`

From Who to What Moves Them: Measuring, Modeling, and Refusing to Overclaim the Drivers of Customer Spend

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

Chapter Information

ABSTRACT

This chapter develops the analysis of relationships among marketing variables: how association is measured, modeled, and kept distinct from causation. It builds linear association through covariance and the Pearson correlation coefficient, then catalogs what correlation hides — nonlinearity, outlier sensitivity, and mixed populations — and develops confounding, defined as a pre-existing common cause, on marketing's most reliable confounder, the calendar. Linear regression is built from the simple case through multiple regression, dummy variables, and interactions, with coefficients read as model-adjusted comparisons rather than levers and interactions read conditionally. Statistical evidence is developed at working level: heteroskedasticity-robust standard errors, confidence intervals, p-values, and variance inflation factors. A section on AI assistance installs the four-point audit of signs, magnitudes, units, and diagnostics. The labs estimate spend and average-order-value drivers on the Chapter 6 feature table, recover the chapter's designed interaction, and verify an AI-fitted model against certified descriptive findings.

KEYWORDS

correlation; covariance; regression; drivers; dummy variables; interaction terms; multicollinearity; confounding; omitted variable bias; causal overclaiming

VERSION AND DATE

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

SUGGESTED CITATION

Mendoza, J. (2026). Relationships, drivers, and regression. In *Applied business analytics for marketing decision-making: Business analytics and data visualization* (Chapter 7, Version 1.0) [Open educational resource]. CC BY 4.0.

project distributed under the modified BSD license. ChatGPT is a product of OpenAI, Claude of Anthropic, Gemini and NotebookLM of Google LLC, and GitHub Copilot of GitHub, Inc. Product names are used for identification only and do not imply endorsement. StyleCraft Collective is a fictional company created for instruction.

COMPANION REPOSITORY

Datasets, notebooks, and figure sources for this chapter: https://github.com/jrmst102/businessanalytics

Chapter Learning Objectives

By the end of this chapter, students should be able to:

1. Distinguish association from causation as a standing discipline, and explain why "driver" is a causal word that correlational evidence places on probation.
2. Compute and interpret sample covariance and the sample Pearson correlation coefficient among marketing variables, stating the direction, the strength, the unit-free character of r, and the difference between the population quantity and the sample estimate the software reports.
3. Identify what correlation misses — nonlinearity, outlier sensitivity, and mixed populations — and apply the always-plot discipline before trusting any correlation.
4. Read a correlation matrix without committing its characteristic misreadings, and explain why a matrix of pairwise correlations is not a map of drivers.
5. Define a confounder as a pre-existing common cause, distinguish it from a variable that is merely correlated with both members of a pair, and recognize seasonality as marketing's standing confounder.
6. Construct and interpret simple linear regression models: the fitted line, the intercept and slope in native units, least squares as a criterion, and residuals as diagnostic evidence.
7. Interpret R^2 as shared-variation fit under its stated conditions, and explain why fit and usefulness are different judgments, anchored to the baseline discipline of Section 2.7.
8. Construct and interpret multiple regression models, read a coefficient as a model-adjusted comparison rather than a lever, and reason about omitted variable bias and its direction, including the qualification that applies once a model holds several predictors.
9. Encode categorical variables with dummy variables and reference categories, interpret segment coefficients against the reference, and specify and interpret an interaction term conditionally.
10. Assess coefficient evidence using heteroskedasticity-robust standard errors, confidence intervals, p-values, and variance inflation factors at working level, and distinguish prediction from explanation as modeling goals.
11. Verify AI-generated regression output against fundamentals — signs, magnitudes, units, fit, and diagnostics — and against certified descriptive findings, documenting the audit per Appendix D.

Chapter 6 ended with a map and a question the map could not answer. The map was the treatment map: four segments, estimated, validated, named by rule, and paired with journeys on the fall calendar. The question came from the sign-off meeting, and it was the natural next one — what actually moves these customers? Which levers, at what strength, with what confidence? Segments describe who; this chapter is about what moves with their behavior, and about the analyst's principal instrument for that question: regression. The route runs from the raw material of relationships — association measured honestly by correlation, and the many ways correlation deceives — through the regression model itself, simple and then multiple, with the categorical machinery that lets StyleCraft's estimated segments enter the model as evidence. Because every step of this route is now one prompt away from an AI assistant, the chapter also installs its own verification discipline — the audit of signs, magnitudes, units, and diagnostics — and its own ethical guardrail, because the characteristic failure of this chapter's tools is not a wrong number but a wrong verb: an association, fluently narrated, wearing the language of cause.

CONCEPT

What This Chapter Is Really About

Chapter 6 argued that grouping is the third analysis, and that an algorithmic grouping is a hypothesis to be validated rather than a fact to be admired. This chapter takes the next step in the same argument: relating is the fourth analysis, and it is the first whose output looks like an answer to the question executives actually ask — what should we do? A correlation of 0.6, a coefficient of $41 per order, a fitted line through a cloud of customers: each is arithmetic about the world as it is, and each will be read, the moment it reaches a meeting, as a claim about what would happen if the world were changed. The entire discipline of this chapter lives in that gap.

The methods are not hard; a modern AI assistant fits a defensible regression in seconds. What cannot be delegated is the reading: whether the signs and magnitudes survive contact with certified facts, whether the model's assumptions survive contact with its residuals, and whether the sentence written about the coefficient claims only what the evidence can carry. An analyst who masters this chapter does not become the person who finds drivers. They become the person who knows exactly how much of "driver" the data has actually earned.

Source: Course concept developed for this guide, informed by Freedman (1991) and Shmueli (2010).

7.1 Marketing Decision Context: Three Levers and One Budget

Chapter 6 left the CRM manager in possession of a signed treatment map: four validated segments — the Urban Loyal Core, the Suburban Occasion shoppers, the Discount-Dependent, and the New/At-Risk — each attached to one of the four journeys the platform can sustain this fall. The map answered who. The sign-off meeting immediately generated the next decision, and this one has a budget attached. Beyond the fixed cost of building the four journeys, StyleCraft's fall retention program carries an incremental budget — the money that funds what the journeys actually do — and the VP of Marketing must allocate it across three candidate levers before the

pre-holiday budget lock in twenty-one days. The three levers on the table: deepen the discount ladder (richer offers, triggered more often), accelerate app adoption (an install campaign plus app-exclusive perks), and increase email frequency for enrolled customers. The CFO will attend the allocation meeting. After the margin conversation that opened this guide in Chapter 1, every dollar of discount is presumed guilty until proven incremental.

Three claims are already circulating in the building, one per lever, and each arrives dressed in numbers. The app claim comes from the platform team's deck: "App users spend 2.4 times more than non-app customers. App adoption is our highest-leverage investment — every install is worth hundreds of dollars in annual spend." The discount claim comes from a finance analysis: "Customers with high discount exposure show below-average total spend. Discounts are training our customers to pay less — cut the ladder, and spend will recover." The email claim comes from the CRM team itself: "Email-engaged customers buy twice as often as the unengaged. Doubling send frequency doubles touches, and touches drive orders." Each claim cites a real pattern; each pattern will reappear, verifiable, in this chapter's labs. And each claim commits, in its second sentence, an inference its first sentence cannot support. The VP has asked the analyst — you — for something specific by the budget lock: an honest drivers analysis of customer spend. Which variables actually move with spend, how strongly, in what units, holding what constant — and, for each of the three claims, a verdict: what the data supports, what it cannot support, and what evidence would settle the difference.

The deliverable has a shape this chapter will fill in. First, the relationships themselves, measured honestly: which of StyleCraft's behavioral and engagement variables are associated with spend, at what strength, and with what caveats about shape, outliers, and the mixed population Chapter 6 showed the base to be. Second, a model: spend regressed on the candidate drivers together, so that each claim can be examined holding the others constant — because the app claim's "2.4 times" compares customers who differ in far more than app ownership, and the meeting deserves to know how much of the gap survives when tenure, recency, engagement, and segment membership are held level. Third, the segments as evidence: the four groups from Chapter 6 enter the model as dummy variables, both because segment membership is itself a candidate driver and because Chapter 5's certified findings about the segments — the suburban average-order-value gap and its decomposition — are the known truths against which any fitted model must be checked. Fourth, and running through everything, the language: the deliverable's sentences must survive a CFO who has learned to ask, of every "driver," whether anyone actually changed anything and watched what happened.

The stakes of the language are not academic. If the app coefficient is read causally and the install campaign is funded on it, the budget buys the customers StyleCraft already has — the young, urban, loyalty-enrolled core who were always going to install the app — dressed up as growth. If the discount association is read causally and the ladder is cut, the Discount-Dependent segment, one quarter of the treatment map, loses the behavioral lever with which its purchases are most visibly associated, three weeks before the holiday, on evidence that has never tested whether the segment responds to that lever at all. The chapter builds the machinery in order.

Sections 7.2 through 7.5 discipline the raw material: association and causation, correlation and its failure modes, confounding and the calendar. Sections 7.6 through 7.10 build the model: simple regression, fit, multiple regression, the categorical machinery, and the reading of statistical evidence. Section 7.11 turns to the AI assistant, which will fit all of this fluently and narrate it causally by default. The labs in Section 7.12 estimate the drivers of spend and average order value on the feature table Chapter 6 built — where, by design, the true coefficient structure is known — and Sections 7.13 through 7.15 rehearse the meeting: the managerial translation, the industry mirror, and the ethics of the verb "drive."

7.1.1 Opening Case Questions

Keep these questions in mind while reading, and return to them after completing the labs.

1. Each of the three lever claims moves from an observed difference to a recommended action. For each claim, state the comparison actually being made in the data, and name one way the compared groups could differ other than the lever itself.
2. The app claim's "2.4 times" is a ratio of group means. What additional facts about the two groups would you want before treating that ratio as even descriptively meaningful — and which chapter of this guide taught each fact?
3. The finance claim reasons from a negative association between discount exposure and spend. Chapter 6's segmentation gives a specific reason this association could exist even if discounts increase every individual customer's spending. State it.
4. The VP asks for "drivers." The CFO asks whether anyone changed anything and watched what happened. What kind of evidence would satisfy the CFO's standard, and why is this chapter's evidence not it?

7.2 Association, Causation, and the Meaning of "Driver"

Everything in this chapter rests on one distinction, and the guide has been preparing it since Chapter 2 named the anatomy of a decision. Two variables are associated when they move together: when knowing one tells you something about the likely value of the other. One variable causes another when intervening on the first changes the second — when the world after the intervention differs from the world without it. Association is a fact about observed data; causation is a claim about what would happen under an action not yet taken. Marketing decisions are actions, so marketing questions are, in their bones, causal — "should we deepen the discount ladder?" is a question about a world that does not yet exist. Marketing data, almost always, records associations. The gap between the question's nature and the data's nature is the standing condition of the marketing analyst's work, and this chapter's tools live entirely on the association side of the gap.

DEFINITION

Association and Causation

Two variables are associated when their values move together in observed data, so that knowing one improves a guess about the other. A variable causes another when changing the first, with all else undisturbed, would change the second. Association is symmetric and observational; causation is directional and interventional. Observed association is consistent with causation in either direction, with a third variable influencing both, or with coincidence — and cannot, by itself, distinguish among them.

Source: Adapted from Simon (1954) and Freedman (1991).

In other words: the data can show that app users spend more; it cannot, alone, show that the app made them do it. The definition's last sentence enumerates the rivals every association carries. The app–spend association could exist because the app increases spending (the platform team's reading); because spending increases app adoption — frequent customers install the app to track orders they were already going to place (reverse causation); because a third variable, engagement with the brand, influences both installs and orders (confounding, Section 7.5's subject); or, in small samples, because of nothing at all. Nothing in the number 2.4 favors any of these. Ruling rivals out requires either design — the experiments of Chapter 11, where random assignment severs every backdoor — or assumptions strong enough to substitute for design, which multiple regression will attempt in Section 7.8 and only partly achieve.

This is why the word "driver," the word the VP used and the word this chapter's title uses, deserves formal probation. In business language, a driver analysis is understood loosely: which variables move with the outcome strongly enough to matter. In causal language, a driver is a lever: push it and the outcome moves. The looseness is not harmless, because the loose sense is what the analysis delivers and the strict sense is what the budget meeting hears. This guide's convention, which the deliverable in Section 7.13 will follow, is to use "driver" only with its probation officer present: a driver, in this chapter, is a variable that predicts the outcome in a specified model, and every sentence that promotes it to a lever must say what additional evidence performed the promotion.

CONCEPT

"Driver" Is a Causal Word on Probation

The deliverable is called a drivers analysis; the evidence is associational. Manage the gap explicitly. In writing and in meetings, let "is associated with," "predicts," and "differs by" carry the findings, and reserve "drives," "lifts," "causes," and "will produce" for claims backed by intervention evidence or clearly labeled assumption. The discipline is not pedantry; it is the difference between the analysis the data performed and the analysis the meeting will remember. Section 7.15 examines what happens — and to whom — when the verbs drift.

Source: Course concept developed for this guide, informed by Freedman (1991).

None of this makes association worthless — the opposite. Association is evidence: it nominates candidate levers, sizes relationships, falsifies confident stories (a lever uncorrelated with the outcome has a steep hill to climb), and disciplines the prioritization of the expensive causal tests to come. Chapter 2's evidence discipline applies: the analyst's job is not to refuse conclusions but to match the confidence of the sentence to the strength of the evidence. The rest of this chapter builds the machinery for measuring association well; the standing distinction rides along in every section, and the ethics section will show it is the chapter's real subject.

7.3 Covariance and Correlation: Putting a Number on Linear Association

Measurement starts, as it did in Chapter 3, with a definition decision: what number should summarize "moving together"? The natural starting point is covariance. Take two variables at the customer grain — frequency and monetary value from the feature table Chapter 6 built. For each customer, measure how far each variable sits from its own mean, and multiply the two deviations. A customer above average on both, or below average on both, contributes a positive product; a customer above on one and below on the other contributes a negative product. Covariance summarizes these products across customers: positive when the variables tend to sit on the same side of their means, negative when they sit on opposite sides, near zero when there is no consistent pattern.

One detail in that summary decides whether the printed number will match the definition, and this guide fixes it now rather than letting the software fix it silently. There are two covariances, and they differ in their denominator. The population covariance is the mean product of paired deviations across a full population: sum the products, divide by the number of members. The sample covariance estimates that quantity from a sample by dividing the sum of products by one fewer than the number of observations. The distinction is not decorative here, because the labs run on pandas, and pandas reports the sample version by default — so a chapter that defines covariance as "the average product" and then prints a number computed with the smaller denominator has taught two different quantities under one name.

DEFINITION

Covariance

The population covariance of two variables is the mean product of their paired deviations from their respective means, positive when the variables tend to deviate in the same direction and negative when they tend to deviate in opposite directions. The sample covariance estimates it from n observations by dividing the sum of the paired deviation products by n − 1. Its magnitude depends on the units of both variables, which makes it a direction detector rather than a strength measure. In this chapter, pandas reports the sample covariance and the sample correlation unless stated otherwise.

Source: Adapted from Rodgers and Nicewander (1988).

Covariance has the right sign and the wrong scale, and the reason echoes Section 6.7's tyranny of units: the products are denominated in the units of both variables at once — order-

dollars, in the frequency–monetary case — so the number's size says as much about measurement units as about the relationship. Chapter 6 solved this problem for distance by standardization, and correlation is the same solution applied to covariance: divide by the spread of each variable. The result is the Pearson correlation coefficient, written r, and the division buys three properties that made it the lingua franca of association. It is unit-free: r between frequency and monetary is the same whether monetary is in dollars or cents. It is bounded: r always lies between −1 and +1, with the endpoints reached only when the points fall exactly on a line. And it is symmetric: r of X with Y equals r of Y with X — a property to remember, because "drivers" language is directional and r is not.

DEFINITION

Pearson Correlation Coefficient (r)

The sample Pearson correlation coefficient is the sample covariance of two variables divided by the product of their two sample standard deviations. Because the same n − 1 denominator appears in the covariance and in each standard deviation, the correlation is unaffected by the choice of denominator even though the covariance is not. It measures the direction and strength of linear association on a fixed, unit-free scale from −1 (perfect negative linear association) through 0 (no linear association) to +1 (perfect positive linear association).

Source: Adapted from Pearson (1895) and Rodgers and Nicewander (1988).

That last property is worth one further sentence, because it explains why the sample-versus-population distinction matters for covariance and not for r. Dividing a covariance by two standard deviations cancels the denominator: whichever convention is used, provided it is used consistently in all three quantities, the ratio is identical. So the covariance a student computes by hand can disagree with pandas in the third digit while the correlation agrees exactly — which is itself a small, memorable lesson about where conventions bite and where they do not, and Lab 7.1 arranges for the student to meet it.

Reading r takes calibration, and Table 7.1 supplies the working scale this guide uses — with the warning that context outranks any fixed ladder. In customer-level marketing data, where individual behavior is noisy by nature, correlations above 0.5 between distinct behaviors are strong findings and correlations near 0.9 should trigger suspicion rather than celebration: behaviorally distinct quantities rarely track each other that tightly, and near-perfect r usually means the two columns are the same fact wearing two names — monetary and its own components, a metric and its numerator, the arithmetic relatives Section 7.10 will flag for the model. The table is introduced here and consulted throughout the labs.

Table 7.1

Reading r in customer-level marketing data

Range of |r|	Working label	Reading in this guide	Standing caution
0.0–0.1	Negligible	No linear association worth a sentence	Could still hide a nonlinear pattern (Section 7.4)
0.1–0.3	Weak	Detectable; matters mainly in aggregate	Easily produced by a shared confounder
0.3–0.5	Moderate	A real behavioral relationship	Plot before believing; check outlier influence
0.5–0.8	Strong	Unusually tight for distinct behaviors	Ask what shared machinery produces it
0.8–1.0	Suspect	Rare between genuinely distinct quantities	Often two names for one fact; check derivations

Three boundary facts complete the working picture. First, r is undefined when either variable has zero variance: the formula divides by both standard deviations, so a variable that never moves has no correlation with anything, and software will report NaN rather than zero. The practical form of this rule is a warning about near-misses — inside a narrow subgroup, a variable with almost no spread yields an r that is numerically defined and substantively meaningless, which Lab 7.1's within-segment exercise is built to expose. Second, r is a customer-grain statistic here, and the grain declaration discipline of Section 3.3 applies with full force: the correlation between frequency and monetary across customers is a different fact from the correlation between daily orders and daily revenue across days, and Section 7.5 will show that the daily version imports the calendar as a hidden third variable. Third, r is a two-variable statistic, and modern practice rarely stops at one pair: the correlation matrix — every pairwise r among a set of variables, arranged in a grid — is the standard first exhibit of a drivers analysis, and it is also the exhibit most reliably misread. The matrix's seductions are specific enough to deserve their own treatment, and they open the next section.

7.4 What Correlation Misses: The Always-Plot Discipline

The correlation coefficient compresses a full scatter of customers into one number, and compression loses information by design. This section catalogs what is lost — and installs the discipline that recovers it, which costs one line of code and has been part of this guide's toolkit since Chapter 4's exploratory charts: plot the data before, or at latest alongside, any correlation you intend to repeat in a meeting.

