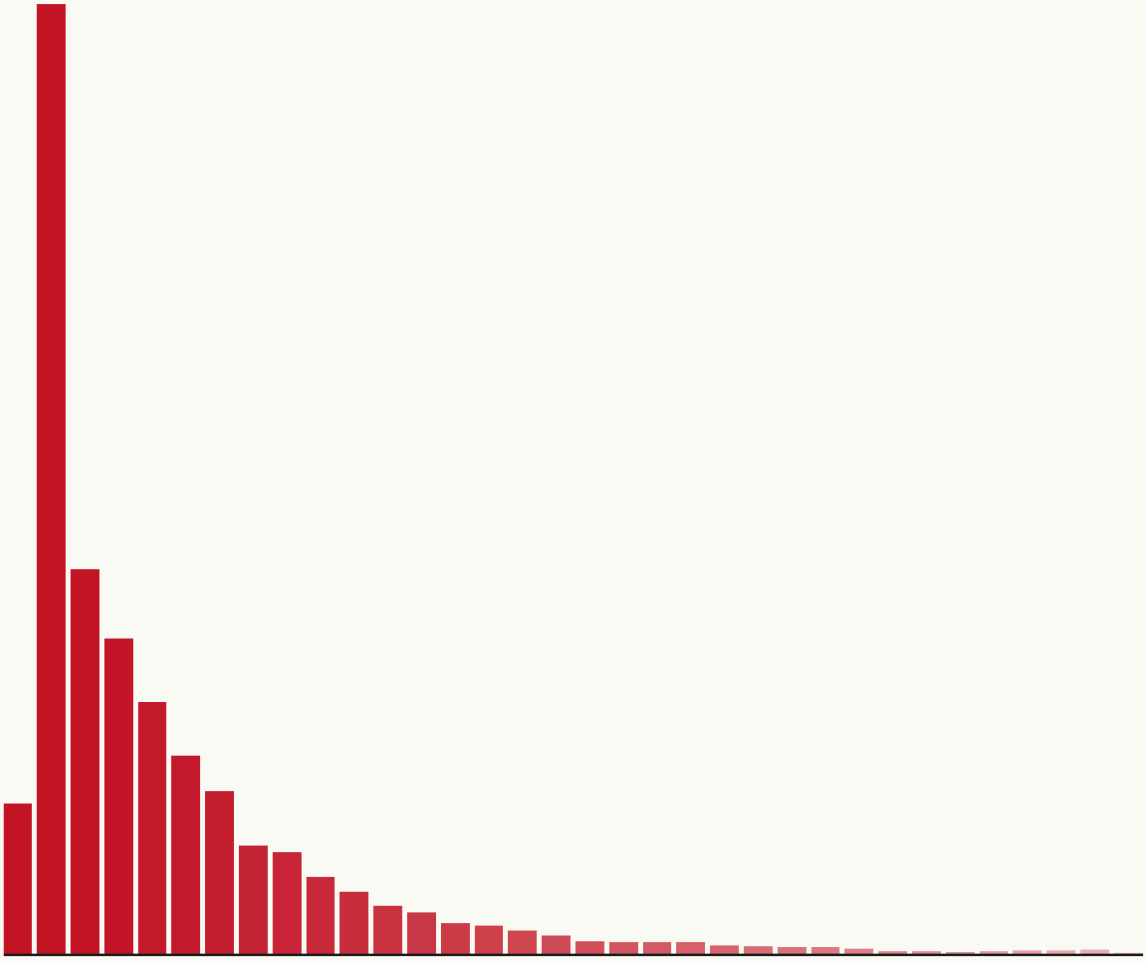
Vibes to Variables

A Methods Package for Students
New to Research, Stats, and Code



ALEX P. LEITH

THIRD EDITION

Vibes to Variables

A Methods Package for Students New to Research, Stats, and Code

Alex P. Leith

About the cover. The shape on the front is real data from this book: the distribution of message lengths across all 35,267 messages in the Twitch chat corpus of November 2018, counted in words. A tall bar at one word falls away into a long tail running out past a hundred. You meet the same distribution again in Chapter 12, drawn properly and read carefully. It is on the cover because it is the honest shape of the thing this book is about: most of what people say in a livestream chat is very short, and a method that assumes otherwise will mislead you.

Third Edition

May 2026

Table of contents

1 Welcome

Vibes to Variables

Third Edition (V2V)

This open educational resource teaches communication and media research methods as a complete arc, from the first spark of curiosity through a published, reproducible study. V2V is written for students who have never opened a stats program and have never written a research paper. It meets them where they are and walks them to a defensible study.

V2V is a methods package, not a textbook with bonuses. The book you are reading is one of several co-equal components:

- **The book** (this site): chapter prose anchored to a real, public dataset.
- **V2V Hub**: the course site that wraps this book with workbooks, cheat sheets, datasets, references, and instructor materials.
- `v2v` **R package**: friction-reducing helpers that make chapter exercises one-line calls instead of multi-step setup.
- **Beyond Vibes**: the graduate supplement, paired with MC 501, that produces a publishable white paper and conference poster.

1.1 How This Book Works

The book is organized into 14 chapters across five parts that mirror the phases of a research project:

1. **Foundation** (Chapters 1 to 3): Reframe research as disciplined storytelling, set up the workspace, ground epistemology and ethics.
2. **Planning** (Chapters 4 to 6): Literature review, theoretical lens, and a research prospectus that locks scope before any code.
3. **Operationalization** (Chapters 7 to 9): Structured listening, operationalization (NOIR), and first R contact through the codebook.
4. **Execution** (Chapters 10 to 12): Sampling and inter-coder reliability, data wrangling, visualization.
5. **Inference and Publication** (Chapters 13 to 14): Hypothesis testing with effect sizes, then a one-click reproducible portfolio rendered to GitHub Pages.

> **i** V2V 3rd Edition
>
> This third edition is a structural rewrite of the second edition: 22 chapters condensed to 14, R deferred to Chapter 9 (so the first eight chapters build conceptual scaffolding before any code), and every example anchored in the Twitch working corpus rather than four separate datasets. **All fourteen chapters are drafted.** The V2V 2nd Edition (22 chapters, music dataset) remains available at the `_archive-v2/` folder in this repository as a transitional reference.
>
> A fourth edition is in progress for Spring 2027. It widens the book from one method to a survey of the methods a research methods course carries, and moves the tool instruction out of the chapters into separate Excel, SPSS and R supplements so the text can be taught in any of them.

1.2 The Dataset

V2V 3rd Edition anchors all examples in the **Twitch working corpus**, a 1.6 GB PostgreSQL dump from a 2018 dissertation: two tables (`chat_log` for IRC messages, `stream_log` for Twitch API snapshots) covering nearly 1,700 channels across 5.85 days of public broadcasting. The `v2v` R package ships row-sampled fixtures so students never touch the full dump:

```
library(v2v)
v2v::twitch_chat()    # sampled chat_log
v2v::twitch_streams() # sampled stream_log
```

The 2nd Edition's `unified_music` dataset (Billboard + Spotify + Genius for 1,792 songs) remains available in the v2v R package as a secondary teaching corpus.

1.3 License

Author: Alex P. Leith, Southern Illinois University Edwardsville

Courses: MC 451 Research Methods in Mass Media (undergraduate); MC 501 Research Methods for Mass Communications (graduate, via the *Beyond Vibes* supplement)

Part I: Foundation

2 Chapter 1: The Science of Storytelling

Listen in Dr. Leith's voice

On a Tuesday afternoon in November 2018, a series of bots was collecting data for a dissertation. Their target was nearly 1,700 Twitch channels. Most of the channels they were monitoring were playing video games. League of Legends, World of Warcraft, Rocket League, Hand Simulator. One of the channels was streaming under the category of Art. This streamer was Bob Ross, a man who had been dead since 1995, painting happy little trees in a public-television loop. Across that week Bob Ross averaged 3,178 concurrent viewers inside the collection. xQc, a variety streamer with the highest chat volume in the collection, averaged 17,363. Both appeared on the same platform, in the same week, drawing very different audiences in very different ways.

Here is the move a social scientist makes. Stare at those two numbers and ask not what they mean, but what they are. They are **data**. Specifically, they are summaries derived from 590,876 stream snapshots collected by an automated process during five days and twenty hours of public broadcasting on Twitch. A snapshot is a single row in a database, six columns wide: channel name, stream title, game category, viewer count, timestamp, and a primary key. Multiply that row by half a million and you have a **corpus**, a body of evidence that licenses some questions and refuses others.

The first pages of a methods textbook are usually where the author tries to convince you that what comes next will be more interesting than it sounds. This book is going to try a different move. The biggest gap in students starting a research methods course is not motivational. It is conceptual. Students who are perfectly comfortable saying "I have data" rarely have a precise account of what data is, where it came from, or what claims it would actually let them make. Half a million stream snapshots is data. So is one student's interview about why she watches Bob Ross. So is a coded transcript. So are seven judges' ratings of a song's emotional valence in a music study from 2008. Data is, broadly, what you collected on purpose because you wanted to be able to defend a claim later. This book teaches you to collect data on purpose, work with it carefully, and defend the claims you build from it.

The skills are portable. The same logic that asks whether gaming streams attract different chat than non-gaming streams also asks whether news framing shapes public opinion, whether ad exposure changes purchase behavior, whether social media use correlates with political polarization. Methods do not belong to a topic. They belong to a way of thinking. The Twitch dataset is the vehicle. The destination is methodological literacy.

2.1 Why we tell stories (and why that matters for science)

The neuroscientist Lisa Feldman Barrett has spent her career dismantling the idea that the brain passively receives information about the world. In *How Emotions Are Made* (Barrett, 2017a), she argues that the brain is fundamentally a prediction machine. It generates models of reality, tests those models against incoming sensory data, and updates its beliefs when its predictions fail. Her theory of constructed emotion proposes that this prediction-error cycle is not a metaphor for the scientific method. It is the cognitive substrate of how minds work (Barrett, 2017b). We are, at a neurological level, hypothesis-testing organisms.

Will Storr, in *The Science of Storytelling* (Storr, 2019), pushes the same framework toward narrative. Stories emerged, he argues, as cognitive tools for managing social complexity. They model cause and effect, simulate outcomes, and let groups coordinate behavior around shared beliefs. When the predictions inside a story fail, we experience cognitive dissonance. The resolution is either changing the story or denying the evidence.

This reframes what research does. We tend to think of science as the opposite of storytelling. Cold. Objective. Stripped of human subjectivity. But the brain does not toggle between "creative mode" and "analytical mode." It runs the same narrative machinery for both. The difference is not in the architecture. It is in the rigor applied to testing the story, and in the public record of evidence that backs it up. Stories without data are anecdotes. Data without stories is a spreadsheet. Research is the discipline of building one out of the other.

2.2 The limits of everyday knowing

Before formalizing what that discipline looks like, it is worth being honest about how we usually make sense of the world. Earl Babbie (2021) identifies several common ways of knowing that work well enough for daily navigation but break down when the stakes demand evidence others can trust.

Tradition is what we have always done. It offers stability and continuity. It also resists updating, and it often rests on nothing more than "this is how it's done."

Authority defers to experts or institutions. This is efficient, often necessary. It is also only as reliable as the expertise itself, which can be misapplied, biased, or compromised by interest.

Common sense feels self-evidently true. It is also culturally bound and frequently contradictory ("look before you leap," "he who hesitates is lost").

Intuition is fast and sometimes insightful, drawing on accumulated experience. It is also shaped by cognitive biases, emotional states, and the availability of recent examples.

These shortcuts are fine for navigating Tuesday. They become problems when we try to build knowledge that others should trust. Public health messaging during the COVID-19 pandemic leaned on authority (government agencies), tradition (past pandemic playbooks), and common sense ("wash your hands"). The underlying scientific questions about transmission, mitigation, and vaccine efficacy demanded systematic testing that sometimes contradicted all three.

Research offers a more disciplined alternative. Not because researchers are smarter or less biased, but because the process itself is designed to expose biases to scrutiny.

2.3 The sacred flaw

Storr (2019) identifies a recurring element in compelling narratives: the **sacred flaw**. This is a deeply held but mistaken belief the protagonist clings to even as evidence mounts against it. The story's tension arises from the inevitable collision between false certainty and reality.

The **null hypothesis** plays this role in research. It is the default story: nothing interesting is happening here. Any pattern you think you see is random noise. The researcher's task is to accumulate evidence so overwhelming that maintaining the null hypothesis becomes untenable. When we "reject the null," we are forcing the data to tell a different story, one that contradicts what we initially assumed.

This framing turns statistical significance from an abstract threshold into a narrative device. A **p-value** of 0.001 does not just mean "statistically unlikely under the null." It means the old story is so incompatible with the evidence that clinging to it requires willful blindness.

Sometimes the move runs in the other direction, and an established story turns out to be the wrong one. The economists Carmen Reinhart and Kenneth Rogoff published a 2010 paper called "Growth in a Time of Debt," which argued that countries whose public debt exceeded 90 percent of GDP grew significantly more slowly than countries below that threshold. The finding became a foundational reference for

austerity policy across Europe and the United States. In 2013, a third-year doctoral student at UMass-Amherst named Thomas Herndon tried to replicate the analysis for a class assignment. He could not get the numbers to come out the same way. When he eventually persuaded Reinhart and Rogoff to share their original spreadsheet, he found an Excel formula that had excluded several countries from the average. With the formula corrected, average growth above the 90 percent debt threshold turned out to be 2.2 percent, not the negative 0.1 percent the original paper had reported (Herndon, Ash, & Pollin, 2014). The cliff disappeared because there was no cliff.

That is a worked example of the discipline. The original paper was not fabricated. It was published, peer-reviewed, and cited in policy debates. It was also wrong, in a way that only became visible when someone took the data seriously enough to ask for the file. Failure of this kind is not a sign that research does not work. It is a sign that it does. The mechanism for catching the error is built into the architecture, and the error gets caught when researchers document their work transparently enough for someone else to retrace it.

Take the Twitch puzzle from a moment ago. The null hypothesis is that there is no systematic relationship between what is on the screen (gaming versus painting) and what happens in the chat. The viewers are different sizes, sure. But the conversation might look identical. Same emote frequency. Same message length. Same patterns of who talks and how often. The null says: you noticed something. That is not yet evidence. The chapters that follow walk through what it takes to move from that hunch to a defensible claim.

2.4 Mapping narrative onto research design

If research is the disciplined work of testing a story against evidence, the components of a study map onto the components of a story. They do, with surprising precision.

2.4.1 Inciting incident: the research problem

Every story begins with disruption. The protagonist's stable world encounters something it cannot fit, and the disruption demands a response. In research, the inciting incident is an **anomaly**, an observation that does not square with existing explanations.

In the Twitch data, the inciting incident might be the Bob Ross puzzle. It might be the fact that loltyler1, a League of Legends streamer, peaked at 78,340 concurrent viewers during one stream that week while giantwaffle, a Rocket League streamer with a steady audience, topped out at 11,119. It might be something more granular: a single message in the chat log that seems out of place, a sudden spike in conversation rate, a pattern of named-streamer references that look like they are addressed to no one in particular.

Inciting incidents work the same way across the social sciences. A political scientist notices that voter turnout rose in a district despite reduced campaign spending. A public relations scholar observes that a corporate apology went viral and made the brand worse. An advertising researcher finds that a product placement in a low-budget show outperformed one in a prestige drama. Each anomaly opens a gap between observation and explanation, and that gap is where the research lives.

What distinguishes a scholarly contribution. Research methods can be understood as disciplined storytelling: you have a claim, you gather evidence, you tell the story the data supports. Graduate-level work adds a second obligation, that your story must be reproducible, falsifiable, and positioned within a cumulative scholarly conversation. A scholarly contribution in communication research does one of four things: (1) tests a theory in a new context; (2) replicates a finding using new data; (3) resolves a

conflicting finding in the literature; or (4) introduces a testable theoretical construct. Before beginning any V2V study at the graduate level, identify which of these your project accomplishes. If you cannot answer precisely, the research question is not yet graduate-level. For your own study, ask which of the four contribution types your current research question best fits, and what you would need to change for it to fit more precisely.

2.4.2 Protagonist: the researcher as detective

In detective fiction, the detective gathers clues, formulates theories, and tests them against evidence. The researcher does the same work. The parallel is not accidental. Both are doing **abductive reasoning**, working backward from observation to the most plausible explanation. Like a detective, the researcher must stay skeptical of convenient narratives and willing to revise theories when the evidence pushes back. The integrity of the investigation depends on this.

2.4.3 Antagonist: confounds, bias, and noise

The antagonist in research is not a person. It is the disorder that obscures truth. **Confounding variables** muddy causal claims. **Sampling bias** makes findings ungeneralizable. **Measurement error** introduces noise. These are not malicious forces. They are the texture of working with messy, real-world data. They function narratively as obstacles the researcher has to overcome through deliberate design.

2.4.4 Rising action: literature review and theory

Before confronting the antagonist, the protagonist needs preparation. The **literature review** is that preparation. Previous studies show what is already known, where the gaps are, and which methods have succeeded or failed. **Theory** provides the conceptual frame, the lens through which findings are interpreted and hypotheses generated.

For a Twitch study, the literature you would turn to first is **uses and gratifications theory** (Katz, Blumler, & Gurevitch, 1973), which proposes that people actively choose media to satisfy specific psychological needs. If livestream chat is meeting needs for connection, performance, or community, those needs should show up in the messages themselves. Chapter 5 walks through how a theory like uses and gratifications gets turned from a paragraph in a journal article into a one-page worksheet that points at columns in a database.

2.4.5 Climax: data analysis

The climax is the moment when the setup pays off. In research, this is the **statistical test**, the point where the accumulated evidence either supports the hypothesis or does not. Everything has led here: the question, the sample, the codebook. Now the analysis runs and the data return a verdict.

2.4.6 Falling action: interpretation and limitations

After the climax, the detective explains what the evidence reveals and acknowledges what is still uncertain. The **discussion section** does the same work. What do the findings mean? Where do they fit in the broader literature? What alternative explanations might still hold? What questions remain open? This is where intellectual honesty becomes the load-bearing virtue. Every study has limits: sample size, measurement precision, generalizability. Acknowledging those limits does not weaken the research. It strengthens it, by showing that the researcher understands the boundary of the claim.

2.4.7 Resolution: implications and future research

A story concludes by showing how the world has changed in light of what we have learned. The **implications section** does this for research. Why do the findings matter? To practitioners, to policymakers, to future scholars. The narrative may be complete, but it opens threads for someone else to pull.

2.5 Anecdote and data: complementary, not oppositional

Mass communication students are skilled storytellers. You know how to find a compelling anecdote, conduct an interview, build a narrative that moves an audience. That is journalism, and it is valuable. But journalism and science do different epistemic work.

Journalism makes the abstract concrete. It humanizes statistics, gives texture to trends, and makes audiences care by showing impact on individual lives. A profile of a single Twitch streamer who built a community out of isolation is far more emotionally resonant than a p-value.

Science establishes generalizability. It asks whether the pattern observed in one case holds across many. It quantifies relationships, controls for confounds, and builds evidence that holds up under skeptical scrutiny.

The tension is productive. Anecdotes generate hypotheses. Data test them. Data identify patterns. Anecdotes explain why those patterns matter. A complete research report uses quantitative findings to establish that the pattern is real and qualitative examples to illustrate what the pattern looks like in practice. The mistake is treating one as a substitute for the other. A moving interview with a Twitch viewer does not prove that a million others had the same experience. A statistically significant finding without human context risks being true and unpersuasive. The best social science holds both in tension.

2.6 What this book teaches

The course gives you a chance to do original research on a real dataset. The dataset is a single week of Twitch chat and stream snapshots collected between November 18 and November 24, 2018. The raw corpus contains 21,964,296 chat messages from 1,695 channels, plus 590,876 snapshots of who was streaming what game to how many people.

That raw corpus is going to play three different roles depending on the question being asked, and the difference between those roles is one of the easiest things to lose track of. The **population** is the 1,690 channels that show up in both the chat log and the stream log during the collection window: every channel that could in principle be analyzed end to end. The **working corpus** is a 50-channel subset stratified by chat volume so that the head, the middle, and the long tail of Twitch are all represented in the sample students actually load into R. The **anchor set** is eight channels inside that 50 (xqcow, forsen, sodapoppin, asmongold, loltyler1, disguisedtoast, giantwaffle, and bobross) that recur as named examples throughout the rest of the book. Bob Ross from a moment ago is one of the anchors. xQc is another.

Population, sample, and exemplar are not interchangeable. A descriptive claim about Twitch in November 2018 has to be defended at the population level. A statistical pattern observed in a sample has to be argued back up to the population it was drawn from. An exemplar like Bob Ross illustrates a phenomenon without proving it. Most of the trouble students get into in their first research project comes from sliding between these three roles without noticing. The book returns to the distinction in every part.

The dataset is a means, not an end. The book studies Twitch not because Twitch is the only thing worth studying, but because it provides a rich, accessible, genuinely interesting sandbox for learning how social science actually works. Every method you apply here transfers to any other content domain. The student

who codes Twitch messages as "directed at the streamer" or "broadcast to the room" using a reliable codebook has also learned to code news frames as "episodic" or "thematic," to classify advertising appeals as "emotional" or "rational," to categorize social media posts as "civil" or "uncivil."

The data license a range of research questions:

- Does chat behavior differ in gaming versus non-gaming streams? Are messages longer, more emote-dense, more directed at the streamer?
- Do bigger audiences produce different chat than smaller ones, or is the per-viewer rate of conversation roughly stable across audience size?
- Does the language viewers use to address a streamer change in moments when the audience is climbing versus when it is falling?
- What does a "regular" of a Twitch channel look like in the data, and how often do the same usernames appear across multiple streams?

A caveat the data itself enforces: five days and twenty hours is not a long time. Questions about how audiences evolve over months or years cannot be answered here. Questions about what was happening on Twitch during that specific window, and what the chat looked like inside it, are well supported. Working within the limits of your data is itself a methodological skill, and Chapter 11 returns to it directly.

The book is organized around the research process itself, divided into five parts that mirror the narrative arc this chapter has been describing.

Part I: Foundations (Chapters 1 to 3) sets up the intellectual and technical infrastructure. You will build habits of mind and workflow, plus the reproducibility principles that distinguish rigorous research from improvised analysis. Chapter 3 takes up research ethics, the constraint that all subsequent work has to honor.

Part II: Planning (Chapters 4 to 6) is where you design the study. Literature review, theoretical framework, research questions, and a project prospectus that maps what you will study, why it matters, and how you will proceed.

Part III: Operationalization (Chapters 7 to 9) is where you turn vague constructs into measurable variables. Sustained observation of the data, the move from qualitative noticing to quantitative coding, the construction of a codebook, and a first hands-on session with R. The book teaches content analysis end to end because depth in one method serves you better than shallow coverage of four, but it does not pretend the other major social-science methods (surveys, experiments, qualitative interviews and focus groups) do not exist. Chapter 5 introduces them as a coherent set when it discusses how theory selection shapes method choice. Chapter 9 stays focused on the first-R-contact work that the rest of the lab depends on. The V2V Hub holds the deeper hands-on material for any of these methods you want to pursue.

Part IV: Execution (Chapters 10 to 12) is the work itself. Sampling, inter-coder reliability, data wrangling, descriptive statistics, visualization. The chapters where you spend the most time in the IDE.

Part V: Inference and publication (Chapters 13 to 14) is the closing arc. You run inferential tests, interpret what they mean, and publish a portfolio that integrates the whole project into a single reproducible document.

A small note on tooling, taken up properly in Chapter 2. This book teaches the canonical version of each task. The companion V2V Hub teaches the practical version, including how to use AI assistants like GitHub Copilot to do parts of the work faster. The textbook focuses on what the discipline expects you

to understand. The Hub focuses on how to execute that understanding in 2026, when an AI assistant is going to be sitting next to your cursor whether or not anyone teaches you what to do with it.

2.7 A note on paradigms

This chapter has presented research through a social scientific lens: formulate hypotheses, test them against data, revise the story based on evidence. This is one way of knowing. It is the primary approach of this book. It is not the only legitimate one.

Interpretive researchers argue that human experience is too complex for hypothesis testing. They seek to understand how people make meaning, using methods like interviews, focus groups, and ethnography. **Critical** researchers argue that research should expose and challenge power structures, not just describe patterns. Both **paradigms** have produced foundational insights in communication and media studies.

Chapter 5 explores these paradigms in more depth. For now, the thing to recognize is that the social scientific approach is a choice. A productive and powerful one. But one of several. The ability to understand and evaluate research from all three paradigms is what separates a methodologically literate scholar from a technician who can run statistical tests but cannot think critically about what those tests mean.

The four pillars of graduate-level practice. The rest of this book introduces four practices central to graduate-level V2V work: (1) *a priori power analysis*, committing to a sample size before data collection; (2) *pre-registration*, committing to hypotheses, measures, and analysis plan before collection; (3) *two-coder reliability*, verifying that your codebook produces consistent results regardless of who applies it; and (4) *audit trail*, keeping a version-controlled record of every transformation from raw data to published figure. These are not optional add-ons. They are entailments of a post-positivist epistemological commitment. The phrase "garden of forking paths" belongs to Gelman and Loken (2013), who show how the many defensible choices open to an analyst inflate false-positive rates even without any conscious fishing. Munafo et al. (2017) locate a related danger in underpowered designs:

> "Low statistical power increases the likelihood of obtaining both false-positive and false-negative results, meaning that it offers no advantage if the purpose is to accumulate knowledge."
>
> Munafo et al. (2017, p. 2)

Their proposed remedy maps directly onto two of the four pillars above:

> "The strongest form of pre-registration involves both registering the study (with a commitment to make the results public) and closely pre-specifying the study design, primary outcome and analysis plan in advance of conducting the study or knowing the outcomes of the research."
>
> Munafo et al. (2017, p. 3)

For your own study, ask where in the V2V workflow the forking-paths risk appears first, and how pre-registration addresses it.

2.8 Looking ahead

Chapter 2 turns to the technical infrastructure that makes research reproducible. VSCode, R, Quarto, Git. Those tools can look intimidating from a distance. Their purpose is simple: they let you document every step of an analysis so others, including your future self, can verify and build on it. Chapter 2 also brings

you face to face with the dataset for the first time. Ten rows of a real collection from a single minute in November 2018, before any cleaning, before any code. The questions you ask of those ten rows are the questions the rest of the book will help you answer at scale.

2.9 References

Babbie, E. R. (2021). *The practice of social research* (15th ed.). Cengage Learning.

Barrett, L. F. (2017a). *How emotions are made: The secret life of the brain*. Houghton Mifflin Harcourt.

Barrett, L. F. (2017b). The theory of constructed emotion: An active inference account of interoception and categorization. *Social Cognitive and Affective Neuroscience*, *12*(1), 1-23. https://doi.org/10.1093/scan/nsw154

Herndon, T., Ash, M., & Pollin, R. (2014). Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff. *Cambridge Journal of Economics*, *38*(2), 257-279. https://doi.org/10.1093/cje/bet075

Katz, E., Blumler, J. G., & Gurevitch, M. (1973). Uses and gratifications research. *Public Opinion Quarterly*, *37*(4), 509-523. https://doi.org/10.1086/268109

Reinhart, C. M., & Rogoff, K. S. (2010). Growth in a time of debt. *American Economic Review*, *100*(2), 573-578. https://doi.org/10.1257/aer.100.2.573

Storr, W. (2019). *The science of storytelling: Why stories make us human, and how to tell them better*. William Collins.

2.9.1 Graduate readings

Gelman, A., & Loken, E. (2013). *The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time*. Department of Statistics, Columbia University.

Munafo, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., & Ioannidis, J. P. A. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, *1*, 0021. https://doi.org/10.1038/s41562-016-0021

3 Chapter 2: The Open Workspace

Listen in Dr. Leith's voice

The Excel error in Chapter 1 was one study. In 2015 the Open Science Collaboration, a group of 270 researchers from around the world, set out to redo 100 published psychology studies. They followed the original methods as closely as the published papers described, recruited similar participants, and ran the same analyses. The results, published in *Science*, became one of the most cited findings in the social sciences of the decade. Only 36 percent of the original results replicated at conventional thresholds of statistical significance, and among the studies that did replicate, the effect sizes were on average about half the magnitude originally reported (Open Science Collaboration, 2015).

The number was striking. The cause was more so. The Collaboration was not finding fraud. They were finding ordinary friction. Published methods sections did not contain enough detail to actually re-run the original analysis. Researchers had to email the original authors for clarifications. Some authors had moved. Some had lost the data. Some had run their analysis in software whose menus had since changed. The

replications were not blocked by malice. They were blocked by the simple fact that the original research had never been built to be retraceable.

This chapter is about building the infrastructure that makes research retraceable. Not in the abstract sense of "good science is reproducible." In the immediate, practical sense of being able to recover, six months from now, exactly what you did to produce a particular number. **Reproducibility** is that property: the analysis can be re-run, by you or by someone else, and it will produce the same result. The tools that make reproducibility possible are an editor, a language, a publishing format, and a version control system. None of those are interesting on their own. Together they form a small ecology that makes every move in a research project visible.

By the end of this chapter you will have installed that ecology on your own machine, run a single setup command to confirm everything works, and looked at the first ten rows of the Twitch dataset the rest of the book will draw on. The installs are tedious. The infrastructure they create is the difference between research you can defend and research you can only describe.

3.1 What gets lost when work is not traceable

The Open Science Collaboration's finding turned a corner that the social sciences had been turning for several years. By 2015 there was already a name for the underlying problem: the **replication crisis**. The crisis was not really a crisis in the sense of a sudden emergency. It was the slow recognition that published findings in psychology, and to varying degrees in economics, biomedicine, and the rest of empirical research, were not as durable as the field had assumed.

The mechanisms behind the unreliability were not exotic. Researchers had been running many analyses and reporting only the ones that worked. Studies with statistically significant findings had been more likely to get published than studies that found nothing, which inflated the visible base rate of "real" effects. Theoretical predictions had been adjusted after looking at the data, which is allowed in some traditions but tends to manufacture significance out of noise. The technical term for the underlying issue is **researcher degrees of freedom**: every analytical decision a researcher makes (which observations to keep, which controls to include, which test to run) is a chance to nudge a result toward "publishable." When the decisions are documented, other people can evaluate whether the choices were reasonable. When the decisions are not documented, they vanish into the methods section as if they had never been made.

The deeper culprit, and the one this chapter cares about, is the **point-and-click problem**. SPSS, Excel, JMP, and most legacy statistical software do their work through menus and dialog boxes. You click Analyze, then Compare Means, then One-Way ANOVA. The result appears. The click history vanishes. If you want to redo the analysis next month, you have to remember which menus you opened, in what order, with what options selected. If a collaborator wants to verify your work, you have to write a description of every click, in prose. Almost no one does this well, and almost no one wants to.

