's verbs: a forecast is *expected*, *projected*, *likely between*; it is never *will be*.

What to record in Appendix D. Beyond the standard record of Section C.3: all three prompts, your predicted leaderboard band beside the actual, the date-order check, the single sealed-holdout result, every override with its owner and review date, and which supplement point caught the most serious problem.

Template C.14 — Draft an Experiment Registration

Owned by. Chapter 11, Section 11.11. The instruction block is that section's, verbatim, generalized with placeholders. The fifteen rows are Table 11.3's.

Use this template when. You are designing a test and want the registration drafted before any data exists. The chronology is the control: this template is only valid if it runs first (Nosek et al., 2018).

What you must decide first.

- What you are testing, stated as a treatment applied to an eligible population and measured on one primary metric.
- The columns you actually have. A guardrail that cannot be computed from them is a design defect, not a missing nicety.
- That you want gaps made visible. The final clause of the prompt is the whole prompt.

Prompt C.21. Draft a preregistration with the gaps made visible

```
I am testing whether [TREATMENT] lifts [PRIMARY METRIC] among [ELIGIBLE
POPULATION]. The columns available to me are: [PASTE SCHEMA].

Draft a preregistration covering every row of the following template —
decision, primary metric with its four elements, arms including what control
receives, treatment dose in number of exposures, assignment mechanism, unit
of randomization, eligible population, sample size with the power
calculation shown, duration, planned analysis, decision rule, stopping rule,
secondary metrics labeled exploratory, and guardrails that are computable
from the columns I have listed.

Where the information I gave you is insufficient to fill a row, write
UNSPECIFIED and ask me; do not fill it yourself.

For every guardrail you propose, name the field it would be computed from.
Do not propose a metric, a dose, or a duration I have not supplied.
```

Likely failure modes.

- The invented design. Asked to design a test, an assistant fills every gap the prompt left — inventing a metric definition, choosing a duration, assuming a unit of randomization, assuming a treatment dose, sometimes assuming an eligibility rule — silently and plausibly, so that the registration reads like a plan you made.
- A treatment dose left implicit, which leaves the break-even lift undefined and therefore the decision rule meaningless.
- Guardrails that name a harm no available column can measure.
- A stopping rule phrased as guidance rather than as a rule.

Audit after the response.

- Count the UNSPECIFIED rows. That count measures your brief, not the assistant.
- Check specifically whether the dose and the guardrail-computability rows were left unspecified or invented.
- Confirm every declared guardrail names the field it would be computed from.
- Confirm the decision rule maps a statistical output to an action, and that the stopping rule fixes the number of looks.
- Date the registration and freeze it. A registration edited after data exist is not one.

What to record in Appendix D. Beyond the standard record of Section C.3: the prompt, the draft, the count of UNSPECIFIED rows, whether the dose was invented or left open, what that count revealed about your own brief.

Template C.15 — Analyze and Stress-Test an Experiment

Owned by. Chapter 11, Section 11.11. Both instruction blocks preserve that section's wording, generalized with placeholders. The supplement is Table 11.7.

Use this template when. The outcome window has closed and you are analyzing exactly what you registered. The registration is pasted back in, because the analysis is graded against it rather than against what now looks interesting (Kohavi et al., 2020).

What you must decide first.

- The interval width you expect, computed from the arm sizes and the baseline rate. An interval far narrower than predicted means the analysis grain slipped.
- The break-even lift, computed in advance from the declared dose. An analyst who selects the contribution figure after seeing the estimate has invented a new way to peek.
- That the primary comparison is intention-to-treat, and that operational losses are reported rather than filtered.
- What the registration says about missing outcomes. A delivery failure is not grounds for removing anyone; an unobserved outcome is a different problem, and it cannot simply sit in the denominator. If the registration is silent, that is the finding, and the analysis stops until it is resolved.