The canonical demonstration is Anscombe's quartet: four small datasets constructed so that the means, variances, correlation (r = 0.816), and fitted regression line are essentially identical

across all four — while the scatterplots show four utterly different worlds: a well-behaved linear cloud; a clean curve that is not a line at all; a perfect line with one outlier that has dragged the fitted slope; and a pillar of identical x-values propped into a "relationship" by a single distant point (Anscombe, 1973). Anscombe built the quartet to argue against the habit of computing without graphing, and half a century later the argument has sharpened, because the computing is now delegated: an AI assistant asked for a correlation matrix returns the numbers, not the shapes, and narrates the numbers fluently. The quartet's lesson, restated for this guide: identical statistics do not mean identical evidence, and the scatterplot is the cheapest audit in analytics.

CONCEPT

Always Plot — the Anscombe Discipline

No correlation, and later no regression, enters a deliverable without its scatterplot having been seen by the analyst. The plot answers four questions r cannot: Is the relationship linear, so that r is even the right summary? Is it produced by a handful of extreme customers? Is it one relationship, or several populations overlaid? Is there any pattern at all in a region that matters — or is the association concentrated where the business isn't? One glance answers all four. The discipline is mechanical, it is the chart-reading skill of Chapter 4 pointed at a new target, and its absence is detectable in a deliverable the way a missing verification log was in Chapter 4: confident numbers, no evidence anyone looked.

Source: Course concept developed for this guide, informed by Anscombe (1973).

Three specific losses matter most on StyleCraft's data, and each is a designed feature of the base the labs will meet. Nonlinearity first: r measures linear association only, and marketing relationships are routinely curved. Recency and future spending, engagement and response, tenure and frequency — each tends to rise or fall steeply in one region and flatten in another, and a straight-line summary of a curve understates the relationship where it is strong and overstates it where it is flat. An r near zero, in particular, does not mean "no relationship"; it means "no linear one," and the plot decides which.

Outlier sensitivity second: r is built from products of deviations, so a single customer far from both means contributes an enormous product — Chapter 5's concentration analysis showed that StyleCraft's top decile holds an outsized share of revenue, and those whales can single-handedly manufacture, inflate, or reverse a correlation among spend variables. The lab's routine, executed in Code 7.5: compute r with and without the top spend decile, and if the story changes, the story was about the whales.

Third, and most important on this dataset: mixed populations. Chapter 6 established that the base contains four segments with different behavioral regimes. A correlation computed across the pooled base averages over regimes that may individually show different relationships — frequency and basket size can be positively related within every segment and negatively related across the pool, because the pool's variation is dominated by the between-segment contrast of frequent small-basket urban customers against infrequent large-basket suburban ones. This is the correlational face of the aggregation lesson of Section 3.3.1 — the claim changes with the grain,

and here, with the population sliced — and the working rule is the one Chapter 5 used for every summary: compute the pooled number, then compute it within segments, and treat disagreement between the two as a finding about structure, not a nuisance.

The correlation matrix concentrates all of these hazards in one exhibit, plus two of its own. Table 7.2 collects the misreadings this guide's students most reliably commit on their first matrix, with the corrections the sections of this chapter supply. The matrix remains the right first exhibit — it is the fastest wide-angle view of a feature table — but it is a screening instrument, and Table 7.2's right-hand column is the protocol for everything it flags.

Table 7.2

Correlation-matrix misreadings and their corrections

Misreading	**Why it is wrong**	**Correction**
"The biggest r is the biggest driver"	r is symmetric association, not directed influence; and the biggest r is often an arithmetic relative	Rank candidates only after Section 7.8's model adjusts for the others; check derivations first
"r ≈ 0, so this variable doesn't matter"	Zero linear association can hide a strong curve or offsetting subgroup patterns	Plot it; compute within segments (Section 7.4)
"These two drivers both correlate with spend, so both add value"	They may carry the same information — multicollinearity in waiting	Check their correlation with each other and their variance inflation factors; Section 7.10
"r = 0.9 — excellent, strong finding"	Near-perfect r between distinct behaviors usually flags a definitional overlap	Trace both columns' derivation rules (Table 6.5) before celebrating
"The matrix shows what to change to move spend"	Every cell is observational; no cell survives Section 7.2's rivals	Treat the matrix as a nomination list for modeling and, later, testing

One habit closes the section, borrowed whole from Chapter 1's workflow: predict before you compute. Before running the matrix in Lab 7.1, the analyst writes the expected sign of every correlation with spend — recency negative, engagement positive, and so on — from marketing logic and from Chapter 5's certified findings. The matrix then grades the predictions, and every surprise is an assignment: a plot, a derivation check, or a within-segment recomputation. Surprises that survive the assignment are findings. This is predict-then-verify (Section 1.7) applied to association, and it is the small-scale rehearsal of the audit this chapter's AI section will formalize.

7.5 Confounding and Spurious Correlation

Section 7.2 listed the rivals behind every association; this section develops the one that does the most damage in marketing, because marketing's operating environment manufactures it on a schedule. An association between X and Y is confounded when a third variable Z, present before either, influences both — creating or distorting a correlation between X and Y that partly or wholly reflects Z's double influence rather than any relationship between X and Y themselves. When the observed association has no direct component at all — when it exists only through Z or through coincidence — it is called spurious: real as a number, empty as evidence (Simon, 1954).

The definition has to be tighter than the one students usually carry, and the looseness is worth naming because it produces real errors. "A variable related to both" is too broad a test. A variable that sits on the pathway between X and Y — a mediator — is related to both, and adjusting for it removes exactly the part of the relationship the analysis was trying to see. A variable that is a common consequence of both — a collider — is also related to both, and adjusting for it can manufacture an association where none existed. This chapter does not develop causal diagrams; at working level the repair is to define the confounder by its position rather than by its correlations, and to treat a double correlation as a flag that starts an inquiry rather than as a verdict that ends one.

DEFINITION

Confounding and Spurious Correlation

A confounder is a pre-treatment variable that influences, or helps determine, both the predictor or exposure and the outcome, creating or distorting their observed association. Being correlated with both variables is a warning sign, not by itself proof that a variable is a confounder: a variable on the pathway between them, or a common consequence of them, is also correlated with both and calls for different handling. An association is spurious when it reflects only the operation of confounders or chance, so that neither variable responds to the other. Detecting confounding requires variables and reasoning beyond the correlated pair itself; nothing in the pair's own correlation reveals it.

Source: Adapted from Simon (1954) and Freedman (1991).

The definition's closing sentence is the operational point: the correlation between X and Y, examined ever more closely, will never confess that Z exists. Confounders are found by thinking about the world, not by staring at the pair — which is why domain knowledge is a technical input to this chapter, not decoration around it. Public collections of absurd examples make the logic memorable — time-series pairs like per-capita cheese consumption tracking engineering doctorates, correlated at 0.9 and connected by nothing but parallel drift (Vigen, 2015) — but the absurd cases are the safe ones, because no one budgets on cheese. The dangerous confounders are the plausible ones, and marketing's most reliable confounder is the calendar.

Here is the seasonality example this guide will reuse for the rest of Part II. Take StyleCraft's daily grain: marketing activity by day and revenue by day, across the 24-month window. Email

volume and revenue will correlate strongly and positively across days. The mechanism is only partly the one the email claim wants: StyleCraft, like every retailer, concentrates its marketing where demand already concentrates — November and December, the designed holiday lift of the dataset's own revenue curve, plus the back-to-school swell. The calendar raises sends (campaign planners schedule into the season) and raises revenue (customers buy for the holidays regardless of email), and the daily correlation gratefully records the two effects as one. A naive read — "days with more email have more revenue; email drives revenue; send more" — is the confounded read, and its recommendation self-destructs in February. The correction pattern generalizes far beyond email: compare within season rather than across it (does extra email correlate with extra revenue among December days?), bring the confounder into the analysis as a variable — which is precisely the job Section 7.8's multiple regression takes up — or change the design so the confounder cannot operate, which is Chapter 11's business. Note what the example does to the email claim from the opening case: the claim's customer-grain version (engaged customers buy more often) and its day-grain version (heavy-send days sell more) are different claims with different confounders, and neither, unexamined, supports the recommendation to double frequency.

Confounding also explains the finance claim's trap, and Chapter 6 already built the machinery to see it. Discount exposure and total spend correlate negatively across the pooled base — the pattern is real and the labs will confirm it. But segment membership is associated with both discount exposure and total spend: the Discount-Dependent segment concentrates its purchases on promotion and spends less in total than the Urban Loyal Core, while the Core buys at full price and spends the most. The pooled correlation records that composition as if it were a consequence — the same between-group mechanism as Section 7.4's mixed-population warning. The segment label earns its place here as a descriptive stratifier, because it exposes the customer composition sitting behind the pooled association. It does not automatically earn the stronger title. Chapter 6 estimated that label from same-window behavioral features that include monetary value and discount share — the very outcome and the very exposure this example is examining — so the label is not an independently measured pre-treatment variable. It is partly constructed from the behaviors whose relationship is under study, and treating it as a clean confounder or a causal control would be reasoning in a circle: controlling for a grouping that was defined, in part, by the exposure and the outcome themselves. This is worth stating plainly because the reflex it corrects is common. An analyst who has just built a segmentation naturally reaches for it as the obvious control, and the reach is right about the arithmetic and wrong about the warrant.

What the segment label does support is a specific and useful list: exposing the composition behind a pooled number, describing heterogeneity, testing whether pooled and within-group patterns disagree, reproducing certified descriptive structure with a new instrument, and improving an explanatory model's descriptive fit. Whether discounts reduce, leave unchanged, or increase any individual customer's spending is simply not answered by the pooled r, and it is not answered by the stratified version either. The within-segment recomputation diagnoses how composition shapes the pooled association; the fitted model with segment dummies (Section 7.9)

is a second descriptive adjustment, useful for examining heterogeneity and insufficient for causal identification. A genuine pre-treatment confounder — the calendar, tenure at the time of exposure, a demographic measured at signup — is a different kind of variable, and Section 7.8 returns to what controlling for one can and cannot buy. The finance claim may still be right. The point of this section is that its current evidence cannot tell.

What correlation cannot do, then, is settle any of the opening case's three claims. What it has done, by the end of Lab 7.1, is discipline them: signs and strengths measured, shapes inspected, whales counted, segments separated, and the calendar named as a standing suspect. The next step is the instrument that examines relationships jointly instead of pairwise — and whose output format, the coefficient, is the one the meeting will actually quote.

7.6 Simple Linear Regression: The Line and Its Coefficients

Correlation answers "do these move together, and how tightly?" The budget meeting needs a different quantity: "if customers differ by one unit of this, how much do they differ in spend?" That question asks for a rate of exchange between variables, denominated in their native units — dollars per order, dollars per day of recency — and the instrument that estimates it is linear regression. The name and the idea are old: Galton (1886), studying heredity, noticed that extreme parents had less extreme children — values "regressing" toward the mean — and the line he drew through that cloud of points became the general tool for summarizing how one variable runs with another. This guide develops it as working machinery, one step above intuition and well short of formalism, per the program's no-prerequisite design.

Simple linear regression models one outcome variable Y — the response — as a straight-line function of one predictor X, plus an error term that absorbs everything the line does not capture. The fitted model is an equation of the form $Y = b_0 + b_1X$, and its two coefficients are the deliverable. The intercept b_0 is the model's estimate of Y when X is zero — sometimes meaningful (predicted spend at zero discount exposure), sometimes a mathematical anchor outside the data's range (predicted spend at zero days of tenure describes a customer who signed up this morning), and the analyst states which. The slope b_1 is the working number: the estimated difference in Y associated with a one-unit difference in X. Its units are always outcome-units per predictor-unit, and this guide's standing rule is that a coefficient is not read until its units are said aloud. A slope of −0.85 in a regression of monetary value on recency_days is not "negative 0.85"; it is "eighty-five cents less total spend per additional day since last purchase, across the observed range" — a sentence with dollars, a direction, a denominator, and a scope.

DEFINITION

Simple Linear Regression

Simple linear regression models a numeric response variable as a linear function of one predictor: an intercept (the estimated response when the predictor is zero) plus a slope (the estimated difference in response per one-unit difference in the predictor), with an error term absorbing what the line does not capture. Coefficients are estimated from data and read in the native units of the variables.

Source: Adapted from James et al. (2021).

Where does the line come from? Any line through the scatter leaves gaps between itself and the actual customers, and each gap has a name: the residual, the customer's actual Y minus the value the line predicts for their X. Positive residual, the model underestimated them; negative, it overestimated. The least squares criterion chooses the one line that makes the sum of squared residuals as small as possible — squared for the same reason inertia squared distances in Section 6.8: large misses are penalized disproportionately, and the arithmetic is computable. Uniqueness carries a condition worth attaching now, because Sections 7.9 and 7.10 will meet the exception. When the model's predictor matrix has full column rank — no predictor is an exact linear combination of the others — least squares yields a unique coefficient vector. When predictors are perfectly redundant, as in the dummy-variable trap, software may still return a fitted result by numerical convention, but the individual coefficients are not uniquely identified and should not be interpreted. The intuition worth keeping is the balance-point image: the least squares line is the scatter's center of gravity stretched into a slope, it passes through the point of means, and — like every mean since Chapter 5 — it is sensitive to extreme points, which is Anscombe's third panel and the whales again. The formalism beyond this is machinery the software owns; what the analyst owns is the criterion and its sensitivities.

DEFINITION

Residual

A residual is the difference between an observation's actual response value and the value the fitted model predicts for it: actual minus predicted. Residuals are the model's itemized account of what it missed, and their patterns — trends, curves, funnels, clusters — are diagnostic evidence about where the model's form is wrong.

Source: Adapted from James et al. (2021).

Residuals deserve the second definition box because they are this chapter's diagnostic instrument, the regression counterpart of Chapter 4's verification log. Software can almost always return a fitted result; whether its coefficients are uniquely identified and substantively meaningful depends on the model's design, and whether the line is an honest summary shows up in what it leaves behind. The residual plot — residuals against predicted values, or against the predictor — should look like weather: patternless scatter around zero. A curve in the residuals means the relationship was not a line (the model form is wrong, not the world); a funnel that

widens with predicted spend means the model misses big spenders by more than small ones, a near-universal feature of revenue data that Chapter 5's skew already predicted, and one with a direct consequence for the uncertainty columns that Section 7.10 will take up; a cluster of large residuals that turns out to be one segment means the single line is averaging over populations that want different lines — the mixed-population lesson again, now visible as diagnostic evidence and fixable with Section 7.9's machinery. Reading residual plots is a chart-reading skill in exactly Chapter 4's sense — the objects were introduced there; this chapter supplies what to look for when the chart is about a model.

Regression and correlation are close arithmetic relatives — for one predictor, the slope is r rescaled by the ratio of standard deviations, so they always agree in sign — but they answer different questions and fail differently. r is unit-free and symmetric: it grades the tightness of the cloud. The slope is unit-bearing and directional in form: it prices the relationship in decision units. That pricing is why the meeting quotes slopes, and why the analyst's caution transfers whole: the regression of Y on X and the regression of X on Y are different lines answering different questions, and nothing in either line knows which variable is in charge — the model's direction is the analyst's framing, not a discovery. Every rival explanation from Section 7.2 survives the fitting of a line untouched.

7.7 How Good Is the Line? R^2, Fit, and Usefulness

A fitted line and honest units are not yet a report card. The standard single-number grade is R^2, the coefficient of determination: the share of the response variable's variation that the model accounts for. The construction is worth seeing once in words. Total variation is the sum of squared deviations of Y around its own mean — how much there is to explain, the baseline ignorance of a model that predicts the mean for everyone. Residual variation is the sum of squared residuals — the ignorance that remains after the model. R^2 is the fraction of the original variation the model removed: R^2 = 1 minus (residual variation ÷ total variation).

Three properties are usually stated as though they were laws of arithmetic, and they are better stated with their conditions attached, because students meet software that reports R^2 under other conventions. For in-sample ordinary least squares models fitted with an intercept to the same observations, R^2 lies between 0 and 1 and cannot decrease when predictors are added to a nested model; and in a simple one-predictor model with an intercept, R^2 equals r^2. Remove the intercept and the arithmetic changes: statsmodels, for example, computes a centered R^2 when a constant is present and an uncentered version when it is absent, so the two are not comparable and the uncentered figure is routinely and misleadingly larger (Seabold & Perktold, 2010). Change the observations, or evaluate the model on data it was not fitted to, and the guarantee that R^2 stays inside the unit interval disappears with it — a fact Chapter 8 will need immediately. The conditions are not fussiness. They are the difference between a number that means what the sentence says and a number that merely looks familiar.

DEFINITION

R^2 (Coefficient of Determination)

R^2 is the proportion of a response variable's total variation, measured around its mean, that a fitted model accounts for: one minus the ratio of residual variation to total variation. It compares the model to the baseline of predicting the mean for every observation. For in-sample ordinary least squares models fitted with an intercept to the same observations, R^2 lies between 0 and 1 and cannot decrease when predictors are added; in a simple one-predictor model with an intercept, it equals r^2. Because it cannot decrease, a rising R^2 is not, by itself, evidence of a better model.

Source: Adapted from Wooldridge (2020) and James et al. (2021); software conventions per Seabold and Perktold (2010).

Notice what R^2 secretly is: a baseline comparison, in exactly Section 2.7's sense. "The model accounts for 34 percent of spend variation" means "against the interpretive baseline of guessing the mean, residual error shrank by a third." This guide flags the connection because baselines are a running thread — Chapter 8 will operationalize them for prediction and Chapter 10 will extend them to time — and because remembering what R^2 is a comparison against inoculates against its two standing misreadings. The first misreading is that high R^2 certifies a good model. It does not: R^2 measures fit to this data, and Anscombe's quartet holds four identical R^2 values over four different truths; a curve, an outlier, or a leaking variable can each buy fit without buying validity. In customer-level data the suspicion runs mostly the other way — an R^2 above roughly 0.9 on individual behavior should trigger the Table 7.1 reflex, because individual humans are not that predictable, and near-perfect fit usually means the model was handed the answer, the trap Code 7.18 demonstrates on purpose.

The second misreading is that low R^2 certifies a useless model. It does not: R^2 grades total variance accounted for, while decisions often turn on the slope. A model accounting for 12 percent of spend variation can still estimate the recency slope precisely enough to price a win-back program — noisy individuals, reliable rates. In other words: fit is a property of the model against this data; usefulness is a property of the model against a decision, and the two judgments can disagree in both directions.

CONCEPT

Fit Is Not Usefulness

R^2 answers one narrow question — how much of the outcome's variation this model absorbs in this sample. Decisions rarely ask that question. They ask whether a specific coefficient is estimated well enough to act on, whether the model's form survives its residuals, and whether the relationship is stable enough to hold next quarter. A high-R^2 model can fail all three; a low-R^2 model can pass all three. Report R^2 — it is the honest disclosure of how much is not accounted for — but never let it stand alone as the verdict, and treat "too good" with the same suspicion as "too poor." The number that flatters the model is the number to audit first.