Code does not have this problem. A script is a description of every step in the analysis, written in a language that the computer can re-run. The Reinhart-Rogoff Excel error from Chapter 1 was caught because the spreadsheet, eventually shared, contained the broken formula on the page. The error would have been caught much faster if the original analysis had been written as code, where every decision is on the page from the start. **Code as documentation** is the central insight: when you write your analysis as code, the code is the methods section.

This is the conceptual move that drives the rest of the chapter. The tools you are about to install are not chosen because they are trendy or because they happen to be what your professor prefers. They are

chosen because each one produces a plain-text record of what you did, which is the precondition for any of your work being checkable later.

3.2 A small ecology of tools

Four pieces of infrastructure cover the work you will do in this course. Each is free, open-source or available at no cost to students, and runs on all major operating systems.

VSCode is the editor. Made by Microsoft, free, and the dominant code editor in industry today. It is where you will write everything: prose, code, configuration files, notes. You can open multiple files in tabs, run code in a terminal, see your Git history, and edit Quarto documents with live previews of the rendered output. It is where most working data analysts already are, which means the muscle memory you build in this course transfers directly into professional contexts.

R is the language. R is the statistical computing language that dominates social science research, and especially mass communication, sociology, political science, and quantitative education research. Python is the other major option. Both are excellent. R is taught here because its packages for the things this book teaches (content analysis, inter-coder reliability, regression, visualization) are deeper and more mature than Python's, and because R's publishing integration with Quarto is tighter. The skills you build in R will transfer to Python if you ever need them; the reverse is also true.

Quarto is the publishing system. Quarto, developed by the team at Posit, takes a single source document (a `.qmd` file) and produces a finished output: an HTML web page, a PDF, a Word document, a slide deck, a book. The source document mixes prose with executable code. When Quarto renders the file, the code runs and the results appear in place. This is what makes a research report reproducible at the document level. The numbers in your discussion section, the tables in your results, the charts in your figures: all come from code that runs every time the document is rendered.

Git and GitHub are the version control system. Git tracks every change to every file in a project. GitHub is one of several services that hosts Git repositories on the web. Together they give you a permanent audit trail of your research work, the ability to roll back to any previous state, and a public-by-default platform for sharing your project with collaborators and the world. Students get the GitHub Student Developer Pack, which includes paid features at no cost. The V2V Hub walks you through claiming it.

Each tool is independently useful. The point of teaching them together is that the combination produces a research workflow where every move you make is documented, every output is reproducible, and every change is recoverable. This is not the standard workflow in most undergraduate methods courses. It is the standard workflow in research-active labs and in industry data science teams, and getting it into your hands now is one of the most durable skills this book can give you.

The audit trail begins at setup. It is tempting to read the setup steps as mere tool choices: install R, create a GitHub repo, connect it. Each is better understood as a *methodological commitment*. Configuring the workspace is the first act of the study, and the decisions made here are what determine whether the analysis can be reconstructed months later.

3.3 Installing the stack

The click-by-click instructions for installing each tool, with screenshots for both Windows and macOS, live on the V2V Hub. The Hub is the place to go when you need moment-to-moment guidance. This section gives you the conceptual map: what gets installed, in what order, and why.

Install in this order:

1. **R.** Visit the Comprehensive R Archive Network (CRAN) at cran.r-project.org and download R for your operating system. Run the installer with default settings. R has no graphical interface of its own to speak of; you install it as a backend that other tools will call.
2. **VSCode.** Visit code.visualstudio.com and download VSCode for your operating system. Run the installer. On first launch, VSCode will offer to sync settings with a Microsoft or GitHub account. You can decline and set up the account later.
3. **The R extension for VSCode.** Open VSCode, click the extensions icon in the left sidebar, search for "R," and install the extension by REditorSupport. This is what connects VSCode to the R installation from step 1.
4. **Quarto.** Visit quarto.org and download Quarto for your operating system. Install with defaults.
5. **The Quarto extension for VSCode.** Back in VSCode's extensions panel, search for "Quarto" and install the extension. You now have live preview and syntax support for `.qmd` files.
6. **Git.** Visit git-scm.com and download Git for your operating system. The default installer options are fine. On macOS, Git is also available through the Xcode command-line tools, which VSCode will offer to install for you the first time you try a Git operation.
7. **A GitHub account.** Visit github.com and create an account if you do not already have one. Use a professional username; you will eventually link work to it that potential employers will see. Verify the email you sign up with. The Student Developer Pack is claimed at education.github.com/pack, and the V2V Hub has the walkthrough.
8. **The `v2v` package.** The Hub has the install command for the current package version. The package ships the Twitch dataset fixtures and a small set of helper functions used throughout the rest of the book.

When everything is installed, open VSCode and create a new file named `setup-check.qmd`. Give it a title in the YAML front matter at the top:

```
---
title: "Setup check"
---
```

Then add an R code chunk that calls the v2v setup validator:

````
```{r}
library(v2v)
v2v::setup()
```
````

Click Quarto's Preview button, or run `quarto preview setup-check.qmd` in the terminal. The `v2v::setup()` function checks that R, Quarto, Git, and the v2v package are all present at acceptable versions. It prints "V2V setup complete" when everything is in place and a diagnostic message when something is missing. If you see anything other than the success message, the Hub's Installing the v2v package guide addresses the common failure modes.

If `library(v2v)` itself fails, the package is not installed yet. It is not on CRAN, so `install.packages("v2v")` will not find it. Install it from GitHub, noting that the repository is named `v2v-r` while the package it provides is named `v2v`:

```
install.packages("remotes")
remotes::install_github("AURA-Lab-SIUE/v2v-r")
```

Git as epistemological infrastructure. Every `git commit` is a time-stamped record of your analysis at a specific moment. Your entire analytical history is reproducible: another researcher can check out any commit and see exactly what your code did at that point. This is not a convenience feature. It is a standard of evidence. Wilson et al. (2017) place change-tracking at the center of reproducible practice:

> "Keeping track of changes that you or your collaborators make to data and software is a critical part of research. Being able to reference or retrieve a specific version of the entire project aids in reproducibility for you leading up to publication, when responding to reviewer comments, and when providing supporting information for reviewers, editors, and readers."
>
> Wilson et al. (2017, p. 12)

Graduate work in V2V treats `git log` as a portion of the methods section. The first commit, before any analysis code is written, should include your data, your codebook, and a `README` that describes the study. For your own study, run `git log --oneline` on a project you completed before learning version control, and ask what a reviewer would want to see that you can no longer reconstruct.

3.4 First contact with the data

The Twitch dataset has been the recurring example since Chapter 1. This is the moment you actually look at some of it. The table below shows the first ten rows of `stream_log`, the table that tracks who was streaming what game to how many people at each snapshot moment during the collection window.

channel	title	game	viewers	date
sodapoppin	`Hello truckers\n44835158804929_2115841973`	Marble It Up!	27,934	1542578200214
xqcow	RANK 1 GAMER ~ NO PLAN (help me)	Just Chatting	19,381	1542578200240
forsen	@forsen, The prince of miracoli ! Giving away artifact keys all day!	Artifact	15,301	1542578200273
giantwaffle	No Smoking in Chat | @GiantWaffle	Rocket League	4,691	1542578200423
sodapoppin	`Hello truckers\n44835158804929_2115841973`	Marble It Up!	28,071	1542578261145

channel	title	game	view-ers	date
xqcow	RANK 1 GAMER ~ NO PLAN (help me)	Just Chat-ting	19,231	1542578261186
forsen	@forsen, The prince of miracoli ! Giving away artifact keys all day!	Artifact	15,181	1542578261211
giant-waffle	No Smoking in Chat | @GiantWaffle	Rocket League	4,701	1542578261409
so-dapop-pin	`Hello truckers\n44835158804929_2115841973`	Marble It Up!	28,203	1542578322302
xqcow	RANK 1 GAMER ~ NO PLAN (help me)	Just Chat-ting	19,231	1542578322337

Take a moment to look at it before reading on. Resist the temptation to do anything with it. Just notice.

A few things should jump out.

The data moves. This is a time series, not a snapshot. Four channels appear repeatedly across the ten rows. Sodapoppin shows up three times. Sodapoppin's viewer count climbs across those three appearances, from 27,934 to 28,076 to 28,203. That is 269 viewers gained in 122 seconds, which is the kind of texture this dataset captures: an audience expanding in close to real time. xQc lost 150 viewers between the first and second snapshot and then held steady. Forsen lost 120. Giantwaffle gained 3. Each of these numbers is a real moment in a stream's life, the moment a clip went viral or a host raid arrived or a viewer's bandwidth dropped. A great many of the most interesting research questions in this book come from interrogating exactly that.

The timestamps look like nonsense. The date column is a thirteen-digit integer that does not visibly resemble a date. The first row shows 1542578200214. The second, taken twenty-six milliseconds later, shows 1542578200240. The difference between the first sodapoppin snapshot and the second is 60,931, which is roughly one minute. This is the storage format used in most production databases, called **Unix epoch time**: the number of milliseconds since midnight on January 1, 1970. It is unreadable as a calendar date, and that is one of the technical problems Chapter 11 will solve. For now, all you need to know is that the dates are in there, and getting them into human-readable form is a wrangling step, not a research question.

There is a hidden newline in one of the titles. Sodapoppin's stream title appears in the table as `Hello truckers\n44835158804929_2115841973`. The `\n` is the conventional way to write a newline character in plain text. In the actual dataset, that position contains a real line break: the streamer's software, at some point, inserted a literal newline into the title field and the data preserved it. Real-world data is messy in exactly this way. Fields contain characters they were not designed for, software pipelines produce artifacts that nobody intended, and a researcher's first job is to look at the data and notice what is actually there. A stray newline in a single title is harmless. The habit of noticing things like that is not.

Titles are themselves data. xQc's title says "RANK 1 GAMER ~ NO PLAN (help me)," which is a joke. Forsen is giving away keys for a card game called Artifact. Giantwaffle's title performs etiquette ("No

Smoking in Chat"). These are content analysis opportunities sitting in a column you might otherwise treat as boring metadata. The book returns to title content in Chapter 12.

Look at the table once more, with these observations in mind. This is what the data looks like before any analysis has been done. You are starting from here.

3.5 The v2v package

The `v2v` package, installed in step 8 of the install sequence, is the course companion. It ships three things.

The first is two **dataset fixtures.** `twitch_chat_sample` is a stratified sample of roughly 35,000 chat messages drawn from the 50-channel working corpus. `twitch_streams_sample` is the full set of stream snapshots for those same 50 channels, approximately 32,000 rows. Once the package is loaded with `library(v2v)`, either fixture loads with `data(twitch_chat_sample, package = "v2v")` or `data(twitch_streams_sample, package = "v2v")`. The full population of 1,690 joinable channels is documented in the package vignette; you will only need the 50-channel subset for course work.

The second is a small set of **helper functions** that wrap common course operations: loading the data, generating codebook skeletons, computing inter-coder reliability metrics, producing diagnostic plots. The helpers are not magic. They exist so that the conceptual move of an exercise is not buried under fifty lines of setup code. You can always do the same work without the helpers; the helpers just let you focus on the methodology while you are learning it.

The third is the **setup validator** you ran a few pages ago: `v2v::setup()`. Run it again any time you suspect an installation issue or have updated R or Quarto and want to confirm the stack still works.

The full documentation for the package is available with `?v2v` in R or at the package website, which is linked from the V2V Hub. Treat the documentation as reference material rather than required reading.

The renv.lock as a methods citation. The `renv.lock` file records the exact version of every R package your analysis depends on. When you publish a white paper, the `renv.lock` is the citation for your computational environment. Without it, a reader attempting replication may get different results because a package updated between your run and theirs. Wilson et al. (2017), whose "good enough" practices this aside builds on, extend the backup discipline to the software a project sits on:

> "Back up (almost) everything created by a human being as soon as it is created. This includes scripts and programs of all kinds, software packages that your project depends on, and documentation."
>
> Wilson et al. (2017, p. 12)

Run `renv::init()` before your first line of analysis code and commit the resulting lock file before you collect your first data point. For your own study, weigh the practical cost of skipping `renv::init()` in a project you intend to publish, in the terms Wilson et al. use to separate "good enough" practices from "best" ones.

3.6 Looking ahead

Chapter 3 takes up the ethical responsibilities that come with collecting and analyzing data about other people. The Twitch dataset is a clean case for thinking through public-by-default information, pseudonymity, and the limits of what informed consent means when it was never asked. Most of the data you

will encounter as a working social scientist will not be as clean, and the framework Chapter 3 builds is meant to handle the harder cases.

3.7 References

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716. https://doi.org/10.1126/science.aac4716

3.7.1 Graduate readings

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLOS Computational Biology*, *13*(6), e1005510. https://doi.org/10.1371/journal.pcbi.1005510

4 Chapter 3: Knowing and Knowing Well

Listen in Dr. Leith's voice

In June 2014, researchers at Facebook and Cornell University published a study in the *Proceedings of the National Academy of Sciences* that became one of the most debated experiments in the history of social science. For one week in January 2012, Facebook had manipulated the News Feeds of nearly 700,000 users without their knowledge. Some users saw fewer positive posts from friends; others saw fewer negative posts. The researchers then measured whether the manipulation affected what users themselves posted. It did. Users exposed to fewer positive posts produced slightly more negative content, and vice versa. The study, led by Adam Kramer, Jamie Guillory, and Jeffrey Hancock, offered evidence of "massive-scale emotional contagion through social networks" (Kramer, Guillory, & Hancock, 2014).

The finding was interesting. The reaction was explosive.

Critics pointed out that nearly 700,000 people had been enrolled in a psychological experiment without informed consent. No one had been told their emotional environment was being manipulated. No one had the opportunity to decline. Facebook argued that its terms of service, which users agree to upon creating an account, authorized the use of data for "internal operations, including troubleshooting, data analysis, testing, and research." The researchers argued that the manipulation was minor, the effect size was tiny, and the study produced valuable scientific knowledge.

The scientific community was divided. Some defended the study as comparable to A/B testing that tech companies perform routinely. Others argued that emotional manipulation, however slight, crosses a line that terms-of-service agreements cannot legitimize. PNAS, the journal that published the study, took the unusual step of appending an "editorial expression of concern" to the article, acknowledging questions about the consent process.

The Facebook study crystallized every major tension in modern research ethics: the boundary between research and product development, the adequacy of passive consent, the obligations of researchers to participants who do not know they are participants, and the question of whether "minimal risk" justifies bypassing standard protections. The case does not have a clean answer. Reasonable people disagree about whether the study was ethical. What is not debatable is that the questions it raised matter, and that every researcher must develop a framework for navigating them.

4.1 Why ethics is not a formality

Students sometimes approach research ethics as a bureaucratic requirement: a form to fill out, a training module to complete, a hurdle between them and the "real" work of data collection. This is understandable.

Ethics review can feel procedural, especially when a study involves publicly available chat messages rather than vulnerable human populations.

But ethics is not a procedure. It is a disposition. It shapes how you think about your relationship to the people whose data, content, or behavior you study. It governs how honestly you report what you find. And it reflects the fact that the history of research includes episodes of genuine harm, episodes serious enough to justify every protection that now exists.

Paradigmatic positioning. Every study operates from a paradigm, a set of assumptions about the nature of knowledge and how it is acquired. Post-positivist communication researchers assume an objective social reality that is incompletely knowable, that measurement error is irreducible but manageable, and that statistical inference allows probabilistic claims about that reality. If that description fits your study, the four methodological commitments (power, pre-registration, reliability, and an audit trail) are not optional; they are logical entailments of the paradigm. If you are working from an interpretive or critical paradigm, V2V's quantitative tools occupy a different role in your methodology, and different obligations apply. Where would you locate your current research question on the ontology–epistemology axis Crotty (1998) describes, and what methodological implications follow from that location?

4.2 How we got here

Modern research ethics regulations exist because researchers harmed people. Not hypothetically. Not in edge cases. Systematically and with institutional support. Understanding this history matters not to assign guilt to the current generation, but to appreciate why the protections exist and what they are designed to prevent.

4.2.1 The Tuskegee syphilis study (1932 to 1972)

For forty years, the United States Public Health Service studied the progression of untreated syphilis in 399 Black men in Macon County, Alabama. The men were told they were receiving free treatment for "bad blood." They were not. Even after penicillin became the standard treatment for syphilis in the 1940s, the researchers withheld it. Twenty-eight men died directly of syphilis, 100 died of related complications, 40 wives were infected, and 19 children were born with congenital syphilis.

The study was not conducted by rogue scientists. It was funded by the federal government, staffed by credentialed researchers, and published in peer-reviewed journals for decades without significant objection from the scientific community. Its exposure in 1972, by Associated Press journalist Jean Heller, led directly to the regulatory framework the United States uses today.

4.2.2 The Milgram obedience experiments (1961)

Stanley Milgram's experiments at Yale examined whether ordinary people would administer what they believed were painful electric shocks to a stranger simply because an authority figure instructed them to do so. Most did. The experiments produced foundational insights about authority and obedience, but they did so by subjecting participants to extreme psychological distress. Participants believed they were causing real harm to another person. Many exhibited signs of acute anxiety, and some experienced lasting psychological effects.

The ethical question was not whether the findings were important. They were. The question was whether the knowledge justified the deception and distress inflicted on participants.

4.2.3 Humphreys and the tearoom trade (1970)

Sociologist Laud Humphreys studied anonymous sexual encounters between men in public restrooms. He recorded participants' license plate numbers without their knowledge, traced their home addresses through police records, and later visited their homes in disguise to conduct "health surveys," gathering personal information the men had no idea was connected to the restroom observations.

Humphreys argued that his research revealed important truths about a hidden population. Critics argued that he violated his subjects' privacy in ways that could have destroyed their lives, marriages, and careers if the data had been exposed.

4.2.4 The common thread

Each case involved researchers who believed their work served a greater good. Each involved participants who were harmed, deceived, or exploited. And each contributed to the recognition that good intentions are not sufficient protection. Researchers need external accountability, formal principles, and institutional oversight.

4.3 The Belmont Report

In 1979, the National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research published *The Belmont Report*, which established three foundational principles for ethical research (National Commission, 1979). These principles now govern virtually all research involving human subjects in the United States.

4.3.1 Respect for persons

People must be treated as autonomous agents capable of making their own decisions. This means **informed consent**: participants must understand what the research involves, what risks it carries, and that they can withdraw at any time without penalty. It also means **protection of vulnerable populations**. People with diminished autonomy (children, prisoners, individuals with cognitive impairments) require additional protections because their ability to provide truly voluntary consent is compromised.

Applied to communication research, if you survey listeners about their emotional responses to a podcast, participants must know they are in a study, understand what they will be asked to do, and be free to stop at any time. If you analyze public social media posts, the question becomes more complex. Did users "consent" to being studied when they posted publicly? The chapter returns to that question shortly.

4.3.2 Beneficence

Researchers must maximize benefits and minimize harms. This involves two obligations. The first is **do no harm**: the research should not cause physical, psychological, social, or economic injury to participants. The second is **maximize the ratio of benefits to risks**: the knowledge gained must justify any risks imposed on participants.

Applied to communication research, most content analysis carries minimal risk because the analyst is interacting with texts, not people. Survey and experimental research can involve psychological discomfort (exposing participants to upsetting media content), social risk (collecting sensitive information that could embarrass participants if leaked), or economic risk (taking participants' time without adequate compensation).

4.3.3 Justice

The benefits and burdens of research must be distributed fairly. No group should bear a disproportionate share of research risks while another group receives the benefits.

The Tuskegee study violated justice because it imposed all the risks on a vulnerable, marginalized population while the benefits, in the form of medical knowledge, accrued to the broader society. In communication research, justice concerns arise when studies about marginalized communities are designed and conducted without input from those communities, or when research on "deviant" media behavior targets specific demographic groups while treating majority behavior as the unmarked norm.

4.4 Institutional review

The Belmont Report's principles are operationalized through **Institutional Review Boards** (IRBs), committees at universities and research institutions that review proposed studies for ethical compliance before data collection begins.

IRBs evaluate whether a study minimizes risks to participants, ensures risks are reasonable relative to anticipated benefits, selects participants equitably, obtains and documents informed consent, monitors data collection for safety, and protects participant privacy and confidentiality.

Not all studies require the same level of scrutiny. Federal regulation distinguishes three tiers.

Exempt review applies to research that poses minimal risk and falls into specific categories defined by federal regulation. Most content analysis of publicly available materials (newspaper articles, song lyrics, television broadcasts, public social media posts) qualifies for exempt status because no human subjects are directly involved.

Expedited review applies to research that involves no more than minimal risk but does not qualify for full exemption. Many surveys and interview studies fall here, particularly those that collect non-sensitive data from adult participants.

Full board review applies to research involving more than minimal risk, vulnerable populations, or deception. Experimental studies that manipulate participants' emotional states, research with children or prisoners, and studies involving sensitive topics (substance use, sexual behavior, criminal activity) typically require full review.

The IRB makes the tier determination, not the researcher. Researchers submit; the board decides. Submitting a proposal you suspect should be exempt does not waste anyone's time, because the determination is the board's to make.

4.5 Walking the Twitch dataset through the framework

The Twitch corpus used in this book is a clean case for ethics review. The cleanness is itself instructive. It shows what a low-friction dataset looks like, and it sets up the contrast cases that show what makes other datasets high-friction.

The Twitch chat log was collected from the public IRC interface that Twitch maintains for chat data. The stream log was collected from the public Twitch API. Both interfaces are public by default; any internet user can connect and read the same data without authentication. The dataset contains no IP addresses, no email addresses, no Twitch internal user IDs, no payment or subscription information, and no whisper (private message) content. Sender usernames are present, but they are **pseudonymous**: a

Twitch username is a chosen handle, not a legal name, and many viewers use names that do not connect to their offline identity.

Walking this through the Belmont principles:

- **Respect for persons.** Participants were not enrolled in a study and did not give informed consent. They posted publicly to a platform whose terms of service describe chat as a public broadcast. The argument for ethical use rests on the public-broadcast nature of the data, not on consent. The Association of Internet Researchers (AoIR) has long argued that "public" does not automatically mean "fair game for research" (Markham & Buchanan, 2012), and the researcher's obligation is to handle the data in ways that respect the implicit expectations of a public chat. In practice, this means not aggregating data in ways that would re-identify individual users, not republishing entire chat logs verbatim, and not amplifying messages whose senders might have a stronger expectation of obscurity than of broadcast.
- **Beneficence.** The risk to senders is minimal because they are pseudonymous, the data are already public, and no analytical move in this book aggregates toward offline identity. The benefit, in the form of scholarship on platform behavior, is reasonable.
- **Justice.** The dataset is not targeting a marginalized population. It samples broadly across Twitch channels, including non-gaming categories. No group is bearing disproportionate research burden.

Most IRBs classify a study using this kind of data as **exempt** because no human subjects are directly involved. Three published manuscripts have used adjacent slices of this collection under exempt-by-design determinations.

The cleanness is the point. Now consider three contrast cases that change the determination.

Same research question, but on Twitch whispers. A whisper is a private message between two users. Even with technical access to whisper logs, the senders had a reasonable expectation of privacy. An IRB would almost certainly require expedited or full review, depending on the sensitivity of the content. The shift from public chat to private message changes the data's ethical status entirely.

Same research question, but on a private Discord server with the same sender population. A private Discord is a closed group with controlled membership. Senders have a stronger expectation of privacy than they do on a public Twitch chat. Most IRBs would require expedited or full review for research on private Discord communication, even when the senders are the same people who chat publicly on Twitch.

Same research question, but on an internal moderation log that included platform-side flags on individual users. Now the data includes platform-internal classifications about people. The risk of harm to identified individuals jumps significantly. Full board review is likely.

The narrow slice of "publicly broadcast chat on a platform whose terms describe it as public, with no platform-internal flags attached" is what makes the dataset for this course exempt-by-design. Most data you encounter in your career will not be that narrow.

4.6 Informed consent in practice

The principle of respect for persons is operationalized primarily through informed consent, the process by which participants voluntarily agree to participate in research after being told what the study involves.

Valid informed consent requires that participants receive a description of the research and their role in it; a description of risks and benefits, including any foreseeable discomfort; a statement that participation

is voluntary and that withdrawal carries no penalty; contact information for the researcher and the IRB; and an explanation of how data will be stored and confidentiality protected.

Consent gets complicated in three recurring ways. **Deception**, where some studies require that participants not know the true purpose of the research, is sometimes justified but always paired with mandatory **debriefing**: researchers must explain the true purpose after participation and give participants the option to withdraw their data. **Passive consent**, like the Facebook study's reliance on terms-of-service agreement, is generally considered insufficient for research purposes; ethicists prefer **active consent**, which requires an affirmative act (signing a form, clicking an explicit agreement to participate in research, verbally confirming participation). **Secondary data analysis**, where you analyze data someone else collected, raises questions about whether the original consent covered your specific use; it is generally acceptable when the data are de-identified, but it is one of the places where contemporary research ethics is still actively evolving.

4.7 Ethics in specific methods

Each major research method carries its own ethical profile.

Content analysis of published media is among the lowest-risk research methods. The analyst is interacting with texts, not with people. Ethical considerations still apply, particularly around how you categorize and interpret content (a content analysis that codes hip-hop lyrics as "violent" without accounting for genre conventions can reinforce harmful stereotypes), cherry-picking examples that confirm a hypothesis while ignoring contradictory cases, and copyright concerns when reproducing extensive content in a final report.

Survey research involves direct interaction with participants, which raises the ethical stakes. Voluntary participation must be genuine; offering course credit is acceptable only if an alternative assignment of equal value is available. Sensitive questions about substance use, sexual behavior, mental health, or illegal activity require careful handling. Data must be stored securely and de-identified.

Experimental research that manipulates participants' experiences carries the highest ethical burden. Manipulation effects must be considered carefully: if you expose participants to distressing media content, you must consider whether the exposure could cause lasting harm. Deception requires debriefing. Control group ethics, particularly in interventional research, can raise fairness questions when a potentially beneficial treatment is withheld from one group.

Digital and social media research occupies an evolving ethical landscape. The "public" problem (data is publicly accessible but the poster may not have anticipated being studied) is the central issue. Aggregation risk (combining individually harmless data points to create identifiable profiles) compounds it. Platform terms of service may prohibit scraping even of public data, which raises legal questions alongside ethical ones. And vulnerable populations online (minors, people in crisis, members of stigmatized groups) may be especially harmed by research exposure, even when their posts are public.

4.8 Ethical writing and reporting

Ethics does not end when data collection is over. How you analyze and report findings carries its own ethical obligations.

Fabrication is inventing data that do not exist. **Falsification** is manipulating data or results to change the outcome. Both are forms of scientific fraud, and both are career-ending if discovered. They are also

more common than the research community likes to admit; surveys of researchers consistently find that a small but meaningful percentage acknowledge engaging in questionable practices (Babbie, 2021).

Selective reporting is a subtler form of dishonesty: running many analyses but reporting only those that produced significant results, or testing multiple hypotheses but presenting only the ones that worked. This practice, sometimes called the file drawer problem, distorts the literature by creating a publication record that systematically overstates the strength of effects. Chapter 2's discussion of researcher degrees of freedom returns here. Selective reporting is the active version of the passive friction the Open Science Collaboration documented in 2015.

HARKing, an acronym for Hypothesizing After the Results are Known, occurs when researchers examine their data, identify patterns, and then write the paper as though they had predicted those patterns all along. This transforms exploratory analysis (which is legitimate) into confirmatory analysis (which requires pre-specification). A "prediction" that was actually a post-hoc observation carries none of the epistemic weight of a genuine a priori hypothesis.

Why pre-registration is a paradigmatic commitment. Pre-registration is often framed as a statistical technique to control false-positive rates. That framing is accurate but incomplete. At the paradigmatic level, pre-registration embodies the hypothetico-deductive model: you derive predictions from theory *before* examining data, then submit those predictions to empirical test. This is the canonical post-positivist epistemological move. Nosek and colleagues (2018) define the commitment in precisely these terms:

> "Preregistration of an analysis plan is committing to analytic steps without advance knowledge of the research outcomes."
>
> Nosek et al. (2018, p. 2601)

The OSF pre-registration form operationalizes that commitment: it requires you to state theory-derived hypotheses, specify measures, and describe the analysis before any data is collected or examined. Is pre-registration compatible with grounded theory or thematic analysis? Make the case for or against, using epistemological rather than practical reasoning.

Even without outright fabrication, researchers face constant temptation to overstate their findings. "Our results suggest" becomes "our results demonstrate." A small effect size gets buried while a large p-value gets highlighted. Limitations are mentioned but minimized. The discussion section tells a cleaner story than the data support.

Honest interpretation means stating what you found, acknowledging what you did not find, and being transparent about the boundaries of your claims. It means writing a limitations section that genuinely grapples with weaknesses rather than performing humility while defending every decision. It means distinguishing between what the data show and what you wish they showed. This is harder than it sounds. It is the core ethical obligation of every researcher.

Epistemological humility and its constraints. Claiming that reality is incompletely knowable does not license sloppy measurement. It licenses *uncertainty quantification*: confidence intervals, power statistics, effect size estimates with precision reported. The four methodological commitments are the methods-level expression of epistemological humility.

4.9 Looking ahead

Chapter 4 opens Part II of the book, where the work shifts from foundations to design. The first move is a literature review: identifying what is already known about a topic and where the gaps in the field are. Literature review is the rising action of a research story. Before you confront your research question, you find out what others have already tried.

4.10 References

Babbie, E. R. (2021). *The practice of social research* (15th ed.). Cengage Learning.

Kramer, A. D. I., Guillory, J. E., & Hancock, J. T. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, *111*(24), 8788-8790. https://doi.org/10.1073/pnas.1320040111

Markham, A., & Buchanan, E. (2012). *Ethical decision-making and internet research: Recommendations from the AoIR Ethics Working Committee (Version 2.0)*. Association of Internet Researchers.

National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. (1979). *The Belmont Report: Ethical principles and guidelines for the protection of human subjects of research*. U.S. Department of Health, Education, and Welfare.

4.10.1 Graduate readings

Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*. SAGE Publications.

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

Part II: Planning

5 Chapter 4: Intelligence Gathering

Listen in Dr. Leith's voice

There is a moment in every research project when you realize the literature is deeper than you thought. You find an article that seems directly relevant to your question. It cites twelve other articles. You track down three of those, and each cites fifteen more. Within an hour you have accumulated thirty sources you should probably read, and the prospect of making sense of them all feels paralyzing.