Prompt C.22. Analyze exactly as registered

```
Here is the registration.

[PASTE THE DATED REGISTRATION]

Analyze the attached file exactly as registered — intention-to-treat, no
other metrics, no subgroups, one analysis. Report the assignment-flow
counts, the control rate, the treatment rate, absolute lift in percentage
points with a 95 percent interval on the difference, relative lift with a 95
percent interval on the ratio computed as a ratio, and the p-value.
Do not interpret.

Do not remove any assigned unit from the assignment-flow table, and do not
redefine the analysis population. Analyze missing outcomes according to the
preregistered rule. If the registration contains no such rule, report the
count by arm, mark the analysis decision UNSPECIFIED, and stop before
computing the effect.
```

Prompt C.23. Argue against your own result

```
Here is the audited estimate and its interval: [PASTE]
The test ran on [POPULATION], over [WINDOW], at a dose of [DOSE].
The rollout under consideration is [SCOPE, DURATION, DOSE].

List every reason this estimate might not transfer to that rollout, and
every contamination path you can construct.

Use only what I have described above; do not assume a fact about the
population, the dose, or the program that I have not stated.
Do not rank the reasons. Do not reassure me. Do not rewrite the estimate.
```

Likely failure modes.

- The observational file analyzed as a trial — the highest-frequency and highest-cost failure, invited by the request itself whenever the prompt says *campaign* instead of *experiment*.
- The uncomplaining test on peeked data. The assistant computes significance for whatever window it is handed and will not ask how many times the data have been examined.
- The per-protocol drift, in which undelivered customers are dropped and the randomized comparison quietly becomes an observational one.
- The segment harvest, testing twenty subgroups and reporting the significant ones at machine speed.
- The lift ambiguity: relative lift quoted without its baseline, or a ratio interval produced by dividing a difference interval by the control rate.
- The naked winner, with no interval, no practical-significance line, no stated dose, and no statement of the tested population.
- The grain slip, analyzing orders or sessions after customers were randomized.
- The power omission, narrating a null as *no effect* rather than as an effect the test could not have detected.

Audit after the response.

- Run Table 8.5's five points, then Table 11.7's three: registration precedes analysis, with every deviation listed as a deviation; assignment integrity, with the flow record reported from randomization through outcome observation and the analysis grain equal to the assignment grain; inference honesty, with absolute lift beside its baseline, a difference interval and a separately constructed ratio interval, and the whole interval compared against the break-even line rather than the point estimate.
- Grade the interval width against your prediction.
- Harvest the adversarial prompt's output into the deliverable's limitations, discarding items with one sentence of justification each.

What to record in Appendix D. Beyond the standard record of Section C.3: both prompts, the assignment-flow counts, the two intervals, the break-even comparison across the whole interval, the adversarial list with the items you kept and discarded.

C.9 Visualization and Communication Templates

Part III changes the shape of the problem. A chart's mechanics and its meaning are produced by the same act, and a sentence about a view is the medium in which the claim is made. There is no later stage at which the meaning gets audited unless the analyst inserts one. The three templates in this section insert it, and all three run the five-step routine of Section C.1.1 rather than the two-prompt pattern.

All three templates repair against Appendix E, which holds the visualization checklist in runnable form. Where an entry says *repair against Appendix E*, the checklist is the instrument and this appendix only routes you to it.

Template C.16 — Ask for Candidate Visual Specifications

Owned by. Chapter 12, Section 12.12. The first prompt preserves that section's wording and adds the candidate framing Table C.2 requires; the second consolidates the palette, footer, and alternative-text uses that Sections 12.12 and 13.11 both name.

Use this template when. You have a question and want a slate of candidate views specified rather than rendered. Specifications can be audited; rendered charts are persuasive in a way their specifications are not, and the human tendency is to evaluate whether a chart is attractive rather than whether it is right.

What you must decide first.

- That every element the assistant returns — the mark, the encodings, the aggregation, the axis ranges, the sort, the palette — is a candidate for your approval and not a decision. Table C.2's last row governs: the analyst sets the axis. A proposed range that does not begin at zero on a length-encoded mark is a defect to catch, not an instruction to follow.
- The schema, pasted. Almost every invented-field failure is a prompt that did not include it.
- The grain of the source table, stated. This is the line that prevents a line-grain average being presented as a per-order quantity.
- The question, in one sentence, as a comparison the reader must make.
- Your prediction, written from the specifications alone: for each candidate, the comparison, the channel carrying it, that channel's rank in the perceptual hierarchy, and the two faults you expect.