Source: Course concept developed for this guide, informed by Shmueli (2010).

7.8 Multiple Regression and the Ceteris Paribus Promise

The opening case's claims all founder on the same rock: each compares customers who differ in many things at once and attributes the difference to one of them. The instrument built for exactly this problem is multiple regression: one response, several predictors, fitted jointly. The model form extends naturally — $Y = b_0 + b_1X_1 + b_2X_2 + \dots$ — and least squares still chooses the coefficients that minimize squared residuals. What changes is the meaning of each coefficient, and the change is the entire reason the method matters: in a multiple regression, b_1 estimates the difference in Y per one-unit difference in X_1 among observations that share the same values of every other predictor in the model. The Latin tag is ceteris paribus — other things equal — and the operation is statistical control (Wooldridge, 2020).

One word has to be handled carefully here, because the natural English for statistical control is misleading. Textbook prose and classroom shorthand both reach for "matched" or "comparable customers," and ordinary least squares does not match anybody. It fits a surface to the whole cloud and reports the surface's slope in one direction. The estimate is therefore a model-based comparison — an adjusted difference computed under the model's assumed functional form — and it can be produced even where few or no real customers share the relevant combination of covariate values. Where that happens, the model is extrapolating across a region the data do not populate rather than comparing genuinely similar customers, and nothing in the printed coefficient distinguishes the two cases. The honest phrasing, used throughout this chapter, is that a coefficient compares customers statistically, holding the included predictors constant within the model's assumed form.

DEFINITION

Multiple Regression

Multiple regression models a numeric response as a linear function of two or more predictors fitted jointly, so that each coefficient estimates the difference in the response per one-unit difference in its predictor, holding the other included predictors constant within the model's assumed functional form. It implements statistical control by adjustment rather than by matching: the comparison is model-based, extends only to the variables the model was given, and may extrapolate where the data contain few observations sharing the relevant combination of values.

Source: Adapted from Wooldridge (2020).

Applied to the app claim, the machinery earns its keep immediately. The raw comparison — app users versus non-users, a 2.4× spend ratio — mixes the app difference with every difference that travels with it: app users are younger, more urban, more loyalty-enrolled, longer-engaged, more email-active. Put spend on the left and app ownership on the right alongside recency, tenure, email opt-in, and segment membership, and the app coefficient now estimates the spend difference between app owners and non-owners adjusted for all of those. On StyleCraft's data the

labs will find that this adjusted difference is far smaller than the raw ratio implied — most of "2.4×" was the Urban Loyal Core's profile traveling with the app, not the app itself. That shrinkage from raw gap to adjusted gap is the most useful single exhibit a drivers analysis produces, and the meeting-ready sentence practically writes itself: "customers with the app spend more; most of that gap reflects who installs the app rather than anything the model can attribute to installing it."

The promise, however, has strict boundaries, and the guide states them as sharply as the promise itself. Ceteris paribus means other included things equal. The model holds constant what it was given and nothing else; every relevant variable not in the model remains free to vary between the compared customers and to smuggle its influence into the coefficients. This failure has a name — omitted variable bias — and a usable direction logic, provided the logic is stated with its scope. In the simplest linear case, one included predictor and one omitted variable, the direction of the bias follows the product of two relationships: the omitted variable's effect on the response, times its association with the included predictor. In a multiple regression the relevant association is narrower: it is the part of the omitted variable's association with the included predictor that remains after the model's other predictors are taken into account. That qualification matters in practice, because a variable that looks strongly related to the predictor in a raw cross-tabulation can be nearly unrelated to it once the model's other controls are in place, and the bias reasoning must follow the adjusted relationship rather than the raw one.

DEFINITION

Omitted Variable Bias

Omitted variable bias is the distortion of a fitted coefficient that arises when a variable related to both the response and an included predictor is left out of the model: the included predictor absorbs credit or blame belonging to the omitted one. In the simplest linear case, the direction of the bias follows the product of the omitted variable's effect on the response and its association with the included predictor. In a multiple regression, the relevant association is the part that remains after the model's other predictors are taken into account.

Source: Adapted from Wooldridge (2020).

If underlying brand engagement increases spend and is positively associated with email activity, then omitting it would tend to bias the email coefficient upward: under those assumptions the fitted model attributes to email part of an association that belongs to enthusiasm. Note the shape of that sentence. The direction follows from stated assumptions about the omitted variable, not from a causal fact the data have established — which is exactly what makes it usable, since the reasoning can be run before the missing variable is ever measured. The direction logic converts anxiety into analysis: for every coefficient the meeting will quote, the analyst lists the plausible omitted variables, reasons the sign of each bias, and reports which way the estimate leans. Statistical control is not a causal machine; it is a better class of comparison, and the honest sentence names both halves.

CONCEPT

The Coefficient Is an Adjusted Comparison, Not a Lever

A multiple regression coefficient describes a model-adjusted difference between customers, computed as though the model's other variables were held constant. It does not describe what happens when you move the variable, for three reasons that never go away: the adjustment extends only to the model's own variable list; the adjustment is a property of the fitted surface rather than of matched pairs of real customers, so it can be reported for combinations the data barely contain; and even a perfectly adjusted difference is still an association — the customers chose their own X. Read every coefficient as "customers who differ by one unit of X differ, on average, by b in Y, holding the listed variables constant" — and treat any restatement beginning "if we increase X" as a new claim requiring new evidence. This box is the chapter's verification theme applied to language, and Section 7.15 shows what its violation costs.

Source: Course concept developed for this guide, informed by Wooldridge (2020) and Freedman (1991).

Two practical notes complete the working level. First, model inputs are governed by measurement, exactly per Section 3.4: numeric predictors enter as they are; ordinal predictors (loyalty tier) enter either as ordered scores — asserting equal steps, an assertion to declare — or as categories per Section 7.9; nominal predictors (segment, metro, channel) must enter as dummy variables, because arithmetic on category codes is the level-of-measurement error Chapter 3 warned against. Second, adding predictors is not free: each addition changes every other coefficient's meaning ("holding constant" now includes the newcomer), R^2 rises mechanically whether or not the addition carries information, and predictors that share information destabilize one another — the multicollinearity of Section 7.10. The model is a specified instrument, not a bucket; every variable in it should be there because the analysis's question, written per Section 2.5's specification discipline, put it there.

7.9 Dummy Variables, Reference Categories, and Interaction Terms

The drivers question at StyleCraft is inseparable from the segments, and the segments are categories: four names, no natural ordering, no units. Regression's arithmetic needs numbers, and the encoding that supplies them honestly is the dummy variable: an indicator column equal to 1 when the observation belongs to a category and 0 otherwise. A categorical variable with c categories enters the model as $c - 1$ dummies; the category left without a dummy is the reference category, and it is not omitted from the analysis — it becomes the analysis's baseline. Every dummy coefficient is then read as an adjusted difference against the reference: with Urban Loyal Core as the reference, the Suburban Occasion dummy's coefficient is the estimated difference in the response between Suburban Occasion customers and Urban Loyal Core customers, holding the model's other variables constant. One category must always be left out — including all c dummies alongside an intercept gives the model two ways to say the same thing, a perfect redundancy that breaks the fitting (the dummy variable trap, multicollinearity in its terminal form).

What the intercept means in such a model needs stating precisely, because the convenient shorthand is wrong as soon as the model contains anything else. It is tempting to say that the reference category's average outcome is carried by the intercept. That is true only in the special case of a model containing nothing but that one set of dummies. Add a numeric predictor and the intercept becomes the model's expected response for the reference category when that numeric predictor equals zero — which may describe no customer who exists. Add a second categorical predictor and the intercept refers to the reference level of both. The reference group's observed mean and the model's intercept are different quantities, and confusing them is how an analyst ends up reporting a negative baseline spend with a straight face.

DEFINITION

Dummy Variable and Reference Category

A dummy variable encodes category membership as an indicator: 1 if the observation is in the category, 0 otherwise. A categorical predictor with c categories enters a regression as c − 1 dummies, and the reference category is represented when all dummies for that predictor equal zero. With an intercept in the model, each dummy coefficient is an adjusted difference from that reference category. The intercept represents the model's expected response for the reference category when all other predictors are at their coded zero or reference values — which is the reference group's observed mean only in the special case of a model containing no other predictors.

Source: Adapted from Wooldridge (2020).

The choice of reference category changes no substance — every model with any reference fits identically and tells the same story — but it changes every printed number and therefore every sentence, so it is chosen for readability and declared. The guide's convention: the reference is the largest or most decision-central category, so that coefficients read as "compared to our core." Table 7.3 shows the encoding for StyleCraft's four segments with Urban Loyal Core as reference, and the reading column is the deliverable skill: students who can compute dummies and cannot read them against the reference produce the most common wrong sentence in applied regression — "the model shows Suburban Occasion customers spend $X" instead of "$X more or less than the Core, adjusted for the model's controls."

Table 7.3

Dummy encoding of the four segments, with Urban Loyal Core as the reference category

Segment (from Chapter 6)	seg_ suburban	seg_ discount	seg_new	How the coefficient reads
Urban Loyal Core	0	0	0	Reference: no coefficient of its own. The model's expected response for this group appears only when every other predictor is also at zero or at its own reference level.

Segment (from Chapter 6)	seg_ suburban	seg_ discount	seg_new	How the coefficient reads
Suburban Occasion	1	0	0	Adjusted difference from the Urban Loyal Core, holding the model's other predictors constant
Discount-Dependent	0	1	0	Adjusted difference from the Urban Loyal Core, holding the model's other predictors constant
New / At-Risk	0	0	1	Adjusted difference from the Urban Loyal Core, holding the model's other predictors constant

Dummies let group membership shift the model's level. The next construction lets it shift the model's slopes — because the sharpest drivers questions are not "does this lever matter?" but "for whom does the relationship differ?" An interaction term is a new predictor built as the product of two others, and its coefficient estimates how the association between one variable and the response differs across values of the other.

Interactions have to be read conditionally, and the reading is mechanical enough to state as a rule. Suppose the model contains app use, loyalty enrollment, and their product, both coded 0 or 1. Then the app coefficient is the app–spend association among customers whose loyalty enrollment equals zero — the non-enrolled. The loyalty coefficient is the loyalty–spend association among customers whose app use equals zero. And the interaction coefficient is the difference between the app association among the enrolled and the app association among the non-enrolled: the app association for enrolled customers is the sum of the app coefficient and the interaction coefficient. Nothing in the model reports an "overall" app association once the interaction is present, and an analyst who quotes the app coefficient as though it were one has quoted a conditional number as an unconditional claim.

DEFINITION

Interaction Term

An interaction term is a predictor formed as the product of two others, whose coefficient estimates how the association between one predictor and the response differs across values of the other. A model with an interaction no longer has one slope per variable; it has slopes that depend on context. Each main-effect coefficient is the association for its variable when the interacting partner equals zero, and the association at any other value of the partner is obtained by adding the interaction coefficient the appropriate number of times.

Source: Adapted from Wooldridge (2020) and James et al. (2021).

The registry example is the one the budget meeting cares about, and this guide names it one way and holds to it: loyalty enrollment × app-use status, with app use serving as the selected channel-engagement indicator. The dataset's specification plants a loyalty-by-channel structure,

and app use is the channel variable StyleCraft's schema actually carries at the customer grain, so the model estimates the planted structure through that indicator rather than through a channel construct the table does not contain. The question the interaction answers is whether the association between app engagement and spend differs by loyalty enrollment — and the honest report of the answer is two conditional associations in dollars, not one significance verdict.

Two standing cautions travel with every interaction. The main-effect coefficients change meaning the moment their interaction enters, which is the conditional reading just given, and it is the single most common source of misquotation in applied regression. And interactions multiply the model's appetite for data: estimating a separate association per group requires enough customers in each combination of the two variables, a substantiality check Chapter 6's segment sizes already supplied and Code 7.15 re-checks by printing the cell counts before the model is fitted. Where an interaction is materially nonzero, the finding to write is descriptive and conditional — "the association between discount exposure and spend differs by loyalty status" — rather than the causal restatement that a slide will reach for, and the exercises test the difference.

7.10 Reading Regression Output as Evidence: p-Values, Confidence Intervals, and Multicollinearity

A fitted model arrives as a table: one row per coefficient, and columns the software prints whether or not anyone can read them. This section supplies the working-level reading of the columns that matter after the estimate itself — the uncertainty columns — and the two structural judgments that govern how much any of it means: whether the predictors undermined each other, and what job the model was built to do. Everything here is calibrated to the analyst's seat: enough to weigh evidence and defend a table in a meeting, on the explicit understanding that the full statistical machinery belongs to later coursework.

Every coefficient is an estimate from one sample of customers, and would come out somewhat differently on another sample — the fitted $41 per order is the center of a range of values the data cannot distinguish, not a constant of nature. The standard error measures that sampling wobble, and the two working instruments are built from it. The 95 percent confidence interval is the honest way to report a coefficient: the range of values compatible with the data at conventional confidence — "$34 to $48 per order" — whose width is the finding about precision, and whose reading follows Chapter 5's spread discipline: a wide interval around a large estimate says "big but poorly pinned down," and a narrow interval around a small one says "small and we know it."

Before either instrument can be trusted, one assumption behind the printed standard errors has to be checked, and Section 7.6 already predicted its failure. Ordinary least squares computes standard errors, by default, on the assumption that the residual variance is the same across the range of fitted values. Revenue data rarely obliges: the residual plot funnels, because the model misses big spenders by more than small ones. That is heteroskedasticity, and its consequence is specific — it does not bias the fitted coefficients, and it does distort the standard errors, so the

intervals and p-values built from them are wrong in a direction the default output does not disclose. The repair is a different covariance estimator, requested at fit time. Statsmodels defaults to a nonrobust estimator and supports several robust alternatives; this guide uses HC3, which performs well in samples of the size marketing analysts actually work with (MacKinnon & White, 1985; Seabold & Perktold, 2010). A chapter that teaches students to expect a residual funnel and then quotes nonrobust intervals two pages later would be teaching the diagnosis and skipping the treatment, so every model in Lab 7.2 is fitted with robust standard errors and Code 7.9 states plainly what changes and what does not: the coefficients and the predictions are identical, and the standard errors, confidence intervals, and p-values are not.

The coefficient's p-value answers a narrower question than either instrument above, and it is worth stating exactly, because the loose version students carry — "the probability of getting an estimate this far from zero by chance" — omits the machinery that produces it. The software computes a test statistic from the coefficient and its estimated standard error, and refers that statistic to a distribution implied by the null hypothesis and the model's assumptions. Small p-values say the data are hard to reconcile with "no association, given this model"; they say nothing about size, importance, or cause. The American Statistical Association's consensus statement, issued after decades of misuse, compresses to three warnings this guide adopts: a p-value is not the probability that a hypothesis is true, statistical significance is not practical significance, and no single p-value should by itself determine a scientific, business, or policy conclusion (Wasserstein & Lazar, 2016). On eight thousand customers, near-trivial associations clear 0.05 easily — at this sample size, significance is table stakes, and the analyst's questions are the interval's location and width in decision units. The asterisk column is where reading starts, not where it ends.

DEFINITION

Coefficient p-Value and Confidence Interval

A p-value is the probability, assuming the stated null hypothesis and statistical model are correct, of obtaining a test statistic at least as extreme as the one observed. For a regression coefficient the null hypothesis is that the coefficient is zero given the model's other variables, and the test statistic is computed from the estimate and its standard error — so the p-value inherits every assumption behind that standard error, including the assumption about residual variance that robust estimators relax. A 95 percent confidence interval is the range of coefficient values compatible with the data at conventional confidence; its center is the estimate and its width is the precision. Together they grade the strength of evidence for a relationship, never the relationship's causal status.

Source: Adapted from Wasserstein and Lazar (2016) and Wooldridge (2020).

The first structural judgment is multicollinearity: predictors that carry overlapping information. When two variables move together tightly — frequency and monetary; email opt-in and app use; tenure and recency in a base that grew in waves — the model struggles to apportion the shared movement between them, and the symptoms are characteristic: coefficients with large standard errors and unstable values that swing when a variable enters or leaves; individually

"insignificant" predictors in a model whose joint fit is strong; occasionally a sign that flips against all descriptive evidence.

The predictor correlation matrix is the first diagnostic, read with Table 7.1's "suspect" row, and it is not sufficient by itself. Pairwise correlations detect only pairwise overlap, and the damaging case is often multi-variable: one predictor is closely approximated by a combination of several others while no single pairwise correlation looks extreme. The instrument that sees this is the variance inflation factor, which measures how much a coefficient's estimated variance is inflated by the predictor's relationships with all the other predictors together (Seabold & Perktold, 2010). This guide declines to publish a universal cutoff, because none survives contact with the literature. VIF is a diagnostic rather than a verdict: large values indicate that the model has difficulty separating a predictor's coefficient from the information carried by the others, and what counts as large depends on how precisely the coefficient needs to be estimated for the decision at hand. The paired instrument is the instability test — refit with and without the twin and watch the survivor's coefficient — which Code 7.17 runs as a specification-sensitivity check.

DEFINITION

Multicollinearity

Multicollinearity is strong overlap in the information carried by a model's predictors — whether between a pair or between one predictor and a combination of several others — which leaves their joint explanatory contribution largely intact but makes the division of that contribution among individual coefficients unstable and imprecise: inflated standard errors, estimate swings across specifications, and occasionally reversed signs. It is diagnosed from predictor correlations, variance inflation factors, and specification-sensitivity checks, and addressed by combining, dropping, or jointly reporting the entangled predictors. Multicollinearity may leave in-sample fit largely intact while destabilizing coefficients; it can also weaken out-of-sample prediction when the relationships among predictors shift.

Source: Adapted from Wooldridge (2020); variance inflation factor per Seabold and Perktold (2010).

The response to a multicollinearity finding is a decision, not a formula: combine the twins into one variable, drop one and say so, or keep both and report the pair's joint story while refusing to interpret their individual split. And one special case is a defect rather than a judgment call: a predictor that is arithmetically derived from the response produces a near-perfect fit that measures the derivation, not behavior. Regressing monetary value on the product of average order value and frequency is the pure form of the error, because that product is monetary by construction (Table 6.5), and Code 7.18 demonstrates it deliberately so that students meet the sensation of a model that is too good. Chapter 8 will name the general disease leakage; here it is caught by the derivation-tracing habit Table 7.2 installed.

The second structural judgment sorts everything this chapter has built, and it is the distinction the rest of Part II stands on: prediction and explanation are different modeling goals, and the same equation serves them differently (Shmueli, 2010). A predictive model has one obligation — accurate outputs on customers it has not seen — and it tolerates much that an explanatory

model cannot: entangled predictors, uninterpretable coefficients, variables that merely correlate with the future. An explanatory model has a different obligation — coefficients an analyst can defend as honest structure — and it tolerates lost accuracy in exchange: variables chosen by the specification's question, collinear twins separated or set aside, every retained coefficient readable in units with its bias direction reasoned. The drivers analysis of this chapter is explanatory work; the spend model of Chapter 8 will be predictive work; and most applied failures of regression trace to answering one goal's question with the other goal's model — quoting individual coefficients out of a kitchen-sink predictive fit, or crippling a forecasting model to keep its coefficients tidy. Table 7.4 sets the comparison, and the lab's final verification check asks which column the opening case's deliverable lives in.