The instinct at this point is to retreat: narrow the scope, focus on a handful of recent articles, and hope that is sufficient. The instinct is understandable and it is wrong. It misunderstands what a **literature review** actually does. A literature review is not a checklist where you demonstrate you have read enough articles. It is a map of intellectual territory, and you cannot map territory by looking at three landmarks and declaring the job done.

The challenge is not just finding sources but finding them systematically, documenting the search so that someone else could repeat it, and knowing when you have read enough to claim with confidence that you understand the landscape. This chapter introduces the **archivist mindset**: the discipline of searching, organizing, and synthesizing scholarship as active cartography rather than passive consumption.

These skills are not specific to any one topic. Whether you are mapping the literature on livestream viewer motivation, news framing and public opinion, health communication campaigns, or algorithmic bias in social media, the process is identical. Search, organize, synthesize, identify the gap.

5.1 The conversation metaphor

Imagine walking into a room where a complex debate has been unfolding for years. The participants are knowledgeable, passionate, and deeply invested. They have built on each other's arguments, challenged assumptions, introduced new evidence, and staked out positions. You have something you want to contribute, an observation you think matters.

If you simply blurt it out without first listening to what has already been said, your contribution will likely be ignored or dismissed as naive. You might be repeating a point made a decade ago. You might be unaware that someone already tested your idea and found it wanting. You might be using terminology in ways that signal you have not done the intellectual work of understanding the debate's history.

To contribute meaningfully, you must first listen. You must understand who the key voices are, what the major points of contention are, where consensus exists, and what questions remain unresolved.

This is what a literature review does. It is the disciplined act of listening to the scholarly conversation before adding your voice to it. And it is not optional. No research project exists in a vacuum. Every study is part of an ongoing dialogue that has been unfolding in journals, books, and conference presentations for years, sometimes decades. The literature review is how you earn the right to ask your question.

5.2 What a literature review accomplishes

A strong literature review does several things at once.

It situates your work. The review demonstrates that you are aware of the broader context and not working in isolation. By connecting your research to established theories and previous findings, you show how your study fits into, and extends, the collective knowledge of the field. This is partly defensive: reviewers and instructors want to know you have done your homework. It is also generative, because seeing how your question connects to existing work often reveals angles you had not considered.

It identifies a gap. Perhaps the most critical function of the review is justifying why your study is necessary, which requires identifying a gap in existing scholarship. By demonstrating a gap, the review answers the "so what?" question. It persuades the reader that your study is not redundant.

It prevents reinventing the wheel. A thorough review ensures you are not proposing a study that has already been done. It is frustrating to develop what feels like an original idea only to discover it was the subject of a dissertation five years ago.

It provides methodological guidance. Beyond findings, the literature offers a wealth of methodological knowledge: validated measurement tools, successful sampling strategies, analytical techniques. You can also learn from others' limitations. Content analysis handbooks such as Krippendorff (2018) and Neuendorf (2017) are particularly valuable here, because they compile decades of best practice for coding, sampling, and reliability.

It refines your research question. Engaging with the literature sharpens a broad interest into a precise, researchable question. You might start with a vague curiosity about "Twitch chat" and, through reading, discover a specific debate about whether viewer engagement is driven by the content on the screen or the social environment around it. The literature provides the concepts, terminology, and frameworks that let you ask a question that is not just interesting but empirically tractable.

The quantitative function of the literature review. Beyond surveying what is known to identify a gap, the literature review has a further function: *effect size extraction*. Before you can calculate the sample size you need (Chapter 6), you need a defensible estimate of the effect size you expect to find. The most defensible source is prior published research. When you read a study using a similar design, extract its effect size (Cohen's d, Cramer's V, correlation r) and note the sample size the authors used. These numbers become the inputs for your power analysis. A literature review that cannot supply effect size estimates has not finished its job. Cohen (1992), whose conventions for small, medium, and large effects the `pwr` package still encodes, identified where this step reliably stalls (he abbreviates effect size as ES):

> "Researchers find specifying the ES the most difficult part of power analysis."
>
> Cohen (1992, p. 156)

The literature review is how you make that difficulty tractable, by borrowing an estimate from studies that have already done the measuring. To practice, choose a published V2V-style study, extract its primary effect size, determine whether the authors provided a power analysis, and use the `pwr` package to calculate the sample size required to detect that effect at 80% power; then work out what you would do when no prior studies report an effect size for your specific research question, describing two defensible strategies for justifying a sample size in that case.

5.3 The search process

Searching for literature happens in phases. Early on you are exploring, trying to get a sense of the terrain. Later you are systematizing, making sure you have found the relevant work and can defend your search strategy to a skeptic.

5.3.1 Phase one: exploratory searching

Start broad. You are trying to answer basic questions. What terms do scholars use to discuss this topic? Who are the key researchers? What journals publish this kind of work? What theories are commonly invoked?

Google Scholar is the place to start. It is less comprehensive than specialized databases, but it is fast, free, and good for getting oriented. Wikipedia, despite not being a citable source, is useful for finding terminology and key concepts; scroll to the references section of a relevant article and follow the citations to actual scholarship. And the reference list of any good article you find is itself a map: the sources an author cites repeatedly are likely foundational works.

At this stage, do not worry about perfect organization. Collect promising sources in **Zotero**, the open-source reference manager this course uses, and tag them liberally.

5.3.2 Phase two: systematic searching

Once you have a sense of the landscape, shift to systematic searching. The goal now is comprehensiveness.

Use academic databases. Communication & Mass Media Complete is the primary database for communication research. PsycINFO is essential if your question touches psychology: emotion, persuasion, cognition, attitudes. Web of Science and Scopus are multidisciplinary databases useful for citation tracking.

Inside those databases, use **Boolean operators** to build precise search strings. AND narrows results, because both terms must appear. OR expands them, because either term can appear. NOT excludes unwanted results. An asterisk is a wildcard that captures variations: `stream*` finds stream, streams, streaming, and streamer.

A search string for livestreaming viewer research might look like this:

```
(livestream* OR "live streaming" OR Twitch) AND
(motivation* OR gratification* OR engagement) AND
(viewer* OR audience* OR chat)
```

Document every search in a plain-text search log: the date, the database, the exact search string, the limiters you applied, the number of results, and how many you kept. The log serves two purposes. It lets you defend your search strategy later, and it helps you refine searches by showing what worked and what did not.

Systematic search over ad hoc search. Ad hoc searches (“I Googled it and found relevant papers”) cannot support power analysis inputs, because you do not know whether the studies you found are representative of the literature. A systematic search uses predefined databases (Communication Abstracts, Web of Science, PsycINFO), predefined search strings, and documented inclusion/exclusion criteria. The search process is itself a methods disclosure: you report it alongside the analysis so that a future researcher can reproduce and update your review.

5.3.3 Phase three: citation chaining

Once you have identified a few highly relevant articles, the keystone articles, use **citation chaining** to map the scholarly network.

Backward chaining means looking at the references cited in a keystone article. These are the foundational works the author built on. **Forward chaining** means using Google Scholar’s “Cited by” feature to

see which newer articles have cited the keystone. This reveals how the conversation has evolved since the keystone was published.

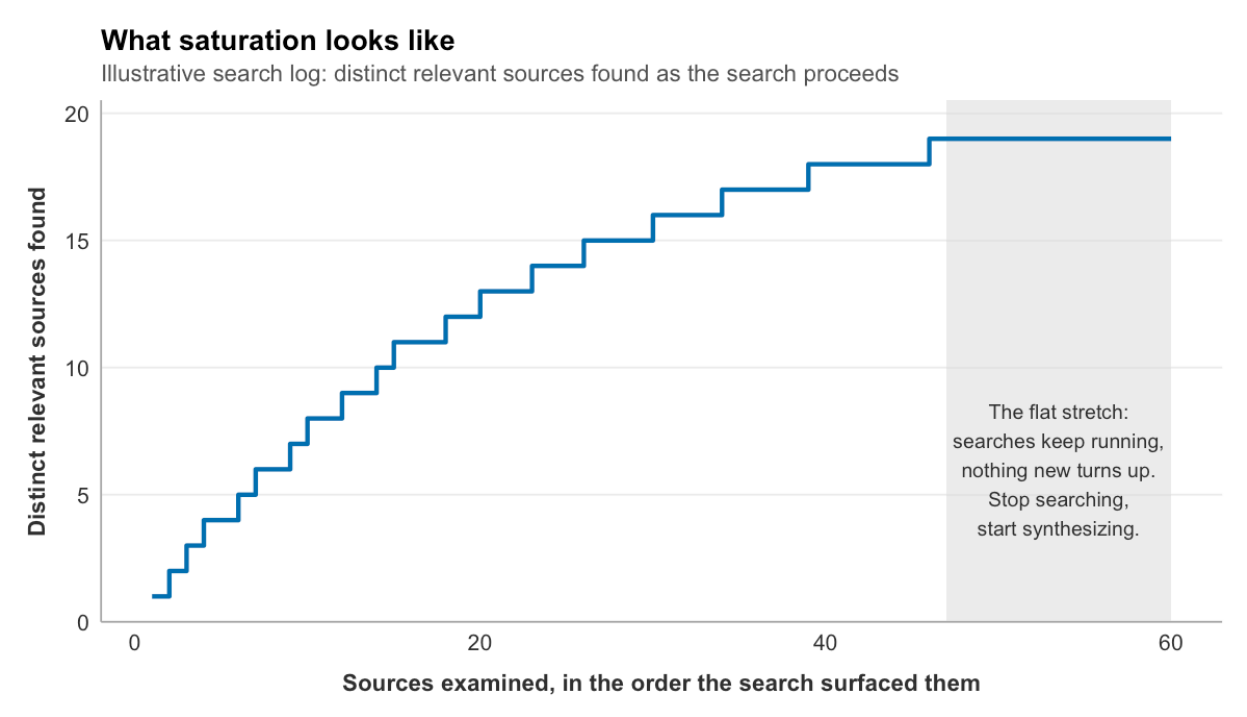
The process creates a snowball effect. One key article leads to ten more, which lead to twenty more. But the growth is strategic, not random. You are following the citation network outward from the work that matters most.

5.4 Recognizing saturation

At some point, new searches yield diminishing returns. You are finding articles that cite the same foundational works you have already read. The arguments sound familiar. The methods are variations on approaches you have seen.

This is **saturation**, the point where additional searching is unlikely to change your understanding of the literature. Saturation does not mean you have read everything, which is impossible. It means you have read enough to identify the major theories, recognize the key debates, understand the common methodological approaches, and articulate what is known and what is still uncertain.

Practically, saturation looks like three consecutive database searches turning up zero new relevant articles, new articles citing the same eight to ten foundational sources you have already read, and an ability to predict what an article will say from its title and abstract. When you reach that point, stop searching and start synthesizing.



5.5 Identifying research gaps

A research gap is not just "something no one has studied." It is a space where inquiry is justified, where the absence of knowledge creates a problem or where contradictions demand resolution. Gaps come in several forms.

A **topical void** is a phenomenon, population, or context no one has studied. There is substantial research on why people watch livestreams, for example, but less that examines how chat behavior itself differs between gaming and non-gaming streams. Be cautious here: what seems novel to you might be covered under different terminology, and thorough searching is the guard against claiming novelty where none exists.

A **methodological gap** exists when a phenomenon has been studied, but with methods that have important limitations. Most livestreaming motivation research relies on self-report surveys, which capture what viewers say motivates them. Behavioral trace data, like the contents of a chat log, could reveal whether what viewers do matches what they say.

A **contradiction** exists when previous studies have produced conflicting findings and your study aims to resolve the inconsistency. Two well-designed studies reaching opposite conclusions is not something to be embarrassed about reporting. It is an opening.

A **theoretical gap** exists when existing work explains a phenomenon through one lens and you believe another lens would be more illuminating. If most research on livestream engagement uses uses-and-gratifications theory, a parasocial-interaction framework (Horton & Wohl, 1956) might explain a part of the picture the dominant lens leaves dark.

5.6 Articulating the gap

When you write the literature review, the gap becomes the hinge of the argument. The structure is consistent regardless of topic:

1. **Establish what is known.** Summarize existing research to show you understand the conversation.
2. **Identify the limitation.** Point out what is missing, contradictory, or inadequately explained.
3. **Argue for your study.** Show how your research addresses that gap.

Known, then limitation, then what this study does. A student studying livestreaming would use different citations than a student studying news framing, but both follow the same logic: research has established X, however most studies share limitation Y, this study addresses that gap by doing Z.

5.7 Organizing sources: the literature map

By now you have dozens of sources in Zotero. The challenge is making sense of them, not as individual articles but as a coherent body of knowledge.

A **literature map** is a document that organizes sources by theme rather than chronologically or alphabetically. Create it as a markdown file in your project, with a heading for each theme and the relevant sources listed beneath it. A literature map for livestreaming research might have a theme for viewer motivations, a theme for community and social dynamics, a theme for the streamer's labor and performance, a theme for content analysis methods, and a running section for the gaps you have identified.

The map does three things. It shows you at a glance which themes are well covered and which are sparse. It forces you to think about how sources relate, which is the beginning of synthesis. And its structure often becomes the structure of your written literature review.

5.8 Synthesis, not summary

A weak literature review reads like a list:

> Hamilton, Garretson, and Kerne (2014) studied Twitch as a participatory community. Sjöblom and Hamari (2017) examined viewer motivations. Hilvert-Bruce and colleagues (2018) studied social motivations for engagement.

Three disconnected summaries. The reader learns what each study was about but not how the studies fit together. Notice also what drives each sentence: the name of whoever wrote the study, rather than the thing that is known.

A strong literature review reads like an argument:

> Researchers have converged on a counterintuitive finding: livestream viewing is motivated less by the content on the screen than by the social experience around it. Twitch streams function as virtual third places, venues where informal communities form and socialize rather than broadcasts to be passively consumed (Hamilton et al., 2014). Put to a survey test through a uses-and-gratifications framework, social and tension-release motivations proved central to why people watched (Sjöblom & Hamari, 2017). The finding extends to engagement specifically: social interaction and a sense of community predicted not just watching but chatting, subscribing, and returning, and viewers of smaller channels were more socially motivated than viewers of large ones (Hilvert-Bruce et al., 2018), which is something this course can investigate directly. Taken together, this body of work suggests that the chat box, not the gameplay, may be where the most important activity on a livestream happens. However, these studies rely almost entirely on self-report surveys, which capture what viewers say motivates them rather than what they actually do. Whether the behavioral traces left in a chat log would corroborate the survey findings remains largely unexamined.

The second version groups sources by theme, identifies the convergence among them, adds nuance, names a methodological limitation they share, and points toward a gap. That is what a literature review demands: not reporting what others found, but constructing an interpretation of the collective evidence. Notice, too, where the gap in that example points. It points directly at the kind of study this course has you do, a content analysis of a real chat log. One more thing in that paragraph is worth copying: where the citations sit. The claim leads each sentence and the citation follows it in parentheses, which is the default form in a review. A name belongs at the front of a sentence only when that study is the point of it: a landmark, a definition you are adopting, or a finding you are about to argue with.

5.9 A minimum viable literature review

If the synthesis above feels intimidating, start smaller. A defensible first pass needs only three sources and four sentences.

1. Source A, one sentence. What did this study find?
2. Source B, one sentence. What did this study find?
3. Source C, one sentence. What did this study find?
4. The gap, one sentence. What do these three sources, together, leave unexplored that your study could address?

A worked example using three livestreaming sources:

> Twitch streams function as virtual third places where informal communities form (Hamilton et al., 2014). Social and tension-release motivations are central to why people watch on Twitch (Sjöblom

& Hamari, 2017). Social interaction predicts not just watching but chatting and subscribing (Hilvert-Bruce et al., 2018). All three studies rely on self-report surveys, leaving the question of whether the same patterns are visible in behavioral chat-log data largely unexamined.

Four sentences, three citations, one gap. That is enough to anchor a prospectus. The synthesis paragraph in the previous section is what this minimum version grows into after another round of reading. It is not the starting point.

5.10 The ethics of citation

Citation is not bureaucratic. It is ethical. When you use someone else's idea, method, or finding, you credit them. This holds even when you are paraphrasing rather than quoting directly.

Specific empirical findings require citation. Theoretical concepts require citation. Methodological approaches require citation. Direct quotes always require citation. Paraphrased ideas require citation unless they are common knowledge in the field. What does not require citation is genuine common knowledge ("Twitch is a livestreaming platform"), your own original interpretations, and your own data and analysis.

When in doubt, cite. Over-citation is rarely criticized. Under-citation damages credibility, and in its more serious forms it is plagiarism, which Chapter 3′s discussion of research integrity treats as the breach of trust that it is.

5.11 Looking ahead

Chapter 5 takes up theory. A literature review tells you what the scholarly conversation has established. A theoretical framework gives you the lens you will use to interpret your own findings. Chapter 5 also addresses how the choice of theory shapes the choice of method, and introduces the major social-science methods as a coherent set, so that committing to content analysis for this course is a decision made with the alternatives in view.

5.12 References

Hamilton, W. A., Garretson, O., & Kerne, A. (2014). Streaming on Twitch: Fostering participatory communities of play within live mixed media. In *Proceedings of the 32nd ACM SIGCHI Conference on Human Factors in Computing Systems* (pp. 1315-1324). Association for Computing Machinery. https://doi.org/10.1145/2556288.2557048

Hilvert-Bruce, Z., Neill, J. T., Sjöblom, M., & Hamari, J. (2018). Social motivations of live-streaming viewer engagement on Twitch. *Computers in Human Behavior*, *84*, 58-67. https://doi.org/10.1016/j.chb.2018.02.013

Horton, D., & Wohl, R. R. (1956). Mass communication and para-social interaction: Observations on intimacy at a distance. *Psychiatry*, *19*(3), 215-229. https://doi.org/10.1080/00332747.1956.11023049

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology* (4th ed.). SAGE Publications.

Neuendorf, K. A. (2017). *The content analysis guidebook* (2nd ed.). SAGE Publications.

Sjöblom, M., & Hamari, J. (2017). Why do people watch others play video games? An empirical study on the motivations of Twitch users. *Computers in Human Behavior*, *75*, 985-996. https://doi.org/10.1016/j.chb.2016.10.019

5.12.1 Graduate readings

Cohen, J. (1992). A power primer. *Psychological Bulletin*, *112*(1), 155-159. https://doi.org/10.1037/0033-2909.112.1.155

Cooper, H., Hedges, L. V., & Valentine, J. C. (Eds.). (2019). *The handbook of research synthesis and meta-analysis* (3rd ed.). Russell Sage Foundation.

6 Chapter 5: Theory as a Lens

Listen in Dr. Leith's voice

Chapter 4 ended on a finding. Across a body of survey research, livestream viewing looks more socially motivated than content motivated. People watch, the studies suggest, for the chat as much as for the game. Take that finding as settled for a moment. It still does not explain itself. Why would the social experience matter more than the content on the screen? What is actually happening when a viewer chooses a stream and stays?

Three researchers could look at the same finding and explain it three different ways. One, drawing on **uses and gratifications theory** (Katz, Blumler, & Gurevitch, 1973), might argue that viewers are active agents seeking to satisfy specific needs, and that a livestream happens to satisfy a social need that a finished, edited video cannot. A second, drawing on **parasocial interaction** (Horton & Wohl, 1956), might argue that viewers form one-sided emotional bonds with the streamer, and that the sense of social connection is the felt experience of that bond. A third, drawing on **social identity theory** (Tajfel & Turner, 1979), might argue that a stream community is a group, that viewers use it to construct part of their identity, and that the chat is a space of belonging.

Each explanation focuses attention on a different mechanism. Uses and gratifications emphasizes the individual's active choice. Parasocial interaction emphasizes the relationship with the streamer. Social identity emphasizes the group. The same empirical pattern generates three different research programs depending on which lens you choose.

This is what theory does. It is not decorative, and it is not an abstract exercise you complete before getting to the real work of analysis. Theory is the architecture that transforms scattered observations into coherent understanding. It tells you what to look for, what to measure, and what would count as evidence for or against your explanation. It is also not specific to any one topic. The same theories that illuminate livestreaming also explain news consumption, political persuasion, brand loyalty, and health behavior. Choosing a theory is choosing a way of seeing.

6.1 Theory is not speculation

In casual conversation, "theory" often means "guess" or "hunch." A theory about why the coffee shop is always crowded on Tuesdays. In research, the term means something more precise. A **theory** is a formal, systematic explanation of relationships between concepts or variables. It organizes knowledge, explains phenomena, and generates predictions. Good theories are logically coherent, meaning the parts fit together without contradiction. They are generalizable, meaning they apply across contexts rather than to one specific case. They are falsifiable, meaning they make predictions that could in principle be proven wrong. And they are generative, meaning they produce new hypotheses and research questions.

Consider social identity theory, developed by Henri Tajfel and John Turner (1979). It proposes that people derive part of their self-concept from the groups they belong to. Group membership creates in-group

favoritism, a preference for "our" people, and out-group derogation, a tendency to distance from "them." When a group's status is threatened, members are motivated to protect it.

This is not speculation. It is a systematic framework built from decades of experimental research, and it generates testable predictions. Applied to livestreaming, it predicts that viewers will rate their preferred stream community more positively than rival communities, especially on subjective dimensions like authenticity. It predicts that when a streamer or community is criticized, regular viewers will respond defensively. Applied to a different domain entirely, it predicts that political partisans will rate news outlets aligned with their party as more credible than opposing outlets, even when the factual content is identical. The theory does not change. The domain does. Each prediction is testable, and that is what makes it theory rather than speculation.

6.2 The lens metaphor

Theory functions like a camera lens. It brings certain elements into sharp focus while leaving others blurred or outside the frame entirely.

If you study livestreaming using **appraisal theory**, which focuses on how individuals cognitively evaluate stimuli, you pay attention to individual psychological responses, the cognitive processes behind them, and personal experience. You design measures that capture individual reactions: surveys asking viewers to rate how a stream makes them feel, or experiments manipulating stream features and measuring responses.

If you study the same phenomenon using social identity theory, you focus on different elements: group-based meanings, collective identity, the cultural context in which a community formed. You design different measures, analyzing how community members talk about the stream as an identity marker, or how community boundaries are policed in chat.

Same phenomenon, different lenses, different research designs. Neither approach is wrong. They are answering different questions because they are guided by different theoretical frameworks. The key is choosing a lens that fits your research question and your data. You cannot use every lens at once, because that produces incoherent blur. You commit to a perspective, knowing it will illuminate certain aspects while leaving others in shadow.

6.3 Three paradigms

The way researchers use theory depends on their broader philosophical commitments, what is called a **paradigm**. Three major paradigms dominate communication research, and each treats theory differently.

The **social scientific paradigm** assumes there is an objective reality that can be observed, measured, and understood through systematic testing. Its approach to theory is deductive: start with theory, derive specific predictions, collect data to test those predictions. Its typical question is "does X cause Y?" A researcher might start with cultivation theory (Gerbner & Gross, 1976), which proposes that heavy media exposure cultivates perceptions consistent with media portrayals, hypothesize that heavy viewers of a particular kind of stream develop particular perceptions, and design a study to test that prediction. If the data support the hypothesis, confidence in the theory strengthens. If not, the theory or the hypothesis needs refinement. This cycle, from theory to hypothesis to data to conclusion to refinement, is how social science progresses.

The **interpretive paradigm** assumes reality is socially constructed and that meaning emerges through interaction and interpretation. Its approach to theory is inductive: start with detailed observations,

identify patterns, build theory from the data. Its typical question is "what does X mean to the people who experience it?" A researcher might immerse themselves in a stream community, analyze how members talk, notice that they engage in shared interpretive labor, and propose a theory that the community builds identity through collaborative sense-making. The theory is the end point, grounded in the data, emerging from the bottom up. Methods common to this paradigm include grounded theory, thematic analysis (Braun & Clarke, 2006), and narrative analysis.

The **critical and cultural paradigm** assumes that power structures shape what counts as knowledge, and that research should critique and challenge inequity. Its approach to theory is to use it as an explicit tool for revealing hidden power dynamics. Its typical question is "whose interests does X serve, and whose voices does it silence?" A researcher studying livestreaming might examine who gets to be a successful streamer, connect the visibility of certain creators to the platform's design and economic structure, and argue that what looks like neutral popularity is shaped by forces that advantage some creators over others. The critical lens transforms description into argument. Other critical lenses include the political economy of media and discourse analysis.

Theory before data as an epistemological commitment. Beyond focusing your research question and guiding your choice of variables, theory serves a further function: it is what you use to derive testable predictions *before you look at the data*. This sequencing (theory → prediction → data → test) is not a formality. It is what distinguishes confirmatory from exploratory research. Nosek and colleagues locate pre-registration at exactly this seam, defining it as the act of fixing the analytic path before any outcome is visible:

> "Preregistration of an analysis plan is committing to analytic steps without advance knowledge of the research outcomes."
>
> Nosek et al. (2018, p. 2601)

For your own study, draft the hypotheses section of an OSF pre-registration, and ask how precisely your theoretical framework allows you to state a directional prediction, and where it is too vague to pre-register.

6.4 Grand theories and middle-range theories

Not all theories operate at the same level of abstraction.

Grand theories explain fundamental aspects of human behavior across all contexts. Marxism holds that all social relations are shaped by economic structures and class conflict. Psychoanalysis holds that human behavior is driven by unconscious desires. Structuralism holds that meaning emerges from underlying universal structures. Grand theories are intellectually powerful, but they are difficult to test empirically. The scope is too broad and the predictions too general to design a study that could prove or disprove them.

Middle-range theories occupy a more useful zone: specific enough to generate testable hypotheses, broad enough to generalize beyond a single case. The sociologist Robert Merton coined the term to describe frameworks that sit between grand theoretical systems and the narrow descriptions of individual studies. Most communication research uses middle-range theories, and every theory in the catalog below is one.

6.5 A catalog of middle-range theories

The theories below are the workhorses of communication research. Each is presented with its foundational citation, its core idea, and an application to livestreaming, but each applies far beyond livestreaming, and the cross-domain examples make that point.

Parasocial interaction. Horton and Wohl (1956) observed that audiences develop one-sided emotional relationships with media figures that feel intimate but lack reciprocity. The theory predicts that people who consume more content from a creator report stronger feelings of connection, that parasocial bonds influence real behavior such as purchasing, and that disruptions like scandals produce feelings of betrayal. In livestreaming, parasocial interaction is the natural lens for studying why viewers donate to streamers they have never met, or why a streamer's absence produces something like grief in a community. Beyond livestreaming, the theory is central to research on influencer marketing and political communication.

Social identity theory. Tajfel and Turner (1979) proposed that group membership shapes self-concept and motivates in-group favoritism and out-group derogation. The theory predicts that people attend preferentially to information that reflects well on their in-group, that threats to group status increase defensive behavior, and that group boundaries are maintained through symbolic markers. In livestreaming, stream communities are rich sites for this lens: viewers do not just watch a streamer, they identify with a community, and that identification shows up in how they talk, what emotes they use, and how they treat outsiders. Beyond livestreaming, social identity theory is foundational in political communication, sports fandom, and brand community research.

Agenda setting. McCombs and Shaw (1972) found that media do not tell people what to think, but they tell people what to think about. By emphasizing certain issues and ignoring others, media shape public priorities. The theory predicts that heavily covered topics are perceived as more important, that the effect is stronger for issues people have less direct experience with, and that elite media influence discourse more than tabloids. In livestreaming, the platform's own structures, the front page, the recommendation system, the category rankings, function as agenda setters, shaping which streamers and games are perceived as relevant. Beyond livestreaming, agenda setting originated in election research and has been applied to health and environmental communication.

Framing theory. Goffman (1974) and later Entman (1993) established that how information is presented, the frame, influences how it is interpreted. The same facts, given different emphasis, produce different conclusions. The theory predicts that episodic framing leads audiences to attribute problems to personal responsibility, that thematic framing leads them to attribute problems to structural causes, and that frames aligning with existing beliefs are more persuasive. In livestreaming, framing is visible in how streamers title and categorize their own broadcasts, and in how media coverage frames streaming itself, as a career, as a waste of time, as a new form of celebrity. Beyond livestreaming, framing theory is central to political, health, and organizational communication.

Uses and gratifications. Katz, Blumler, and Gurevitch (1973) argued that audiences are active, not passive: people choose media to satisfy specific needs, including information, entertainment, social connection, and identity construction. The theory predicts that people select media that fulfill current needs, that different media satisfy different gratifications, and that media use changes when needs change. In livestreaming, uses and gratifications is the lens behind much of the viewer-motivation research from Chapter 4. Sjöblom and Hamari (2017) explicitly used it to study why people watch others play video games. Beyond livestreaming, the theory has been applied to every medium from radio to TikTok.

Cultivation theory. Gerbner and Gross (1976) proposed that long-term, cumulative exposure to media content shapes audiences' perceptions of social reality, so that heavy consumers develop beliefs more consistent with media portrayals than with real-world conditions. The theory predicts that heavy viewers overestimate the prevalence of what they see, that the effect is strongest where audiences lack direct experience, and that cultivation is gradual rather than the result of single exposures. In livestreaming, a cultivation question might ask whether heavy viewers of a particular genre of stream develop skewed perceptions of how common certain behaviors are. Beyond livestreaming, cultivation originated in television research and remains most associated with it.

6.6 Trying two lenses on the same data

The fastest way to feel what a lens does is to apply two of them to the same fragment of data. Consider one short stretch of chat from a Twitch broadcast:

> **DragoBoi89:** xQc actually playing well today wtf
>
> **emotelord:** PogU PogU
>
> **realnamedfan:** @xqcow do u remember when u said hi to me last week
>
> **streamfan22:** chat is this real

Through the **uses and gratifications** lens, you ask what needs these viewers are meeting in the moment. DragoBoi89 is satisfying a need for evaluative commentary on performance. emotelord is satisfying a need for collective emotional expression. realnamedfan is satisfying a need for direct social acknowledgement from the streamer. streamfan22 is satisfying a need for collective sense-making with other viewers. The lens directs your attention to motivations and to the variety of needs a single chat satisfies. All four viewers are roughly equally interesting because each represents a different gratification.

Through the **parasocial interaction** lens, you ask what the messages reveal about the viewer-streamer bond. DragoBoi89′s comment treats xQc as a familiar figure whose performance can be assessed in casual register. realnamedfan's @-direct address paired with a memory claim about a prior acknowledgement is a textbook parasocial move: the viewer is treating a one-sided interaction as a continuing relationship. emotelord and streamfan22 are not addressing the streamer at all, so the lens does not foreground them.