Prompt C.24. Request candidate view specifications

```
I have [TABLE] at [GRAIN], with these columns [PASTE THE SCHEMA]. I want to
answer this question: [QUESTION]. Propose three candidate views. For each,
state the mark type, every field-to-channel encoding, the aggregation and
the grain one mark represents, the axis ranges, the sort order, and the
palette. Do not use any column not in the schema I pasted; if the question
requires a field I have not given you, say so instead of assuming it. Do not
render or describe conclusions.

Treat every element as a proposal for my approval. For any axis range that
does not begin at zero, state the reason in one line; do not choose a range
to make a difference look larger.
```

Prompt C.25. Request the palette, the footer, and the alternative text

```
For the view specified as [PASTE THE CHOSEN SPECIFICATION], produce three
things and nothing else.

1  A color-blind-safe [CATEGORICAL | SEQUENTIAL | DIVERGING] palette as an
   explicit list of hex values, with a note on how each pair reads in
   grayscale.
2  The footer line, using only the values below: source table, date range,
   filters applied, grain of aggregation, and extract refresh date.
   [SUPPLY THE FIVE VALUES]
3  Alternative text giving the view's purpose, population, units, the
   major pattern, and any material exception — without adding a causal,
   predictive, or evaluative claim the view does not support.

Do not use a red-to-green palette to carry an ordered field.
Do not add a value I have not supplied.
```

Likely failure modes.

- The truncated axis to order, produced whenever the request contains a phrase like *make the difference clear*.
- The grain slip — a per-order metric computed from line-grain rows, which is a correct average of the wrong thing.
- The default pie for a composition question.
- The palette mismatch in both directions, with an accessibility omission alongside it, since nothing in the request will have mentioned color-vision deficiency.
- The invented field, referencing a column that would be natural for the question and does not exist.
- The wrong-purpose chart — a trend answer to a comparison question, which will look entirely reasonable.
- The undisclosed filter, which leaves no trace in the image.
- The default sort, left in place constantly.
- The overclaiming title, containing a verb the view has not earned.

Audit after the response.

- Build the specification exactly as written, including the faults. A specification corrected silently during construction cannot be audited.
- Run the three transferable points of Table 8.5 and all three of Table 12.5, line by line, on the built view.
- Compare the faults you found against the faults you predicted, and record both the misses and the false alarms. A fault you predicted and did not find is as informative as one you missed.
- Repair against Appendix E, one mechanic at a time, with a one-sentence note per repair recording what changed in the *reading* rather than what changed in the chart.

What to record in Appendix D. Beyond the standard record of Section C.3: the request prompt with the schema you pasted, the candidate specifications, any column the assistant invented

despite the schema, your written predictions, the audit, your grade of your own predictions including false alarms, the repairs with their reading notes.

Template C.17 — Audit a Chart or Tableau View

Owned by. Chapter 12, Section 12.12, adversarial use. The checklist it applies is Appendix E.

Use this template when. A view exists — yours, a colleague's, or one an assistant drafted — and you want every way it could mislead a reader enumerated before it circulates. Handed a description of a finished view and asked what could go wrong, an assistant produces a checklist that will often contain items the builder missed.

What you must decide first.

- That you will describe the view completely, including the filter state and the footer. An audit of a partial description audits a partial chart.
- What the reader is supposed to conclude. The audit is about the gap between that and what the view supports.
- Your own one-minute check first: read the axis floor, name the grain, name the comparison. Run it before you ask, so the response is graded rather than trusted.

Prompt C.26. Enumerate how a finished view could mislead

```
Here is a complete description of a finished view.
  Mark type: [ ]
  Every field-to-channel encoding: [ ]
  Aggregation, and the grain one mark represents: [ ]
  Axis ranges, both axes: [ ]
  Sort order: [ ]
  Palette and its type: [ ]
  Filter state, including defaults: [ ]
  Title, subtitle, and footer as written: [ ]
The reader is meant to conclude: [ONE SENTENCE]

List every way this view could mislead that reader. Organize the list under
these headings and nothing else:
  grain and aggregation
  scale and baseline
  encoding, and the comparison the reader must make
  color and accessibility
  filter state, and what is not shown
  the title's verb, and every number in it

For each item, state the change in what the reader would conclude, not the
change to the chart.
Use only what I described. Do not assume a field, a filter, or a value I did
not state. Do not propose a redesign. Do not comment on taste.
```

Likely failure modes.

- Objections about taste rather than about reading — font, color preference, placement — which are unfalsifiable and therefore unhelpful.

- Assumed filters, fields, or values you did not describe, producing defects the view does not have.
- A redesign offered in place of a defect list, which moves ownership of the encodings.
- Silence about the omissions, which are the hardest class: an undisclosed filter leaves no trace in an image and will not be surfaced unless the prompt asks for it by name.