Table 7.4

Prediction and explanation as distinct modeling goals

Dimension	Explanation (this chapter)	Prediction (Chapter 8)
Question	How do these variables relate to the outcome?	What will the outcome be for new cases?
Success measure	Defensible coefficients: signs, units, bias reasoning	Out-of-sample accuracy against a baseline
Variable selection	By the specification's question	By predictive contribution
Multicollinearity	A serious problem — it muddles attribution among coefficients	Often tolerable in-sample, but a risk when predictor relationships shift out of sample
Coefficient reading	The deliverable	Optional; often declined
Characteristic failure	Causal overclaiming (Section 7.15)	Leakage and overfitting (Chapter 8)

7.11 AI as a Regression Assistant

Chapter 6 marked the point where AI assistance began delivering whole analytic acts; this chapter is where the acts start arriving in the language of decisions. Prompted with the feature table and "what drives spend?", a current assistant will select variables, fit models, format the coefficient table, and — this is the part that changes the analyst's job — narrate the results in fluent managerial prose, complete with recommendations. The narration is the hazard. Code errors announce themselves; a regression that runs, fits, and is narrated with confident causal language fails silently, and it fails in the exact register the meeting will quote. The division of labor therefore follows the pattern set in Section 6.12, with this chapter's twist: the assistant may draft the machinery, and the analyst audits not only the numbers but the verbs.

Where the assistant genuinely helps, use it. Drafting the modeling code — formula syntax, dummy encoding, the residual-plot boilerplate, the tidy coefficient table with intervals — is legitimate delegation of mechanics this chapter has specified, provided the variable list comes from your written specification rather than the assistant's initiative. Generating the audit exhibits themselves — predicted-sign checklists, with-and-without-whales correlation runs, specification-sensitivity refits — is exactly the tedious-but-mechanical work assistants accelerate. Translating a finished, verified coefficient table into plain-language draft sentences is useful, provided every sentence is then checked against the verb discipline of Section 7.2. And the assistant is a genuinely good brainstorming partner for candidate confounders — "what variables might influence both discount exposure and total spend?" — because enumerating rivals is a breadth task, and breadth is what it has.

What must never be delegated are the judgments, and the failure modes are characteristic enough to name. The confident narration: asked for drivers, assistants return causes — "app adoption drives a $212 spend lift" — because fluent causal prose saturates their training data; the arithmetic may be flawless while the verb is unearned. The silent specification: the assistant chose which variables entered, which were dropped, what the reference category was, and whether an interaction was included — choices that change every number — and reported none of them as choices. The plausible-but-impossible coefficient: a sign or magnitude that contradicts certified facts (a positive recency coefficient; an intercept implying negative spend; a segment gap triple anything Chapter 5 measured), delivered with the same serene formatting as a correct one. The units slip: narrating a coefficient without units, or with the wrong ones — "recency matters most because its coefficient is largest" when recency is in days and frequency is in orders, the standardization confusion of Section 6.7 reborn in a coefficient table. The default covariance estimator: a model fitted on visibly funneling residuals and reported with nonrobust intervals, because nothing in the default workflow connects the diagnostic to the inference. The diagnostics skip: no residual plot, no mention of the funnel or the curve, because the default workflow fits and reports but does not examine. The unconditional reading of a conditional coefficient: a main effect quoted as an overall association while an interaction sits in the same model. And the significance theater: p-values below 0.05 presented as importance, exactly the misreading this section retired.

The audit routine that answers them is this chapter's verification theme, compact enough to run on every AI-fitted model and every colleague's model — four points, in order, before any narration is accepted. Table 7.5 states it as a working checklist; the AI in Practice box operationalizes it as a prompt-and-audit pattern.

Table 7.5

The four-point audit of a fitted regression

Point	The check	Fails when
1. Signs	Every coefficient's direction against written predictions and certified descriptives (Chapter 5)	A sign contradicts known structure with no confounding story offered
2. Magnitudes	Every coefficient's size against hand-computable anchors: group means, Chapter 5 gaps, plausible ranges	A coefficient implies differences no descriptive table shows, in either direction, with no explanation grounded in the model's variables
3. Units	Every coefficient restated aloud in outcome-units per predictor-unit; comparisons only on common scales; conditional coefficients read conditionally	A narration compares raw coefficients across different units, or quotes a main effect from a model containing its interaction as though it were unconditional
4. Diagnostics	Residual plots inspected; the covariance estimator matched to what the residuals show; variance inflation factors read; R^2 read against Section 7.7; specification sensitivity spot-checked	Fit is reported but residuals were never looked at, or intervals are quoted from a nonrobust fit on funneling residuals

AI IN PRACTICE

The Regression That Must Survive Its Audit

A two-prompt pattern with the audit between. Prompt one, the specification: paste the variable list with derivation rules (Table 6.5), the declared response, the reference category, the covariance estimator, and the exact model formula from your written spec, then ask for mechanics only — "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^2; change nothing in the specification; narrate nothing yet."

Then run the four-point audit of Table 7.5 yourself: check every sign against the predictions you wrote before fitting; check magnitudes against Chapter 5's certified gaps; say every coefficient in its units, and read any coefficient whose interaction is in the model as the conditional quantity it is; read the residual plot and confirm the covariance estimator matches what it shows.

Prompt two, only after the model survives: "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." Audit the verbs in the reply as strictly as the numbers — assistants reintroduce causal language at the narration step even when the numbers are clean. Predictions first, per Section 1.7; every exchange documented per Appendix D; and the analyst of record signs the sentences, not the software that drafted them.

Source: Course concept developed for this guide, informed by Wasserstein and Lazar (2016) and Shmueli (2010).

The deeper continuity: in Chapter 6 the assistant proposed structure and the analyst tested whether the structure was real; here the assistant proposes relationships and the analyst tests whether the relationships are honestly stated. The labs that follow are built so the audit has teeth. The dataset's specification planted the coefficient structure the labs recover — which variables carry spend, at what strengths, with one designed interaction — so an AI-fitted model's claims can be graded against a known answer, and Exercise 7.6 does exactly that with a deliberately underspecified prompt.

7.12 Hands-On Application in Python and Google Colab

The preceding sections built the instruments; this section points them at the opening case, in two labs that mirror the chapter's two halves. Lab 7.1 disciplines the raw material: correlations on the feature table, predicted before computed, plotted before believed, and stress-tested against whales and mixed populations. Lab 7.2 builds the deliverable: the drivers models for spend and average order value, with segment dummies, robust inference, the designed interaction, and the four-point audit run against an AI-fitted version. Both labs consume the customer_features table built and reconciled in Lab 6.1 — the analysis date remains June 30, 2026, the window remains the full 24 months — plus, in Lab 7.2, two roll-ups from the certified transactions file. AI assistants may draft any code cell (Appendix C has templates; Appendix A covers Colab

mechanics); every output is predicted before it is computed, and every exchange is documented per Appendix D.

Two conventions hold across every cell. Models are fitted with heteroskedasticity-robust standard errors, per Section 7.10, because this chapter predicts a residual funnel before it fits anything. And the four segments enter as explicitly constructed indicator columns with short names rather than through a formula shortcut, so that every coefficient label in the printed output is one a student can read aloud — the formula expression C(segment, Treatment(reference="Urban Loyal Core")) produces exactly the same fit with longer labels, and either is correct.

7.12.1 Lab 7.1, Part A: Correlation by Hand in Miniature

The miniature is Lab 6.1's ten-customer feature table, reused so every number is hand-checkable. The question: how strongly do frequency and monetary value move together? Write your prediction first — most students, reasoning "more orders means more revenue," predict r above 0.7.

Code 7.1. Covariance and correlation on the miniature feature table

```
import numpy as np
import pandas as pd

mini = pd.DataFrame({
    "customer_id": ["C001", "C002", "C003", "C004", "C005",
                    "C006", "C007", "C008", "C009", "C010"],
    "frequency": [4, 2, 3, 1, 3, 1, 1, 2, 1, 2],
    "monetary":  [196.00, 400.00, 69.60, 44.00, 127.00,
                  242.00, 24.00, 126.00, 38.00, 312.00],
}).set_index("customer_id")

# The two covariances of Section 7.3, computed side by side so that the
# denominator is a visible choice rather than a silent default.
dev_f = mini["frequency"] - mini["frequency"].mean()
dev_m = mini["monetary"] - mini["monetary"].mean()
deviation_products = dev_f * dev_m
n = len(mini)

print(mini.mean().round(2))
print("sum of the ten deviation products:",
      round(deviation_products.sum(), 2))
print("sample covariance    (n - 1):",
      round(deviation_products.sum() / (n - 1), 2))
print("population covariance    (n):",
      round(deviation_products.sum() / n, 2))
print("pandas .cov():", round(mini["frequency"].cov(mini["monetary"]), 2))
print("pandas .corr():", round(mini["frequency"].corr(mini["monetary"]), 3))

assert np.isclose(mini["frequency"].mean(), 2.0)
assert np.isclose(mini["monetary"].mean(), 157.86)
assert np.isclose(deviation_products.sum(), 240.60, atol=0.01)
# pandas reports the SAMPLE covariance, which is what Section 7.3 declared.
assert np.isclose(mini["frequency"].cov(mini["monetary"]),
                  deviation_products.sum() / (n - 1))
```

Expected output: means of 2.00 orders and 157.86 dollars; a deviation-product sum of 240.60; a sample covariance of 26.73 against a population covariance of 24.06; pandas reporting 26.73; a correlation of 0.199; and four assertions passing silently.

Input: ten literal customers. Transformation: paired deviations, their products, and two divisions. Output: the two covariances and the correlation. The cell exists to make one distinction concrete before it can cause trouble at scale: the two covariances differ by roughly 11 percent on ten rows, and the correlation is indifferent to the choice because the same denominator appears in the covariance and in both standard deviations and cancels. A student who computes the population covariance by hand and then compares it with the pandas figure has found a discrepancy that is a convention rather than an error — which is exactly the discrimination this guide's verification habit is meant to develop.

VERIFICATION CHECK

Before you run: compute r by hand from the printed table. The means are frequency 2.0 and monetary 157.86. The ten products of paired deviations sum to 240.60, so the sample covariance is 240.60 ÷ 9 = 26.73. The sample standard deviations are 1.054 orders and 127.34 dollars. Therefore r = 26.73 ÷ (1.054 × 127.34) ≈ 0.199. Write down each intermediate figure, and predict which of the two printed covariances pandas will match.

After you run: every figure reconciles to the cent, pandas matches the sample covariance rather than the population one, and all four assertions pass. Then confront the surprise: r ≈ 0.20 — weak, by Table 7.1 — against the strong association almost everyone predicts. Two observations resolve it, and both belong in the notebook. First, C002 and C010 — the two big-basket occasion shoppers — sit exactly at the mean on frequency, so they contribute zero to the covariance while inflating monetary's standard deviation, diluting r. Second, C006, the lapsed one-time $242 basket, pulls in the negative direction. The miniature contains two customer worlds, and the pooled r averages across them.

Investigate if: your hand-computed covariance disagrees with the printed one by roughly a ninth. That is the n versus n − 1 convention, not an arithmetic error, and the fix is to state which quantity you meant. A disagreement of any other size is arithmetic, and the deviation-product column is where to look.

Code 7.2. The labeled miniature scatterplot

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(6.5, 4.5))
ax.scatter(mini["frequency"], mini["monetary"], s=45)
for cid, row in mini.iterrows():
    ax.annotate(cid, (row["frequency"], row["monetary"]),
                xytext=(5, 4), textcoords="offset points", fontsize=8)

# The mean lines turn the plot into the covariance calculation: points in
# the upper-right and lower-left quadrants contribute positive products.
ax.axvline(mini["frequency"].mean(), linewidth=0.8, linestyle="--")
ax.axhline(mini["monetary"].mean(), linewidth=0.8, linestyle="--")
ax.set_xlabel("frequency (orders in the window)")
ax.set_ylabel("monetary (total revenue, $)")
ax.set_title("Ten customers, two worlds")
plt.tight_layout()
plt.show()
```

Expected output: a ten-point scatter with every customer labeled and two dashed mean lines dividing it into quadrants; C002 and C010 sit on the vertical mean line, and C006 sits alone in the upper-left quadrant.

Input: the miniature. Transformation: none — this is a rendering. Output: the picture that makes the arithmetic obvious. The mean lines are the point of the cell rather than decoration: they partition the plot into the four quadrants of the covariance calculation, so a customer's contribution can be read off its position. Points on a mean line contribute nothing, which is why

the two largest spenders in the table move r not at all in the positive direction while widening the vertical spread that r divides by. This is the entire argument of Section 7.4, experienced in the order statistics always should be: number, surprise, plot, understanding.

7.12.2 Lab 7.1, Part B: The Matrix at Full Scale, Stress-Tested

Part B moves to the full customer_features table. Before computing anything, write the predicted sign of the correlation between monetary and each of: recency_days, frequency, aov, discount_share, store_share, orders_per_month, revenue_per_month, app_user, and email_opt_in — with one sentence of reasoning each, drawing on Chapter 5's certified findings and Chapter 6's segment profiles. The two engagement flags belong in that list and in the matrix: the opening case's app and email claims are claims about exactly those columns, and a matrix that examines behavior while omitting the engagement indicators cannot speak to two of the three levers.

Code 7.3. Completeness and type checks before the matrix

```
corr_features = [
    "monetary",
    "frequency",
    "recency_days",
    "aov",
    "discount_share",
    "store_share",
    "orders_per_month",
    "revenue_per_month",
    "app_user",
    "email_opt_in",
]

# .corr() drops incomplete pairs silently and column by column, so a matrix
# built on partly missing columns quietly reports each cell on a different
# population. Check first; declare a policy if anything is missing.
missing = cf_full[corr_features].isna().sum()
print(missing[missing > 0] if missing.any() else "no missing values")

# Convert the Boolean flags deliberately. Left as Booleans they still
# correlate, but statsmodels will later label them app_user[T.True],
# which will not match the coefficient names this chapter uses.
cf_full[["app_user", "email_opt_in"]] = (
    cf_full[["app_user", "email_opt_in"]].astype(int))

assert cf_full[corr_features].notna().all().all(), (
    "declare a missing-data policy before calling .corr()")
assert set(cf_full["app_user"].unique()) <= {0, 1}
assert set(cf_full["email_opt_in"].unique()) <= {0, 1}

print(cf_full[corr_features].dtypes)
```

Expected output: the message "no missing values", then ten numeric dtypes with app_user and email_opt_in reported as integers, and three assertions passing.

Input: the certified feature table. Transformation: a completeness audit and one deliberate type conversion. Output: a feature list safe to correlate. The missingness check is not ceremony.

Pandas computes each cell of a correlation matrix on the rows where that particular pair is complete — the documented behavior of DataFrame.corr, not an accident of this build (pandas development team, 2026) — so a single partly missing column produces a matrix whose cells describe different populations while presenting them as one exhibit, a grain violation in Section 3.3's sense hiding inside a routine call. The assertion above is the response: declare and resolve a missing-data policy before the call, rather than discovering afterward that row 4 of the exhibit was computed on a different customer base than row 7. The type conversion is the second half of the same discipline: a Boolean column correlates perfectly well, and the formula API in Lab 7.2 will rename it in a way that breaks every coefficient sentence this chapter teaches, so the conversion happens once, here, where it can be seen.

Code 7.4. The predicted-then-computed correlation matrix

```
corr = cf_full[corr_features].corr().round(2)

print("correlations with monetary, weakest to strongest:")
print(corr["monetary"].drop("monetary").sort_values())
print()
print(corr)

# Two structural properties, asserted rather than assumed: a correlation
# matrix is symmetric and its diagonal is exactly 1.
assert np.allclose(np.diag(corr.values), 1.0)
assert np.allclose(corr.values, corr.values.T)
```

Expected output: a sorted column of nine correlations with monetary, then a ten-by-ten symmetric matrix with ones on the diagonal, and two assertions passing.

Input: the ten checked features. Transformation: pairwise sample correlations. Output: the screening exhibit of Section 7.4. Grade your predictions against the sorted column first, one by one, before reading the full matrix — the sorted column is the drivers question in its most naive form, and seeing it before the corrections land is what makes the corrections stick. The exact values await the certified data, but the designed structure fixes the pattern your predictions should anticipate: recency negatively associated with monetary, frequency and the two per-month rates positively, and aov's correlation with frequency negative across the pooled base, because the base's variation is dominated by the contrast between frequent small-basket urban customers and infrequent large-basket suburban ones. That negative aov–frequency cell is the segments speaking through a correlation matrix, and it belongs in your notebook's notes as the chapter's first mixed-population exhibit at full scale.

Two cells deserve immediate suspicion by Table 7.1's standards rather than celebration. The correlation between monetary and revenue_per_month will be high because one is the other divided by exposure, and the correlation between monetary and aov will be substantial because aov is monetary divided by frequency. Neither is a finding about customers; both are arithmetic relatives, and the derivation-tracing habit is what tells them apart from a behavioral result.

Code 7.5. Whale sensitivity: correlations with and without the top spend decile

```
cut = cf_full["monetary"].quantile(0.90)
trimmed_base = cf_full[cf_full["monetary"] < cut]

pooled = cf_full[corr_features].corr()["monetary"].drop("monetary")
trimmed = trimmed_base[corr_features].corr()["monetary"].drop("monetary")

compare = pd.DataFrame({
    "all customers": pooled.round(3),
    "top decile removed": trimmed.round(3),
})
compare["change"] = (compare["top decile removed"]
                     - compare["all customers"]).round(3)

removed = len(cf_full) - len(trimmed_base)
assert removed > 0
assert len(trimmed_base) + removed == len(cf_full)

print(f"top-decile cut at ${cut:,.2f}; {removed:,} customers removed "
      f"({removed / len(cf_full):.1%} of the base)")
print(compare.reindex(
    compare["change"].abs().sort_values(ascending=False).index))
```

Expected output: the cut point in dollars and the count removed, then nine rows sorted by the size of the change, with the largest movements on the spend-related features.

Input: the feature table and the concentration cut of Section 5.9. Transformation: the same correlation column computed twice, on the full base and with the top spend decile removed. Output: a movement report. Read the change column as an influence diagnostic, not as a correction: removing the whales does not produce the true correlation, because the whales are real customers and a substantial share of the revenue the chapter is trying to explain. What the column establishes is which stories depend on them. A correlation that survives the trim is a statement about the base; a correlation that halves or reverses is a statement about roughly one customer in ten, and it must be reported as such or not at all.

Code 7.6. Pooled and segment-faceted scatterplots of discount exposure and spend