Same four messages, two different stories. The uses-and-gratifications lens treats all four viewers as equally interesting. The parasocial lens makes realnamedfan the central case and pushes emotelord and streamfan22 to the periphery. Neither reading is wrong. They are answering different questions, and the prospectus you write in Chapter 6 will commit to one of them.

6.7 Variables: the building blocks of hypotheses

Theories become testable through **variables**, measurable concepts that can take on different values.

An **independent variable** is what you manipulate, in an experiment, or measure as the predictor, in a correlational study. It is the proposed cause. In livestreaming research, an independent variable might be the stream's category type (gaming or non-gaming), the streamer's tenure on the platform, or a stream's concurrent viewer count.

A **dependent variable** is the outcome you measure, the thing you think is influenced by the independent variable. It might be the number of chat messages per minute, viewer retention over the course of a broadcast, or the rate of subscriptions and donations.

Consider a simple hypothesis: in non-gaming streams, a higher proportion of chat messages are directed at the streamer than in gaming streams. The independent variable is category type, gaming or non-gaming. The dependent variable is whether a message is directed at the streamer or broadcast to the room. This is testable. You classify streams by category, code a sample of messages for whether each one addresses the streamer, and run a statistical test to see if the predicted relationship holds. The same structure works in any domain. News stories framed around individual human interest will generate more social media engagement than stories framed around systemic data: the independent variable is frame type, the dependent variable is engagement, and the logic is identical.

Two further concepts refine the picture. A **mediator** explains how or why an independent variable affects a dependent variable; it is the mechanism in between. If stream category affects the proportion of directed messages, the mediator might be the perceived pace of the stream: non-gaming streams may feel more conversational, and that conversational feel may be what actually invites viewers to address the streamer directly. A **moderator** changes the strength or direction of a relationship; it answers the question "under what conditions?" The relationship between category type and directed-message rate might itself be moderated by audience size, holding on small streams but weakening on large ones. Mediators and moderators are how research moves from "X relates to Y" toward "X relates to Y, through this mechanism, under these conditions."

6.8 Applying theory to your research

Selecting a theory is not arbitrary. It should fit your research question, your data, and the kind of explanation you are seeking. The process has a consistent shape.

First, identify your research question, and be specific. "Livestreaming and community" is too broad. "Does the proportion of chat messages directed at the streamer differ between gaming and non-gaming streams?" is focused enough to act on. Second, map the question to a theoretical domain: psychological mechanisms point toward appraisal theory, social and group processes point toward social identity theory or uses and gratifications, questions about media influence on perception point toward cultivation, agenda setting, or framing. Third, review how others have studied similar questions, which is where the literature review from Chapter 4 pays off; convergence on a particular theory signals it is a productive framework for the domain. Fourth, derive specific hypotheses or research questions by letting the theory generate predictions. Fifth, design measures that operationalize the theoretical concepts, translating abstract ideas like identity or salience into variables you can actually measure. That last step, operationalization, is the work of Chapters 7 and 8.

HARKing and why it undermines inference. HARK stands for Hypothesizing After Results are Known. It occurs when a researcher examines data, notices a pattern, then writes up the report as if the pattern had been predicted in advance. The statistical test reports a *p*-value, but because the hypothesis was generated by the data rather than before it, that *p*-value does not carry its nominal meaning. Nosek and colleagues describe the inferential cost of blurring the line between predicting an outcome and explaining one after the fact:

> "Failing to appreciate the difference can lead to overconfidence in post hoc explanations (postdictions) and inflate the likelihood of believing that there is evidence for a finding when there is not."
>
> Nosek et al. (2018, p. 2600)

Pre-registration is the structural solution: you submit your theory-derived hypotheses, variables, and analysis plan to a public registry (OSF) before data collection begins. The registry entry is time-stamped and permanent. For your own study, consider how Nosek et al. distinguish pre-registration from Registered Reports: what additional protection does a Registered Report provide, and which communication journals currently offer that format?

What a V2V pre-registration includes. (1) Research question and directional hypotheses; (2) unit of analysis, platform, and data collection window; (3) the codebook or extracted variables; (4) planned statistical tests and inferential criteria; (5) sample size justification (from Chapter 6). OSF's standard template walks through each section.

6.9 Different theories call for different methods

Choosing a theory is closely tied to a decision this book has so far left implicit: the choice of **method**. A theory tells you what to look for. A method is how you go and look. And different theories, because they ask different kinds of questions, tend to call for different methods. It is worth seeing the major social-science methods as a set before this book commits, deliberately, to one of them.

Surveys measure what people report about themselves: their attitudes, their motivations, their self-described behavior. A survey is the natural method for a uses-and-gratifications question, because uses and gratifications is a theory about what audiences need and seek, and the most direct way to learn what someone seeks is to ask them. The livestreaming motivation research from Chapter 4 is survey research. The limitation of surveys is the gap between what people say and what they do: a viewer can sincerely report a motivation that does not match their actual behavior.

Experiments isolate cause by manipulating one variable while holding others constant. An experiment is the natural method for a sharp causal question, the "does X cause Y?" of the social scientific paradigm. If you want to know whether a particular stream feature causes a change in viewer behavior, you manipulate that feature, hold the rest constant, and measure the result. The limitation of experiments is artificiality: the control that makes causal inference possible also makes the setting less like the world the behavior normally happens in.

Qualitative methods, including in-depth interviews, focus groups, and ethnography, capture meaning, process, and lived experience. They are the natural methods for the interpretive paradigm's question, "what does X mean to the people who experience it?" If you want to understand how a stream community understands itself, you talk to its members and observe it at length. The limitation of qualitative methods is that they are not built to generalize in the statistical sense; they trade breadth for depth.

Content analysis systematically quantifies the features of communication content itself: the messages, the texts, the artifacts. It is the natural method for questions about what is in the content, asked at a scale too large to eyeball. It is also the method this course teaches end to end, for three reasons. The first is fit: the Twitch dataset is content, millions of chat messages and stream records, and content analysis is the method native to that kind of data. The second is pedagogical completeness: content analysis lets you practice the full research pipeline, operationalization, sampling, reliability, analysis, on a single dataset, without the separate apparatus of human-subjects recruitment. The third is transfer: the disciplines content analysis

demands, a precise codebook, tested reliability, systematic sampling, are the same disciplines that make every other method work. Learning one method deeply is the fastest way to understand what all of them are trying to do.

Seen together, these four methods are not a random toolbox. They line up with the paradigms from earlier in this chapter. The social scientific paradigm, with its deductive testing of predictions, leans on surveys and experiments. The interpretive paradigm, building theory from the ground up, leans on qualitative methods. Content analysis is the flexible case: it can serve a social scientific study that counts and tests, or an interpretive study that reads for meaning, depending on how its codebook is built. A method is not just a procedure. It is the operational expression of a way of seeing, which is why the choice of method and the choice of theory are, in the end, the same decision approached from two directions.

This is not a claim that content analysis is the best method. It is a claim that it is the right method for this dataset and this course, and that committing to it is a decision worth making with the alternatives in view rather than by default.

6.10 Looking ahead

Chapter 6 turns the theory and the literature into a plan. A research prospectus is the document that states a research question, situates it in the literature, names a theoretical framework, and commits to a method, all before any data is collected. It is where the foundations of the first half of this book become a concrete proposal for the second half.

6.11 References

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Entman, R. M. (1993). Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, *43*(4), 51-58. https://doi.org/10.1111/j.1460-2466.1993.tb01304.x

Gerbner, G., & Gross, L. (1976). Living with television: The violence profile. *Journal of Communication*, *26*(2), 172-199. https://doi.org/10.1111/j.1460-2466.1976.tb01397.x

Goffman, E. (1974). *Frame analysis: An essay on the organization of experience*. Harvard University Press.

Horton, D., & Wohl, R. R. (1956). Mass communication and para-social interaction: Observations on intimacy at a distance. *Psychiatry*, *19*(3), 215-229. https://doi.org/10.1080/00332747.1956.11023049

Katz, E., Blumler, J. G., & Gurevitch, M. (1973). Uses and gratifications research. *Public Opinion Quarterly*, *37*(4), 509-523. https://doi.org/10.1086/268109

McCombs, M. E., & Shaw, D. L. (1972). The agenda-setting function of mass media. *Public Opinion Quarterly*, *36*(2), 176-187. https://doi.org/10.1086/267990

Sjöblom, M., & Hamari, J. (2017). Why do people watch others play video games? An empirical study on the motivations of Twitch users. *Computers in Human Behavior*, *75*, 985-996. https://doi.org/10.1016/j.chb.2016.10.019

Tajfel, H., & Turner, J. C. (1979). An integrative theory of intergroup conflict. In W. G. Austin & S. Worchel (Eds.), *The social psychology of intergroup relations* (pp. 33-47). Brooks/Cole.

6.11.1 Graduate readings

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

7 Chapter 6: The Prospectus

Listen in Dr. Leith's voice

There is a particular kind of paralysis that strikes partway through a research project. You have read thirty articles. You have notes scattered across several documents. You have identified a few interesting theories and several possible research questions. You can see connections everywhere, and the project feels simultaneously too small, just one study, and too large, impossible to finish.

The problem is not a lack of ideas. It is a lack of focus. The difficult choices that turn a cloud of possibilities into a concrete plan have not been made yet.

A **research prospectus** forces those choices. It is a short document, roughly one page, that states what you are studying, why it matters, and how you will do it. By the end of this chapter you will be able to write one. The prospectus is not busywork. It is a diagnostic. If you cannot articulate your project clearly on a single page, you do not yet understand it well enough to begin. And the single hardest element to get right, the element the rest of the prospectus is built around, is the research question.

7.1 The research question

The hardest part of research is not analyzing data or writing the final report. It is asking the right question. Most students can learn to run a statistical test or code content systematically with practice. What is harder is the intellectual work that precedes analysis: formulating a question that is specific enough to answer, broad enough to matter, and grounded enough in existing knowledge to avoid reinventing the wheel.

Consider two questions. Question A: "How does livestreaming affect people?" Question B: "Does the rate of chat messages per viewer differ between gaming and non-gaming streams?" Question A is a career-spanning research program disguised as a single question. It is so broad that it is essentially unanswerable within any realistic scope. Question B is researchable. It specifies what will be measured, what will be compared, and what evidence would answer it. You can design a study around it, and you can know when you are done. The craft of research question formulation is the craft of getting from A to B.

A strong research question meets five criteria.

It is specific. Weak questions use vague language that could mean different things to different people. "How does social media influence politics?" leaves "influence," "social media," and "politics" all undefined. A stronger version names the platform, specifies the outcome, and bounds the population.

It is measurable. Some questions use concepts that resist **operationalization**, the translation of an abstract idea into a concrete, observable variable. "Do authentic streamers build better communities?" cannot be studied until "authentic" and "better" are operationalized into something you can actually observe and record.

It is answerable within your constraints. Some questions are sound in principle but impossible given your resources, timeline, or skills. A study requiring decades of data, or interviews with hundreds of participants, is not a semester project no matter how good the question.

It is not already definitively answered. Your literature review from Chapter 4 should reveal whether your question is genuinely open. If a relationship has been tested dozens of times with consistent results, asking it again without a new angle, a new population, a new moderator, contributes little.

It matters. The question should have intellectual or practical significance: it should contribute to theory, resolve a contradiction, or have applied value. "Do streams that start on Tuesdays draw more viewers than streams that start on Thursdays?" might be answerable, but even a clear answer would be trivia rather than insight.

A template helps produce questions that meet these criteria: some combination of *does there exist*, plus a *specific variable or concept*, plus a *relationship, pattern, or difference*, plus a *bounded population or context*, plus, where relevant, the *confounds being accounted for*. "Is there a relationship between stream category and chat message rate among channels in the Twitch working corpus?" is built from exactly those parts.

7.2 Narrowing: scope creep and the funnel

The greatest threat to student research is the Everything Study. It announces itself in phrases like "I want to study the effect of social media on society" or "I am going to analyze streaming platforms." These are not research projects. They are career-spanning programs that would occupy teams of scholars for decades.

Scope creep happens when you try to answer too many questions at once, invoke multiple theories that pull in different directions, attempt to analyze data too large or complex for the timeline, or keep adding one more thing to the study. The solution is relentless narrowing: drilling down until the project fits the constraints of time, resources, and skill you actually have.

Narrowing is a process, and it looks like a funnel:

- Iteration 1: "I am interested in Twitch chat." Too broad. What aspect? Which streams?
- Iteration 2: "I want to study how viewers participate in chat." Better, but still vast.
- Iteration 3: "I want to analyze whether chat participation differs across kinds of streams." Getting there. Which kinds? Measured how?
- Iteration 4: "I want to examine whether chat message volume differs between gaming and non-gaming streams." Much clearer.
- Iteration 5: "I will draw a stratified sample of chat messages from the 50-channel Twitch working corpus, code each message as directed at the streamer or broadcast to the room, and test whether the proportion of directed messages differs between gaming and non-gaming streams, framing chat as a site of social gratification."

Each iteration sacrifices breadth for depth. By the fifth version, the project can be completed in a semester. It will not answer everything about Twitch chat, but it will answer something rigorously. The hard part is accepting that narrowing is not weakness. It is discipline. And the logic is domain-general: a study of news framing or health campaigns or organizational communication runs through the same funnel, swapping the topic while keeping the narrowing logic intact.

Certain narrowing mistakes recur often enough to name. The **impossible comparison** asks a question confounded by self-selection: "are people who watch Twitch lonelier than people who do not?" compares groups that differ in many ways besides the one being studied. The **circular question** is tautological: "do popular streamers attract large audiences because people want to watch them?" defines its terms in a loop. The **false binary** forces a choice the data does not require: "is it the streamer or the game that

matters?" assumes you must pick one, when the better question asks how much each contributes and whether they interact.

7.3 From concepts to operational definitions

Theory gives you abstract concepts. Research requires concrete variables. The bridge is operationalization, and it usually involves a choice among imperfect options.

Consider **parasocial relationship**, the one-sided emotional bond between a viewer and a media figure (Horton & Wohl, 1956). It is not directly observable. You could operationalize it three ways. A self-report scale asks viewers to rate agreement with statements like "I feel like this streamer understands me"; it captures subjective experience directly but is subject to social desirability bias. Behavioral indicators count observable actions like donations, subscription length, or hours watched; behavioral data can be more reliable than self-report, but a behavior does not always reflect the bond you think it does. Language analysis codes viewer-created content, like chat messages, for parasocial markers: expressions of intimacy, emotional disclosure, direct address to the streamer. It uses naturalistic data and reveals how viewers actually talk, but it is labor-intensive and misses viewers who feel strongly without posting.

Consider **chat sentiment**, the emotional valence of chat content. Automated sentiment analysis scores messages with software: fast and replicable, but blind to sarcasm, irony, and the heavily coded vocabulary of Twitch emotes. Human coding trains coders to categorize sentiment holistically: it captures context and nuance, but it is labor-intensive and requires inter-coder reliability testing. Dimensional coding scores several dimensions at once, such as valence and arousal: it captures emotional complexity better, but it demands a more careful codebook.

In each case there is no single right answer. You choose the operationalization that fits your research question, your data access, and your resources, and you defend why the proxy you chose is valid. That logic is identical for any abstract construct, whether it is media trust, political polarization, or brand loyalty.

7.4 Question types: goals, questions, and hypotheses

Not all research asks the same kind of question, and the goal shapes every later decision.

Exploratory research asks "what is going on here?" It investigates a new or poorly understood phenomenon and generates hypotheses rather than testing them. It is often qualitative and inductive, and its output is patterns, themes, and preliminary frameworks. **Descriptive research** asks "what does the landscape look like?" It documents the characteristics of a population or phenomenon, often quantitatively, and its output is frequencies, distributions, and prevalence estimates. **Explanatory research** asks "why does this happen?" It tests relationships and mechanisms, it is hypothesis-driven and deductive, and its output is support or disconfirmation of predictions. Research on a topic often moves through these stages in order, from exploratory to descriptive to explanatory, and you do not need to do all three, but you should know which one you are attempting.

The goal determines whether you pose a research question or a hypothesis. A **research question** is interrogative and open-ended. Use one when you are exploring or describing, when theory does not make a clear prediction, or when the literature is too contradictory to predict confidently. A **hypothesis** is a declarative statement predicting a relationship. Use one when theory makes a specific prediction, when previous research suggests a likely outcome, or when you are testing a causal claim.

Behind every hypothesis sits a **null hypothesis**, which predicts no relationship or no difference. The null is the skeptical default position, and it is what statistical tests actually evaluate. If the data are sufficiently inconsistent with the null, you reject it in favor of your alternative hypothesis. This is the same logic Chapter 1 introduced as the sacred flaw: the question is not whether you can tell a story with your data, but whether the data could have come from a world where your effect does not exist.

7.5 Topic, theory, and data

A viable research project requires alignment across three elements: the topic, the theory, and the data. If any one is mismatched with the others, the study collapses.

The topic is what you are studying, and it must be bounded by population, time, medium, or context. "Livestreaming" is not a topic. "Chat participation in the 50-channel Twitch working corpus" is. The theory is the explanatory framework that guides interpretation, and it must predict patterns you can actually observe in your data. The data source is the evidence you will analyze, and it must be accessible within your timeline, sufficient to detect the patterns you are looking for, and appropriate to the question.

Misalignment is easiest to see in an example. Suppose the topic is parasocial relationships between viewers and streamers, the theory is the political economy of media, which explains structural forces like corporate consolidation and platform incentives, and the data is a content analysis of chat messages. The theory and the data are not in conversation. Political economy explains industry structure; chat messages are individual expression. The theory cannot explain the patterns the data would show. Now realign: keep the topic, but switch the theory to parasocial interaction (Horton & Wohl, 1956), which is about one-sided emotional bonds, and the pieces fit. The theory predicts that viewers expressing stronger bonds will behave differently in chat, the data captures that expression, and topic, theory, and data are finally talking to each other.

7.6 Writing the prospectus

The prospectus is a contract with yourself and your instructor. It commits you to a specific, focused project and demonstrates that you have thought through its feasibility. It has six components: a descriptive title that hints at the key variables; one to three focused research questions or hypotheses, not five and not ten; a two-to-three-sentence theoretical framework that names the theory and explains how it relates to the question; a two-to-three-sentence statement of the gap in the literature, citing a few key sources; a three-to-four-sentence method overview covering the data, the coding or measurement, and the number of cases; and a one-to-two-sentence statement of the expected contribution.

Assembled, a model prospectus for this course might read like this:

> **Title.** Directed and broadcast chat in gaming and non-gaming streams: A content analysis of the Twitch working corpus.
>
> **Research question.** Does the proportion of chat messages directed at the streamer differ between streams in gaming categories and streams in non-gaming categories, among the 50 channels in the Twitch working corpus?
>
> **Theoretical framework.** Uses and gratifications theory (Katz, Blumler, & Gurevitch, 1973) holds that audiences actively select media to satisfy specific needs, including social ones. If a substantial part of livestream viewing is socially motivated, the form of that participation, whether viewers

address the streamer directly or talk to other viewers in the room, is one observable trace of the gratification being sought, and stream type may shape which form predominates.

Gap in the literature. Research on livestreaming viewer motivation is dominated by self-report surveys (Sjöblom & Hamari, 2017; Hilvert-Bruce et al., 2018), which capture what viewers say motivates them rather than what they do. Whether the behavioral traces in a chat log corroborate the survey findings, and specifically whether directed and broadcast forms of participation vary with stream type, is largely unexamined.

Method. This study will draw a stratified sample of 1,500 chat messages from the 50-channel Twitch working corpus shipped with the course, balanced across gaming and non-gaming streams. Each message is the unit of analysis. Messages will be coded into three mutually exclusive categories: directed at the streamer, broadcast to the room, or unclassifiable. The codebook will be tested for intercoder reliability against a second coder on a 10 percent subsample, using Krippendorff's alpha as the reliability metric. The proportion of directed messages will then be compared between the two stream-type groups.

Expected contribution. The study tests whether a finding established in survey research, that livestream viewing is more socially motivated than content motivated, leaves a visible trace in coded chat behavior, contributing to uses-and-gratifications scholarship on livestreaming.

That is roughly two hundred and fifty words. It fits on one page, every sentence carries weight, and every component of the method (the sample, the unit of analysis, the coding scheme, the reliability check, the comparison) is named.

A graduate prospectus commits to a specific sample size backed by a quantitative justification, and that justification is the a priori power analysis. The number of cases named in the method section is not asserted but defended.

A different theory points the same student to a different prospectus. Here is the same dataset framed through parasocial interaction:

Title. Markers of parasocial bond in viewer chat: A content analysis of large-audience and small-audience streams in the Twitch working corpus.

Research question. Do chat messages on streams with smaller audiences show a higher rate of parasocial language than chat messages on streams with larger audiences?

Theoretical framework. Parasocial interaction theory (Horton & Wohl, 1956) proposes that audiences develop one-sided emotional bonds with media figures that feel intimate despite the absence of reciprocity. Hilvert-Bruce and colleagues (2018) found that viewers of smaller channels reported being more socially motivated than viewers of larger ones, which predicts that observable parasocial markers should appear more frequently in the chat of smaller streams.

Gap in the literature. Parasocial interaction has been studied primarily through self-report scales completed by audience members (Hilvert-Bruce et al., 2018). Whether parasocial markers are visible in the unprompted language audiences produce in real time, and whether they vary with stream audience size, has not been examined in a chat-log corpus.

Method. This study will draw a stratified sample of 1,000 chat messages from the 50-channel working corpus, split between channels in the top quartile of average viewer count and channels in the bottom quartile. Each message is the unit of analysis. Three parasocial markers will be coded as present or absent in each message: direct second-person address to the streamer, expressions of familiarity (use of the streamer's first name or known nicknames), and emotional disclosure. Intercoder reliability will be tested on a 10 percent subsample using Krippendorff's alpha. The rate of each marker will be compared between the two audience-size groups.

Expected contribution. The study extends parasocial-interaction research from self-report measures to behavioral trace data, testing whether Hilvert-Bruce and colleagues' (2018) finding about smaller-audience viewers leaves a measurable trace in coded chat behavior.

Two prospectuses, same dataset, different theory, different unit-of-analysis choices in the codebook, different comparison structure. Neither is "more correct." Each is a defensible plan, and each demonstrates the same underlying discipline: a question, a lens, a gap, a method that can be executed in a semester, a contribution scoped to what the study can deliver.

If both prospectuses feel within reach, read the method paragraphs once more before you commit. The first codes one variable. The second codes three. Three variables means three codebook decisions to defend, three reliability checks to run, and three results to write up, which is the better study to read and the heavier semester to execute. Pick the one your curiosity actually points to, and then check that you also picked the workload you can carry. The point of the prospectus is to make that trade-off visible to you before week 7 rather than after week 11.

7.6.1 A note before you write yours

The prospectus is pre-statistics work. You do not yet need to know which test you will run, what your p-value will be, or what R code will execute the comparison. Chapters 10 through 13 cover the analysis side. What the prospectus asks you to commit to is the structure of the study: a research question, a theory, a unit of analysis, a coding scheme, and the structure of the comparison you will make. The phrase "will be compared" in the method paragraph is genuinely all you need at this stage. The hard work in the prospectus is conceptual, not computational. If the stats half of the course is what you are dreading, the prospectus is not where that dread should land yet.

Certain prospectus problems recur. Multiple unrelated questions are really several separate studies; pick one. A theory-data mismatch pairs a framework with evidence it cannot explain. Vague methods ("I will look at some streams and see what emerges") cannot be evaluated for feasibility. And the Everything Study returns here in prospectus form, promising to analyze all of Twitch across all of its history.

It is easier to see those problems in a draft than in a list. A common first attempt looks something like this:

Title. Twitch and Society.

Research question. How does Twitch chat affect viewers' sense of community, and what is the role of parasocial relationships and social identity in shaping their behavior across different streams and genres? Also, are streamers becoming more popular over time?

Theoretical framework. I will use parasocial interaction theory, social identity theory, and the political economy of media to understand how Twitch shapes its audience.

Method. I will look at some chat logs and see what patterns emerge.

Expected contribution. This will contribute to our understanding of streaming.

Every recurring problem is present. The title names no variables. The research question is actually three or four questions stacked on top of each other. The framework piles up three theories that pull in different directions and that cannot all be operationalized on the same data. The method is vague. The contribution is generic. A reader cannot tell what the study is, what it would measure, or how to know when it is done.

The same student, after one narrowing pass, might revise to this:

Title. Direct-address language in gaming and non-gaming Twitch streams: A content analysis of the working corpus.

Research question. Does the rate of chat messages that directly address the streamer differ between gaming-category and non-gaming-category streams in the Twitch working corpus?

Theoretical framework. Uses and gratifications theory (Katz, Blumler, & Gurevitch, 1973) holds that audiences select media to satisfy specific needs, including the need for social interaction. If a stream's category shapes whether the social experience on offer centers on the streamer or on the group of fellow viewers, direct address to the streamer is one observable trace of which gratification audiences are pursuing.

Gap in the literature. Uses and gratifications research on livestreaming has relied primarily on self-report (Sjöblom & Hamari, 2017; Hilvert-Bruce et al., 2018), capturing what viewers say they want from the medium rather than what behavior the medium elicits. Whether direct-address rates in real-time chat vary systematically with stream type, as a behavioral test of the gratification proposition, has not been examined.

Method. A stratified sample of 1,000 chat messages from the 50-channel working corpus will be drawn, balanced across gaming and non-gaming streams. Each message is the unit of analysis and will be coded as direct-address or not, with intercoder reliability tested on a 10 percent subsample using Krippendorff's alpha. The rate of direct-address messages will be compared between the two categories.

Expected contribution. The study tests one observable prediction of uses-and-gratifications theory in behavioral chat data, complementing the self-report tradition with a directly observed alternative.

The revised version picks a single question, a single theory that the data can speak to, a method that can actually be executed in a semester, and a contribution scoped to what the study can deliver. The weak version was not lazy. It was trying to do too much. Narrowing is the move that converts ambition into a defensible plan.

Before finalizing, run the prospectus through a feasibility test. Can you access the data? For this course, yes: the working corpus ships with the `v2v` package. Can you analyze it in the time available? Fifty channels is manageable; the full population of 1,690 is not, for one person in a semester. Does your method match your question? Content analysis answers questions about what is in the content; it does not directly answer questions about why viewers feel what they feel. And have you controlled scope creep? If you keep adding variables, populations, or time periods, narrow until it hurts a little, then narrow a bit more.

7.7 The prospectus as pre-registration

The prospectus serves a function that extends well beyond this course. In professional research, a practice called **pre-registration** involves publicly committing to your hypotheses, methods, and analysis plan before collecting or analyzing data. Services like the Open Science Framework let researchers timestamp these commitments, which makes it possible to verify that a study was designed before its results were known.

Why this matters connects directly to Chapter 2. Simmons, Nelson, and Simonsohn (2011) demonstrated that researchers who do not commit to an analysis plan in advance can, consciously or not, adjust their hypotheses, variables, and analytical decisions until a result reaches significance. Their paper is the source of the term Chapter 2 introduced: researcher degrees of freedom. Pre-registration closes those degrees of freedom by making the plan inspectable. If the final paper reports different hypotheses than the pre-registration, that discrepancy is visible, and it raises questions worth raising.

Your prospectus functions as an informal pre-registration. It documents what you planned to study and how you planned to study it, before you began coding and analyzing. When you are later tempted to add variables, chase an unexpected finding, or quietly reframe the study after seeing the data, the prospectus is the record of what you originally set out to do.

Statistical power is the probability that your study will detect an effect of a given size, assuming that effect is real. A study with 50% power has a coin-flip chance of detecting the effect even if it exists. The conventional minimum threshold is 80% power. Below that threshold, a significant result is less credible, because underpowered studies have inflated false-positive rates and overestimated effect sizes. That 80% figure is a convention rather than a derived optimum, and Lakens cautions against treating the number as settled:

> "the default recommendation to aim for 80% power lacks a solid justification."
>
> Lakens (2022, p. 6) Lakens (2022) treats justification by resource constraints as a last resort: for your own study, ask whether it is ethically defensible to run a design you know is underpowered, and under what conditions.

Once you have an effect size estimate from the literature (Chapter 4) and a significance level ($\alpha = .05$), the `pwr` package computes the required n:

```
library(pwr)
# Two-group t-test, medium effect (d = 0.5), 80% power
pwr.t.test(d = 0.5, sig.level = 0.05, power = 0.80, type = "two.sample")
```

This output, $n = 64$ per group, becomes the sampling plan justification in your prospectus. If you cannot feasibly collect that many observations, the research question must be revised, the design changed for greater power, or the limitation explicitly justified in writing. For your own prospectus, run this analysis with the effect size you extracted in Chapter 4 and report the required n at both 80% and 90% power, noting how the ten-point increase changes what you would need to collect.

7.8 Looking ahead

Chapter 7 begins Part III, where planning becomes practice. Before you can code a single chat message, you have to know the data the way a researcher knows it: not as an abstraction described in a prospectus,

but as a texture you have spent real time inside. Chapter 7 is about that immersion, the structured listening that turns a dataset into something you understand well enough to operationalize.

7.9 References

Hilvert-Bruce, Z., Neill, J. T., Sjöblom, M., & Hamari, J. (2018). Social motivations of live-streaming viewer engagement on Twitch. *Computers in Human Behavior*, *84*, 58-67. https://doi.org/10.1016/j.chb.2018.02.013

Horton, D., & Wohl, R. R. (1956). Mass communication and para-social interaction: Observations on intimacy at a distance. *Psychiatry*, *19*(3), 215-229. https://doi.org/10.1080/00332747.1956.11023049

Katz, E., Blumler, J. G., & Gurevitch, M. (1973). Uses and gratifications research. *Public Opinion Quarterly*, *37*(4), 509-523. https://doi.org/10.1086/268109

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359-1366. https://doi.org/10.1177/0956797611417632

Sjöblom, M., & Hamari, J. (2017). Why do people watch others play video games? An empirical study on the motivations of Twitch users. *Computers in Human Behavior*, *75*, 985-996. https://doi.org/10.1016/j.chb.2016.10.019

7.9.1 Graduate readings

Lakens, D. (2022). Sample size justification. *Collabra: Psychology*, *8*(1), 33267. https://doi.org/10.1525/collabra.33267

Part III: Operationalization

8 Chapter 7: Structured listening

Listen in Dr. Leith's voice

In Chapter 2 you looked at ten rows of the `stream_log` and watched Sodapoppin's audience climb: 27,934 concurrent viewers, then 28,076 a minute later, then 28,203 a minute after that. Two hundred sixty-nine viewers in a hundred and twenty-two seconds. The dataset records that climb to the second. What the dataset cannot tell you is what the climb was.