Audit after the response.

- For every item, decide whether it is a defect, a judgment call, or noise. Appendix E distinguishes stop-ship failures from judgment calls; use its labels.
- Repair the stop-ship items against Appendix E, one mechanic at a time.
- For each repair, write one sentence on what changed in the reading. *A reader now sees a quarter that moved slightly, rather than a quarter that appeared to jump twelvefold* is the form.
- Confirm the footer carries source, window, filters, grain, and refresh date before the view leaves the workbook.

What to record in Appendix D. Beyond the standard record of Section C.3: the description you supplied, the defect list, which items you accepted and which you discarded with one sentence each, the repairs with their reading notes.

Template C.18 — Draft and Test Executive Communication

Owned by. Chapter 13, Section 13.11. The first instruction block preserves that section's wording, generalized with placeholders; the second renders its rewrite request as a template.

Use this template when. You have a verified view or a finished recommendation and need language for it. This is the narrowest delegation in the guide: the assistant may generate candidates, and the analyst must verify every one against the view it sits on, one sentence at a time, before any of them reaches a canvas.

What you must decide first.

- What the view actually shows, in its own units, including the values a reader can see. Supply nothing else — in particular, do not paste the chapter's conclusions, because the assistant will use them and the title will contain a number the view does not.
- The population the view covers, and its boundary. A statement about four mature stores must not become a statement about a format.
- Your prediction: how many of the candidates will overclaim, and in which of the three shapes — causal or predictive verb, generalized population, imported number.
- For the rewrite prompt, the reader: role, what she knows, and how long she will spend.

Prompt C.27. Generate candidate assertion titles with verification lines

```
Here is a description of one view from a dashboard: [DESCRIPTION IN THE
VIEW'S OWN UNITS, INCLUDING THE REFERENCE LINE, THE POPULATION IT COVERS,
AND THE VALUES A READER CAN SEE].

Write twelve candidate titles as complete sentences.

Do not use any number that I have not given you. Do not assert a cause, a
forecast, or a payback period. Do not generalize beyond the [N] [STORES /
CUSTOMERS / PERIODS / EXPERIMENTS] represented in the view.
For each candidate, state in a second line exactly which marks in the view a
reader would have to look at to verify it.
```

Prompt C.28. Rewrite for a stated reader without losing a finding

```
Rewrite the paragraph below for [READER: ROLE, WHAT SHE ALREADY KNOWS, HOW
LONG SHE WILL SPEND ON IT].

[PASTE YOUR PARAGRAPH]

Keep every interval, every threshold, every reversal condition, and every
qualification of population, window, and dose. If you shorten a sentence,
do not remove a clause that states a limit.

Use only what is in the paragraph above. Do not introduce a number, a
claim, or a qualification that is not already there.

Return the rewrite, and beneath it a complete list of every clause you
removed, merged, or reworded, with the original wording beside the new one.
```

Likely failure modes.

- The overclaiming title, in its three shapes. It is the modal failure, because a more interesting sentence is better writing and the assistant is optimizing for writing.
- The confident recommendation, which drops the interval, the threshold, and the reversal criterion while keeping the tone.
- The disappeared uncertainty. Asked to make a paragraph clearer, an assistant removes the interval before it removes anything else, because the interval is the clause that complicates the sentence.
- The generic audience, produced by any prompt that does not name the reader. Pasting the audience analysis of Section 13.2 into the rewrite prompt changes the output substantially and costs one paragraph.
- The invented finding — a connective sentence asserting a relationship between two views that no view establishes. It reads as synthesis; it is confabulation, and it is caught only by the rule that every sentence must be traceable to a specific view.

Audit after the response.

- Take each candidate, cover the description you supplied, look only at the built view, and decide whether a reader could confirm the sentence from the marks. Reject on the first

failure rather than repairing, because repairing reintroduces your own knowledge into a sentence that must stand without it.

- Grade the overclaim count and shapes against your prediction, recording misses and false alarms.
- For the rewrite, diff the two versions clause by clause rather than reading the new one. The missing clause is invisible in a text that reads well.
- Show the surviving title on its view to someone who has seen neither, ask what it says, and confirm her paraphrase is the claim you intended rather than the stronger one next door. The reader's first interpretation is the usability evidence, not her opinion of the design (Krug, 2014).
- Run C1 through C3 of Table 13.6 on the finished artifact.