```
segments = sorted(cf_full["segment"].unique())

fig, axes = plt.subplots(1, len(segments) + 1,
                         figsize=(3.6 * (len(segments) + 1), 3.4),
                         sharex=True, sharey=True)

pooled_r = cf_full["discount_share"].corr(cf_full["monetary"])
axes[0].scatter(cf_full["discount_share"], cf_full["monetary"],
                s=4, alpha=0.25)
axes[0].set_title(f"Pooled base\nr = {pooled_r:.2f}, n = {len(cf_full):,}")
axes[0].set_ylabel("monetary ($)")

for ax, seg in zip(axes[1:], segments):
    part = cf_full[cf_full["segment"] == seg]
    r = part["discount_share"].corr(part["monetary"])
    ax.scatter(part["discount_share"], part["monetary"], s=4, alpha=0.35)
    ax.set_title(f"{seg}\nr = {r:.2f}, n = {len(part):,}")

for ax in axes:
    ax.set_xlabel("discount_share")

plt.tight_layout()
plt.show()
```

Expected output: five panels sharing one pair of axes — the pooled base first, then one per segment — each captioned with its own correlation and customer count.

Input: the feature table and the Chapter 6 segment labels. Transformation: one pooled scatter and four conditional ones, on shared axes. Output: the always-plot discipline applied to the finance claim. Shared axes are the whole design: the panels are comparable only if the scales are identical, and a faceted exhibit with free axes is the most common way a mixed-population plot conceals the very structure it was drawn to reveal. Write one sentence per panel describing what the pooled plot concealed, and note in particular whether any segment's cloud occupies a region of discount_share that the other segments barely reach — because a coefficient estimated across the pooled base will be an average over regions where the segments do not overlap, which is Section 7.8's extrapolation warning arriving one section early and in visual form.

Code 7.7. Within-segment correlations against the pooled value

```
rows = []
for seg, g in cf_full.groupby("segment"):
    rows.append({
        "segment": seg,
        "customers": len(g),
        "r_discount_monetary": round(
            g["discount_share"].corr(g["monetary"]), 3),
        "r_app_monetary": round(g["app_user"].corr(g["monetary"]), 3),
        "mean_discount_share": round(g["discount_share"].mean(), 3),
        "mean_monetary": round(g["monetary"].mean(), 2),
    })
within = pd.DataFrame(rows).set_index("segment")

assert within["customers"].sum() == len(cf_full)

print("pooled discount-monetary r:",
      round(cf_full["discount_share"].corr(cf_full["monetary"]), 3))
print(within)
```

Expected output: the pooled correlation on one line, then one row per segment carrying its size, its two within-segment correlations, and the two segment means that explain the pooled figure.

Input: the feature table grouped by segment. Transformation: the same two correlations computed inside each group, beside the group means. Output: the confounding-by-composition exhibit and the finance claim's rebuttal. The two mean columns are why the cell prints them: if the segment with the highest mean discount_share also has the lowest mean monetary, the pooled negative correlation is substantially a statement about which segment a customer belongs to, and the within-segment correlations say what remains once that composition is set aside. Note the direction of the conclusion carefully. Weak or reversed within-segment correlations do not establish that discounts raise spending; they establish that the pooled correlation cannot be read as evidence that discounts lower it.

VERIFICATION CHECK

Before you run: predict the direction of the change. If segment composition is doing the work Section 7.5 claims, the pooled negative discount–spend correlation should weaken, vanish, or reverse within segments. Write the prediction down, and write the alternative outcome too: if the association survives at full strength inside every segment, the composition story explains less than claimed and that is equally a finding.

After you run: the pooled figure, the four within-segment figures, and the two mean columns are all in the notebook, and one insight statement per Section 5.9 records the comparison, its magnitude, its connection to the budget decision, and a verb that stops at the evidence. The whale comparison from Code 7.5 sits beside it, so that a correlation that both depends on the top decile and dissolves within segments is flagged twice rather than once.

Investigate if: a within-segment correlation is computed on a segment with very few customers, or on a segment whose discount_share barely varies. Both produce unstable correlations that look like findings, and the customers column and the faceted plot from Code 7.6 are where to check. A correlation inside a group whose predictor has almost no spread is arithmetic noise, not a weak relationship — and in the limiting case, where a segment's discount_share takes a single value, r is not small but undefined, since its denominator is zero. Software may report NaN there; read it as "the question cannot be asked of this group," not as "no relationship."

7.12.3 Lab 7.2, Part A: The Composition Features and the Anchor Model

Lab 7.2 builds the deliverable in four parts, moving from one predictor to the full specified model. The response variables are monetary (total spend) and aov, both defined per Table 6.5. Two additional features join from the certified transactions file, built with Chapter 4's groupby mechanics and reconciled before use: units_per_order (total units ÷ distinct orders) and occasionwear_share (revenue on occasionwear lines ÷ total revenue, joining products for the is_occasionwear flag). Declare both derivation rules in the notebook before computing them.

Code 7.8. Build and reconcile the two composition features

```
analysis_date = pd.Timestamp("2026-06-30")
window_start = analysis_date - pd.DateOffset(months=24)

# Certified figures for the DECLARED WINDOW, copied from the Chapter 4
# verification log before anything is computed. Fill these in from your
# own log: a build that reconciles only against itself cannot catch an
# error it inherited.
CERT_REVENUE    = ...   # total revenue in the window
CERT_PURCHASERS = ...   # distinct customers with an order in the window

window = transactions_clean[
    (transactions_clean["order_date"] > window_start)
    & (transactions_clean["order_date"] <= analysis_date)]

# The occasionwear flag lives on the product dimension, so the join is
# validated: many lines to one product, and nothing may go unmatched.
lines = window.merge(products[["product_id", "is_occasionwear"]],
                     on="product_id", how="left", validate="many_to_one")
assert len(lines) == len(window), "the join changed the row count"
assert lines["is_occasionwear"].notna().all(), "unmatched product_id"

comp = (lines
        .assign(occasion_revenue=lambda d: d["line_revenue"]
                .where(d["is_occasionwear"], 0.0))
        .groupby("customer_id")
        .agg(units=("quantity", "sum"),
             orders=("order_id", "nunique"),
             revenue=("line_revenue", "sum"),
             occasion_revenue=("occasion_revenue", "sum")))

comp["units_per_order"] = comp["units"] / comp["orders"]
comp["occasionwear_share"] = comp["occasion_revenue"] / comp["revenue"]

model_data = cf_full.join(
    comp[["units_per_order", "occasionwear_share"]], how="left")

# Reconciliation: a feature built from the same window on the same file
# must agree with the certified feature table, or one of them is wrong.
assert len(model_data) == len(cf_full)
assert (comp["orders"].reindex(cf_full.index) == cf_full["frequency"]).all()
assert np.allclose(comp["revenue"].reindex(cf_full.index),
                   cf_full["monetary"], atol=0.01)
assert len(cf_full) == CERT_PURCHASERS
assert np.isclose(comp["revenue"].sum(), CERT_REVENUE, atol=0.01)

tol = 1e-9
assert model_data[["units_per_order",
                   "occasionwear_share"]].notna().all().all()
assert (model_data["units_per_order"] >= 1 - tol).all()
assert model_data["occasionwear_share"].between(-tol, 1 + tol).all()

print(model_data[["units_per_order", "occasionwear_share"]]
      .describe().round(3))
```

Expected output: an eight-row summary for the two new features, with units_per_order at or above 1.0 throughout and occasionwear_share bounded by 0 and 1, and nine assertions passing.

Input: the certified line-grain file and the product dimension. Transformation: a validated join, a conditional revenue column, one grouped aggregation, and two ratios. Output: the two composition features Chapter 5's decomposition named, joined to the modeling table. The reconciliation assertions are the point of the cell, and they run at two levels. The first pair recomputes orders and revenue per customer from the same window that produced the certified feature table, so the two must agree exactly. The second pair checks both against the Chapter 4 verification log, because a build that reconciles only against the feature table will happily reproduce an error the feature table already contains — the internal check catches this chapter's mistakes, and the certified check catches the ones inherited from upstream. The share needs a tolerance rather than a bare comparison with 1, for the reason Chapter 6 met: a share summed from rounded line revenues can land a fraction above one in floating point.

Note the ordering discipline the join enforces. The composition features are built from the transaction file, not from the feature table, because units and occasionwear status live at line grain and cannot be recovered from customer totals — the same grain argument that sent Chapter 6 back to transactions for discount_share. Building them here rather than in Chapter 6 is a deliberate scope decision: they exist to serve this chapter's average-order-value model, and a feature table should not accumulate columns for analyses nobody has specified.

Code 7.9. The anchor model, with nonrobust and robust standard errors side by side

```
import statsmodels.formula.api as smf

m1_default = smf.ols("monetary ~ recency_days", data=model_data).fit()
m1 = smf.ols("monetary ~ recency_days",
             data=model_data).fit(cov_type="HC3")

comparison = pd.DataFrame({
    "coefficient": m1.params.round(4),
    "SE (nonrobust)": m1_default.bse.round(4),
    "SE (HC3)": m1.bse.round(4),
})

# HC3 changes the estimated standard errors, confidence intervals, and
# p-values. It does NOT change the fitted coefficients or the predictions.
assert np.allclose(m1.params, m1_default.params)
assert np.isclose(m1.rsquared, m1_default.rsquared)
assert np.allclose(m1.fittedvalues, m1_default.fittedvalues)

print(comparison)
print("R-squared (identical under both):", round(m1.rsquared, 3))
print("\n95% interval on recency_days, HC3:")
print(m1.conf_int().loc["recency_days"].round(3))
print("95% interval on recency_days, nonrobust:")
print(m1_default.conf_int().loc["recency_days"].round(3))
```

Expected output: a two-row table whose coefficient column is identical under both fits and whose two standard-error columns differ; one R-squared value; and two intervals on the recency slope that share a center and differ in width.

Input: the modeling table. Transformation: the same regression fitted twice, under two covariance estimators. Output: the difference robust inference makes, isolated. Read the slope aloud in units — dollars of 24-month spend per additional day since last purchase — and write the interval sentence per Section 7.10. Then read the three assertions as the chapter's claim in executable form: the estimate, the fit, and every prediction are untouched by the choice of estimator, and everything used to judge the estimate's precision is not. An analyst who reports the coefficient from one fit and the p-value from the other has mixed two answers to two different questions.

Code 7.10. Residual diagnostics: the funnel, measured rather than eyeballed

```
fig, axes = plt.subplots(1, 2, figsize=(11, 4))

axes[0].scatter(m1.fittedvalues, m1.resid, s=4, alpha=0.25)
axes[0].axhline(0, linewidth=0.8)
axes[0].set_xlabel("fitted spend ($)")
axes[0].set_ylabel("residual ($)")
axes[0].set_title("Residuals against fitted values")

for seg in segments:
    mask = (model_data["segment"] == seg).values
    axes[1].scatter(m1.fittedvalues[mask], m1.resid[mask],
                    s=4, alpha=0.4, label=seg)
axes[1].axhline(0, linewidth=0.8)
axes[1].set_xlabel("fitted spend ($)")
axes[1].set_title("The same residuals, colored by segment")
axes[1].legend(fontsize=7, markerscale=2)

plt.tight_layout()
plt.show()

# A funnel is a claim about spread, so measure the spread. Residual
# standard deviation by decile of fitted value turns the picture into a
# number the Verification Check can be written against.
spread = (pd.DataFrame({"fitted": m1.fittedvalues, "resid": m1.resid})
          .assign(decile=lambda d: pd.qcut(d["fitted"], 10, labels=False,
                                           duplicates="drop"))
          .groupby("decile")["resid"]
          .agg(residual_sd="std", n="size")
          .round(2))
print(spread)
```

Expected output: two residual plots sharing a scale, and a ten-row table in which the residual standard deviation rises materially from the lowest fitted decile to the highest.

Input: the fitted anchor model. Transformation: two renderings and one grouped spread calculation. Output: the diagnostic evidence that justifies both the robust estimator already used and the fuller model about to be fitted. The left panel is the classic exhibit; the right panel is the one that changes the specification, because visible clustering by segment is the data arguing that four groups do not share one line. The decile table converts an impression into a measurement, which matters for two reasons: an eyeballed funnel is exactly the kind of judgment that quietly

disappears when the analysis is delegated, and a rising residual standard deviation is the concrete fact that makes the case for HC3 in a sentence a non-technical reader can follow.

7.12.4 Lab 7.2, Part B: The Specified Spend Model

Part B fits the specified spend model: monetary on recency_days, tenure_days, app_user, email_opt_in, and discount_share, with the four segments entering as dummies against the Urban Loyal Core reference per Table 7.3. Before running, write the predicted sign of every coefficient and one anchor magnitude — the raw app-user spend gap, which Code 7.11 prints beside the model so the shrinkage can be read directly.

Code 7.11. Segment dummies and the specified spend model

```
# Table 7.3 as code. The reference category gets no column: it is
# represented when all three indicators are zero.
SEG_REF = "Urban Loyal Core"
SEG_CODES = {
    "Suburban Occasion": "seg_suburban",
    "Discount-Dependent": "seg_discount",
    "New / At-Risk": "seg_new",
}
for name, col in SEG_CODES.items():
    model_data[col] = (model_data["segment"] == name).astype(int)

seg_cols = list(SEG_CODES.values())
# Exactly one indicator on for a non-reference customer, none for the
# reference: the encoding is exhaustive and mutually exclusive.
assert model_data[seg_cols].sum(axis=1).isin([0, 1]).all()
assert (model_data[seg_cols].sum(axis=1) == 0).sum() == (
    model_data["segment"] == SEG_REF).sum()

spend_formula = (
    "monetary ~ recency_days + tenure_days + app_user + email_opt_in"
    " + discount_share + seg_suburban + seg_discount + seg_new")

m2 = smf.ols(spend_formula, data=model_data).fit(cov_type="HC3")
print(m2.summary().tables[1])
print("R-squared:", round(m2.rsquared, 3))

# The anchor magnitude: the raw gap the platform deck quoted.
raw = model_data.groupby("app_user")["monetary"].mean()
print(f"\nraw app-user spend gap:   ${raw.loc[1] - raw.loc[0]:,.2f}")
print(f"raw app-user spend ratio: {raw.loc[1] / raw.loc[0]:.2f}x")
print(f"adjusted app coefficient: ${m2.params['app_user']:,.2f}")
```

Expected output: a coefficient table with nine rows including the three segment indicators; an R-squared value; and three lines contrasting the raw app-user gap and ratio with the adjusted coefficient, the adjusted figure being substantially the smaller.

Input: the modeling table. Transformation: three indicator columns and one fitted model. Output: the drivers table the meeting will quote, and the single comparison that disciplines it. The two assertions make Table 7.3's encoding executable rather than merely described: every non-reference customer carries exactly one indicator, and the count of all-zero rows equals the

reference segment's size. That second assertion is the one that catches a mis-specified reference — the most common silent error in dummy coding, and one that changes every printed number without changing the model's fit.

Read the app comparison in the order printed. The raw ratio is a difference between two groups of customers who differ in everything that travels with app ownership. The adjusted coefficient is a model-based comparison holding recency, tenure, email engagement, discount exposure, and segment membership constant — and the gap between the two numbers is the chapter's central exhibit, because it prices how much of a widely quoted business fact is profile rather than product.

Code 7.12. The coefficient table as a deliverable: estimates, intervals, and units

```
UNITS = {
    "Intercept": "dollars, at zero on every predictor",
    "recency_days": "dollars per additional day since last purchase",
    "tenure_days": "dollars per additional day of tenure",
    "app_user": "dollars, app users vs. non-users",
    "email_opt_in": "dollars, opted-in vs. not",
    "discount_share": "dollars per 1.00 increase in discount share",
    "seg_suburban": "dollars vs. the Urban Loyal Core",
    "seg_discount": "dollars vs. the Urban Loyal Core",
    "seg_new": "dollars vs. the Urban Loyal Core",
}

def coefficient_table(fit, units=None, digits=2):
    """A coefficient table a meeting can read: estimate, interval, units."""
    low, high = fit.conf_int()[0], fit.conf_int()[1]
    out = pd.DataFrame({
        "estimate": fit.params.round(digits),
        "ci_low": low.round(digits),
        "ci_high": high.round(digits),
        "interval_width": (high - low).round(digits),
        "p_value": fit.pvalues.round(4),
    })
    if units:
        out["reads as"] = [units.get(i, "") for i in out.index]
    return out

table = coefficient_table(m2, UNITS)

# An interval that does not contain its own estimate is a construction bug.
assert (table["ci_low"] <= table["estimate"]).all()
assert (table["estimate"] <= table["ci_high"]).all()

print(table)
```

Expected output: a nine-row table carrying, for every coefficient, its estimate, its HC3 interval, the interval's width, its p-value, and a plain-language units string.

Input: the fitted model. Transformation: a reshaping of the results object. Output: the exhibit that goes in the deliverable. The units column is the cell's argument. Section 7.6 required that a coefficient not be read until its units are said aloud, and a units dictionary written by hand before

the table is printed forces the analyst to say them once for every predictor — including the two that trip people most reliably. A dummy's units are dollars relative to the reference, not dollars absolutely. And discount_share is a proportion, so its coefficient describes a difference of one full unit of share, which is the entire range from no discounted revenue to all of it; quoting it as though it were a percentage point is a hundred-fold error that no software will catch.

The interval width column earns its place for the reason Section 7.10 gave: at this sample size nearly everything clears conventional significance, so the p-value column separates almost nothing and the width column separates a great deal. Sort the table by width and the coefficients the analysis actually pins down separate themselves from the ones it merely signs.

Code 7.13. Variance inflation factors

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from patsy import dmatrix

# VIF is computed on the model's design matrix, including the intercept
# column, and then reported for the predictors only.
X = dmatrix(spend_formula.split("~", 1)[1], data=model_data,
            return_type="dataframe")

vif = pd.Series(
    [variance_inflation_factor(X.values, i) for i in range(X.shape[1])],
    index=X.columns, name="VIF").drop("Intercept")

assert (vif >= 1 - 1e-6).all(), (
    "a VIF below 1 means the design matrix was built wrong")

print(vif.sort_values(ascending=False).round(2))
print("\npredictor correlations, for comparison:")
print(model_data[["recency_days", "tenure_days", "app_user",
                  "email_opt_in", "discount_share"]].corr().round(2))
```

Expected output: eight variance inflation factors sorted from largest to smallest, each at least 1, followed by the five-by-five correlation matrix of the numeric and indicator predictors.

Input: the specified model's design matrix. Transformation: one variance inflation factor per predictor. Output: the multicollinearity diagnostic that pairwise correlations cannot supply. Print the two together, as the cell does, because the comparison is the lesson: a predictor can show no alarming pairwise correlation and still carry a raised variance inflation factor, because the factor measures its relationship with all the other predictors at once. The three segment indicators are the natural place to see it, since they are mutually exclusive by construction and therefore related to one another whatever their pairwise correlations look like.

Resist the cutoff. This guide publishes none, and the reason is practical rather than pedantic: the number that matters is whether a particular coefficient is estimated precisely enough for the decision in front of you, which is a question the interval width in Code 7.12 answers directly and a threshold answers only by proxy. Read a large factor as an instruction — check this coefficient's stability across specifications, and consider reporting it jointly with its entangled partners rather than separately — which is exactly what Code 7.17 does.