It could have been a raid, another streamer ending a broadcast and sending their audience over. It could have been a clip going viral on Reddit, pulling in strangers. It could have been the streamer returning from a break, or a game update, or simply the slow accumulation of a good night. The row looks the same in every case: a number, then a larger number, then a larger number still.

The only way to know what a climb like that means is to have watched enough live Twitch to recognize the shapes these numbers make. Someone who has spent real time on the platform can look at a viewer curve and read it. A raid has a signature, a near-vertical jump, often with a wave of near-identical greetings in chat. A viral clip has a different signature, a slower swell of viewers who do not know the room's conventions. The dataset will not label these for you. You bring the labels, and you can only bring them if you have done the looking.

This chapter is about that looking. Before you can turn the Twitch dataset into variables, before you build a codebook and code a single message, you need to know the medium as lived experience rather than as an abstraction. The discipline has a name in qualitative research: **immersion**, sustained and systematic attention to a subject before analysis begins. For a content analyst working with Twitch, immersion takes the form of **structured listening**. You watch streams, you read chat, you take notes, and you do it with the same disciplined attention a music researcher brings to a song or an ethnographer brings to a field site. The dataset is historical, collected across just under six days in 2018. The platform is live in front of you right now, and the behaviors the dataset records, chat moving, audiences swelling and ebbing, streamers working a room, are still there to be watched. Watching them is the work of this chapter.

8.1 Why structured observation matters

Consider two ways to begin a content analysis of Twitch chat.

In the first, you have the dataset. You have a prospectus, and it commits you to comparing how chat behaves in gaming and non-gaming streams. You move straight to a coding scheme: each message is either directed at the streamer or broadcast to the room. You hand the scheme to two coders. They read the chat log as plain text and sort each message into one bucket or the other. You analyze the result.

The problem is not that you have measured nothing. The problem is that you do not know what you have measured. A coder who has never watched Twitch will read the message "LULW" and have no idea that it is an emote, a reaction of laughter, and not a typo or a username. They will read a wall of the same phrase repeated by thirty different accounts and not recognize it as copypasta, the coordinated spam that follows a raid or a notable moment. They will see "@sodapoppin same" and not know whether "same" is agreement with something the streamer said or a stock reply the room produces reflexively. The coding scheme sorts every message into a bucket, confidently, and the buckets do not mean what you think they mean.

In the second approach, you do the same work, but first you watch. Before any coding scheme exists, you spend hours on live streams across the kinds of channels your dataset contains. You read chat as it

scrolls. You notice that chat has conventions, that emotes carry meaning that their letter-strings do not, that a message addressed to the streamer by name is often performing for the room rather than expecting a reply. Then you build a coding scheme that reflects what is actually there. The scheme is slower to produce. It is also the difference between a finding and an artifact.

This is the logic the anthropologist Clifford Geertz called **thick description**: a record of behavior rich enough to capture not just what happened but what it meant in context (Geertz, 1973). A thin description of the chat log counts messages. A thick description knows that the count is made of raids and inside jokes and reflexive replies, and it codes accordingly. Structured listening is how a thin dataset becomes something you can describe thickly. The principle is not specific to Twitch. A researcher coding immigration coverage would read dozens of articles before fixing categories; a researcher studying advertising would watch scores of commercials first. The medium changes. The discipline does not.

8.2 Three modes of attention

Structured listening is not idle viewing. It is active, systematic observation, guided by your research question but open to what you did not expect. It works best in three modes, run in sequence.

Casual watching is initial exposure. Open several streams across the range of the platform: a large gaming channel, a small one, a Just Chatting stream, an art or music stream. Do not take notes. Do not try to code anything. Just watch, and let the medium be unfamiliar. What surprises you? What feels like a convention you do not yet understand? This is the browsing phase, and its only goal is to replace your assumptions about Twitch with exposure to it.

Focused watching is thematic attention. Now choose a few streams and watch them with specific dimensions in mind. The design brief for this kind of observation names three worth tracking, and they are a good place to start. **Chat behavior**: how fast does chat move, who is it addressed to, how does its pace change? **Viewer trajectories**: does the concurrent-viewer count climb, hold, or fall, and what happens on the stream when it moves? **Host moves**: what does the streamer, the host of the room, actually do? Do they read chat aloud, react to it, ignore it, steer it? Watch the same stream more than once with a different one of these dimensions in focus each time. Take brief notes after each pass. You are still describing, not coding.

Analytical watching is pattern recognition. Watch a larger number of streams while asking what recurs. Do certain chat behaviors cluster with certain kinds of stream? Are there category-specific conventions, so that chat in a competitive game looks different from chat in an art stream? What cases resist easy description? This is where the contours of your eventual coding scheme start to appear, though you are still documenting patterns rather than forcing categories onto them.

The sequence, familiarize then focus then analyze, holds for any medium. If you were observing news coverage you would skim, then read closely, then read for pattern. The modes are the same; only the material differs.

8.3 Field notes

Observation without a record is just watching television. Research observation produces **field notes**: written, dated, specific accounts of what you saw and what it made you think. Field notes are not polished prose. They are thinking on paper, and their value is that they capture an observation while it is fresh, before it hardens into something you assume you always knew.

Four kinds of field note are worth writing, and the distinction among them is a distinction of purpose.

An **observational note** records what you saw. "Watched a mid-size Just Chatting stream for forty minutes. Chat moved in bursts, near silence while the streamer talked, then a flood every time they asked a question or paused. The flood was mostly short messages, many of them the same two or three emotes. Chat speed seems tied to whether the streamer leaves an opening."

A **methodological note** records a measurement problem. "If I am coding messages as directed at the streamer or not, the bursts after a question are a problem. They are answers to the streamer, so directed, but they are also performance for the room. A coder reading the log as text cannot see the question that prompted them. The codebook may need a rule about messages that respond to streamer prompts."

A **theoretical note** connects an observation to a framework from your reading. "The burst-after-a-question pattern fits uses and gratifications (Katz, Blumler, & Gurevitch, 1973). If viewers are there for social-integrative reasons, the streamer's question is an invitation to participate, and the flood is the gratification being taken up. This suggests chat volume is not a steady trait of a stream but a response to host moves."

A **comparative note** records how cases differ. "Two streams, same hour. The gaming stream's chat ran constant and fast, mostly reacting to the game. The art stream's chat ran slow and conversational, with viewers talking to each other as much as to the streamer. Whatever I end up coding, gaming and non-gaming chat are not the same object."

Keep the notes in a single plain-text document, dated, one entry per observation session. The V2V Hub provides a field-notes template that gives the four note types a consistent structure. Set yourself a rhythm, a note every fifteen or twenty minutes of observation, so the record keeps pace with the watching. The notes do not need to be good. They need to exist, because the codebook you build in the next chapter will be assembled almost entirely from what they contain.

Two-coder reliability planning begins here. As you take field notes, identify candidate categories, and notice edge cases, you are doing something else at the same time: identifying *where two coders will disagree*. Structured immersion is not only preparation for a codebook. It is the first pass at the reliability problem, and field notes that mark your own uncertainty are its earliest record. The stakes are what make this early work worth doing. As Hayes and Krippendorff argue, reliability is the precondition for a study's findings to count for anything at all:

> "Conclusions from such data can be trusted only after demonstrating their reliability."
>
> Hayes and Krippendorff (2007, p. 77)

8.4 Manifest and latent content

Structured listening trains a particular kind of judgment, and naming it now will make the next chapter easier. Content analysis distinguishes between two layers of meaning in any message, a distinction Krippendorff (2018) places at the center of codebook design.

Manifest content is the surface: features that are explicitly present and that any trained coder can identify with high agreement. Does a chat message contain an `@` mention? Is it shorter than ten characters? Does it contain a known emote token? How many words is it? Manifest features are countable almost mechanically, and reliable coding of them is mostly a matter of a clear rule.

Latent content is the underlying meaning, the part that requires interpretation. Is a message friendly or hostile? Is it directed at the streamer or performing for the room? Is "first" a genuine claim or a running

joke? Is the mood of chat, taken as a whole, celebratory or restless? Latent features cannot be read off the surface. Two careful coders can disagree about them in good faith.

This is the precise reason structured listening comes before coding. You cannot code latent content reliably until you have built the interpretive framework that turns a subjective impression into a defensible judgment, and that framework is built by watching. The coder who has spent twenty hours on live Twitch knows that a wall of one emote after a big play is celebration, not noise. They learned it the way you are about to learn it: by watching it happen, again and again, until the pattern was obvious. Immersion is how latent content becomes codeable.

Why disagreement identification matters before the codebook. Most content-analytic codebook failures come not from poorly defined categories but from underspecified decision rules for boundary cases. Watch a Twitch stream and code a comment as "hostile," and another coder watching the same stream might code it as "sarcastic," because the category boundary was not drawn precisely enough. Structured immersion is when that ambiguity first becomes visible. Field notes should document not just *what* you observe but *where* you are uncertain, because those locations are exactly where a second coder will need explicit guidance. After completing your immersion log, identify three categories where you anticipate the most disagreement with a second coder, and for each write one candidate decision rule that would resolve the ambiguity.

8.5 From observation to sharper questions

Structured listening does more than prepare you to code. It tells you whether your research question survives contact with the medium.

You arrived at this chapter with a prospectus. Suppose it commits you, as the model prospectus in Chapter 6 did, to comparing chat that is directed at the streamer with chat that is broadcast to the room. Observation will do one of two things to that question. It may confirm it: you watch, and the directed-versus-broadcast distinction turns out to be real, visible, and worth measuring. Or it may complicate it. You may notice, watching chat closely, that a great deal of it is neither directed at the streamer nor broadcast to the room in general, but aimed at another viewer: a reply, an argument, an inside joke between regulars. The two-way distinction you committed to on paper turns out to have a third term in practice.

That is not a failure of the prospectus. It is the cheapest possible moment to learn something the prospectus could not have known. A distinction discovered now becomes a category in the codebook you build next chapter. A distinction discovered after coding becomes a reason to start coding over.

By the end of structured listening you should be able to state two or three **candidate research questions**, grounded in what you actually saw rather than in what you assumed before you watched. If your prospectus question came through observation intact, the candidates are sharper, more operationalizable versions of it. If observation showed that the prospectus question cannot be answered with what the medium offers, the candidates are its honest replacements. Either way, the question you carry into operationalization is a question observation has tested, not just one theory proposed.

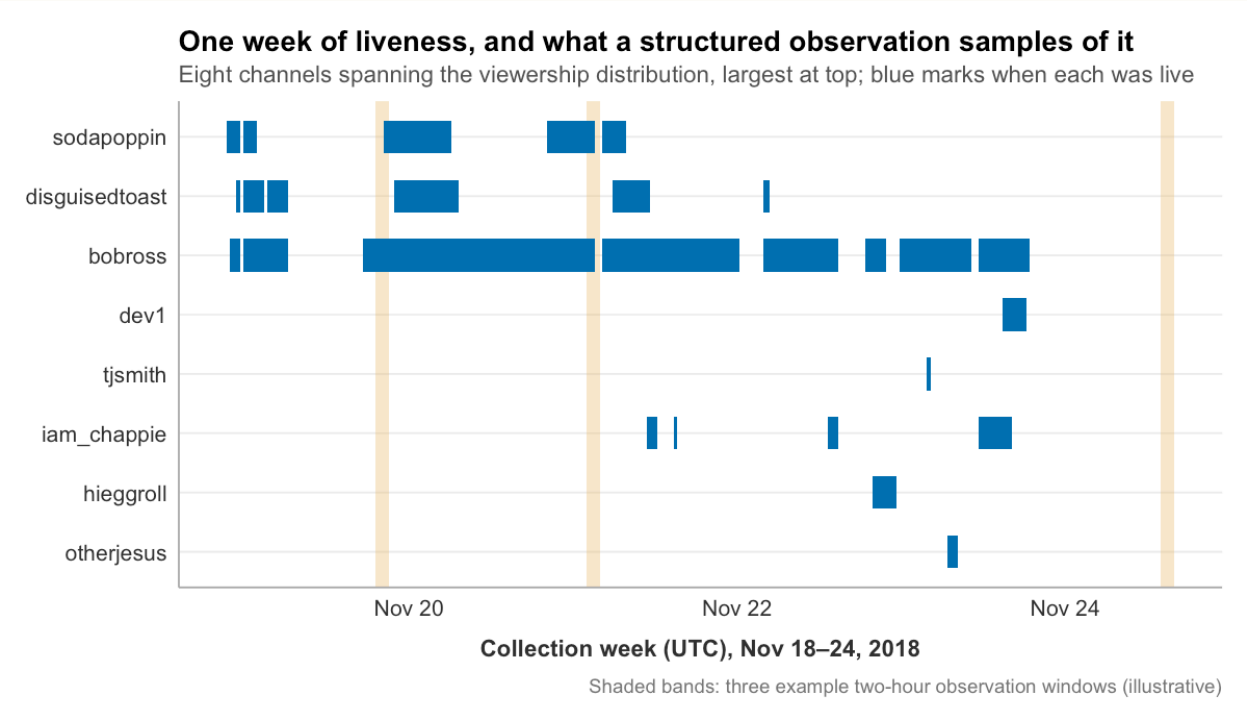
8.6 Sampling your observation

You cannot watch all of Twitch, and you do not need to. What you need is a **structured sample** of observation that reflects the range of the thing you will study.

Watching ten streams that are all large competitive-gaming channels will teach you a great deal about large competitive-gaming channels and mislead you about everything else. The Twitch dataset this book

uses spans a deliberate range: fifty channels drawn from across the platform's volume distribution, gaming and non-gaming, the heavily populated and the nearly empty. Your observation should span a comparable range. Watch big channels and small ones. Watch gaming streams and Just Chatting and art. Watch at different hours, because a stream at peak time and the same stream in a quiet hour are not the same room.

The goal is not volume but coverage. Twenty streams chosen to span the variety of the platform will teach you more than fifty streams that are all the same kind. This is the same principle that governed the literature search in Chapter 4, where the point of a wide search was to be sure you had seen the shape of the whole conversation, and it is the same principle that will govern statistical sampling in Chapter 10. Representativeness is a discipline that shows up at every stage of a study, and observation is the first stage it shows up in.



8.7 Documenting edge cases

Some of what you observe will resist every category you are tempted to draw. Those moments are not annoyances. They are the most useful thing observation produces, and they deserve their own record: an **edge-case log**, kept alongside your field notes.

An edge case on Twitch might be a message made entirely of emotes, with no words at all. Is that directed at the streamer, broadcast to the room, or something a content analysis of text should treat as a separate kind of object? It might be a copypasta, a block of text every regular recognizes and pastes without reading, which is technically a message but not in any normal sense something a viewer wrote. It might be a message in a language you do not read, or an obvious bot, or a message that is grammatically addressed to the streamer but is plainly a joke for the room. For each, log the case, what makes it ambiguous, and the possible ways a codebook could handle it.

These cases are not exceptions to be ignored. They are the raw material of the **decision rules** that make a codebook work. The model prospectus in Chapter 6 included an "unclassifiable" category; the edge-case

log is where you find out what actually lands in it. A codebook written without an edge-case log will meet these messages for the first time during coding, when it is expensive to handle them. A codebook written with one has already decided.

Planning for the reliability pilot. The formal intercoder reliability protocol (Chapter 8) requires a training phase and a reliability test on a separate data subset. The training phase has two coders apply the codebook independently to the same units, then reconcile their disagreements, and effective training requires the genuinely ambiguous cases you identified during immersion. Your field notes double as a training-set seed: the hardest-to-categorize observations become the calibration material. Krippendorff argues that unitizing (deciding what counts as a single unit of analysis) is a prior reliability problem most researchers underestimate; as you log edge cases, ask what your unit of analysis is and what makes consistent unitizing difficult.

8.8 Knowing when to stop

Immersion does not last forever. At some point structured listening gives way to structured coding, and the signs that you have reached that point are recognizable.

New streams stop surprising you. You can name three to five dimensions that clearly matter for your research question. Your edge-case log has enough entries to write decision rules from. Your field notes have begun to repeat themselves, confirming patterns rather than turning up new ones. This is the same idea Chapter 4 called **saturation**. There it meant that new literature searches stopped yielding new sources. Here it means that new observation stops yielding new patterns. In both cases saturation is not the claim that you have seen everything. It is the more modest, more useful claim that you have seen enough for the next step to rest on solid ground.

What structured listening leaves you with is four things: conceptual clarity about what your variables actually mean in this medium, a set of candidate categories for each one, the beginnings of the decision rules that will resolve ambiguous cases, and the contextual knowledge to recognize when a surface feature is about to mislead you. Those four things are precisely the raw material of a codebook. Assembling them into one is the next chapter's work.

8.9 Looking ahead

Chapter 8 is where vibes become variables, the move the whole book is named for. You have watched the medium and filled a field-notes document with what you saw. Chapter 8 takes that record and turns it into a codebook: a set of variables, each with defined values, each with a level of measurement, each with decision rules for the cases that do not sort themselves. It is the last step before the data, and the first step that the data will hold you to.

8.10 References

Geertz, C. (1973). *The interpretation of cultures: Selected essays*. Basic Books.

Katz, E., Blumler, J. G., & Gurevitch, M. (1973). Uses and gratifications research. *Public Opinion Quarterly*, *37*(4), 509-523. https://doi.org/10.1086/268109

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology* (4th ed.). SAGE Publications.

8.10.1 Graduate readings

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, *1*(1), 77-89. https://doi.org/10.1080/19312450709336664

9 Chapter 8: From vibes to variables

Listen in Dr. Leith's voice

Somewhere in the field-notes document you built in Chapter 7 is an entry like this one: "Chat in the art stream felt calmer and more conversational than chat in the gaming stream. People talked to each other, not just to the streamer." That sentence is a real observation. You earned it by watching. It is also, as written, useless to a coder.

"Calmer" is not a category anyone else can apply. "More conversational" is not something you can count. If you handed that field note to two people and asked them to sort a thousand chat messages by it, they would produce two different thousand-message sortings, and you would have no way to say which was right. The observation is true and the observation is unmeasurable, and the distance between those two facts is the distance this chapter closes.

The book is named for this chapter. **Operationalization** is the disciplined process of turning a vibe, a holistic and subjective impression, into a variable, an explicit and replicable measurement. It is the central methodological act of content analysis. Done well, it preserves what you noticed during immersion while making it something a stranger could measure the same way you did. Done badly, it produces numbers that look like findings and are actually artifacts. By the end of this chapter you will have a draft codebook: the document that turns your research question into a set of variables precise enough to code.

9.1 The operationalization gap

Take the variable the running example has been circling since Chapter 6: the target of a chat message, who it is aimed at. Chapter 7′s observation complicated the simple two-way split, so call the categories directed at the streamer, broadcast to the room, directed at another viewer, and unclassifiable.

A vague attempt at operationalizing this would be: "Code each message by who it is for." The problem surfaces the moment a coder meets a real message. A viewer types "same." Who is that for? A viewer types "LULW." A viewer types "no way that just happened." Each of these could plausibly be aimed at the streamer, at the room, or at another viewer, and a coder with only "code by who it is for" to go on will simply guess. Two coders will guess differently. The variable is named but not operationalized.

A better attempt specifies observable criteria. "A message is **directed at the streamer** if it contains an at-mention of the streamer's channel name, or is a second-person address ('you', 'your') that responds to something the streamer said or did in the preceding moments. It is **directed at another viewer** if it contains an at-mention of a non-streamer account, or is a reply to a specific prior message. It is **broadcast to the room** if it is a reaction or comment with no specific addressee. It is **unclassifiable** if none of these criteria settle the question." Now a coder has something to apply. The criteria are observable, the categories are defined, and two coders working from them will land in the same place far more often than two coders working from a vibe.

That is the work. Operationalization specifies, in advance and in writing, exactly what observations count as evidence for a concept. It is not unique to Twitch. A scholar coding immigration coverage must turn "this article feels sympathetic" into a defined frame variable with observable criteria. A health researcher must turn "this ad feels fear-based" into a category a coder can apply. The medium changes; the act does not.

9.2 Conceptual and operational definitions

Every variable needs two definitions, working together.

A **conceptual definition** says what the variable means in the abstract. It draws on your reading and your theory, and it answers one question: what is this variable meant to capture? For message target, a conceptual definition might read: "The intended addressee of a chat message, reflecting whether the message functions as participation with the streamer, with the broader chat collective, or with a specific other viewer." That is clear about the idea. It is not yet a recipe.

An **operational definition** is the recipe. It says exactly what a coder does to assign a value, in enough detail that a stranger could follow it and measure the same thing you measured. The criteria in the previous section, the at-mention rule, the second-person-address rule, the reply rule, are the operational definition of message target. The conceptual definition tells you what the variable is for. The operational definition tells you what counts as evidence.

You need both, and they discipline each other. A conceptual definition with no operational definition is an idea you cannot measure. An operational definition with no conceptual definition is a procedure that has lost track of why it exists, which is how a study ends up precisely measuring something nobody wanted to know. Writing the two side by side, for every variable, is the first concrete step toward a codebook.

9.3 How operationalization goes wrong

Four failures recur often enough to be worth naming, because naming them is the fastest way to catch them in your own work.

The first is **conflating concepts**: treating two related but distinct ideas as if they were one. A study that measures a streamer's community by counting chat messages has conflated community with activity. A large channel can have a fast chat that is also shallow and anonymous, thousands of strangers reacting in parallel and never to each other. Volume is real, but it is audience activity, not community, and the fix is to say so plainly: either defend message volume as a deliberate, narrow proxy or measure community more directly.

The second is the **unjustified proxy**: using an indirect measure without explaining why it stands in for the thing you care about. Measuring stream quality by viewer count is the standard example. Viewer count reflects discoverability, time of day, the popularity of the game being played, and a channel's existing momentum, none of which is quality. A proxy is not forbidden, but an undefended one is, and the fix is to defend it or replace it.

The third is **oversimplification**: collapsing a concept with several dimensions into a single indicator that misses most of it. Chat engagement is not only how many messages appear. It is also who is talking, whether they are talking to each other, and in what spirit. A message count captures one dimension and silently discards the rest. The fix is either to measure the other dimensions too or to state honestly that the study addresses message volume specifically, not engagement in full.

The fourth is the **unmeasurable definition**: an operational definition that cannot actually be carried out. "A message is sincere if the viewer genuinely means it" defines sincerity in terms of something no coder can observe. The fix is to operationalize through observable indicators, the surface features that, taken together, are the best available evidence for the thing you cannot see directly.

The thread running through all four is the same. An operational definition is a promise that someone else could follow your recipe and measure what you measured. Each mistake breaks that promise in a different way.

9.4 Levels of measurement

Once a variable is operationalized, it has a **level of measurement**, and that level determines what you are later allowed to do with it. The four levels are easy to remember as **NOIR**: nominal, ordinal, interval, ratio.

The `stream_log` table is a convenient place to see all four, because its columns happen to span the range. Recall the preview from Chapter 2: channel, title, game, viewers, date.

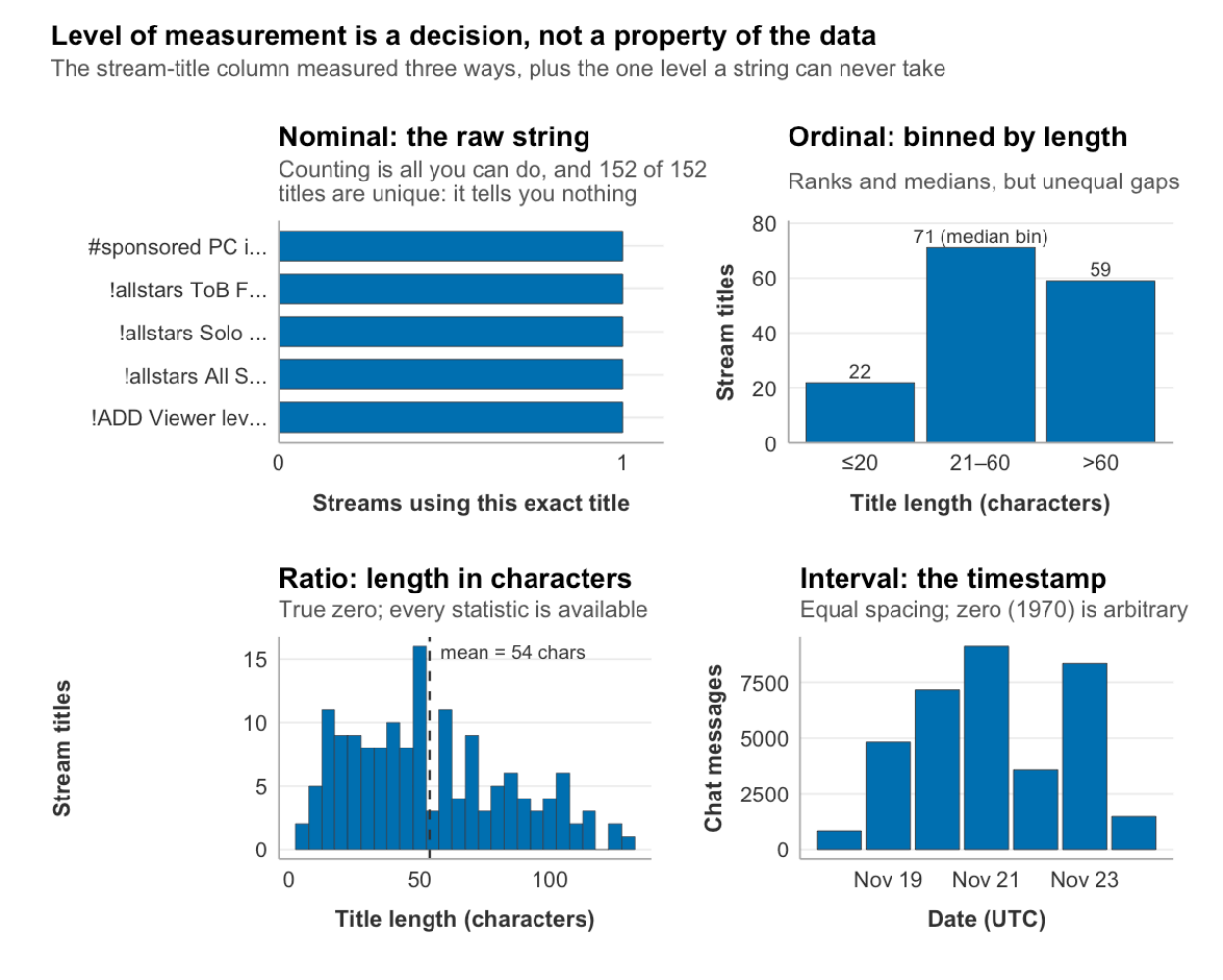
A **nominal** variable sorts cases into categories that have no inherent order. The `game` column is nominal: Fortnite, Just Chatting, Art, Hearthstone are categories, and no arithmetic relates them. Fortnite is not greater than Art. The `channel` column is nominal too, and so, in its raw form, is `title`: a stream title is a string, and one string is not ranked above another. With a nominal variable you can count how often each category appears and identify the most common one, but you cannot average it. The average of Fortnite and Art is not a category.

An **ordinal** variable sorts cases into categories that do have an order, but where the distance between ranks is not constant. If you took stream titles and binned them by length into short, medium, and long, length-bin would be ordinal: long outranks short, but the gap from short to medium is not guaranteed to equal the gap from medium to long. With an ordinal variable you can rank cases and find the median, but the mean is not strictly meaningful.

An **interval** variable has equal, constant distances between values, but its zero point is arbitrary rather than meaning "none." The `date` column is the example, and it carries a caveat worth stating now because Chapter 11 will spend real effort on it. The dataset stores `date` as a bigint count of milliseconds since midnight on January 1, 1970. Once converted to a timestamp, it is interval-level: the distance from one minute to the next is constant across the whole column, but the zero point, 1970, does not mean "no time." It is a convention. You can meaningfully subtract two timestamps to get a duration. You cannot meaningfully say one timestamp is "twice" another.

A **ratio** variable has equal intervals and a true zero that genuinely means "none." The `viewers` column is ratio: zero viewers means no one is watching, and because that zero is real, a stream with 28,000 viewers genuinely has twice the audience of one with 14,000. Ratio variables permit the full range of arithmetic. Most counts are ratio-level: viewers, message length in characters, number of messages.

Notice what happened with stream title. As a raw string it is nominal. Measured as a character count it is ratio, because a title can have zero characters and a true zero exists. Binned into short, medium, and long it is ordinal. One column, three possible levels, depending entirely on how you operationalize it. Level of measurement is not a property the data hands you. It is a property of the decision you make.



Levels matter because they govern the statistics available to you later. A nominal variable can be tested for association with a chi-square test. A ratio variable can be averaged, correlated, and entered into the kind of test Chapter 13 runs. If you operationalize an interesting variable at the nominal level when it could have been ratio, you have quietly narrowed what your study can conclude before you have coded a single case.

9.5 Two kinds of variable

The `stream_log` columns make NOIR concrete, but they also illustrate something about where variables come from, and the distinction matters for what comes next.

Some variables arrive in the dataset ready-made. `viewers` is already a number in a column. `game` is already a category. For variables like these, the operationalization work is mostly classification: decide the level of measurement, decide whether you will use the column raw or transform it, and move on. There is no coding to do, because the platform already did it.

Other variables do not exist until you build them. Message target is not a column in `chat_log`. Nothing in the dataset records whether a message was aimed at the streamer or the room. That variable exists only after a human coder reads each message and assigns a value, and a coder can only do that consistently if

you have given them a precise, written set of instructions. For variables like these, the operationalization work is most of the work, and the instructions have a name. They are a codebook.

The rest of this chapter is about the second kind. But first, the standard those instructions are built to meet.