What to record in Appendix D. Beyond the standard record of Section C.3: both prompts, the twelve candidates with their verification lines, your predicted overclaim count and shapes beside the actual, the clause-by-clause diff with every removal classified as legitimate compression or deletion of a finding, the reader's verbatim paraphrase.

Section 13.11 names two further failure modes that this entry deliberately does not carry, because neither prompt above can produce or repair them. The **layout without a hierarchy** is a dashboard defect, and its fix is to supply the KPI hierarchy of Section 13.4 as an input rather than to audit a sentence. The **accessibility and provenance omissions** are omissions rather than errors, and they are caught by the export and delivery section of Appendix E. Template C.19 picks up both at the deliverable level.

C.10 Review and Close

One template belongs to no single chapter. Four chapters name the same use — the assistant as an adversary — and each says it is where the tool earns most: Section 2.8 on the specification, Section 11.11 on a causal estimate, Section 12.12 on a finished view, Section 13.11 on a finished dashboard. Template C.19 generalizes them to a whole deliverable, and it is the last thing you run before you sign.

Template C.19 — Conduct an Adversarial Final Review

Owned by. Sections 2.8, 11.11, 12.12, and 13.11, which each name adversarial review as the highest-value delegation. The defect categories consolidate the audit tables of Section C.3.

Use this template when. The deliverable is finished and you are about to submit or circulate it. Run this before the last read-through, not instead of it.

What you must decide first.

- Whether the deliverable may leave your environment at all. This template asks you to paste a finished artifact, which is the largest single disclosure in the appendix. Run a manual or approved local privacy screen **first**. Do not use this template on identifiable, confidential, contractually restricted, or proprietary material unless the assistant operates

inside an environment authorized for it. The final defect heading asks the assistant to find sensitive data; by then it has already been supplied, so that heading is a backstop and not the control.

- That you want a defect list and not a rewritten deliverable. A rewrite at this stage transfers authorship of the claims.
- What evidence stands behind each figure — design, window, population, method, and the interval or baseline. Without it the review can only check internal consistency.
- Which of the defects, if found, would stop submission. Deciding that in advance stops the list being triaged by how tired you are.

Prompt C.29. Adversarial review of a finished deliverable

```
Here is a finished deliverable: [PASTE THE MEMO, THE TITLES, THE TABLE OF
FIGURES, AND THE LIST OF VIEWS WITH THEIR GRAINS].

Here is the evidence behind it: [DESIGN, WINDOW, POPULATION, AND METHOD FOR
EACH CLAIM, AND THE INTERVAL OR BASELINE FOR EACH FIGURE].

Act as a skeptical reviewer who will be blamed if this is wrong. Produce a
numbered defect list under these headings:
  claims not supported by the evidence described
  a missing baseline or comparison
  a grain that changes between a figure and the sentence about it
  a metric used with two different definitions
  causal or predictive verbs the evidence does not license
  uncertainty stated nowhere
  a number in the prose that appears in no exhibit
  anything a reader could reasonably conclude that the evidence does not
    support
  sensitive or identifying data visible anywhere (a backstop; the privacy
    screen runs before this prompt, not inside it)
  missing AI-use documentation

For each defect, quote the sentence or name the exhibit.
Use only the deliverable and the evidence described above; do not assume
context I have not supplied.
Do not rewrite anything. Do not rank or score the deliverable.
Do not raise an objection that would apply equally to any deliverable.
End with the three questions you would ask the analyst before signing.
```

Likely failure modes.

- Generic review comments that would apply to any deliverable.
- Invented defects, produced by assuming context you did not supply.
- Silence on omissions, which is the category the review is most valuable for and least likely to volunteer.
- A rewritten version supplied helpfully alongside the list.

Audit after the response.

- Triage every item into defect, judgment call, or noise, and record one sentence for each discard.

- Repair the defects. Re-run the specific audit from Table C.4 that governs whatever you changed — a repair is a new artifact and inherits its own audit.
- Answer the three closing questions yourself, in writing. If you cannot, the deliverable is not finished.
- Confirm the four questions of Section 1.9 can be answered yes for every claim that survives.

What to record in Appendix D. Beyond the standard record of Section C.3: the deliverable version reviewed, the defect list, the triage with a sentence per discard, the repairs and the audits they triggered, your answers to the three closing questions.