VERIFICATION CHECK

Before you run: this is the four-point audit of Table 7.5, run on your own model before it is ever run on an assistant's. Write the predicted sign of all eight coefficients. Compute the raw app-user gap by hand from a two-line groupby and predict whether the adjusted coefficient will land above, near, or far below it. Predict the direction of the three segment coefficients against Chapter 5's certified descriptive gaps, and predict which two predictors will carry the largest variance inflation factors.

After you run: signs first — every coefficient against your written predictions, with any surprise earning a confounding story or a flag rather than a shrug. Magnitudes next: compare the adjusted segment coefficients with the certified descriptive gaps for sign, approximate scale, and substantive plausibility. An adjusted difference may come out smaller or larger than the raw difference; a larger one is not evidence of an error, because opposing covariate differences can suppress part of an association until the model removes them. Either direction requires an explanation grounded in the variables that entered the model. Then units: restate three coefficients aloud, including one dummy and discount_share. Then diagnostics: the residual funnel should be reduced relative to Code 7.10 but not gone, the variance inflation factors should be read, and the interval widths should be the basis of any claim about precision.

Investigate if: a segment coefficient exceeds its descriptive gap by a wide margin and no covariate story explains it, or the app coefficient fails to shrink at all against the raw gap. The first is worth tracing through Code 7.17's specification sensitivity; the second usually means a control that was supposed to carry the profile — segment membership most often — is absent, misspelled in the formula, or constant within the sample.

7.12.5 Lab 7.2, Part C: Average Order Value and Chapter 5's Decomposition

Part C turns to average order value, and to the chapter's designed payoff: a certified descriptive finding, remeasured by a new instrument. Chapter 5 decomposed the suburban average-order-value premium into basket size and occasionwear mix. Two models reproduce that finding in regression form — one with the segments alone, one with the two composition features added — and the comparison between them is the deliverable.

Code 7.14. The segment-only and composition-controlled AOV models

```
# The certified descriptive gap from the Chapter 5 verification log --
# suburban mean AOV minus Urban Loyal Core mean AOV, in dollars.
CERT_AOV_GAP_SUBURBAN = ...

aov_simple = smf.ols(
    "aov ~ seg_suburban + seg_discount + seg_new",
    data=model_data).fit(cov_type="HC3")

aov_controlled = smf.ols(
    "aov ~ seg_suburban + seg_discount + seg_new"
    " + units_per_order + occasionwear_share",
    data=model_data).fit(cov_type="HC3")

# A segment-only model IS the difference in group means. Checking that
# identity is how you know the encoding and the reference are right.
means = model_data.groupby("segment")["aov"].mean()
descriptive_gap = means["Suburban Occasion"] - means[SEG_REF]
assert np.isclose(aov_simple.params["seg_suburban"],
                  descriptive_gap, atol=0.01)
assert np.isclose(aov_simple.params["Intercept"], means[SEG_REF], atol=0.01)
# ...and the recomputed gap must match the certified one.
assert np.isclose(descriptive_gap, CERT_AOV_GAP_SUBURBAN, atol=0.01)

s_lo, s_hi = aov_simple.conf_int().loc["seg_suburban"]
c_lo, c_hi = aov_controlled.conf_int().loc["seg_suburban"]

print(f"descriptive AOV gap (Chapter 5): ${descriptive_gap:,.2f}")
print(f"segment-only coefficient:
${aov_simple.params['seg_suburban']:,.2f} "
      f"[{s_lo:,.2f}, {s_hi:,.2f}]")
print(f"with composition controls:
${aov_controlled.params['seg_suburban']:,.2f} "
      f"[{c_lo:,.2f}, {c_hi:,.2f}]")
# NOT a share of the gap: this is the proportional change in ONE
# coefficient between two specifications. It can exceed 100% or go
# negative if the coefficient crosses zero.
proportional_reduction = (
    1 - aov_controlled.params["seg_suburban"]
      / aov_simple.params["seg_suburban"]
)
print(f"\nproportional reduction in the suburban coefficient "
      f"after adding the composition variables: "
      f"{proportional_reduction:.1%}")
print(f"R-squared, segment only: {aov_simple.rsquared:.3f}")
print(f"R-squared, controlled:   {aov_controlled.rsquared:.3f}")
```

Expected output: the descriptive gap and the segment-only coefficient agreeing to the cent, a controlled coefficient much closer to zero with an interval that may include it, the proportional reduction in the suburban coefficient, two R-squared values, and three assertions passing.

Input: the modeling table with its composition features. Transformation: two fitted models sharing a reference category. Output: Chapter 5's decomposition in coefficient form. The three assertions are the cell's verification spine, and they are worth pausing on. The third is the one that reaches outside this chapter: the gap the model reproduces must equal the gap Chapter 5

certified, so a segment label that drifted or a population that changed is caught here rather than in a meeting. The first two are worth pausing on for a different reason. A regression of a response on a complete set of category indicators with an intercept reproduces the group means exactly: the intercept is the reference group's mean, and each indicator's coefficient is that group's mean minus the reference's. In a model containing nothing else, therefore, the intercept and the reference mean coincide — the special case Section 7.9's definition named. Add the two composition variables and the intercept stops being anybody's mean, which is the general case the definition insisted on.

The wording of the finding is where this part earns its place in the chapter, and the temptation is specific. It is tempting to write that the controlled model shows the suburban premium runs through composition. That sentence describes a causal pathway — a mediation claim — and the two fitted models do not establish one. What the evidence supports is a statement about adjustment: after adjusting for units per order and occasionwear share, the estimated suburban difference becomes much smaller, so these variables statistically account for most of the observed average-order-value gap in this dataset. That formulation agrees with Chapter 5 exactly, survives a CFO's follow-up question, and does not claim a mechanism the analysis has not tested.

7.12.6 Lab 7.2, Part D: Interactions, Consistency, Sensitivity, and Leakage

Part D estimates the dataset's designed interaction, tests a claim the managerial section would otherwise have made on faith, and closes with the two audits that keep a coefficient table honest: a specification sweep and a deliberate leakage demonstration.

Code 7.15. The loyalty-by-app interaction and its two conditional associations

```
# .ne("None") treats a missing tier as NOT equal to "None", which
# would silently file unknown customers as enrolled. Resolve first.
assert model_data["loyalty_tier"].notna().all(), (
    "resolve missing loyalty_tier values before defining enrollment"
)
model_data["loyalty_enrolled"] = (
    model_data["loyalty_tier"].ne("None").astype(int))

# An interaction estimates a separate association per group, so check the
# group sizes BEFORE fitting. Empty or thin cells produce a coefficient
# with an interval wide enough to contain any story anyone wants.
cells = pd.crosstab(model_data["loyalty_enrolled"], model_data["app_user"])
print(cells)
MIN_CELL = 30  # course screening threshold, not a universal guarantee
assert (cells.values >= MIN_CELL).all(), (
    "one interaction cell falls below the course screening threshold"
)

interaction = smf.ols(
    "monetary ~ app_user * loyalty_enrolled + recency_days + tenure_days"
    " + discount_share + seg_suburban + seg_discount + seg_new",
    data=model_data).fit(cov_type="HC3")
print(interaction.summary().tables[1])

def linear_combination(fit, terms, label):
    """Estimate and interval for a sum of coefficients (Section 7.9)."""
    contrast = np.zeros(len(fit.params))
    for t in terms:
        contrast[list(fit.params.index).index(t)] = 1.0
    test = fit.t_test(contrast)
    est = float(np.ravel(test.effect)[0])
    lo, hi = np.ravel(test.conf_int())[:2]
    print(f"{label}: ${est:,.2f}  [{lo:,.2f}, {hi:,.2f}]")
    return est

not_enrolled = linear_combination(
    interaction, ["app_user"],
    "app association, NOT loyalty-enrolled")
enrolled = linear_combination(
    interaction, ["app_user", "app_user:loyalty_enrolled"],
    "app association, loyalty-enrolled    ")

# The conditional reading of Section 7.9, made executable.
assert np.isclose(enrolled, interaction.params["app_user"]
                  + interaction.params["app_user:loyalty_enrolled"])
```

Expected output: a two-by-two table of cell counts, a coefficient table containing app_user, loyalty_enrolled, and app_user:loyalty_enrolled, then two conditional app associations in dollars with their HC3 intervals, and three assertions passing.

Input: the modeling table. Transformation: one derived indicator, one cell-count check, one fitted model with a product term, and two linear combinations. Output: the two numbers the meeting can act on. Two guards run before the model. The first is a missing-value check on loyalty_tier, and it exists because the derivation immediately below it is booby-trapped:

.ne("None") evaluates a missing tier as not equal to "None", so any customer whose tier never arrived would be filed as enrolled without a word of complaint — Chapter 4's two-kinds-of-null problem reappearing as a silent recoding. The second is the cell-count check, and it is not boilerplate either: an interaction asks the data for a separate association in each combination of two variables, and a combination holding a handful of customers will supply one, complete with a confident-looking point estimate and an interval nobody reads. Read the threshold for what it is. Thirty is this course's screening floor, chosen so a lab cell has enough customers to estimate a slope at all; it is not a statistical sufficiency boundary, and clearing it guarantees nothing about precision. The confidence intervals printed below still decide what the sample actually supports, and a cell that passes the screen can easily produce an interval too wide to act on.

The two printed associations are the deliverable, and the interaction coefficient itself is not. A meeting sentence built on "the interaction was significant" tells a VP nothing actionable; two dollar figures with intervals — the app association among non-enrolled customers, and the app association among enrolled customers — tell the four journeys something they can be designed around. Note also what the conditional reading does to the main-effect row in the printed table: app_user is now the association among the non-enrolled only, and quoting it as the overall app association would be exactly the misreading Table 7.5's third point exists to catch.

Code 7.16. Is the email association the same in every segment?

```
email_formula = (
    "monetary ~ email_opt_in + seg_suburban + seg_discount + seg_new"
    " + email_opt_in:seg_suburban + email_opt_in:seg_discount"
    " + email_opt_in:seg_new"
    " + recency_days + tenure_days + app_user + discount_share")

email_model = smf.ols(email_formula, data=model_data).fit(cov_type="HC3")

# One joint test of the three differences, then the four conditional
# associations. The joint test asks whether the associations differ at
# all; the four numbers say what they are.
diff_terms = [p for p in email_model.params.index
              if p.startswith("email_opt_in:")]
R = np.zeros((len(diff_terms), len(email_model.params)))
for row, term in enumerate(diff_terms):
    R[row, list(email_model.params.index).index(term)] = 1.0
joint = email_model.f_test(R)
print("joint test that all three email-by-segment differences are zero:")
print(f"  F = {float(np.ravel(joint.fvalue)[0]):.2f}, "
      f"p = {float(np.ravel(joint.pvalue)[0]):.4f}")
print()

for seg, col in [(SEG_REF, None)] + list(SEG_CODES.items()):
    terms = ["email_opt_in"]
    if col:
        terms.append(f"email_opt_in:{col}")
    linear_combination(email_model, terms, f"email association, {seg:<20}")
```

Expected output: one joint F test with its p-value, then four conditional email associations in dollars with HC3 intervals — one per segment, the first being the Urban Loyal Core reference.

Input: the modeling table. Transformation: a model carrying one interaction per non-reference segment, one joint test, and four linear combinations. Output: the evidence behind a managerial sentence that would otherwise be an assumption. This cell exists because of a specific discipline failure worth naming. A draft of this chapter's managerial section reported that the email association is consistent across segments — a claim about four quantities that no model in the chapter had estimated. Either the analysis is run or the claim is withdrawn, and running it is cheap.

Read the two outputs in order and let them do different jobs. The joint test asks a single question: are the three segment differences all zero? A large p-value there does not establish that the associations are identical — failing to detect a difference is not evidence of sameness, particularly where the segments differ in size and the smallest ones carry the widest intervals. The four conditional estimates are what the deliverable reports, and the honest summary names both the common direction and the spread of the intervals around it. Section 7.13 writes that sentence.

Code 7.17. Specification sensitivity: how much does the app coefficient depend on the model?

```
SPECS = {
    "A. app only": "monetary ~ app_user",
    "B. + behavior, no segments":
        "monetary ~ app_user + recency_days + tenure_days"
        " + email_opt_in + discount_share",
    "C. the specified model": spend_formula,
    "D. + purchase intensity":
        spend_formula + " + orders_per_month",
}

rows = []
for name, formula in SPECS.items():
    fit = smf.ols(formula, data=model_data).fit(cov_type="HC3")
    lo, hi = fit.conf_int().loc["app_user"]
    rows.append({
        "specification": name,
        "app_coefficient": round(fit.params["app_user"], 2),
        "ci_low": round(lo, 2),
        "ci_high": round(hi, 2),
        "r_squared": round(fit.rsquared, 3),
        "predictors": int(fit.df_model),
    })

sensitivity = pd.DataFrame(rows).set_index("specification")

# R-squared cannot fall as predictors are added to a NESTED model on the
# same rows (Section 7.7). A and B and C are nested; C and D are nested.
assert sensitivity.loc["C. the specified model", "r_squared"] >= \
       sensitivity.loc["B. + behavior, no segments", "r_squared"]
assert sensitivity.loc["D. + purchase intensity", "r_squared"] >= \
       sensitivity.loc["C. the specified model", "r_squared"]

print(sensitivity)
```

Expected output: four rows showing the app coefficient shrinking substantially from specification A to specification C, a further movement at D, non-decreasing R-squared down the nested sequence, and two assertions passing.

Input: the modeling table. Transformation: four nested fits, each reporting the same coefficient. Output: the instability report that turns a single number into a range of defensible numbers. This is the specification-sensitivity check Section 7.10 named as multicollinearity's paired instrument, and it is also the honest answer to the question every drivers deliverable invites: how much does this depend on your choices? Specification D is the pointed one, because orders_per_month is close kin to the response and to several predictors at once; watch what it does to the app coefficient's interval as well as to its point estimate, and note that a rising R-squared accompanies it either way.

The two assertions encode Section 7.7's conditions rather than a general law. R-squared cannot fall when predictors are added to a nested model fitted to the same observations — which is why the comparison being asserted is B against C and C against D, and why specification A,

which is nested inside all of them, is left to the eye. If either assertion ever fails, the models are not fitted on the same rows, and a silently dropped observation is the first place to look.

Code 7.18. The leakage audit: a model handed its own answer

```
# Chapter 6 defined aov = monetary / frequency (Table 6.5). Their product
# is therefore monetary itself, exactly. Predict both R-squared values
# before running this cell.
additive = smf.ols("monetary ~ aov + frequency", data=model_data).fit()
product = smf.ols("monetary ~ I(aov * frequency)",
                  data=model_data).fit()

print("additive model, monetary ~ aov + frequency")
print("  R-squared:", round(additive.rsquared, 4))
print("product model, monetary ~ I(aov * frequency)")
print("  R-squared:", round(product.rsquared, 8))

identity_error = (model_data["aov"] * model_data["frequency"]
                  - model_data["monetary"]).abs().max()
print("\nlargest |aov x frequency - monetary|:", f"{identity_error:.2e}")

# The additive model fits strongly and NOT perfectly: a sum is not a
# product. The product model reconstructs the response by construction.
assert additive.rsquared < 0.999
assert product.rsquared > 0.9999
assert identity_error < 1e-6
```

Expected output: a strong but imperfect R-squared for the additive model, an R-squared of essentially 1 for the product model, an identity error at floating-point scale, and three assertions passing.

Input: the modeling table. Transformation: two regressions of the response on its own components. Output: the sensation of a model that is too good, produced on purpose. The two specifications are here together because the difference between them is the lesson. Adding aov and frequency as separate additive predictors fits strongly — they are the response's own factors — but it does not reconstruct the response, because a linear combination of two variables is not their product. Multiply them and the model is handed monetary under another name, and R-squared goes to one up to floating-point error.

Name what happened in the notebook, in the words the deliverable would use: the too-good model did not discover customer behavior; it was handed the outcome under another name. Then generalize the detection rule, because the next instance will not be this obvious. Trace every predictor's derivation before it enters a model, exactly as Table 7.2's fourth row instructed for correlations; treat any R-squared above roughly 0.9 on individual customer behavior as a defect report rather than a result; and remember that the identity holds here to floating-point precision only because Chapter 6's build does not round aov — a version that rounded to cents would produce an R-squared microscopically below one, which is a fine illustration of how nearly invisible a leaking variable can be. Chapter 8 will name the general disease and equip you to detect it where no arithmetic identity announces it.

VERIFICATION CHECK

Before you run: predict both R-squared values in Code 7.18, in writing, before executing it — the additive model and the product model. Predict the two conditional app associations of Code 7.15 in dollars: what should the app–spend association be among customers who are not loyalty-enrolled, and how should it differ among the enrolled if loyalty and app engagement reinforce each other as the dataset's specification designed? For Code 7.16, predict whether the joint test will detect differences across segments, and say what you would conclude from either answer before you know which one you get.

After you run: the two conditional associations are written in dollars with their intervals, one sentence each, and the interaction coefficient is not quoted on its own anywhere in the deliverable. The email section reports four conditional estimates and the joint test, with a sentence that does not convert a large p-value into a claim of sameness. The specification table shows the app coefficient's range across four defensible models, and the deliverable quotes that range rather than a single number wherever the choice of specification moves it materially.

Investigate if: the product model's R-squared is not essentially 1. Either aov was rounded somewhere upstream, in which case the identity error printed by the cell will be at cents scale rather than floating-point scale, or the two columns are not the ones Table 6.5 defines. Both are derivation questions, and both are answered in the feature table rather than in the model.

Close Lab 7.2 with the AI round trip. Give an assistant the raw feature table and the deliberately loose prompt "find what drives spend and tell me what to do," then run Table 7.5's audit on the reply, grade its verbs against Section 7.2's registry, and file the exchange per Appendix D. The comparison between the assistant's narration and your Part B through Part D findings — same data, same arithmetic, different discipline — is the lab's last exhibit and the chapter's argument in miniature. Deliverables for both labs: the notebook with all eighteen cells executed, the written predictions made before each model was fitted, the reconciliation output from Code 7.8, the coefficient table with intervals and units, the specification-sensitivity table, the two conditional interaction associations in dollars, and the AI-Use Appendix documenting every assistant exchange, including at least one delegated step you corrected and why.

7.13 Marketing Interpretation and Managerial Insight

The lab's outputs are coefficient tables; the budget lock runs on sentences. This section translates — and, per this guide's standing practice, it does so partly by exhibiting a wrong managerial reading and correcting it, because this chapter's characteristic misreading is the most expensive one in the guide so far: a misread summary misinforms a meeting, a misread segmentation runs for a season, but a misread driver reallocates a budget and then defends the reallocation with the very analysis that caused it.

The wrong reading will be offered in the allocation meeting by someone reasonable, probably while pointing at your own Code 7.12 table: "The model says app users spend $140 more than comparable non-users. We have several thousand customers without the app. Fund the

install campaign — at $140 a head, it pays for itself many times over." Every number in the sentence can be exactly right, and the conclusion does not follow. The coefficient is a model-adjusted comparison between existing app users and non-users; it is not a forecast of what happens when a non-user is persuaded to install. The customers who chose the app differ from those who did not in everything the model cannot see — enthusiasm for the brand, planned future purchases, the underlying engagement that Section 7.8's omitted-variable logic says is inflating the coefficient right now. Installing the app does not install the enthusiasm.