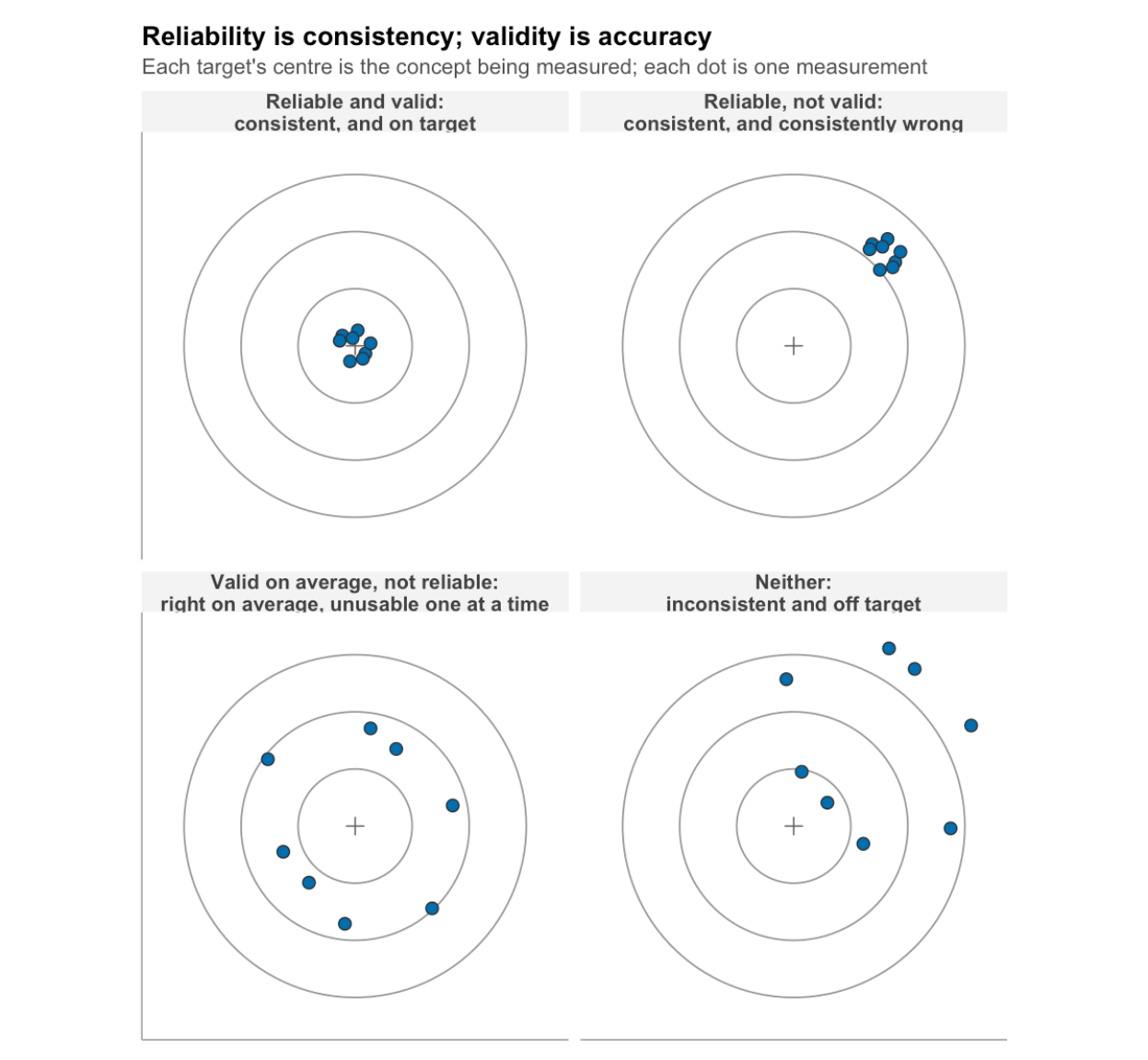
9.6 Reliability and validity

An operational definition can be perfectly precise and still be bad measurement. Two criteria tell you whether your measurement is any good: reliability and validity.

Reliability is consistency. A measure is reliable if it produces the same result under the same conditions. A bathroom scale that reads 150, then 162, then 147 as you step on and off without changing is unreliable, and its readings are worthless regardless of your actual weight. For content analysis, the form of reliability that matters most is **inter-coder reliability**: if two trained coders apply your codebook to the same messages independently, do they agree? If they do not, the codebook is not yet measuring anything stable, and no amount of analysis downstream will repair that. Inter-coder reliability is quantified with statistics, Cohen's kappa and Krippendorff's alpha among them, that correct for the agreement two coders would reach by chance alone. Chapter 10 covers those statistics and how to act on them; for now, the point is conceptual. A codebook's job is to make independent coders agree.

Validity is accuracy. A measure is valid if it captures the thing it claims to capture. Validity and reliability are independent. A scale calibrated ten pounds light is perfectly reliable, it reads the same every time, and perfectly invalid, it is wrong every time. The reverse can also happen: a measure can be accurate on average but so inconsistent that any single reading is unusable. Validity has several forms worth knowing by name. **Face validity** asks whether the measure looks, on inspection, like it assesses the intended concept. **Content validity** asks whether it covers all of the concept and not just a convenient slice. **Construct validity**, the most demanding, asks whether the measure behaves the way theory says it should, relating to other variables as predicted.

One principle ties the two together and is worth holding onto: reliability is necessary but not sufficient for validity. A measure that is inconsistent cannot be accurately capturing anything, so unreliability rules validity out. But a measure can be perfectly consistent and still measure the wrong thing. You need both, and you get reliability first, because a codebook is the instrument that produces it.



The formal intercoder reliability protocol. Your codebook is the operational heart of the study, but a codebook that works for one coder is not necessarily a reliable instrument: it is not finished until a second coder has applied it to a sample of your data and the agreement has been measured and reported. A complete V2V reliability protocol has four steps. *First*, the second coder receives the codebook (no walk-through, no hints) and codes a training set of 20–30 units independently. *Second*, the two coders compare their coding and discuss every disagreement; no new production data is coded at this stage. *Third*, the codebook is revised to add decision rules for all identified disagreements. *Fourth*, the second coder codes a fresh reliability sample, separate from the training set, independently, and the agreement on this fresh sample is the κ that gets reported. Lombard et al. (2002) recommend reporting more than one reliability statistic; for your own design, decide which statistic you would report alongside κ, and why. Hayes and Krippendorff (2007) state the standard this protocol exists to meet in a single line:

> "Conclusions from such data can be trusted only after demonstrating their reliability."

Hayes and Krippendorff (2007, p. 77)

9.7 The codebook

A **codebook** is the complete set of instructions for coding your data. It is not paperwork you assemble after the fact to satisfy a methods requirement. It is the instrument itself.

The most useful way to think about a codebook is as source code. If your coding process were a computer program, the codebook would be the program: it specifies every operation, handles every conditional, and ensures that two processors, here two human coders, running the same program on the same input produce the same output. Without it, you are coding from memory and intuition, and memory and intuition are exactly what reliability requires you to eliminate. Neuendorf (2017) describes the codebook as the heart of a content analysis, and the description is not decorative. The clarity of the codebook is the single strongest predictor of whether your coders will agree.

A complete codebook has five parts.

The **unit of analysis** states exactly what one coded case is. This must be unambiguous. "Code the chat" is not a unit of analysis. "The unit of analysis is a single chat message, defined as one row of the `chat_log` table: one sender, one message string, one timestamp" is. If the unit is vague, every count you later report is vague.

The **variables and their categories** list what you are measuring and what values each variable can take. For each variable you give the conceptual definition, the operational definition, every category, and a short description of each category.

The **decision rules** tell coders what to do with the cases that do not sort themselves. This is where the edge cases from your Chapter 7 observation log earn their keep. Every ambiguous case you wrote down becomes a rule.

The **examples** give two or three prototypical cases for each category, so coders have a reference for what good coding looks like and a set of cases to train on.

The **special cases** section records recurring complications. It is usually empty in a first draft and fills in as you pilot the codebook, which is Chapter 10's work.

Two principles govern the categories of any variable. They must be **exhaustive**, meaning every case can be coded, which is what the "unclassifiable" or "other" catch-all guarantees. And they must be **mutually exclusive**, meaning every case fits exactly one category. If a message can honestly be placed in two categories at once, the categories overlap, and overlap destroys reliability because two coders will split on it forever. The fix is either sharper category definitions or a decision rule that assigns precedence. A catch-all is necessary, but if more than roughly one case in ten lands in it, the category scheme is incomplete and needs work.

9.8 A codebook for the chat study

Here is a draft codebook for the running example, the study the prospectus in Chapter 6 committed to and the observation in Chapter 7 sharpened. It has three variables, which is the floor for a workable study, and it is deliberately a draft: the special-cases section is thin because piloting has not happened yet.

Codebook: chat message target and form Unit of analysis. A single chat message, defined as one row of `chat_log`: one sender, one message string, one timestamp. Each message is coded independently of the messages around it, except where a decision rule explicitly says otherwise.

Variable 1: message target. Conceptual definition: the intended addressee of the message. Operational definition: after reading the message, the coder assigns one category. Categories: **directed at streamer** (contains an at-mention of the streamer's channel name, or is a second-person address responding to the streamer); **directed at another viewer** (contains an at-mention of a non-streamer account, or is a reply to a specific prior message); **broadcast to the room** (a reaction or comment with no specific addressee); **unclassifiable** (none of the above criteria settle the question). Level: nominal.

Variable 2: message length. Conceptual definition: the verbal extent of the message. Operational definition: the count of characters in the message string, including spaces and emote tokens. This is a derived measure, computed rather than judged. Level: ratio.

Variable 3: contains emote. Conceptual definition: whether the message uses Twitch's emote vocabulary. Operational definition: coded yes if the message contains at least one token from the project's emote list, no otherwise. Level: nominal.

Decision rules. Rule 1, precedence: if a message both at-mentions the streamer and performs for the room, code it directed at streamer; the explicit address takes precedence. Rule 2, emote-only messages: a message consisting only of emote tokens is coded broadcast to the room for target unless it contains an at-mention. Rule 3, copypasta: a recognized copypasta block is coded by its content like any other message; its status as copypasta is not itself a category. Rule 4, non-English messages: code target if the addressee is determinable from at-mentions or structure; otherwise code unclassifiable. Rule 5, bot accounts: messages from known bot accounts are coded unclassifiable for target and noted for possible exclusion.

Three variables, spanning two manifest measures a coder can apply almost mechanically and one latent measure that requires real judgment, spanning nominal and ratio levels, every category exhaustive and mutually exclusive, every decision rule traceable to something observation actually turned up. That is a draft codebook. It is enough to start.

The threshold and the calculation. Before production coding begins, the agreement between coders has to clear a bar: the standard threshold for Cohen's κ is ≥ 0.70 (Landis & Koch, 1977), and below it the codebook requires further revision. The reliability sample should contain at least 50 coded units; for variables with low base rates (rare codes), 100+ units are preferred. `v2v::reliability()` computes κ with interpretation labels. For your own study, simulate the scenario where two coders agree on 40 of 50 units: run `v2v::reliability()`, read the κ it returns, decide whether this codebook clears the 0.70 threshold, and work out what you would revise if it does not.

9.9 Looking ahead

Chapter 9 is first contact with the data inside R. You have a research question, a theoretical frame, a prospectus, a field-notes document, and now a draft codebook. What you have not yet done is open the dataset in the tool you will analyze it with. Chapter 9 is where the planning stops and the code begins: you will load `chat_log` and `stream_log`, look at their structure, and confirm that the data you have designed a study around is the data you actually have.

9.10 References

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology* (4th ed.). SAGE Publications.

Neuendorf, K. A. (2017). *The content analysis guidebook* (2nd ed.). SAGE Publications.

9.10.1 Graduate readings

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, *1*(1), 77–89. https://doi.org/10.1080/19312450709336664

Lombard, M., Snyder-Duch, J., & Campanella Bracken, C. (2002). Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research*, *28*(4), 587–604. https://doi.org/10.1111/j.1468-2958.2002.tb00826.x

10 Chapter 9: The rulebook and first workspace

Listen in Dr. Leith's voice

Here is a strange thing about the work of the last eight chapters. You have a research question, a theoretical frame, a prospectus, a field-notes document, and a draft codebook. You have, in other words, designed an entire study around the Twitch dataset. And you have never actually seen it.

You have been told what it contains. Chapter 1 described a collection of nearly 1,700 channels recorded across just under six days of November 2018, Bob Ross streaming under Art while most of the platform played games. Chapter 2 showed ten rows of it. You have taken the rest on trust. This chapter is where the trust ends and the looking begins. By the end of it you will have typed a few lines of R, and the dataset will have stopped being a description and become an object sitting in front of you, one you can inspect column by column.

This is also the chapter where the book changes character. For eight chapters there has been no code, on purpose. Planning a study before writing a line of it is the discipline that makes the code worth writing. But the planning is done. From here forward the work is done in R, and Chapter 9 is the gentle start: it opens a workspace, it loads the data, and it looks. It runs no analysis and computes no result. First contact is its own task, and it is worth doing slowly.

10.1 The workspace

Before the data, the workspace. Chapter 2 argued that reproducible research lives or dies on traceability: a study someone else can rerun is a study built from files in known places, not from commands typed once and forgotten. A **project** is how that happens. It is a single folder that holds everything one study needs, the data, the code, the codebook, the written report, so that the whole study travels together and nothing depends on a file that lives only on your laptop.

The `v2v` package scaffolds one for you. A single call does it:

```
v2v::new_portfolio("twitch-portfolio")
```

Note the `v2v::` in front of the function name. It is not decoration. It tells R, and tells you, that `new_portfolio()` comes from the `v2v` package specifically, the set of tools this book ships. Every `v2v::` call in the chapters ahead carries that prefix for the same reason: so that the package never becomes a hidden assumption, and you always know which tools are the book's and which are R's own.

Running `new_portfolio()` creates a folder with a structure like this:

```
twitch-portfolio/
├── twitch-portfolio.Rproj
├── _quarto.yml
├── README.md
├── codebook.qmd
├── analysis.qmd
├── data/
└── .Rprofile
```

You did not have to decide where files go, what to name them, or how to wire them together. That is the point. Where to put things is a real source of friction for someone opening a data project for the first time, and an hour lost to it is an hour not spent on the study. The scaffold removes the decision.

One more call confirms the workspace is sound:

```
v2v::setup()
```

`setup()` checks that R, the packages this book depends on, and the project's environment are all present and the versions agree. You met it once already, in Chapter 2, as the final step of installing the stack. It returns here as a habit worth keeping: run it when you open a project, and you find out about a missing piece before it interrupts you mid-analysis.

10.2 The rulebook becomes a document

Look again at the scaffold. One of the files is `codebook.qmd`.

In Chapter 8 you drafted a codebook on paper: a unit of analysis, three operationalized variables, a set of decision rules. That draft was an argument about how the study would measure things. Now it becomes a document in the workspace, the project's **rulebook**, the reference every later step answers to. Move the Chapter 8 draft into `codebook.qmd`, give it a version line and a date, and render it the way Chapter 2 had you render a one-page Quarto document. Rendering a codebook is not busywork. A codebook that renders cleanly is a codebook with no broken structure, and a rendered codebook is one a second coder can actually read.

Calling this "finalizing" the codebook is slightly optimistic, and it is worth being honest about that. Chapter 10 will pilot the codebook, two coders applying it to the same messages, and piloting routinely sends you back to edit a category or sharpen a rule. What Chapter 9 finalizes is not the content of the rulebook but its status: it stops being a sketch in your notes and becomes the official instrument of the project, versioned, rendered, and ready to be tested. A draft you can revise is exactly what you want walking into a pilot.

10.3 First contact with the data

Now the data. Two lines bring it in.

```
library(tidyverse)

chat <- v2v::twitch_chat()
streams <- v2v::twitch_streams()
```

The first line loads the tidyverse, the collection of R packages this book uses for everything from inspecting data to drawing figures (Wickham et al., 2019). The next two lines load the dataset itself. `v2v::twitch_chat()` returns the chat data; `v2v::twitch_streams()` returns the stream data. The arrow, `<-`, is R's way of giving the result a name, so `chat` and `streams` now refer to the two tables for the rest of the session.

Notice that the data arrives through a function call, not as a file you opened or a spreadsheet you imported. This is deliberate. A new R user's first encounter with data is often a pile of path errors, the file is somewhere the code cannot find, the format is not what was expected. `twitch_chat()` and `twitch_streams()` are written to remove that entire category of problem: they reach into the `v2v` package, find the data that ships inside it, and hand it back, with no path for you to get wrong. Called with no arguments, as above, each returns the full working sample. They also accept arguments for later, when you want only certain channels or games, but first contact is the no-argument version. Load everything, and look.

It is worth being clear about what "everything" means here. This is not the entire 2018 collection. The full dump runs to tens of millions of rows and well over a gigabyte, too much to ship inside a package or to open comfortably on a laptop. What `v2v` ships, and what you just loaded, is a working sample drawn from the fifty-channel corpus Chapter 1 described: a subset large enough to be real and small enough to be fast. Every example in the chapters ahead runs against this sample.

A word about what happens when this does not work. Sooner or later a line of R you type will return an error instead of a result: a misspelled function name, a package not loaded, a stray bracket. This is normal. It happens to everyone who writes code, and it is not a sign that you are unsuited to the work. The V2V Hub keeps a common-errors cheat sheet for exactly these moments, listing the errors a new R user meets most often and what each one is telling you. From this chapter forward, when a line breaks, that cheat sheet is the first place to look. An error is information, not a verdict.

Data provenance as a methodological disclosure. First contact is a moment of exploration (load, inspect, describe), but it rests on a prior step: documenting the *provenance* of the data before the first line of analysis code runs. When you publish a study, you are making a claim about specific data collected under specific conditions at a specific time. "I collected Twitch chat data" is not a methods section. "I collected 14 days of Twitch chat from the top 10 most-viewed streams in the *Just Chatting* category, using Twitch API v2, between 2024-01-15 and 2024-01-28, saving raw JSON responses to `data/raw/` before any processing" is. The difference matters because replication requires exact knowledge of what was done. Munafo and colleagues describe what goes missing when a study is discoverable but its methods are not legible:

> "Poor usability reflects difficulty in evaluating what was done, in reusing the methodology to assess reproducibility, and in incorporating the evidence into systematic reviews and meta-analyses."
>
> Munafo et al. (2017, p. 4)

For the v2v package dataset, write a four-sentence data provenance statement that would satisfy a journal methods section, and note what information you would need that the package documentation does not provide.

10.4 Looking at the data

The data is loaded. The first question is simply what it looks like, and the tidyverse function for that is `glimpse()`, which prints every column of a table, its type, and its first few values.

```
glimpse(chat)
```

```
Rows: 35,267
Columns: 5
$ id      <int> 89, 551, 1033, 2094, 2803, …
$ channel <chr> "sodapoppin", "xqcow", "forsen", "sodapoppin", "xqcow", …
$ sender  <chr> "madzee", "prometheanow", "spectre155", "itsjer", "laserfru…
$ message <chr> "??????????????????????", "TriEasy Clap TriEasy Clap TriE…
$ date    <dbl> 1542578127023, 1542578135562, 1542578143610, 1542578157677…
```

Five columns, 35,267 rows, and each row is one chat message. `id` is a number that uniquely labels the message. `channel` is the channel the message was sent in. `sender` is the account that sent it. `message` is the text itself. `date` is when it was sent.

Two things in that output reward a closer look. The first is `message`. The values are real, and they are not tidy. The first message is twenty-three question marks. The second is the word "TriEasy Clap" repeated over and over, a viewer pasting the same phrase many times. These are not errors in the data. They are what Twitch chat actually contains, and your codebook will have to meet them as they are. The second is `date`. It is a column of enormous numbers, marked `<dbl>`, R's label for a numeric value. Chapter 8 classified this column as interval-level and flagged the catch: the number is a count of milliseconds since 1970, not a date anyone can read. Converting it into a real timestamp is a headline lesson of Chapter 11. For now it is enough to notice that the column is there and that, as it stands, it is unreadable.

The stream table is the same kind of object with different columns.

```
glimpse(streams)
```

```
Rows: 32,276
Columns: 6
$ id      <int> 4, 10, 18, 61, 105, …
$ channel <chr> "sodapoppin", "xqcow", "forsen", "giantwaffle", "sodapoppin…
$ title   <chr> "Hello truckers\n44835158804929_2115841973", "…", …
$ game    <chr> "Marble It Up!", "Just Chatting", "Artifact", "Rocket Leag…
$ viewers <int> 27934, 19381, 15300, 4697, 28076, …
$ date    <dbl> 1542578200214, 1542578200240, 1542578200273, 1542578200423…
```

Six columns, 32,276 rows, and each row is one snapshot of one stream at one moment: its `title`, the `game` it was filed under, its `viewers` count, and the `date` of the snapshot. The first row should look familiar. It is Sodapoppin's channel, playing Marble It Up!, with 27,934 viewers, the exact row Chapter 2 used to open its discussion of reproducible data. The fifth row is the same channel a few minutes later at 28,076 viewers, the climb Chapter 2 asked you to interpret. You are now looking at the source of that example directly.

The `title` column shows the same messiness as chat. The first title has a line break inside it and a long string of digits, which is not what a tidy title looks like but is what some streamers actually type. Real data is like this, and noticing it now, before any analysis, is the entire reason for first contact.

Beyond seeing the columns, you can count things. The tidyverse verb `count()` tallies how often each value of a column appears. Asking it about `game` shows what the streams are filed under:

```
streams %>% count(game, sort = TRUE)
```

```
# A tibble: 60 × 2
   game                                n
   <chr>                           <int>
 1 Art                              5298
 2 Fortnite                         3682
 3 Just Chatting                    2474
 4 Hearthstone                      2157
 5 Pokemon: Let's Go, Pikachu!/Eevee!  1967
 6 League of Legends                1952
 7 Heroes of the Storm              1473
 8 Counter-Strike: Global Offensive    1346
 9 Rocket League                    1321
10 World of Warcraft                 920
# ℹ 50 more rows
```

The `%>%` is the tidyverse **pipe**: it takes the thing on its left and feeds it to the function on its right, so the line reads as an instruction in order, take `streams`, then count its games. The result is itself a small table, sixty rows, one per game, sorted with the most frequent at the top. Sixty categories is more variety than a study can examine one by one, and the count makes the shape of that variety visible: a handful of games account for most of the snapshots, and a long tail of fifty more accounts for the rest. That shape will matter in Chapter 12, when you have to decide how to draw a chart that does not drown in sixty lines.

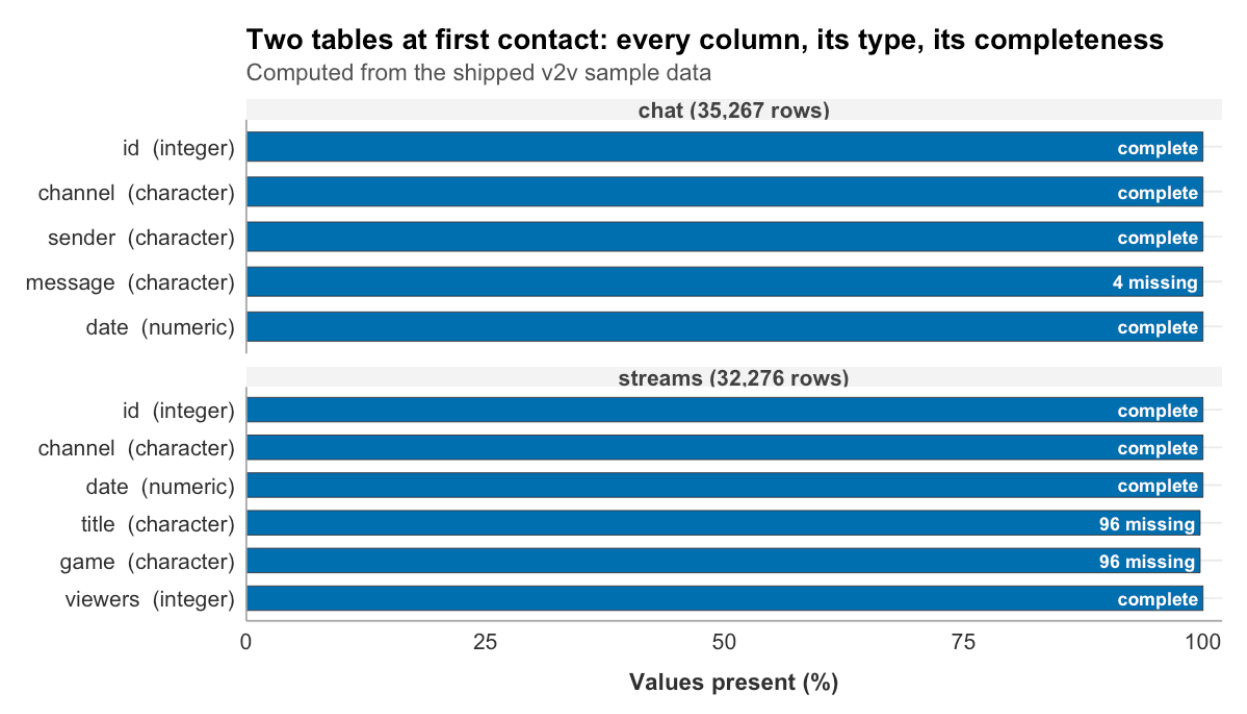
`count()` is also a quick way to confirm the data is the corpus you expect. Chapter 1 described a working corpus of fifty channels. A related function, `n_distinct()`, counts how many different values a column holds, and asking it about `channel` checks that claim directly:

```
n_distinct(chat$channel)
```

```
[1] 50
```

Fifty distinct channels in the chat table, and `n_distinct(streams$channel)` returns the same fifty. The data you loaded is the corpus the book has been describing since Chapter 1, no more and no less.

There is one further way to look, worth knowing even though this chapter does not lean on it. Typing a table's name on its own, just `chat`, prints its first rows in a compact form. Running `View(chat)` opens the table in a spreadsheet-style viewer you can scroll. Of the options, `glimpse()` is usually the most useful for seeing a table's shape at a glance, but printing, `View()`, `glimpse()`, and `count()` are all doing the same basic thing: looking, without changing anything.



None of this changed the data. Looking is the whole of what this chapter has done with it. The chat table and the stream table are exactly as they were loaded. Inspection comes before transformation, and Chapter 9 stays entirely on the inspection side of that line.

10.5 Reading the data against the codebook

First contact has one job left, and it is the most important one. Open the codebook next to the `glimpse()` output and ask a blunt question: does this data actually support the study the codebook describes?

Walk the three variables. **Message target** is to be coded from each message, using the message text and, for at-mentions and replies, the sender information. The chat table has a `message` column and a `sender` column. Supported. **Message length** is the character count of the message string. There is a `message` column. Supported. **Contains emote** is read from the message text. There is a `message` column. Supported. The study the codebook designed is a study this data can carry. That is not a small thing to confirm, and it is far better to confirm it now than to discover a missing column halfway through coding.

First contact also surfaces what the codebook will have to work harder at. The messages in the `glimpse()` output are exactly the edge cases Chapter 7′s observation log anticipated and Chapter 8′s decision rules were written for. The message of twenty-three question marks is a message with no obvious target, the kind Rule 1 and the “unclassifiable” category exist to catch. The repeated “TriEasy Clap” is copypasta, which Rule 3 already addresses. Short tokens like the ones you can expect throughout the chat are emote-only messages, the subject of Rule 2. The codebook did not anticipate these in the abstract. It anticipated them because Chapter 7 had you watch for them, and here they are, in the first five rows.

One genuine wrinkle is worth carrying forward. Among the first messages is one addressed to a streamer: it begins “@xQcOW”. The `channel` column for that same row reads “xqcow”. The at-mention and the channel name are the same name in different casing, the display name as a viewer typed it against the lowercase login name the data stores. The message-target rule that keys on “an at-mention of the

streamer's channel name" will have to account for that difference. This is precisely the kind of refinement that belongs in the codebook now, while it is still the project's living rulebook and before a single message has been coded. First contact has done its work: the study is buildable, and the codebook knows a little more than it did.

What the provenance record includes. For studies using the provided v2v package dataset, provenance is embedded in the package and stable. For studies using fresh API data, the record must include: (1) data source and API version; (2) date and time range of collection; (3) any filters, keywords, or channel selectors applied; (4) format in which raw data was saved; and (5) the git commit at which raw data was frozen (tagged as immutable). Item 5 connects directly to the audit trail: never re-run data collection to overwrite a file that a downstream script already depends on. If a journal reviewer asked you to share the raw data for a study using the v2v package dataset, consider what you would share and what you would withhold.

10.6 Looking ahead

You have a workspace, a rulebook installed in it, and the data loaded and inspected. Chapter 10 puts the rulebook to the test. It draws a sample of chat messages, the specific messages your study will actually code, and asks the question every content analysis has to answer before its results mean anything: when two coders apply the codebook independently, do they agree? Sampling and inter-coder reliability are the subject, and the codebook you finalized here is what goes on trial.

10.7 References

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

10.7.1 Graduate readings

Munafo, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., & Ioannidis, J. P. A. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, *1*, 0021. https://doi.org/10.1038/s41562-016-0021

Part IV: Execution

11 Chapter 10: The sample

Listen in Dr. Leith's voice

Suppose you sat down to code the chat fixture by hand, one message at a time, the codebook open beside you. Read the message, decide who it is aimed at, record the code, move to the next. At a brisk and probably unsustainable pace of one message every ten seconds, the 35,267 messages in the fixture would take just under a hundred hours. The full 2018 collection, all twenty-two million messages, would outlast your degree.

So you will not code all of it. You will code a **sample**: a smaller set of messages, drawn carefully enough that what you learn from coding it holds for the larger collection it came from. Drawing that sample well is the first half of this chapter.

The second half is a test. The codebook you finalized in Chapter 9 is an argument about how messages should be classified, but an argument on paper is not yet a working instrument. A codebook earns that status only when a second person can pick it up, apply it to the same messages you would, and reach the same codes. Whether your codebook clears that bar is not something you can know by reading it. You have to test it. This chapter draws the sample and runs the test, and the test is the part that can send you back to Chapter 8.

11.1 Why you sample

Chapter 1 drew a distinction between a population and a sample, and it is worth recovering here because the rest of the chapter rests on it. The **population** is the full set of cases you want your conclusions to be about. For the chat study, that is something like all chat messages sent in the fifty-channel corpus during the collection week. The sample is the subset you actually examine.

Sometimes you do not need a sample. Coding every case is called a **census**, and it is the right move when the population is small enough to handle whole. The stream fixture is close to this: at roughly 32,000 snapshots it ships in its entirety, no sampling involved, because the snapshot count for fifty channels is small enough to keep all of it.

Chat is the opposite situation. There is far too much of it to code by hand, and so a census is off the table. This is the ordinary condition of content analysis. The point of a sample is to make the work finite without making the conclusions worthless, and that depends entirely on how the sample is drawn. A sample of 1,500 messages can support a sound study or a misleading one. The difference is method.

11.2 Why a simple random sample is not enough

The most familiar way to draw a sample is **simple random sampling**: give every message in the population an equal chance of selection and draw the number you need. It has one large virtue. It is unbiased, in the precise sense that no message is favored over any other, so the sample carries no systematic tilt built in by the researcher.