C.11 Anti-Patterns

Every chapter from Chapter 4 onward prints a deliberately loose prompt and grades what comes back, because reproducing the failure is how the failure becomes legible. Those prompts are collected in Table C.6 with what each one leaves unspecified and the minimum addition that makes the task auditable. They are worth running once each: an exchange in which the assistant performed badly and you caught everything is a better artifact than one in which it performed well.

Table C.6

Weak requests, what they leave unspecified, and the minimum repair

The weak request	What is unspecified	Minimum repair
"Analyze this dataset and give me insights."	The decision, the unit of analysis, the comparison, and what would count as an answer	Template C.1, then Template C.8. State the decision and the output grain; ask for candidate views, not findings
"Clean this file." (Exercise 4.6)	Which step, which remedy, what may be dropped, and what the totals should be afterward	Template C.5. One step, one named remedy, predicted row count and revenue total before the code
"Summarize this sales data and tell me what it means." (Exercise 5.6)	The output grain, every denominator, the measure of center, and the totals it must reconcile to	Template C.7. Add the certified totals and forbid causal verbs
"Segment these customers and describe the segments." (Exercise 6.6)	The features and their derivation rules, the population, the number of clusters, whether inputs are scaled, and what the names may assert	Template C.9, in two prompts, with the validation artifacts required before any name is written

The weak request	What is unspecified	Minimum repair
"Find what drives customer spend and tell me what to do about it." (Exercise 7.6)	The specification, the reference category, the covariance estimator, and the verbs the evidence licenses	Template C.10. Supply the formula; forbid narration in the first prompt
"Build me the best possible model — maximize accuracy." (Exercises 8.6, 9.6)	The frame, the label with its four dates, the snapshot wall, the baselines, and the decision metric	Templates C.11 and C.12. Seven declarations, the label verbatim, both baselines, and the test set untouched
"Forecast our daily revenue for the next eight weeks — maximize accuracy." (Exercise 10.6)	The three windows, the seasonal structure, the calendar, the metric's denominators, and the intervals	Template C.13. Selection separated from the grade; annual fitting forbidden; calendar supplied
"Check whether the campaign worked, and see which customer segments responded best." (Lab 11.2)	Whether the data are experimental, what was registered, how many looks have occurred, and whether the subgroups were chosen before or after	Templates C.14 and C.15. Register first; paste the registration back in; one analysis, no subgroups
"Make a chart showing how much the suburban stores have grown, and make the growth clear." (Lab 12.2)	The schema, the grain, the axis range, and what "clear" is permitted to do to the axis	Template C.16. Paste the schema, state the grain, ask for a specification rather than a chart
"Make this recommendation more confident." / "Make this punchier."	Which clauses may be cut. The interval goes first, because it is the clause that complicates the sentence	Template C.18, second prompt. Require the removed-clause list and diff rather than read
"Fix the code."	The error, the expected result, the data types, and what the cell is allowed to change	Template C.6. Paste the traceback and the expectation; forbid unrequested additions

The pattern across all eleven is the same. Each weak request delegates a decision the requester did not know she was delegating, and the assistant makes it silently because nothing in the request suggested it was open. The repair is never longer prompting for its own sake; it is naming the decision.

CONCEPT

Fluent Is Not Verified

An assistant-produced analytics failure often does not look like a failure. It looks like finished work: a table that adds up within itself, a model whose fit statistic is excellent, a chart that is well designed, a paragraph that reads better than the one you wrote. Presentation quality does not establish correctness, because presentation quality is what the tool optimizes.

So before any response leaves your desk, answer Section 1.9's four questions in order. Does the output answer the question I specified? Does one row still represent what I said it represents? Can I reproduce at least one number by hand? Can I state in one sentence what this result does not show? A no anywhere is the finding.

Source: Course concept developed for this guide, informed by National Institute of Standards and Technology (2024) and Dell'Acqua et al. (2023).

C.12 Template Index and Exclusion Register

Table C.7 is the chapter-indexed prompt finder. It maps each chapter's AI-assistant section and its AI in Practice box to the templates that generalize them, so that a student working through a chapter can find the template without reading this appendix from the front.