The corrected reading, which the deliverable should print under the table: "Customers with the app spend $140 more than otherwise similar customers without it, adjusting for recency, tenure, email engagement, discount exposure, and segment. Some of that difference may reflect what the app enables; some may reflect selection into app use, and this analysis cannot split the two. The raw gap the platform deck quoted was several times larger, and the model attributes most of it to profile rather than to the product. Across four defensible specifications the estimate ranges from $X to $Y, which is the range the decision should carry. If the install campaign is funded, it should be funded as a test — Chapter 11's instrument, not this chapter's — with a holdout that would tell us, one quarter from now, what a caused install is actually worth." That paragraph — association stated, rivals named, sensitivity disclosed, decisive evidence specified — is what an honest drivers deliverable sounds like, and it is a stronger meeting position than the overclaim, because it cannot be dismantled by the CFO's first question.

The same discipline settles the other two claims, and the deliverable should give each its verdict paragraph. The discount claim dissolves under the lab's within-segment exhibit from Code 7.7 and the faceted plot from Code 7.6: the pooled negative association reflects which customers use discounts more than what discounts do, and discount exposure is more strongly associated with the Discount-Dependent segment's purchasing pattern than with any change in it. Cutting the ladder on this evidence would withdraw the lever most visibly associated with a quarter of the treatment map's purchasing, on a correlation that composition substantially accounts for, and without any test of how that quarter responds. The recommendation that survives: hold the ladder, and test depth variations within the Discount-Dependent journey, where the lever's marginal effect is the live question.

The email claim earns the mildest verdict, and it is now a verdict the chapter has actually earned rather than assumed. Code 7.16 estimated the email association separately in each segment. Report what it found in the deliverable's own words: email engagement is positively associated with spend in every segment, in the specified model, with the four conditional estimates falling in a band the deliverable should print; the joint test of whether those associations differ is reported alongside them; and a large p-value on that test does not establish that the association is identical across segments, because failing to detect a difference is not evidence of sameness, especially where the smaller segments carry the widest intervals. What the chapter cannot supply from any of this is a price for additional sends — the day-grain version of the claim was disqualified by the calendar in Section 7.5, and no model here varies send frequency at all. Doubling frequency on this evidence would be paying twice for the same

association; the honest sentence funds the content of the engaged journeys, not the volume, pending a frequency test.

The average-order-value finding gets a fourth paragraph and its own careful verb. After adjusting for units per order and occasionwear share, the estimated suburban difference becomes much smaller — these two variables statistically account for most of the observed premium in this dataset. Say it that way. The tempting version, that the premium runs through composition, asserts a pathway, and a pathway is a causal structure that two fitted models cannot establish. The adjustment claim is exactly as useful to the meeting, agrees with Chapter 5's certified decomposition, and does not require a retraction when someone asks how it was tested.

Two further translation disciplines round out the deliverable. First, magnitudes travel in decision units and with their intervals: "$0.85 less per day of recency, interval $0.70 to $1.00" prices a win-back trigger's timing; "recency was highly significant" prices nothing. Every quoted coefficient carries units, direction, the held-constant list in one clause, and its interval — the four-element metric discipline of Section 3.5.1, reborn for model outputs — and where the interval was computed with robust standard errors, which in this chapter is everywhere, the deliverable says so once in a methods line rather than in every sentence.

Second, the model's silences are reported alongside its statements: the drivers model says nothing about levers not in the data (creative quality, price perception, service), nothing about individuals (these are adjusted differences averaged over customers), nothing about combinations of predictor values the base barely contains, and nothing about the future. The one-page deliverable for the VP therefore has a fixed anatomy: the four verdict paragraphs; the coefficient table with intervals, in units; the composition exhibits that discipline the discount claim; the two conditional app associations from the interaction, in dollars, because that is the finding the four journeys can act on this fall; the specification-sensitivity range for any coefficient the recommendation leans on; and a closing line that names the next instrument — the questions this analysis cannot answer are precisely the ones an experiment can, and the cheapest of those experiments should be in the field before the next budget cycle. Drivers analysis, honestly delivered, does not end arguments; it upgrades them.

7.14 Business Analytics in Practice

This section turns from the fictional case to three ways regression is used — and misread — in professional marketing analytics, where it is at once the most institutionalized method in the field and the one whose outputs are most routinely pushed past what they can bear. The second and third vignettes are composites of recurring professional patterns rather than reports about a single named organization.

7.14.1 Marketing Mix Modeling

Marketing mix modeling is regression's flagship deployment and one of the oldest continuous analytics practices in the industry. It descends directly from the marketing-mix concept (Borden, 1964): regress a brand's sales, usually weekly and by market, on its marketing inputs — media

spend by channel, price, promotion, distribution — plus the confounders this chapter taught you to name first: seasonality, holidays, competitor activity, macro conditions. The fitted coefficients become the currency of budget season, translated into contribution charts and response curves that answer, channel by channel, what a marginal dollar returned. For decades the method lived mostly at large consumer-goods firms with the data and patience for it; the privacy era revived it broadly, because as cookie deprecation, platform walled gardens, and tracking limits degraded user-level attribution, an approach that needs no individual tracking — aggregate spend in, aggregate sales out — became newly attractive.

Practitioners at the center of that revival have been candid about the method's fragilities, and the list reads like this chapter's table of contents at industrial scale: limited variation in the spend data, strong correlation among channels that are planned together, selection effects when spend chases demand, and results acutely sensitive to specification (Chan & Perry, 2017). Three of those deserve unpacking, because they are the difference between a model that is fitted and a model that is governed.

The first is that the specification is governed rather than merely chosen. In a mature practice the variable list, the functional forms for diminishing returns and carryover, the treatment of price and distribution, and the set of control variables are agreed in advance, documented, and changed through a review process — because a coefficient that moves when a control is added is a coefficient whose value is partly an artifact of a modeling choice, and an ungoverned specification lets that choice be made, unrecorded, by whoever ran the model last. The second is the specific fragility of the time controls. Seasonality, holiday timing, and the promotion calendar are the confounders that matter most, and they are also the ones most entangled with the media plan, since campaigns are scheduled into exactly the weeks when demand is highest; a model with too coarse a time control credits the season to the media, and a model with too fine a time control absorbs the media effect into the calendar. Neither error announces itself in the fit statistics. The third is collinearity among channels: media budgets move together, up in the fourth quarter and down in the first, so the model is asked to separate contributions that the data present as a bundle — Section 7.10's problem, with a budget attached to its answer.

One distinction governs how the output may be used, and it is this chapter's distinction wearing an industrial suit. A marketing mix model used for historical attribution answers a descriptive question: given what we spent and what happened, how does the model apportion the sales we observed? A claim that moving next year's budget from one channel to another will cause a particular return is a different claim, resting on the assumption that the fitted relationship would survive an intervention that changes the very spend pattern the model was estimated on. Careful teams state which of the two they are doing, and increasingly they pay for the upgrade the same way this chapter recommends: deliberate spend experiments — geographic holdouts, staged rollouts — whose results calibrate and correct the regression. The practice lesson: the world's most consequential marketing regressions survive not because the fitting is clever but because the checking is institutional.

7.14.2 Confounding in the Wild

The second vignette is a composite of a pattern retail analytics teams meet regularly. A national retailer's team estimated discount elasticity from two years of transaction data: how much does volume respond when price is cut? The first model said, absurdly, that discounts barely mattered — and a specification review found the reason. Promotions clustered in December, when volume would have surged anyway; and the deepest discounts were routinely applied to the slowest merchandise, so discount depth was correlated with weak demand by managerial construction. The calendar pushed the estimate up; the clearance policy pushed it down; the two biases nearly canceled into a number that looked precise and meant nothing.

The repaired analysis compared like with like — within-season, within-category, promotion versus non-promotion weeks for the same items — and the elasticity that emerged was several times larger and finally decision-grade. The lesson is Section 7.5 operating at line-item scale: before trusting any observational coefficient, ask who chose the variable's values and why — because in business data the treatment is almost never assigned at random; it is assigned by someone's policy, and the policy is the confounder. It is also a lesson about the danger of two biases in opposite directions, since a badly confounded estimate does not always look wrong. Sometimes it looks reasonable, which is worse.

7.14.3 Key Driver Analysis

The third vignette is key driver analysis, the regression genre most students will meet first: customer-satisfaction and brand trackers, where survey attributes — ease of use, value for money, feels like a brand for me — are regressed on overall satisfaction or intent, and the coefficients are relabeled impact scores and sorted into a bar chart. The genre is useful, because it prioritizes attention across dozens of attributes that no one can weigh by intuition. It is also where every pathology this chapter names tends to appear at once, which makes a driver deck a good exam for a new analyst.

The attributes are strongly intercorrelated, because a satisfied customer rates everything higher, so the individual impact scores are multicollinearity's unstable division of shared credit, per Section 7.10 — and the ranking that the bar chart implies is precisely the quantity that instability makes unreliable. The causal direction is asserted rather than established: satisfied customers may rate value higher because they are satisfied, which is the reverse-causation rival from Section 7.2 sitting in the survey instrument itself. And the standard chart's language — improving this attribute by one point will lift satisfaction by X — is Section 7.8's coefficient-as-lever fallacy printed as a recommendation. Careful practitioners keep the genre honest by reporting attribute groups rather than pretending to separate entangled twins, by triangulating stated importance against modeled association, and by writing the deck's verbs in association language. Which mostly means that the analyst in the room did to the vendor's regression exactly what Table 7.5 does to an assistant's.

7.14.4 In Your First Analyst Job

In your first analyst job, these vignettes compress into one expectation: you will spend more meeting-room minutes defending regressions from misreading than building them. The mix-model budget chart, the elasticity estimate, the driver bar chart — each will arrive in your inbox with causal language already attached, and the marginal value you add is rarely a better fit; it is the audit of signs, magnitudes, units, and diagnostics, the confounder named before the decision, the specification recorded so that next quarter's model is comparable to this one, and the verb corrected before the slide ships. The statistics of this chapter are a solved problem in every software stack on earth. The judgment — what the coefficient can and cannot carry — is the job.

7.15 Ethics, Causal Overclaiming in Stakeholder Language

The Business Analytics in Practice section described analysts defending regressions from misreading. This section examines the defense as an ethical obligation, extending the guide's running discussions — data use (Section 1.13), problem framing (Section 2.12), measurement design (Section 3.13), cleaning as editorial power (Section 4.14), honest summarization (Section 5.14), and differential treatment (Section 6.16) — to the act this chapter adds: stating what moves what. The chapter's ethics is not a caution about a side effect. Overclaiming is the technique's characteristic temptation, because association analysis produces artifacts that look exactly like causal answers, and the analyst is the only person in the chain positioned to know the difference.

Begin with how overclaiming actually happens, because the mechanism is rarely a lie. It is drift. The analyst writes "app ownership is associated with $140 higher spend, holding recency, tenure, engagement, and segment constant." The deck version becomes "app users spend $140 more." The executive summary becomes "the app drives $140 per customer." The board slide becomes "app adoption is a $1M growth lever." Each rewrite is shorter, each was made in good faith by someone optimizing for clarity, and the final sentence claims something the analysis never measured. No one decided to deceive; the qualifications were simply the least memorable part of every handoff, and compression removed them in order. The drift is worst under exactly two conditions this chapter creates: when the finding is quantified, because numbers survive compression and caveats do not, and when the finding flatters an existing plan, because the app team was always going to propose the install campaign. AI assistance intensifies both — Section 7.11 showed that assistants narrate causally by default, which means the first draft of the deck sentence now often arrives pre-drifted, and the analyst's role shifts from author to editor of claims.

Why does it matter, beyond precision for its own sake? Because overclaimed drivers move real money onto weak evidence, and the costs land asymmetrically — the asymmetric-cost lens of Section 2.7, applied to language. Fund the install campaign on the drifted claim, and the cost of being wrong is a wasted budget and a metric that quietly fails to move, discovered quarters later and attributed to execution. Cut the discount ladder on the drifted claim, and the cost lands on the Discount-Dependent segment — a quarter of the treatment map loses the lever most

associated with its purchasing, on evidence that never tested how it responds — and on the P&L in the holiday quarter. And there is a second-order cost, paid by the discipline itself: every quantified overclaim that later fails burns credibility that the next honest analysis must spend to be believed. The analyst's overclaim is borrowed against colleagues' future evidence.

The discipline this guide installs has three parts, all rehearsed in this chapter and none of them new machinery. First, the verb registry of Section 7.2, applied at the point of writing: "is associated with," "predicts," and "differs by" for this chapter's findings; "drives," "causes," and "will lift" reserved for evidence that earned them, with the earning stated. The registry is a writing rule, which means it is auditable — a reviewer can grade a deliverable's verbs against its methods in five minutes, and Exercise 7.7 does. Second, the drift defense: the analyst controls the source document, so the source document carries the qualification inside the sentence, not in a footnote — "app users spend $140 more than otherwise similar non-users, a difference this analysis cannot attribute to the app itself" survives one more round of compression than the same content footnoted. Anticipating compression is part of writing the sentence. Third, the escalation duty: when a drifted claim surfaces in a meeting — someone else's slide, quoting your number with an upgraded verb — the analyst of record corrects it, with the same professional obligation as correcting a wrong number. The correction is rarely comfortable and is the whole point of having an analyst of record: Chapter 1 defined the role as accountability for what the analysis actually establishes, and the verb is part of what it establishes.

One boundary keeps the discipline honest in the other direction, because refusing all causal language is its own failure — an analyst who answers every question with "correlation is not causation" has abandoned the field to whoever will overclaim. The evidence for a relationship comes in grades: raw association; association with confounders named; association robust to controls and to specification, which is what Code 7.17 measures; association consistent with a designed mechanism; and, above them all, experimental evidence. Honest stakeholder language locates the finding on that ladder and states what would move it up a rung: "this is a controlled association that holds across four specifications; the December test would make it causal evidence." The analyst's job is not to withhold judgment forever; it is to price the judgment correctly and name the purchase that would upgrade it. That framing — evidence has a price list, and experiments are how you pay — is the bridge on which this chapter ends.

CONCEPT

The Verb Is Part of the Finding

A drivers deliverable makes two kinds of claims at once: what the numbers are, and what kind of evidence they constitute. The second claim lives in the verbs, and it is as checkable as the first. Before any deliverable ships: read every sentence that contains a coefficient and grade its verb against the method that produced it; put the qualification inside the sentence the meeting will quote, because compression strips everything outside it; and when your number surfaces later wearing a stronger verb, correct it as you would correct a wrong figure. An analysis can be arithmetically perfect and still be wrong in the only sentence anyone remembers.

Source: Course concept developed for this guide, informed by Freedman (1991) and Wasserstein and Lazar (2016).

7.16 Chapter Summary

This chapter took up the question Chapter 6's treatment map could not answer — what moves these customers? — and equipped the budget-lock decision with an honest drivers analysis. The main point is the standing distinction the chapter opened with and never set down: association is what marketing data records, causation is what marketing decisions require, and the analyst's value lies in measuring the first rigorously while refusing to let it impersonate the second. "Driver" entered the chapter as a word on probation, and it leaves the same way — usable, but only with its evidence named.

The chapter's machinery arrived in two movements. The measurement movement built covariance — with the population quantity and its sample estimate distinguished, because the software reports one of them and the definition had better name which — and then the Pearson correlation coefficient, whose division by two standard deviations makes it indifferent to the convention that the covariance is not. It then cataloged what the correlation cannot buy: nonlinearity, outlier sensitivity, and mixed populations, each answered by the always-plot discipline of Anscombe's quartet, and each demonstrated on StyleCraft's base, where a hand-computed r of 0.20 between frequency and monetary value taught the mixed-population lesson on ten rows. Confounding gave the rivals of every association their mechanism, with the confounder defined by position — a pre-existing common cause — rather than by the loose test of being correlated with both, and with the calendar installed as marketing's standing confounder. Segment composition entered on different terms: an estimated segment label built from same-window behavior is a descriptive stratifier that exposes what a pooled number is averaging over, not a clean pre-treatment control, and the chapter was careful to claim only the first.

The modeling movement built regression from the fitted line upward: coefficients read aloud in native units, least squares as a criterion with a mean's sensitivities, residuals as the model's confession, and R^2 as a baseline comparison whose familiar properties hold under stated conditions — in-sample, with an intercept, on the same observations, in nested models. Multiple

regression delivered the ceteris paribus reading and its strict boundaries: other included things equal, an adjustment computed from a fitted surface rather than a matching of real customers, and an estimate that can extrapolate where few customers share the relevant combination of values. Omitted variable bias supplied the direction logic for reasoning about everything excluded, with the qualification that in a multiple regression the relevant association is the one that survives the model's other predictors. Dummy variables and reference categories brought the four segments into the model as evidence, with the intercept correctly described as the expected response at every predictor's zero rather than as the reference group's mean. The interaction term let one association differ across groups, read conditionally and reported as two dollar figures rather than one significance verdict.

The evidence-reading section calibrated the uncertainty columns at working level and connected them to the diagnostic that precedes them: a funneling residual plot is heteroskedasticity, heteroskedasticity distorts standard errors without biasing coefficients, and the repair is a robust covariance estimator requested at fit time — so every model in the labs was fitted with HC3, and the labs printed the nonrobust and robust standard errors side by side to show exactly what changes and what does not. The p-value was defined through the test statistic it is computed from, under the null hypothesis and the model's assumptions, and read with the American Statistical Association's three warnings attached. Multicollinearity was diagnosed with variance inflation factors as well as pairwise correlations, deliberately without a universal cutoff, and described honestly: it may leave in-sample fit largely intact while destabilizing coefficients, and it can weaken out-of-sample prediction when the relationships among predictors shift. Prediction and explanation were separated as modeling goals with different tolerances.

The AI section named the assistant-era failure modes — the confident causal narration, the silent specification, the plausible-but-impossible coefficient, the default covariance estimator on funneling residuals, the unconditional reading of a conditional coefficient — and installed the four-point audit of signs, magnitudes, units, and diagnostics that the labs then rehearsed. Eighteen numbered cells carried the argument into executable form: the two covariances computed side by side, the labeled miniature scatter with its mean lines, a correlation matrix preceded by the completeness check that makes it one exhibit rather than several, whale sensitivity, faceted and within-segment views of the discount claim, the composition features built and reconciled against certified totals, robust and nonrobust inference contrasted, the funnel measured rather than eyeballed, the specified model with its encoding asserted, a coefficient table carrying units, variance inflation factors, Chapter 5's decomposition remeasured, the designed interaction reported as two conditional associations, the email claim tested rather than assumed, a specification sweep, and a leakage demonstration in which a model handed the outcome under another name returns an R^2 of one. The interpretation, practice, and ethics sections carried the findings into the meeting room, where the verbs proved to be part of the finding and the analyst of record proved to be their editor.