For Twitch chat, that virtue is not enough, and the reason is the shape of the data. Chat volume is not spread evenly across channels. It is severely concentrated. In the full population the busiest channel logged more than a million messages during the collection week; the median channel logged around 1,440; the quietest logged one. A distribution like that is called **right-skewed**, a small number of enormous values and a long tail of small ones, and it has a hard consequence for sampling. Draw messages at random from that population and the overwhelming majority will come from the handful of giant channels, because

that is where the overwhelming majority of messages are. The small and middling channels, which is to say most of Twitch, would barely appear in the sample at all.

The same problem shows up along a second dimension. Forty-three of the fifty channels in the corpus are gaming channels. Only a handful are not: Bob Ross's painting channel is the clearest case. A simple random sample of messages would be drawn almost entirely from gaming chat, simply because that is where almost all the messages are. For a study whose central comparison is gaming against non-gaming chat, that is close to fatal. The sample would have plenty of one side of the comparison and almost nothing of the other.

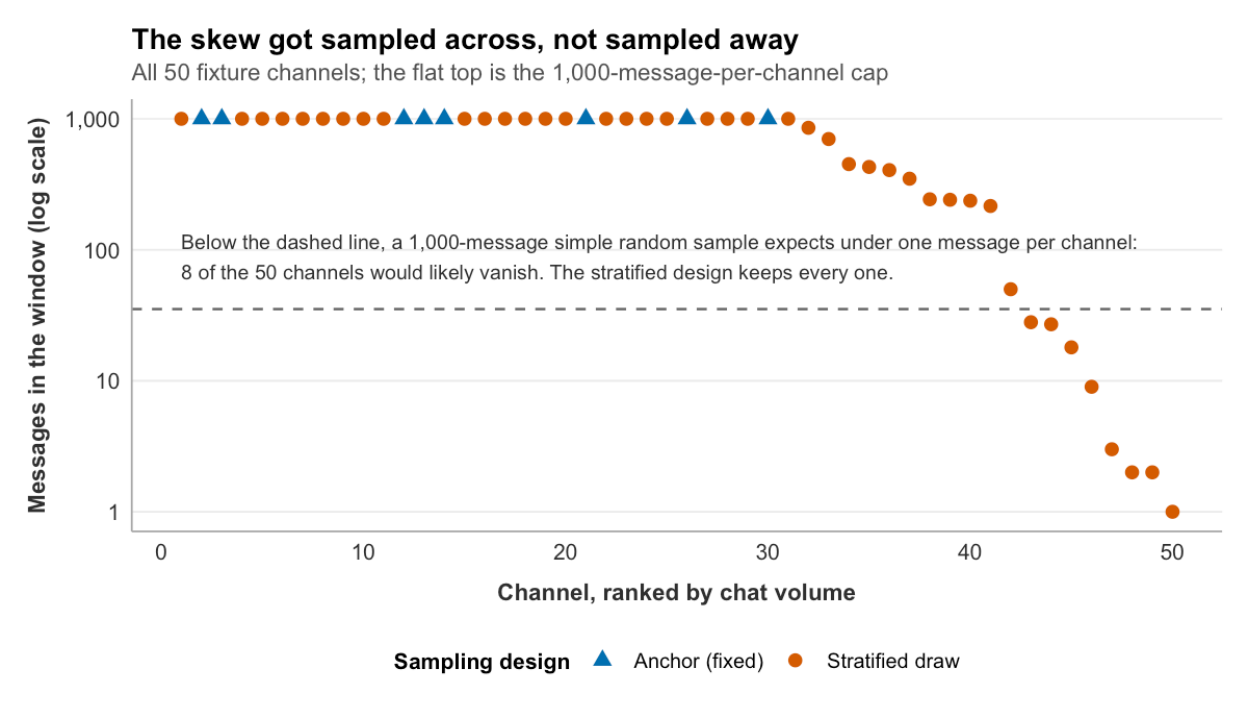
This is the trap of simple random sampling on skewed data. It is unbiased and still unrepresentative of the things you care about. It faithfully reproduces the population's imbalance at sample size, when what the study needs is enough of every group to compare them.

11.3 Stratified sampling

The fix is **stratified sampling**. Instead of drawing from the population as one undifferentiated pool, you first divide it into meaningful subgroups, called **strata**, and then draw a sample from within each stratum separately. Every stratum is guaranteed a place in the final sample, in whatever proportion you decide, rather than in whatever proportion the population happens to hand you.

The strata have to be chosen to matter for the research question. For the chat study the natural stratifying variable is the channel. Dividing the population by channel and drawing a fixed number of messages from each guarantees that the small channels appear in the sample at the same strength as the giant ones, which is exactly what a simple random draw would not do. Because gaming and non-gaming status is a property of the channel, stratifying by channel also rescues the study's central comparison: every non-gaming channel is in the sample by construction, not by luck.

It is worth knowing that the fixture you loaded in Chapter 9 is itself the product of a stratified design. Its fifty channels were not the fifty busiest. Eight were chosen as fixed anchors for pedagogical variety, Bob Ross among them as the deliberate non-gaming case. The other forty-two were drawn by sorting the rest of the population into ten bands by chat volume and sampling from every band, so that the corpus spans the full range from the busiest channels to the quietest rather than just the head of the distribution. The skew did not get sampled away. It got sampled across, on purpose.



One feature of stratified sampling matters when the strata are uneven in size. If you ask for thirty messages per channel but a channel only ever produced nine, there is no way to draw thirty. The sensible rule, and the one the tools here follow, is to take everything that channel has. In the fixture, nineteen of the fifty channels hold fewer than a thousand messages, and a few hold only a handful. For those small strata, a request for a fixed number simply returns all of it. That is the right floor: a stratum contributes its full weight or, if it is too small, its full self.

Purposive sampling does not fix inadequate power. A common misconception is that purposive sampling (selecting "most relevant" units) compensates for a small *n*. It does not. Purposive sampling improves construct validity; it has no effect on statistical power. The two concerns address different inferential problems, and conflating them produces a study that is internally coherent but statistically indefensible.

11.4 Drawing the sample

With the logic settled, the draw itself is one function call. `v2v::sample_messages()` takes the data, the variable to stratify by, and the number to draw from each stratum.

```
set.seed(409)

coding_sample <- v2v::sample_messages(chat, by = channel, n = 30)
```

The `by = channel` argument makes each channel a stratum. The `n = 30` argument asks for thirty messages from each. With fifty channels that aims at roughly 1,500 messages, a sample large enough to support the study and small enough to code by hand, with the small channels contributing all they have where thirty is more than they hold.

The first line, `set.seed()`, is not optional, and it is worth being clear about why. Drawing a sample is a random process: run it twice and you get two different samples. `set.seed()` fixes the starting point of R's random number generator, so that the "random" draw comes out identically every time the code is run. Without it, your sample is unreproducible, and an unreproducible sample undermines everything Chapter 2 argued for. Anyone who reruns your analysis, including you in six months, would be working with different messages and could not check your results against the originals. The specific number is arbitrary; 409 has no meaning. What matters is that a number is set and recorded, so the draw is frozen.

`coding_sample` now holds the messages the study will code. The sampling half of the chapter is done. The harder question is whether the codebook is ready to code them.

The conversation between your prospectus and your sample. Sampling is not only a practical tradeoff about capturing the variation in your dataset. Your sample must also be large enough to detect the effect you expect with adequate power. Your prospectus power analysis (Chapter 6) specified a minimum n at your chosen effect size and power level, and Chapter 10 is where that number meets the realities of your dataset. If your power analysis requires 128 coded units and your platform generates 10,000 units in your collection window, stratified sampling solves the problem. If your platform generates only 80 units, you face a constraint: the study is underpowered, and you must either change the design (longer collection window, multiple platforms, or a revised research question targeting a larger expected effect) or explicitly justify the limitation in the prospectus. For your own study, identify the minimum n from your Chapter 6 power analysis and the maximum n available from your dataset; if they conflict, document the conflict and propose one design modification that resolves it, and weigh the ethical implication of publishing a study you know is underpowered because adequate data were unavailable, along with how the limitations section should handle it. Lakens (2022) is blunt about what honesty demands once the available data fall short of the design:

> "Do not try to spin a story where it looks like a study was highly informative when it was not. Instead, transparently evaluate how informative the study was given effect sizes that were of interest, and make sure that the conclusions follow from the data."
>
> Lakens (2022, p. 11)

11.5 The pilot test

You finalized the codebook in Chapter 9, and Chapter 9 was careful about the word. It finalized the codebook's status, not its content. The reason for that hedge is the **pilot test**, and this is where it comes due.

A pilot test is a trial run of the codebook before the real coding begins. A subset of the sample, a hundred messages is a reasonable size, is coded not by one person but by two, working from the same codebook. Then their codes are compared. The question the comparison answers is the one that decides whether the codebook is an instrument or just a document: when two people apply these rules independently, do they produce the same classifications?

That word, independently, carries the whole logic. The two coders must work in genuine isolation, no conferring, no comparing notes, no checking each other as they go. The reason is not suspicion. It is that the pilot is measuring the codebook, not the coders. If the two of them discuss a hard message and agree on how to code it, their later agreement on similar messages measures their memory of that conversation, not the clarity of the codebook. Only independent coding tests what the study actually needs to know,

which is whether the written rules are clear enough to carry a classification on their own, without their author standing beside the coder to interpret them.

Chapter 8 introduced the idea of **reliability** and promised that the way to measure it would come later. This is later. Reliability is consistency, the property that the codebook produces the same answer regardless of who is holding it, and the pilot test is how reliability stops being a hope and becomes a number.

11.6 Why raw agreement is not enough

The obvious way to turn the pilot into a number is to count how often the two coders matched. If they agreed on the code for 74 of the 100 pilot messages, that is 74 percent agreement, and 74 percent sounds respectable.

It is also misleading, and seeing why is the key idea of the chapter. Some agreement between two coders happens for no good reason at all. It happens by luck.

Cohen's 1960 paper that introduced the standard fix made the point with a sharp example. Imagine two clinicians independently sorting patients into two categories, and suppose each one, rather than examining anyone, simply assigns the categories blindly at the rates the categories happen to occur. The two of them, coordinating on nothing, would still agree a large share of the time, purely because there are only so many categories to land on. Agreement produced that way is worth nothing. It reflects the number of categories and how often each is used, not the soundness of any judgment.

Raw agreement cannot tell that empty agreement apart from the real kind. It counts every match the same, the ones that reflect a clear shared rule and the ones that are coincidence. When the categories are few, or one category is far more common than the others, the coincidental matches pile up fast, and raw agreement drifts upward on their strength. A figure of 74 percent might reflect a genuinely clear codebook. It might also be mostly luck. The number alone does not say.

What you need is a measure that starts from the agreement actually observed, estimates how much agreement chance alone would have produced, and credits the codebook only with the difference. That is what a reliability statistic does.

11.7 Cohen's kappa and Krippendorff's alpha

Two such statistics cover most of content analysis, and the `v2v::reliability()` function computes either one, selected by its `method` argument.

Cohen's kappa is the first, introduced in the 1960 paper just mentioned (Cohen, 1960). It is built for the common case: two coders, applying a nominal codebook. Its logic is exactly the correction described above. Conceptually, kappa is the observed agreement minus the agreement expected by chance, divided by the room for improvement that chance left on the table:

```
kappa = (observed agreement - expected agreement) / (1 - expected agreement)
```

When the coders do no better than chance, the numerator is zero and kappa is zero, whatever the raw agreement looked like. When they agree perfectly, kappa is one. A codebook is credited only for the agreement it produced beyond luck.

Krippendorff's alpha is the second, and it is the more general instrument (Krippendorff, 2018). Where Cohen's kappa assumes exactly two coders and nominal categories, alpha accommodates any number of

coders, any level of measurement from nominal to ratio, and datasets with missing codes. For a study with three coders, or one whose central variable is ordinal, alpha is the appropriate choice; for the standard two-coder nominal pilot, kappa and alpha will tell much the same story, and kappa is the more familiar choice for that case. `v2v::reliability(coder_a, coder_b, method = "alpha")` computes alpha; `method = "kappa"` computes kappa.

Either way the result is a number between roughly zero and one, and a number needs a standard to be read against. The most cited one comes from Landis and Koch (1977), who attached descriptive labels to ranges of agreement: values around 0.0 to 0.2 they called slight, 0.2 to 0.4 fair, 0.4 to 0.6 moderate, 0.6 to 0.8 substantial, and above 0.8 almost perfect. Those labels are a vocabulary, not a law, and published content analysis tends to apply a stricter working rule: a reliability coefficient below about 0.70 is generally treated as too low to proceed on, with 0.80 a more comfortable floor. The exact threshold is a judgment call that depends on the stakes of the study. The principle is not: a codebook that cannot clear a defensible threshold is not ready to be used.

11.8 A worked reliability check

To see what a failed check looks like, and why codebooks fail it, consider a deliberately stark example.

Imagine the codebook included a loosely worded variable. It asked coders to flag the pilot's "high-energy" messages, the pure expressive reactions, but it never pinned down what that meant in operational terms. Two conscientious coders, each trying to apply the instruction, each settle on a concrete rule of their own. Coder A decides a high-energy message is one carrying a Twitch emote, a token like LULW or OMEGALUL, and flags messages on that basis. Coder B decides a high-energy message is one that shouts, and flags every message containing a word in all capitals. Both rules are reasonable readings of "high-energy." Both coders believe they are coding the same variable.

They code the first hundred messages of the sample independently, and their codes go into `v2v::reliability()`:

```
v2v::reliability(pilot$coder_a, pilot$coder_b, method = "kappa")
```

```
Inter-coder reliability: Cohen's kappa

  Messages coded   100
  Cohen's kappa    0.236
```

The two coders agreed on 74 of the 100 messages, the same 74 percent that looked respectable a few pages ago. But chance alone, given how often each coder flagged a message, would have produced agreement on about two-thirds of them. Run that through the formula, observed agreement of 0.74 against expected agreement of about 0.66, and the result is roughly 0.24. The function reports it as 0.236.

By the Landis and Koch labels, 0.236 is fair agreement. By any working threshold for publishable content analysis, it is a failure, far below 0.70. And the diagnosis is exact. The raw agreement of 74 percent was almost all coincidence. Emote-bearing messages and all-caps messages overlap a fair amount in Twitch chat, enough that two coders keying on those different features would land on the same answer often, without ever coding the same concept. The loose codebook let two careful people measure two different things while believing they measured one. That is what kappa exposed and raw agreement concealed.

11.9 When reliability fails: revising the codebook

A kappa of 0.236 is not a result to report. It is an instruction, and the instruction is to stop and fix the codebook before any real coding happens.

The fix begins with diagnosis. The two coders, who until now have worked in isolation, finally meet and go through the messages they classified differently, one by one. The disagreements are the data. They show precisely where the codebook ran out of guidance: a category whose boundary is fuzzy, a kind of message the rules never anticipated, a term that two readers can take two ways. In the worked example the diagnosis is plain, because the variable was never operationally defined at all. The repair is to define it: state in the codebook exactly what counts as a high-energy message, with the kind of explicit decision rules Chapter 8 built for the message-target variable, so that two coders are no longer free to invent their own.

Then the codebook, now revised, is piloted again. New coders or the same coders, a fresh subset of messages, another independent coding, another reliability coefficient. The cycle, pilot, diagnose, revise, repeat, continues until the coefficient clears the threshold. Only then does full coding of the sample begin.

This is why Chapter 9 finalized only the codebook's status and not its content. A codebook is a living instrument right up until a pilot test certifies it, and a pilot test has the standing to send it back for revision however many times that takes. The discipline is to treat a low reliability coefficient as useful news rather than an obstacle. It has told you, before you invested weeks in coding, that the instrument was not yet sound. That is the pilot test working exactly as intended.

11.10 Looking ahead

You now have a sample drawn and, once it has cleared its pilot, a codebook trusted to classify it. What you do not yet have is data in a form ready for analysis. The chat and stream tables are still separate, the timestamps are still unreadable integers, and the variables the study needs, message length and the gaming context of each message, do not yet exist as columns. Chapter 11 takes that on. It is the wrangling chapter: the one where the raw tables are reshaped, joined, and cleaned into the single tidy dataset every later analysis depends on.

11.11 References

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37-46. https://doi.org/10.1177/001316446002000104

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology* (4th ed.). SAGE Publications.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159-174. https://doi.org/10.2307/2529310

11.11.1 Graduate readings

Lakens, D. (2022). Sample size justification. *Collabra: Psychology*, *8*(1), 33267. https://doi.org/10.1525/collabra.33267

12 Chapter 11: Wrangling the data

Listen in Dr. Leith's voice

The chat table you loaded in Chapter 9 cannot answer your research question. Not yet. The study asks whether chat behaves differently on gaming and non-gaming channels, and it operationalizes that through

message length. But look at the table. There is no column for message length. There is no column saying whether a message came from a gaming channel or a non-gaming one. The timestamps are unreadable runs of digits. And the information that would settle the gaming question is not even in the chat table; it sits in a separate stream table that the chat table has no link to.

This is the ordinary situation at this stage of a study, and closing the gap is the work of this chapter. **Data wrangling** is the process of turning the data you collected into the data you can analyze: reshaping it, cleaning it, deriving the variables your codebook named, and joining separate tables into one. It is unglamorous, it is rarely mentioned in the published paper, and it is most of the work. A wrangling chapter looks like a sequence of small technical steps. What it is actually doing is building the table that every later result depends on.

12.1 The verbs of data wrangling

The tidyverse handles wrangling through a small set of functions, each of which does one clear thing to a data frame. They are usually called the **dplyr** verbs, after the package they live in, and five of them carry most of the work.

`filter()` keeps rows that meet a condition and drops the rest. To look at only Bob Ross's messages:

```
chat %>% filter(channel == "bobross")
```

`select()` keeps or drops columns:

```
chat %>% select(channel, sender, message)
```

`mutate()` adds a new column, or changes an existing one, computed from the columns already there. `arrange()` reorders the rows by a column's values. And `group_by()` together with `summarize()` collapses many rows into one per group, which is how you compute a per-channel or per-game figure.

The reason these verbs combine so well is a shared design: each one takes a data frame and returns a data frame. That is what lets the pipe string them into a sequence, the output of one verb flowing in as the input of the next. A wrangling pipeline is just verbs chained this way, each step a small readable transformation, the whole chain carrying the data from raw to ready. The rest of this chapter is four such transformations.

The wrangling script as a methods section. A wrangling script is a *documented transformation chain*: a sequence of decisions, each recorded in version-controlled code, that connects raw data to analysis-ready data. Every `mutate()`, `filter()`, and `group_by()` is a methodological decision. You decided which cases to exclude and why. You decided how to handle missing values. You decided how to aggregate timestamp data into time windows. When a reviewer asks "how did you handle cases where the streamer's username appeared in the chat?", the answer is in the wrangling script, but only if the script is version-controlled and decision points are commented. For your own study, identify three wrangling decisions a reviewer might challenge, and for each add a comment stating the decision rule and its justification.

12.2 Making the timestamp readable

Start with the timestamps, because Chapter 9 already flagged them. The `date` column is a column of very large numbers, and Chapter 8 classified it as interval data with a catch: the number is not a date in any form a person can read.

What the number actually holds is a count of milliseconds elapsed since midnight on the first of January, 1970. That reference moment is the **Unix epoch**, the zero point nearly all computers use to keep time, and counting from it is how a timestamp becomes a single integer. The integer is precise and entirely unreadable. Turning it back into a date is a conversion, and `mutate()` is the verb for it:

```
chat <- chat %>%
  mutate(timestamp = as.POSIXct(date / 1000, origin = "1970-01-01", tz = "UTC"))
```

Two parts of that line carry the lesson. `as.POSIXct()` is the function that turns a number into a date-time value, given the origin to count from. And `date / 1000` is the part it is easy to get wrong. `as.POSIXct()` expects a count of seconds, but the `date` column is a count of milliseconds, a thousand times finer. Hand it the raw number and it will place every message tens of thousands of years in the future. Dividing by 1,000 converts milliseconds to seconds first. It is a small arithmetic step and the single most common way this conversion fails.

Because it is the most common failure, the course package ships a wrapper that performs the division and the conversion together, so the mistake cannot happen by accident:

```
chat <- chat %>%
  mutate(timestamp = clean_dates(date))
```

That is the same operation, not a shortcut around it. `clean_dates()` exists to be read: run `body(clean_dates)` and you will find the line above. Use whichever you prefer in your own work, but learn the arithmetic first, because the day you meet a millisecond timestamp outside this course there will be no wrapper waiting for it.

The result is a real date-time. Setting `date` and the new `timestamp` side by side shows the transformation:

```
chat %>% select(date, timestamp) %>% head(3)
```

```
# A tibble: 3 × 2
           date timestamp
          <dbl> <dttm>
1 1542578127023 2018-11-18 21:55:27
2 1542578135562 2018-11-18 21:55:35
3 1542578143610 2018-11-18 21:55:43
```

The same instant, now legible: the evening of the 18th of November, 2018. The column type has changed too, from `<dbl>`, an ordinary number, to `<dttm>`, a date-time. That new type is what makes the column useful. From a real date-time you can pull the hour of day, the day of week, the calendar date, none of which can be read off the raw integer. Chapter 12 will use exactly that to chart how chat volume rises and falls across the hours of the day.

12.3 Measuring message length

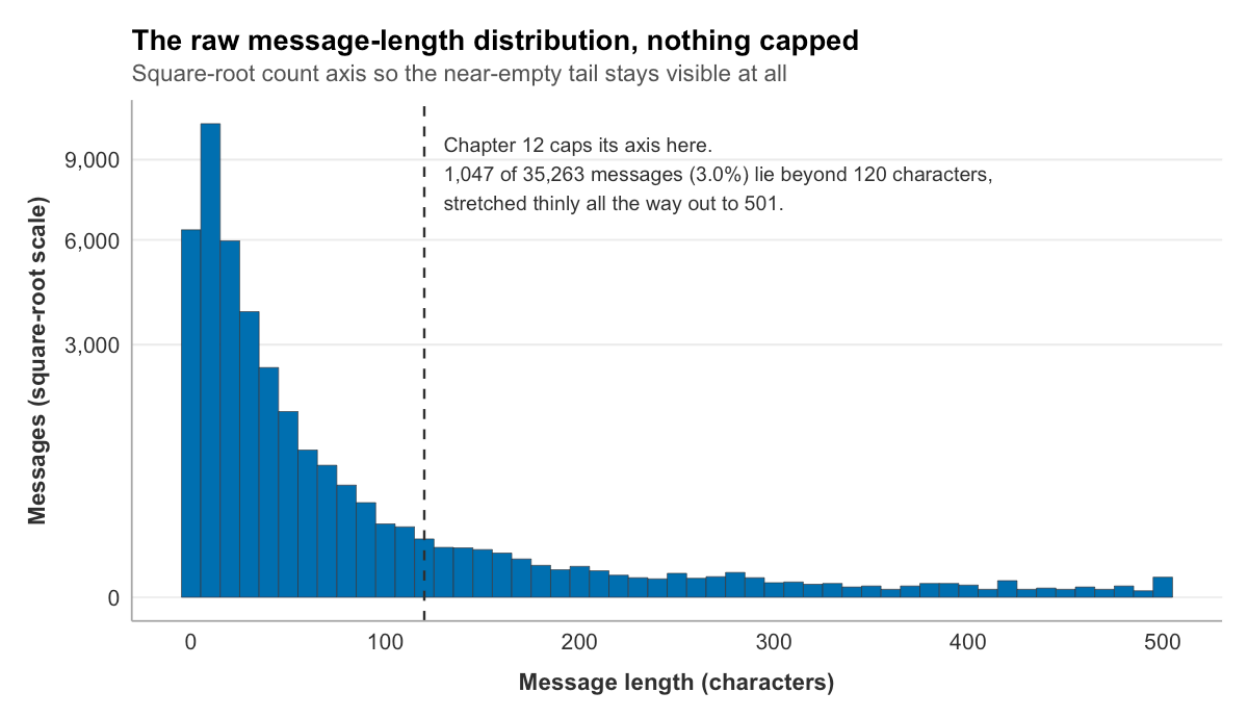
The second transformation creates a variable the codebook named but the data does not yet contain. Message length, the character count of a message, is `mutate()` again, this time with `str_length()`, which counts the characters in a string:

```
chat <- chat %>%
  mutate(message_length = str_length(message))
```

```
chat %>% select(message, message_length) %>% head(3)
```

```
# A tibble: 3 × 2
  message                                  message_length
  <chr>                                             <int>
1 "?????????????????????????"                         23
2 "TriEasy Clap TriEasy Clap TriEasy Cla…"            441
3 "cmonBruh"                                            8
```

`message_length` is a ratio variable, in the Chapter 8 sense: a true zero, equal intervals, a count you can average and compare. The three values above also preview something. A single-word message of eight characters, a row of punctuation at twenty-three, and a block of copy-pasted text at 441 are all real Twitch chat, and the distance between them is large. That spread is the raw material of the study, and Chapter 12 will look hard at its shape.



12.4 Labeling gaming and non-gaming

The third transformation is the one the central research question turns on. Every message needs a label: did it come from a gaming channel or a non-gaming one? That label does not exist in the chat table, and building it takes more work than the last two, because the information lives in the other table.

Gaming or non-gaming is best understood as a property of the channel, fixed by what the channel mostly streams. Bob Ross's channel streams painting; it is a non-gaming channel, and every message in it is a

non-gaming message. So the label is built in three steps: find what each channel mostly streams, decide whether that counts as gaming, and attach the decision to every message from that channel.

The first step reaches into the stream table. Each channel appears there many times, once per snapshot, each snapshot carrying the `game` it was streaming. The channel's dominant category is the **modal** category, the one that appears most often:

```
channel_type <- streams %>%
  filter(!is.na(game)) %>%
  count(channel, game) %>%
  group_by(channel) %>%
  slice_max(n, n = 1, with_ties = FALSE) %>%
  ungroup()
```

The pipeline reads in order. Drop snapshots with no recorded category. Count how many snapshots each channel logged in each category. Then, within each channel, keep only the single most frequent category. The result is one row per channel, naming what that channel streamed most.

The second step is a judgment, and it should be visible rather than buried. Twitch sorts streams into categories, and some of those categories are not games: painting under Art, unstructured talk under Just Chatting, and others. Listing them explicitly states the rule the study is using:

```
nongaming_categories <- c(
  "Art", "ASMR", "Beauty & Body Art", "Creative", "Food & Drink",
  "IRL", "Just Chatting", "Makers & Crafting", "Music",
  "Music & Performing Arts", "Science & Technology",
  "Sports & Fitness", "Talk Shows & Podcasts", "Travel & Outdoors"
)

channel_type <- channel_type %>%
  mutate(is_gaming = !(game %in% nongaming_categories)) %>%
  select(channel, is_gaming)
```

`game %in% nongaming_categories` asks whether a channel's dominant category is on the non-gaming list; the `!` negates it, so `is_gaming` is `TRUE` for every channel whose main category is a game and `FALSE` otherwise. `channel_type` is now a compact lookup table: one row per channel, one column saying gaming or not.

The third step is the **join**. A join brings columns from one table onto another by matching on a shared key, here the channel name. A `left_join()` keeps every row of the chat table and attaches the matching `is_gaming` value to it:

```
chat <- chat %>%
  left_join(channel_type, by = "channel")
```

A join is only as good as the match between its keys, and that is worth a word of warning. Twitch's chat protocol writes channel names with a leading "#", so the same channel can appear as `#bobross` in one source and `bobross` in another. Join two tables whose keys disagree like that and every row fails to match, silently. The `v2v` fixture has already normalized this, the "#" stripped before the data ever reached you,

but raw Twitch data would need that cleaning step first. When a join returns nothing, mismatched keys are the first thing to check.

One result of the join deserves attention. Always inspect a column you just derived:

```
chat %>% count(is_gaming)
```

```
# A tibble: 3 × 2
  is_gaming     n
  <lgl>     <int>
1 FALSE      3457
2 TRUE      31309
3 NA          501
```

Two channels in the corpus streamed without ever having a category recorded: every one of their stream snapshots had a missing `game`. With no categories at all, there is no modal category, so those channels never made it into `channel_type`, and the `left_join()` left their messages with `is_gaming` set to `NA`. That is the correct outcome, not a bug. The data genuinely does not say whether those 501 messages came from gaming channels, and an honest `NA` records that the study cannot classify them. They will sit out the gaming comparison rather than being guessed into one side of it.

12.5 The analysis-ready table

Four transformations, run in sequence, have rebuilt the chat table:

```
chat <- v2v::twitch_chat() %>%
  mutate(
    timestamp = as.POSIXct(date / 1000, origin = "1970-01-01", tz = "UTC"),
    message_length = str_length(message)
  ) %>%
  left_join(channel_type, by = "channel")
```

A final `glimpse()` shows what the table has become:

```
Rows: 35,267
Columns: 8
$ id             <int> 89, 551, 1033, 2094, 2803, …
$ channel        <chr> "sodapoppin", "xqcow", "forsen", "sodapoppin", "xqcow…
$ sender         <chr> "madzee", "prometheanow", "spectre155", "itsjer", "la…
$ message        <chr> "??????????????????????", "TriEasy Clap TriEasy Cla…
$ date           <dbl> 1542578127023, 1542578135562, 1542578143610, 1542578…
$ timestamp      <dttm> 2018-11-18 21:55:27, 2018-11-18 21:55:35, 2018-11-18…
$ message_length <int> 23, 441, 8, 12, 206, …
$ is_gaming      <lgl> TRUE, FALSE, TRUE, TRUE, FALSE, …
```

This is what wrangling was for. The table is now **tidy**, in the precise sense the term carries in data science (Wickham, 2014): each row is one observation, a single chat message; each column is one variable; and everything the study needs sits in this one place. The unreadable integers have become timestamps.

Message length and gaming status, named in the codebook and absent from the raw data, are now columns. The chat and stream tables, once disconnected, are joined.

None of this was analysis. Not a single result has been computed. But every result that follows, every figure in Chapter 12 and every test in Chapter 13, will be computed from this table, which is why the care spent building it is not optional. A wrangling mistake does not announce itself. It travels quietly into every chart and every statistic downstream. The reward for getting this chapter right is that nothing later has to be unwound.

Tidy data as the target state. Wickham's tidy data principles (one observation per row, one variable per column, one value per cell) are not just a preference. They are the structural assumption that every `tidyverse` function makes about its inputs and produces about its outputs. Wickham (2014) defines the target in one sentence:

> "Tidy data is a standard way of mapping the meaning of a dataset to its structure."
>
> Wickham (2014, p. 4)

Wrangling that produces tidy data is wrangling that can be audited: the structure itself is a transparency claim.