Table C.7

Templates by chapter

Chapter	AI-assistant section and box	Templates
1	Sections 1.7 and 1.9; "Review Before You Rely." No prompt is printed; the box is the general four-question audit	The audit in Section C.3 applies to every template
2	2.8 AI as a Decision-Support Assistant; "Ask for the Other Side"	C.1, C.2
3	3.9 AI as a Measurement and Documentation Assistant; "Draft the Dictionary, Audit the Draft"	C.3, C.4
4	4.10 AI as a Data Preparation Assistant; "Delegate the Code, Never the Count"	C.5, C.6
5	5.10 AI as a Descriptive Analytics Assistant; "The Summary That Must Add Up"	C.7, C.8
6	6.12 AI as a Segmentation Assistant; "The Cluster Solution That Must Survive a Refit"	C.9
7	7.11 AI as a Regression Assistant; "The Regression That Must Survive Its Audit"	C.10

Chapter	AI-assistant section and box	Templates
8	8.11 AI as a Predictive-Modeling Assistant; “The Pipeline That Must Survive Its Audit”	C.11
9	9.11 AI as a Classification Assistant; “The Classifier That Must Survive Its Audit”	C.12
10	10.11 AI as a Forecasting Assistant; “The Forecast You Priced Before You Ran It”	C.13
11	11.11 AI as an Experimentation Assistant; “Preregister With the Assistant, Then Audit the Analysis Against the Registration”	C.14, C.15, C.19
12	12.12 AI as a Chart-Drafting Assistant; “Predict the Faults Before You Look”	C.16, C.17
13	13.11 AI as a Communication Assistant; “Draft the Titles, Then Prove Them”	C.18, C.19

Table C.8 records what this appendix does not contain and why. A library that silently omits a template looks identical to one that never considered it, and the reasons are worth stating because several of them are decisions rather than oversights.

Table C.8

Exclusion register

Not included	Reason
A template for Chapter 1	Chapter 1 has no AI-assistant section and prints no reusable prompt. Its contribution is the four-question audit of Section 1.9, which governs every template here, and the compact record of Table 1.6, which every entry invokes.
Separate templates for the four segmentation tasks the blueprint lists	Chapter 6 prints one specification prompt and states the second in prose. Publishing four templates would promise a division of labor the chapter does not teach. Template C.9 carries both prompts.
Separate templates for the four executive-communication artifacts	Chapter 13 prints one title prompt and one rewrite request. Template C.18 carries both; the memo structure and the audience-testing protocol belong to Sections 13.10 and 13.12 and are not prompts.
A template for error analysis	Sections 8.11 and 9.11 both print a one-line error-analysis question. Neither needs a template; both are folded into the audit steps of Templates C.11 and C.12.
Templates for alternative text and footer drafting	Sections 12.12 and 13.11 name both as good uses. They are one prompt between them and are folded into Template C.16.

Not included	Reason
Prompts that ask an assistant to interpret, recommend, or conclude	No chapter authorizes one, and eleven chapters explicitly forbid it at the point where it would occur.
A prompt for the deliberately loose requests	The eleven weak requests in Table C.6 are reproduced as anti-patterns to be run and graded, not as templates to be adapted.
Vendor-specific or model-specific phrasing	Every template is written to be tool-neutral. Where a chapter names a tool, it is because the course requires it, not because the prompt depends on it.
A rewritten "good" version of each weak prompt	The chapters ask students to author those rewrites (Exercises 4.6, 5.6, 6.6). Supplying them here would answer a graded exercise.

One dependency is recorded rather than excluded. Templates C.16 and C.17 repair against Appendix E, and Template C.17's triage uses its pass, caution, and stop labels. Chapter 12 invokes Appendix E eight times and Chapter 13 once, so the dependency is the chapters' rather than this appendix's; but until Appendix E is published, those two entries point forward rather than across.

References

Dell'Acqua, F., McFowland, E., III, Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., & Lakhani, K. R. (2023). *Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality* (Harvard Business School Technology & Operations Management Unit Working Paper No. 24-013). https://doi.org/10.2139/ssrn.4573321

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data, 6*(4), Article 15. https://doi.org/10.1145/2382577.2382579

Kohavi, R., Tang, D., & Xu, Y. (2020). *Trustworthy online controlled experiments: A practical guide to A/B testing.* Cambridge University Press. https://doi.org/10.1017/9781108653985

Krug, S. (2014). *Don't make me think, revisited: A common sense approach to web usability* (3rd ed.). New Riders.

National Institute of Standards and Technology. (2024). *Artificial intelligence risk management framework: Generative artificial intelligence profile* (NIST AI 600-1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.600-1

Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences, 115*(11), 2600–2606. https://doi.org/10.1073/pnas.1708274114