Looking ahead, the drivers model was built to explain: its coefficients were the deliverable, and their honesty was the discipline. The next question is the one the model was not built for —

how well do these patterns hold on customers the model has never seen? The moment a model's outputs, rather than its coefficients, become the product, the obligations change: targets and horizons must be defined, data must be split, baselines must be beaten, and a new class of silent failure — the model that grades itself on information it should never have had, of which Code 7.18 was a deliberately obvious instance — becomes the central danger. Chapter 8 takes up predictive modeling for marketing decisions, where Part II's machinery turns from explaining the past toward scoring the future, and where the too-good result replaces the wrong verb as the red flag an analyst is paid to notice. Further out, Chapter 11 supplies the instrument this chapter kept naming and could not use: experimental design, where random assignment severs the backdoors that every observational coefficient in this chapter had to reason around, and where the verbs this chapter rationed can finally be earned.

7.17 Exercises for Practice and Homework

The following exercises practice the chapter's main habits: predict signs before computing, plot before believing, read every coefficient in its units against its reference, read conditional coefficients conditionally, reason omitted-variable-bias directions, match the covariance estimator to the residuals, run the four-point audit on every fitted model, and grade every verb against its evidence. They are organized into two groups. Core chapter practice is the required path and should be completed by every student, and it holds the two homework submissions from which your instructor will assign a subset; Assignment #3, Drivers and Predictive Modeling, draws on the homework sets of this chapter and Chapter 8. In-class activities are prepared for discussion rather than submitted. Each exercise also carries its assignment label — required practice, homework submission, or in-class discussion — so that instructors can assign selectively.

7.17.1 Core Chapter Practice

Exercise 7.1 Concept Check (Required Practice)

Answer each in two or three sentences, in your own words.

1. State the difference between association and causation, and give the opening case's app claim as an example of the gap between them.
2. Why is covariance a direction detector but not a strength measure, and what operation converts it into one? State also which covariance pandas reports, and explain why the same choice does not change the correlation.
3. An analyst reports r = 0.02 between tenure and average order value and concludes the two are unrelated. Name two distinct ways this conclusion could be wrong, citing the section that covers each.
4. Define a confounder by its position rather than by its correlations. Then explain why "related to both variables" is too broad a test, naming the two other kinds of variable that pass it and stating why adjusting for each causes a different problem.

5. State the ceteris paribus reading of a multiple regression coefficient, and then state three distinct boundaries on it — one about the variable list, one about what the adjustment actually is, and one about the regions of the data where it may be extrapolating.
6. Why must one category be excluded when a categorical variable enters a regression as dummies, and what does the intercept represent in a model that also contains a numeric predictor?
7. A model contains app_user, loyalty_enrolled, and their product. State what each of the three coefficients means, and write the expression for the app association among loyalty-enrolled customers.
8. State the three familiar properties of R^2 together with the conditions under which they hold, and give one situation in which a reported R^2 would not satisfy them.
9. A coefficient has p = 0.001 and a 95 percent confidence interval of $0.40 to $0.62 per unit. Write the honest one-sentence report, state precisely what quantity the p-value is the probability of, and name the misreading the p-value alone invites.
10. A residual plot funnels outward as fitted spend rises. Say what that is called, what it does and does not distort, and what the analyst should change about the fit. Then state what stays exactly the same after the change.
11. What does a variance inflation factor measure that a pairwise correlation matrix cannot, and why does this guide decline to publish a cutoff for it?
12. State the difference between a model built to explain and a model built to predict, and name one practice that is a failure in one and tolerable in the other.

Exercise 7.2 Predict the Signs (Required Practice)

Before running Lab 7.1 Part B, write the predicted sign (positive, negative, or near zero) of the correlation between monetary and each of: recency_days, frequency, aov, discount_share, store_share, orders_per_month, revenue_per_month, app_user, and email_opt_in — with one sentence of reasoning each, citing Chapter 5's certified findings or Chapter 6's segment profiles where they apply. Mark in advance which two of the nine you expect to be arithmetic relatives of monetary rather than behavioral findings, and say how you would tell the difference from the matrix alone. After running Code 7.4, grade every prediction and write one paragraph on your largest miss: what did the data know that your reasoning did not, or what did the pooled computation conceal that Section 7.4's tools would reveal?

Exercise 7.3 Read the Coefficient Aloud (Required Practice)

For each fitted result below, write the single honest sentence that reports it — units, direction, held-constant clause, and a verb that stops at the evidence. Then write the overclaimed version a deck would produce, and underline the exact words that changed the claim.

1. In the Code 7.11 spend model: recency_days coefficient −0.85, HC3 interval −1.00 to −0.70.
2. In the Code 7.11 spend model: seg_suburban coefficient −180, HC3 interval −220 to −140, with Urban Loyal Core as the reference.

3. In the Code 7.11 spend model: discount_share coefficient −310, HC3 interval −390 to −230. Be careful with the units on this one; state what a one-unit difference in discount_share actually is.
4. In the Code 7.14 controlled AOV model: seg_suburban coefficient 4, HC3 interval −2 to 10.
5. In the Code 7.15 interaction model: app_user coefficient 45, app_user:loyalty_enrolled coefficient 60, both intervals excluding zero. Report the two conditional associations rather than the three coefficients, and explain why the deck should never quote the 45 on its own.

Exercise 7.4 Hand Arithmetic on the Miniature (Required Practice)

Using only the ten-customer miniature of Code 7.1 and no code: verify the sample covariance of 26.73 between frequency and monetary by computing the ten deviation products and summing them, and state what the population covariance would be and why the two differ. Explain in one sentence why C002 and C010 contribute exactly zero to the sum despite being the two largest spenders. Recompute r with C006 removed and state what the change reveals about single-observation influence. Sketch the scatterplot by hand with both mean lines drawn, label the quadrant each customer falls in, and mark the region that would mislead a straight-line summary. Close with one sentence connecting the exercise to Anscombe's argument.

Exercise 7.5 The Full Drivers Deliverable (Homework Submission)

Complete Labs 7.1 and 7.2 on the certified files and assemble the VP's one-page deliverable per Section 7.13: the four verdict paragraphs on the app, discount, email, and average-order-value findings; the coefficient table with HC3 intervals and a units column; the pooled-versus-within-segment discount exhibit with its faceted plot; the two conditional app associations from the designed interaction, in dollars; the specification-sensitivity range for any coefficient your recommendation leans on; and the closing line naming the experiment that would upgrade the weakest verdict. State once, in a methods line, that all intervals are heteroskedasticity-robust, and say why. Submissions are graded on the verdict paragraphs' verbs and the audit evidence as heavily as on the code.

Exercise 7.6 AI Regression Audit (Homework Submission)

Give an AI assistant the customer_features table and this deliberately loose prompt, verbatim: "Find what drives customer spend and tell me what to do about it." Then audit the response with Table 7.5's four points. Check every reported sign against your Exercise 7.2 predictions. Check every magnitude against a hand-computed anchor — group means, Chapter 5 gaps — and note whether any adjusted figure exceeds its descriptive counterpart, which is permitted but requires an explanation grounded in the model's variables. Restate every quoted coefficient in its units, flag any cross-unit comparison, and flag any main effect quoted unconditionally from a model that contains its interaction. Determine whether the assistant examined residuals, which covariance estimator it used, and whether it connected the two. Separately, list every causal verb

in the reply and grade each against Section 7.2's registry. Report what specification the assistant chose — variables, reference category, interactions — and whether it disclosed choosing. Document the full exchange per the AI-use documentation template in Appendix D, and conclude with two sentences on which audit point caught the most serious problem.

7.17.2 In-Class Activities

Exercise 7.7 Grade the Verbs (In-Class Discussion)

Each sentence below quotes a defensible analysis and states an indefensible conclusion, or states a defensible conclusion in indefensible language — or is honest as written. Classify each, rewrite the failures, and cite the section that governs your verdict.

1. "Customers acquired through influencer channels spend 30 percent more; shifting acquisition budget to influencers will raise average customer value 30 percent."
2. "Email-engaged customers buy more often, holding segment and tenure constant; email engagement is associated with higher frequency, and a frequency test would tell us whether more email causes more buying."
3. "The interaction term was significant at $p < 0.01$, confirming that loyalty drives app value."
4. "Suburban customers' AOV premium runs almost entirely through basket size and occasionwear mix." Classify this one carefully: the arithmetic behind it is correct and the verb still fails. Name what kind of claim "runs through" makes, and rewrite it as the adjustment claim the two models actually support.
5. "R^2 was 0.94, so the spend model is excellent and its coefficients can be trusted."
6. "The email association is consistent across all four segments — the interaction test came back insignificant."
7. "We cannot say anything about discounts because correlation is not causation."

Exercise 7.8 Find the Flaw in the Drivers Analysis (In-Class Discussion)

Each scenario below contains at least one flaw from this chapter. Name it, cite the section, and state the repair.

1. An analyst regresses monetary on aov, frequency, and recency, reports $R^2 = 0.96$, and recommends the model as near-perfect. Explain both why the fit is so high and why it is not exactly 1, and state what specification would make it exactly 1.
2. A deck ranks drivers by raw coefficient size: "recency (−0.85) matters less than app ownership (+140)."
3. A weekly-grain analysis finds sends and revenue correlate at 0.8 and recommends doubling December send volume.
4. A model of spend includes all four segment dummies plus an intercept, and the software drops one "for technical reasons" the analyst does not mention.
5. An analyst inspects the residual plot, notes the funnel, writes it up as a limitation, and then quotes the default confidence intervals from the same fit in the deliverable.

6. Two engagement variables are each insignificant in the full model, so the analyst concludes engagement does not matter — although the model fits far worse when both are removed, and both carry variance inflation factors above anything else in the table.
7. A colleague reports the discount–spend association is negative "even after controls," but the control list omits segment membership; state the likely bias and its direction using Section 7.8's logic, and state what the direction rule requires you to reason about that a raw cross-tabulation of segment against discount exposure would not settle.
8. An analyst adjusts for a variable measured after the exposure of interest — customers' post-campaign satisfaction scores — on the grounds that it correlates with both the campaign and spend, and reports that the campaign association disappeared.

7.18 Glossary of Terms

This glossary includes only the terms introduced in this chapter. Each definition is tied to the sources used in the chapter rather than added for decoration.

Association. The tendency of two variables to move together in observed data, so that knowing one improves a guess about the other; symmetric, observational, and consistent with multiple causal structures (adapted from Simon, 1954; Freedman, 1991).

Causation. The relation in which changing one variable, with all else undisturbed, would change another; directional and interventional, and never established by association alone (adapted from Simon, 1954; Freedman, 1991).

Ceteris paribus. The "other things equal" reading of a multiple regression coefficient: the estimated difference in the response per unit difference in a predictor among observations alike on the model's other included variables, computed as an adjustment under the model's assumed functional form rather than by matching observations (adapted from Wooldridge, 2020).

Confidence interval (coefficient). The range of coefficient values compatible with the data at a stated confidence level, centered on the estimate; its width is the finding about precision, and its validity depends on the standard error it was built from (adapted from Wasserstein & Lazar, 2016; Wooldridge, 2020).

Confounder. A pre-treatment variable that influences, or helps determine, both the predictor or exposure and the outcome, creating or distorting their observed association. Correlation with both variables is a warning sign rather than proof: a variable on the pathway between them, or a common consequence of them, is also correlated with both and requires different handling (adapted from Simon, 1954; Freedman, 1991).

Correlation matrix. The grid of pairwise Pearson correlations among a set of variables; a screening exhibit whose characteristic misreadings — ranking drivers, celebrating near-perfect cells, treating cells as levers — this chapter catalogs, and whose cells are computed pair by pair over the observations complete for that pair, so incomplete columns silently change the population behind each figure (behavior per pandas development team, 2026).

Covariance. The mean product of two variables' paired deviations from their means in a population; estimated from a sample by dividing the sum of the paired deviation products by n −

1. Positive for same-direction movement, negative for opposite, with a magnitude that depends on both variables' units (adapted from Rodgers & Nicewander, 1988).

Dummy variable. An indicator predictor equal to 1 for membership in a category and 0 otherwise, by which categorical variables enter regression; c categories require c − 1 dummies (adapted from Wooldridge, 2020).

Heteroskedasticity. Residual variance that differs systematically across the range of fitted values, visible as a funnel in a residual plot. It does not bias the fitted coefficients and it does distort the standard errors computed under the default assumption, so the intervals and p-values built from them are unreliable until a robust covariance estimator is used (adapted from Wooldridge, 2020; MacKinnon & White, 1985).

Interaction term. A predictor formed as the product of two others, whose coefficient estimates how the association between one predictor and the response differs across values of the other; once it is present, each main-effect coefficient is the association for its variable when the interacting partner equals zero (adapted from Wooldridge, 2020; James et al., 2021).

Least squares. The estimation criterion that chooses regression coefficients to minimize the sum of squared residuals. The solution is unique when the predictor matrix has full column rank; under perfect redundancy among predictors, software may still return a fitted result whose individual coefficients are not identified. Like every mean, it is sensitive to extreme observations (adapted from James et al., 2021).

Multicollinearity. Strong overlap in the information carried by a model's predictors, whether pairwise or between one predictor and a combination of others, which leaves joint explanatory power largely intact while making individual coefficients imprecise and unstable across specifications (adapted from Wooldridge, 2020).

Multiple regression. The modeling of a numeric response as a linear function of two or more predictors fitted jointly, so that each coefficient carries a ceteris paribus reading with respect to the model's other variables (adapted from Wooldridge, 2020).

Omitted variable bias. The distortion of an included predictor's coefficient when a variable related to both the response and that predictor is left out of the model. In the simplest linear case its direction follows the product of the omitted variable's effect on the response and its association with the included predictor; in a multiple regression the relevant association is the part remaining after the model's other predictors are taken into account (adapted from Wooldridge, 2020).

p-value (coefficient). The probability, assuming the stated null hypothesis and statistical model are correct, of obtaining a test statistic at least as extreme as the one observed; for a regression coefficient, the null hypothesis is that the coefficient is zero given the model's other variables. It is a measure of surprise under that hypothesis, not of size, importance, or truth (adapted from Wasserstein & Lazar, 2016).

Pearson correlation coefficient (r). The sample covariance of two variables divided by the product of their sample standard deviations: a unit-free, symmetric measure of linear association

ranging from −1 to +1, unaffected by the denominator convention that changes the covariance (adapted from Pearson, 1895; Rodgers & Nicewander, 1988).

R^2 (coefficient of determination). The proportion of a response's variation around its mean that a fitted model accounts for. For in-sample ordinary least squares fitted with an intercept to the same observations, it lies between 0 and 1, cannot decrease when predictors are added to a nested model, and equals r^2 in the simple one-predictor case; it grades fit to the sample and never, by itself, usefulness for a decision (adapted from Wooldridge, 2020; James et al., 2021).

Reference category. The category of a categorical predictor represented when all of its dummy variables equal zero, against which every dummy coefficient is read as an adjusted difference. With an intercept in the model, the intercept is the expected response for the reference category when all other predictors are at their coded zero or reference values — the reference group's observed mean only when the model contains nothing else (adapted from Wooldridge, 2020).

Residual. An observation's actual response minus the model's predicted value; the itemized record of what the model missed, whose patterns are the primary diagnostic evidence about model form (adapted from James et al., 2021).

Robust standard error (heteroskedasticity-consistent). A standard error computed under an estimator that does not assume constant residual variance, requested at fit time and used throughout this chapter's labs in its HC3 form. It changes the standard errors, confidence intervals, and p-values, and leaves the fitted coefficients and predictions unchanged (adapted from MacKinnon & White, 1985; Seabold & Perktold, 2010).

Simple linear regression. The modeling of a numeric response as an intercept plus a slope times one predictor, with coefficients estimated by least squares and read in the native units of the variables (adapted from James et al., 2021).

Spurious correlation. An observed association that reflects only the operation of confounders or chance, so that neither variable responds to the other (adapted from Simon, 1954).

Standard error (coefficient). The estimated sampling variability of a fitted coefficient — how much the estimate would move across repeated samples from the same population — and the quantity from which confidence intervals and test statistics are built. It describes the precision of the estimate, not the spread of the underlying variable (adapted from Wooldridge, 2020).

Variance inflation factor. A diagnostic measuring how much a coefficient's estimated variance is inflated by the predictor's relationships with all the other predictors together, and therefore able to detect overlap that no pairwise correlation reveals. It is a diagnostic rather than a verdict, and this guide publishes no universal cutoff for it (adapted from Seabold & Perktold, 2010; Wooldridge, 2020).

7.19 Further Readings

Students who want additional background may begin with the following readings. Foundations of the association–causation discipline are listed first, methods second.

- Anscombe (1973) for the four-dataset argument that statistics without graphs deceive — three pages, one figure, and the origin of this chapter's always-plot discipline.
- Freedman (1991) for the classic essay on what regression can and cannot establish — "shoe leather" as the name for the evidence-gathering that no model replaces; the extended companion to Sections 7.2 and 7.15.
- Shmueli (2010) for the definitive statement of prediction and explanation as distinct modeling goals; the foundation under Section 7.10 and the doorway to Chapter 8.
- Wasserstein and Lazar (2016) for the American Statistical Association's consensus statement on p-values — short, official, and the corrective to most of what students have absorbed about significance.
- James et al. (2021), Chapter 3, for the standard accessible treatment of linear regression, one level of formality above this chapter.
- Chan and Perry (2017) for a candid practitioner account of media mix modeling's challenges — the industrial-scale version of every fragility this chapter names.

7.20 References

Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician, 27*(1), 17–21. https://doi.org/10.1080/00031305.1973.10478966

Borden, N. H. (1964). The concept of the marketing mix. *Journal of Advertising Research, 4*(2), 2–7.

Chan, D., & Perry, M. (2017). *Challenges and opportunities in media mix modeling*. Google Inc. https://research.google/pubs/challenges-and-opportunities-in-media-mix-modeling/

Freedman, D. A. (1991). Statistical models and shoe leather. *Sociological Methodology, 21*, 291–313. https://doi.org/10.2307/270939

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *Journal of the Anthropological Institute of Great Britain and Ireland, 15*, 246–263. https://doi.org/10.2307/2841583

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://doi.org/10.1007/978-1-0716-1418-1

MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics, 29*(3), 305–325. https://doi.org/10.1016/0304-4076(85)90158-7

pandas development team. (2026). *pandas.DataFrame.corr* [Software documentation]. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London, 58*, 240–242. https://doi.org/10.1098/rspl.1895.0041

Rodgers, J. L., & Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician, 42*(1), 59–66. https://doi.org/10.1080/00031305.1988.10475524

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference* (pp. 92–96). https://doi.org/10.25080/Majora-92bf1922-011

Shmueli, G. (2010). To explain or to predict? *Statistical Science, 25*(3), 289–310. https://doi.org/10.1214/10-STS330

Simon, H. A. (1954). Spurious correlation: A causal interpretation. *Journal of the American Statistical Association, 49*(267), 467–479. https://doi.org/10.2307/2281124

Vigen, T. (2015). *Spurious correlations*. Hachette Books.

Wasserstein, R. L., & Lazar, N. A. (2016). The ASA's statement on p-values: Context, process, and purpose. *The American Statistician, 70*(2), 129–133. https://doi.org/10.1080/00031305.2016.1154108

Wooldridge, J. M. (2020). *Introductory econometrics: A modern approach* (7th ed.). Cengage Learning.