Never overwrite raw data. `data/raw/` is read-only after the first commit. Your wrangling script reads from `data/raw/` and writes to `data/processed/`. This separation means you can reproduce the processed dataset from scratch by re-running the wrangling script against the immutable raw data. Overwriting raw data severs this guarantee. To test the guarantee on your own study, re-run your wrangling script from a fresh R session against the raw data: if it does not produce the same processed dataset without manual intervention, find the undocumented dependency you missed.

12.6 Looking ahead

The data is tidy and the variables exist. The next move is to look at them. Chapter 12 turns the analysis-ready table into figures: how chat volume moves across the hours of the day, how viewership splits across game categories, and how the distribution of message length compares between gaming and non-gaming channels. It is the chapter where the study's central question finally meets a picture of the answer.

12.7 References

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1-23. https://doi.org/10.18637/jss.v059.i10

13 Chapter 12: Visualizing the narrative

Listen in Dr. Leith's voice

Chapter 11 ended with a single tidy table, `analysis`, a little over thirty-five thousand rows wide. That table is a complete and faithful record of the chat study's data, and it is almost impossible to read. Nobody can scroll thirty-five thousand rows and come away knowing whether gaming chat runs longer than non-gaming chat, or when in the day the channels are busiest. The numbers are all present. The pattern is not visible.

A figure is the instrument that makes the pattern visible. It takes a column of numbers too long to hold in the mind and turns it into a shape the eye can read at a glance: a line that rises, a bar that towers over its neighbors, a distribution that leans hard to one side. This chapter builds three such figures from the analysis table. Each one answers a question the study has been carrying since its prospectus, and each one is a different kind of chart, chosen to fit a different kind of question.

The chapter's title is a claim about what that work is for. A figure does not merely display data. A well-made figure carries an argument: it shows the reader what the analyst found and why it matters. Visualization is not decoration applied after the analysis. It is part of the analysis, and it is where the study's narrative first becomes something other people can see.

13.1 The grammar of graphics

The figures in this chapter are built with **ggplot2**, the R package that has become the standard tool for statistical graphics (Wickham, 2016). What makes ggplot2 worth learning is not a catalog of chart types. It is a single idea, borrowed from a book by Leland Wilkinson called *The Grammar of Graphics* (Wilkinson, 2005): that every statistical graphic can be described as the same small set of parts, combined in different ways.

Three parts do most of the work. The first is the **data**, the table being shown. The second is a set of **aesthetic mappings**, which say how columns of that table correspond to visual properties of the plot: this column to the horizontal position, that column to the vertical position, a third to color. The third is a **geometry**, the kind of mark used to draw the data, a line or a bar or a point. Choose the data, map the columns to aesthetics, pick a geometry, and a plot is specified.

In code the three parts sit in a recognizable skeleton:

```
ggplot(data, aes(x = ..., y = ...)) +
  geom_line()
```

`ggplot()` names the data and, inside `aes()`, the aesthetic mappings. `geom_line()` is the geometry. The `+` is ggplot2's way of stacking layers: each piece is added to the plot, and a figure is assembled by composition rather than written all at once. Changing one part changes the figure without disturbing the rest. Swap `geom_line()` for `geom_col()` and the same data is drawn as bars. That is the payoff of a grammar. A handful of interchangeable parts covers an enormous range of figures, and learning the parts is learning all of them at once.

The three figures that follow are three settings of the same grammar.

13.2 A line for change over time

The first question is about viewership. Across the collection week, how did the audience for these fifty channels move, and how was it split across the games being played?

When the horizontal axis is time and the vertical axis is a quantity that rises and falls, the natural geometry is the **line chart**. A line connects one moment to the next, and the eye reads the climb and the drop as a single continuous motion. That is what makes a line right for change over time and wrong for unordered categories: it implies that the points are in sequence and that the space between them is real.

The data needs shaping first, the kind of dplyr work Chapter 11 covered. The stream snapshots are grouped into six-hour buckets, the long tail of games is collapsed so that only the five most common categories keep their names and the rest become "Other," and the viewer counts are summed within each bucket:

```
library(tidyverse)

viewers_over_time <- streams %>%
  filter(!is.na(game)) %>%
  mutate(
    timestamp = as.POSIXct(date / 1000, origin = "1970-01-01", tz = "UTC"),
    six_hour  = floor_date(timestamp, "6 hours"),
    category  = fct_lump_n(game, n = 5)
  ) %>%
  group_by(six_hour, category) %>%
  summarise(total_viewers = sum(viewers), .groups = "drop")
```

With the data in shape, the figure itself is short. The aesthetic mappings send the time bucket to the x-axis, the summed viewers to the y-axis, and the game category to color, so each category gets its own line:

```
ggplot(viewers_over_time, aes(x = six_hour, y = total_viewers, color = category)) +
  geom_line(linewidth = 0.9) +
  labs(
    title = "Concurrent viewers by game category",
    x = "Date (UTC, six-hour buckets)",
    y = "Total viewers in bucket",
    color = "Category"
  ) +
  v2v::scale_colour_v2v() +
  v2v::theme_v2v()
```

The last line is worth a note. `v2v::theme_v2v()` applies a single consistent visual theme, the fonts, spacing, and gridlines of a publication-ready figure, in one call. Every figure in this chapter ends with it, so the three look like members of one set rather than three unrelated charts.

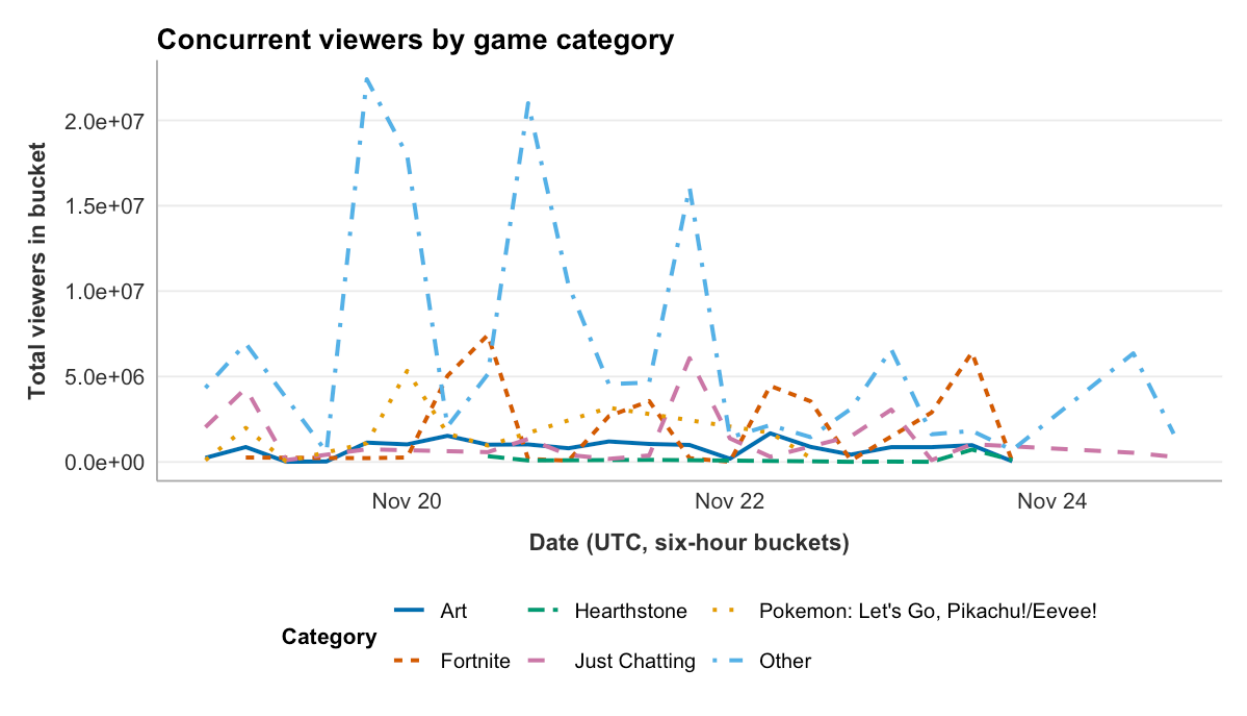


Figure 1: Total concurrent viewers across the focal fifty channels, summed within six-hour UTC buckets and split by game category, over the November 2018 collection week.

The figure has an obvious headline. The line that dominates the plot is "Other," the catch-all holding every game outside the top five, and it spikes far above any named category. That fact can be read two ways, and a careful analyst holds both. One reading is substantive: the viewership in this corpus is genuinely spread across a long tail of games, with no single title commanding the audience the way a casual observer might expect. The other reading is a caution about the chart's own choices: lumping more than fifty categories into one bucket all but guarantees that bucket will be the largest, so part of what the figure shows is an artifact of where the line between "named" and "Other" was drawn. There is also a plain design cost. Because "Other" is so large, the five named lines are pressed into the bottom of the plot, and the comparison a reader might most want, how Fortnite moved against Just Chatting across the week, is hard to make. A dominant series can crowd out the rest, and noticing that is part of reading a figure honestly.

13.3 A bar for counts

The second question is about timing. The chat data spans a little under six days; across the hours of the day, when are the channels busiest?

Hour of day is not a continuous sweep the way a date is. It is a set of twenty-four discrete categories, and the thing being measured for each is a simple count of messages. For counts across discrete categories, the natural geometry is the **bar chart**: one bar per category, its height the count, the bars sitting side by side so the eye can compare them.

The data is one short pipeline. The hour is pulled out of each message's timestamp, and the messages are counted within each hour:

```
chat_by_hour <- analysis %>%
  mutate(hour = hour(timestamp)) %>%
  count(hour, name = "messages")

ggplot(chat_by_hour, aes(x = hour, y = messages)) +
  geom_col(fill = "#2f7d8a") +
  labs(
    title = "Chat volume by hour of day",
    x = "Hour of day (UTC)",
    y = "Messages in sample"
  ) +
  v2v::theme_v2v()
```

`geom_col()` is the geometry that draws a bar whose height is a value already computed, which is what `count()` produced.

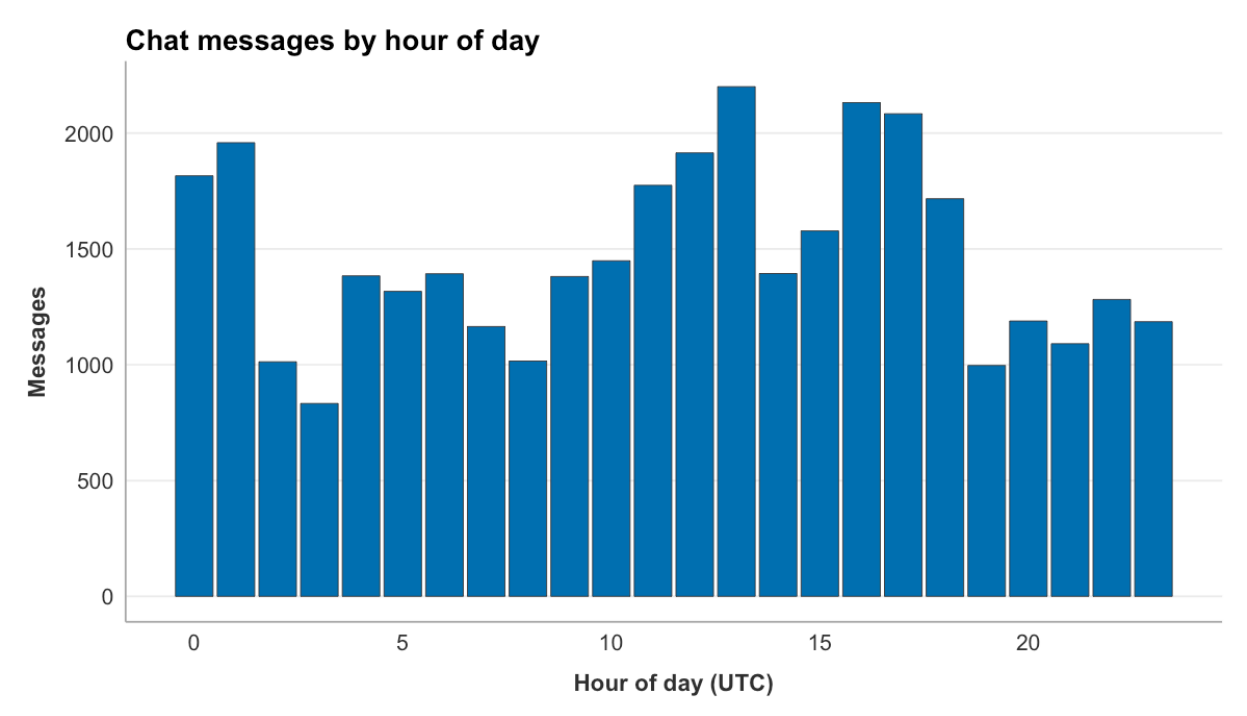


Figure 2: Total chat messages in the sample by hour of day in UTC, across the collection window.

The figure shows a clear daily pulse. Chat is busiest through the UTC midday and afternoon, peaking at 13:00 with 2,201 messages, and quietest in the small hours, bottoming out at 03:00 with 833, a swing of well over two to one between the loudest hour and the quietest. The shape is not mysterious. Twitch's audience in 2018 was concentrated in the Americas and Europe, and the UTC midday and afternoon are the waking, screen-facing hours across that span, while 03:00 UTC is the middle of the night for most of it. The point worth taking is less the particular peak than what the bar chart does: it takes twenty-four numbers and makes a rhythm visible in one glance, which is exactly the job a bar chart is for.

13.4 A histogram for shape

The third question is the study's central one. The chat study set out to compare gaming and non-gaming channels through message length. So: how long is a chat message, and does the answer depend on the kind of channel?

This question is not about a total or a trend. It is about a **distribution**, the shape of how a single variable's values spread out, where they cluster and where they thin. The geometry for a distribution is the **histogram**. A histogram slices the range of a numeric variable into equal intervals, called **bins**, and draws a bar for each showing how many values fall inside it. The result is a picture of the variable's shape.

The figure overlays two histograms, one for gaming channels and one for non-gaming, so the shapes can be compared directly. Messages with no gaming label, the unmatched rows Chapter 11 set aside, are dropped first, and message lengths are capped at 120 characters for the view:

```
msglen <- analysis %>%
  filter(!is.na(is_gaming)) %>%
  mutate(length_shown = pmin(message_length, 120))

ggplot(msglen, aes(x = length_shown, fill = is_gaming)) +
  geom_histogram(binwidth = 5, position = "identity", alpha = 0.55) +
  labs(
    title = "Message length, gaming versus non-gaming channels",
    x = "Message length (characters, capped at 120)",
    y = "Messages",
    fill = "Gaming channel"
  ) +
  v2v::scale_fill_v2v() +
  v2v::theme_v2v()
```

Two choices in that code shape the figure and both have to be disclosed. The `binwidth = 5` sets each bar to cover five characters; a wider bin smooths the shape and a narrower one roughens it, and the number is a judgment the analyst makes and reports. The `pmin(message_length, 120)` caps the displayed length at 120 characters, so every message longer than that lands in the final bar. Twitch's real message limit is far higher, 500 characters, so that last bar is a genuine pile-up of long messages compressed into one place. Capping keeps the bulk of the distribution legible instead of stretched thin by a few very long outliers, but a cap that is not announced is a quiet distortion. The axis label says "capped at 120" for exactly that reason.

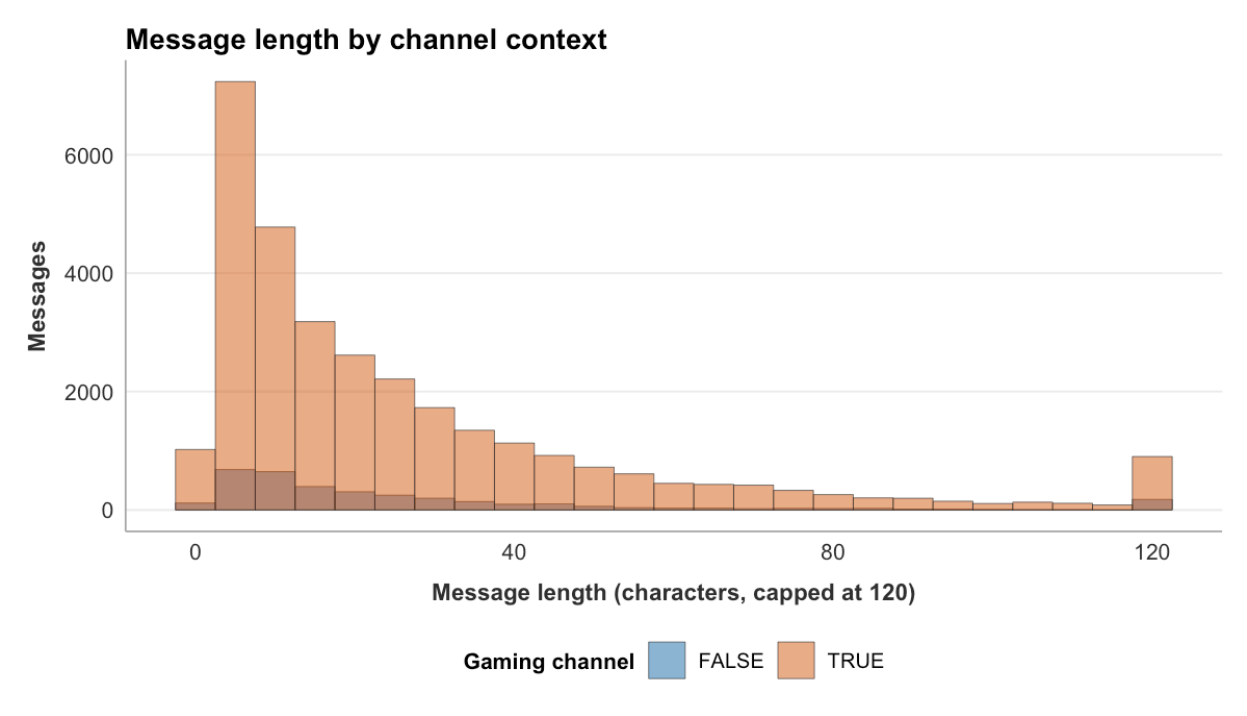


Figure 3: Overlaid histograms of message length for gaming and non-gaming channels, bin width five characters, lengths capped at 120.

The figure shows, first, a shape both groups share. Each distribution is heavily **right-skewed**: a tall stack of very short messages at the left, then a long thin tail trailing off to the right. Twitch chat is mostly brief, a word or an emote, with a minority of longer messages stretching the range. This is the most important thing the histogram reveals, and it is invisible in any single summary number.

It matters because of what the summary numbers say. The descriptive statistics for the two groups, the kind of table `v2v::pretty_table()` formats cleanly, set the comparison up:

```
msglen %>%
  group_by(is_gaming) %>%
  summarise(
    n      = sum(!is.na(message_length)),
    mean   = mean(message_length, na.rm = TRUE),
    median = median(message_length, na.rm = TRUE),
    sd     = sd(message_length, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  v2v::pretty_table()
```

is_gaming	n	mean	median	sd
FALSE	3,457	33.70	16	60.68
TRUE	31,305	28.49	17	38.47

The gaming row counts 31,305 rather than the 31,309 messages that came from gaming channels, because four messages in the corpus have no text at all and therefore no length. Counting rows where the statistics skip four of them would make the `n` column describe a different set of messages from the columns beside it.

Read the `mean` column and a story jumps out: non-gaming messages, the `FALSE` row, average 33.70 characters against 28.49 for gaming messages. Non-gaming chat looks meaningfully longer. Now read the `median` column, the length of the exactly-middle message in each group, and the story collapses: 16 characters for non-gaming, 17 for gaming, all but identical, and pointing the other way. The typical message is the same length in both kinds of channel.

The histogram explains how both things can be true at once. The mean is sensitive to a long tail in a way the median is not: a relatively small number of very long messages pulls an average upward while leaving the middle value untouched. The non-gaming group has the heavier tail, which its much larger standard deviation, 60.68 against 38.47, records directly. So the gap in means is real arithmetic, but it is the work of the tail, not of the typical message. Lean only on the mean and you would report that non-gaming chat is longer. The picture, and the median beside it, show that the everyday message is the same and the difference lives in the extremes.

This is what it means to treat descriptive statistics as a visual setup rather than a verdict. A mean is one number standing in for thousands, and which thousands it stands in for is something only the distribution can show. It also leaves a precise question for the next chapter. There is a gap of about five characters between the two means. Is that gap a real difference between gaming and non-gaming channels, or the kind of wobble that any two samples would show? A figure can frame that question. It cannot answer it.

Effect size reporting with every comparison. A good figure conveys the pattern in the data without distortion, and descriptive reporting adds a further requirement on top of that: every figure that displays a comparison must include the effect size of that comparison. A bar chart showing that one group scored higher than another is not a scientific claim; it is a visual impression. The effect size quantifies the magnitude of that difference in a scale-invariant unit. Cohen's d reports how many standard deviations apart two group means are; Cramer's V reports the strength of association between two categorical variables after accounting for marginal distributions. These values let future researchers use your descriptives as inputs for their power analyses, closing the cumulative science loop that Chapter 4 opened. The case for making the practice routine begins from the standing effect sizes hold in the evidence itself:

> "Effect sizes are the most important outcome of empirical studies."
>
> Lakens (2013, p. 1)

Lakens (2013) argues that routine effect size reporting would improve meta-analyses in communication research; for your own work, ask what practical barriers stand in the way of adopting the convention in a field where effect sizes are rarely reported in standard write-ups.

13.5 Figures other people can read

Three figures are built. Before they are finished, they have to be made readable, and not only by the analyst who made them.

The first duty is **color**. The viewer-count figure tells its categories apart by color, and the histogram tells its two groups apart by color, and roughly one in twelve men has a form of **color-vision deficiency** that

makes some color pairs, red against green most notoriously, hard or impossible to distinguish. A figure whose meaning rests on color alone is a figure that some readers cannot read. There are two defenses, and good practice uses both. Choose a colorblind-safe palette, a set of colors picked to stay distinct across the common kinds of color blindness. And do not let color carry the message by itself: separate the lines by more than hue, or split groups into their own panels, so the figure still works in grayscale.

The second duty is the **alt text**, a short written description of the figure attached to it in the document's source. A reader using a screen reader cannot see the figure at all; the alt text is what they get instead, and a figure with none is, to them, simply missing. Good alt text is not the caption repeated. It states what kind of chart the figure is, what is on each axis, and what the figure shows, the takeaway, in a sentence or two. Each figure in this chapter carries one. In a Quarto document the alt text is written into the figure's `fig-alt` attribute, and writing it is not an afterthought. Being forced to say in one sentence what a figure shows is a good test of whether the figure shows anything at all.

A figure that is honest about its choices, legible without color, and described for readers who cannot see it is a figure that does its job for everyone. That is the standard a publication-quality figure is held to, and the three built here are meant to meet it.

Reporting conventions. For every group comparison in a visualization, report the effect size in the figure caption or a companion table. For group means: Cohen's d with a 95% confidence interval. For cross-tabulations: Cramer's V. For correlations: r and its CI. The `v2v::run_t_test()` and `v2v::run_chi_square()` functions compute all of these automatically and return formatted output ready for the caption. For two figures in your current analysis, report the effect size you would include in the caption; if one effect is small by Cohen's (1988) conventions, consider what that implies for the interpretive claim you planned to make.

13.6 Looking ahead

The histogram left a precise question on the table. The means of the two groups differ by about five characters, and the distributions behind those means are skewed, overlapping, and built from very different numbers of messages. Whether that five-character gap is a real difference or only sampling noise is not something the eye can settle. Chapter 13 takes it up. It is the inference chapter, where the study moves from describing what the sample looks like to testing what the difference is likely to mean.

13.7 References

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-24277-4

Wilkinson, L. (2005). *The grammar of graphics* (2nd ed.). Springer. https://doi.org/10.1007/0-387-28695-0

13.7.1 Graduate readings

Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for *t*-tests and ANOVAs. *Frontiers in Psychology*, *4*, 863. https://doi.org/10.3389/fpsyg.2013.00863

Part V: Inference and Publication

14 Chapter 13: Making the call

Listen in Dr. Leith's voice

Chapter 12 ended on a number and a doubt. The number was a gap: across the sample, gaming channels averaged 28.49 characters per chat message and non-gaming channels 33.70, a difference of a little over five characters. The doubt was whether that gap means anything.

The doubt is not idle. A sample almost always shows some gap. Split any group of people in two at random, measure their heights, and the two averages will differ a little, not because the halves are truly different but because no two samples come out identically. The five-character gap between gaming and non-gaming chat could be a real difference between the two kinds of channel. It could also be the ordinary wobble that any two samples produce. Looking at the histogram cannot tell the two apart, because both possibilities produce a gap.

Settling that doubt is the job of **inferential statistics**, and it is the work of this chapter. The chapter's title names what is at stake. After describing a sample, a researcher eventually has to make a call: is the pattern in the data a real feature of the world, or is it noise that will not survive the next sample? Chapter 13 is where the chat study makes that call about its central finding.

14.1 What a test actually asks

Start with what the question really is. The study did not measure those 34,762 messages because anyone cares about those particular messages. It measured them to learn something about gaming and non-gaming chat in general, the broad populations the sample stands in for. The sample is a means; the population is the point. And a sample, however carefully drawn, is not the population. It is a partial view, and partial views are noisy.

So the honest question is not "is there a gap in the sample," because there plainly is. It is "is the gap in the sample large enough that chance alone is an unconvincing explanation for it." That is the question a **hypothesis test** answers, and it answers it through a move that feels backward the first time.

The test does not start from the difference you suspect is real. It starts from its opposite, called the **null hypothesis**: the assumption that there is no real difference at all, that gaming and non-gaming chat have the same true mean length, and that the five-character gap is nothing but sampling noise. The test then asks a single question of the data: if the null hypothesis were true, how surprising would a gap this large be?

The logic is the logic of a skeptic. Imagine testing whether a coin is weighted by flipping it a hundred times. You would not try to prove it crooked directly. You would assume it fair, the null hypothesis, work out how often a fair coin would produce a result as lopsided as the one you got, and if that turned out to be a rare event, conclude that the fair-coin assumption is hard to believe. A hypothesis test treats the chat data the same way. Assume nothing is going on, measure how surprising the data are under that assumption, and if they are surprising enough, reject the assumption.

14.2 Choosing the test

There is not one hypothesis test. There are many, and which one applies depends on the kind of data in hand. This is where the measurement levels from Chapter 8, the nominal, ordinal, interval, and ratio distinctions, stop being vocabulary and start doing work.

The chat study's question pairs one continuous variable with one categorical split. Message length is a ratio variable, an exact count of characters. Gaming status is a nominal variable with two values, gaming and non-gaming. A continuous outcome compared across two groups is the textbook setting for the **t-test**, and that is the test this chapter runs.

A different question would call for a different test. Suppose the study asked instead whether a channel's gaming status is associated with its audience size, pairing `is_gaming` with the `viewer_band` variable built in Chapter 11. Those are two categorical variables, and the test for an association between two categorical variables is the **chi-square test**, available as `v2v::run_chi_square()`. The study does not pursue that question here, but the point generalizes: the variables choose the test. The V2V Hub keeps a test-decision-tree that lays out the full mapping, and consulting it before running anything is the habit to build. A test applied to the wrong kind of data produces a number, and the number is meaningless.

14.3 Running the test

The test itself is one function call. `v2v::run_t_test()` takes the data, the variable to compare, and the variable that splits it into groups:

```
v2v::run_t_test(analysis, value = message_length, group = is_gaming)
```

```
Welch two-sample t-test
message_length by is_gaming

  Mean, gaming         28.49 characters
  Mean, non-gaming     33.70 characters
  Difference           -5.21 characters

  t                    -4.94
  df                   3768.7
  p-value              < .001
  Cohen's d            -0.13  (negligible)

  Gaming and non-gaming channels differ in mean message length by a
  statistically significant but negligible amount.
```

The test is named in the first line as a **Welch** two-sample t-test, and the choice of the Welch version is deliberate. The ordinary t-test assumes the two groups have the same spread, and Chapter 12 showed plainly that they do not: the non-gaming standard deviation, 60.68, is far larger than the gaming standard deviation of 38.47. Welch's t-test drops that assumption, which is why it is the safer default whenever two groups might vary by different amounts.

The rest of the output is the result, and it has three parts worth reading slowly. The `t` value, −4.94, measures the gap between the means in units of its own uncertainty: roughly, how many widths of sampling noise separate the two averages. A value near zero would mean the gap is small compared to the noise; a value of about five means it is large compared to the noise. The sign is only direction, negative because gaming, the first group, sits below non-gaming. The `df`, or degrees of freedom, is a number the test uses to turn `t` into a probability; Welch's test computes it from the two groups' sizes and spreads, which is why it is not a round number. And the `p-value` is the answer to the question the whole test was built to ask.

14.4 Reading the p-value

The **p-value** reported here is "< .001," and stating exactly what that means takes care, because the p-value is the most misread number in statistics.

It means this: if gaming and non-gaming chat truly had the same mean message length, so that the null hypothesis were true, then drawing a sample with a gap at least as large as the observed five characters would happen less than one time in a thousand. The data are, in other words, very surprising under the assumption that nothing is going on. By the convention that researchers apply, a p-value below the threshold of .05 counts as **statistically significant**, and a p-value this small clears that bar easily. Chance alone is an unconvincing explanation for the gap.

That convention deserves a note of its own. The .05 threshold is exactly that, a convention, with no deep justification; a p-value of .049 and one of .051 describe almost identical evidence, and treating one as a discovery and the other as a non-event is a habit the field is right to question (Wasserstein & Lazar, 2016).

What matters more is being precise about what the p-value does not say. It is not the probability that the null hypothesis is true; the null is either true or false, and the p-value is a statement about data, not about hypotheses. It is not the probability that the result is a fluke. And, most important for what follows, it is not a measure of how large the difference is, or of how much the difference matters. A small p-value says the gap is probably not zero. It says nothing at all about whether the gap is big.

Justify your α. The .05 threshold is a convention, not a law. Run ten tests on one dataset without correction and you expect at least one false positive by chance alone, so declaring the number of tests and any correction procedure in advance, as pre-registration requires, is what keeps that inflation in check. Report the α you chose, state why it fits your design, and apply it consistently. The American Statistical Association's guidance frames the deeper point, that a threshold is a convention rather than a verdict:

> "Scientific conclusions and business or policy decisions should not be based only on whether a p-value passes a specific threshold."
>
> Wasserstein & Lazar (2016, p. 131)

Lakens et al. (2018) argue that different research contexts justify different thresholds; for your own study, ask what would justify using $\alpha = .01$ rather than .05.

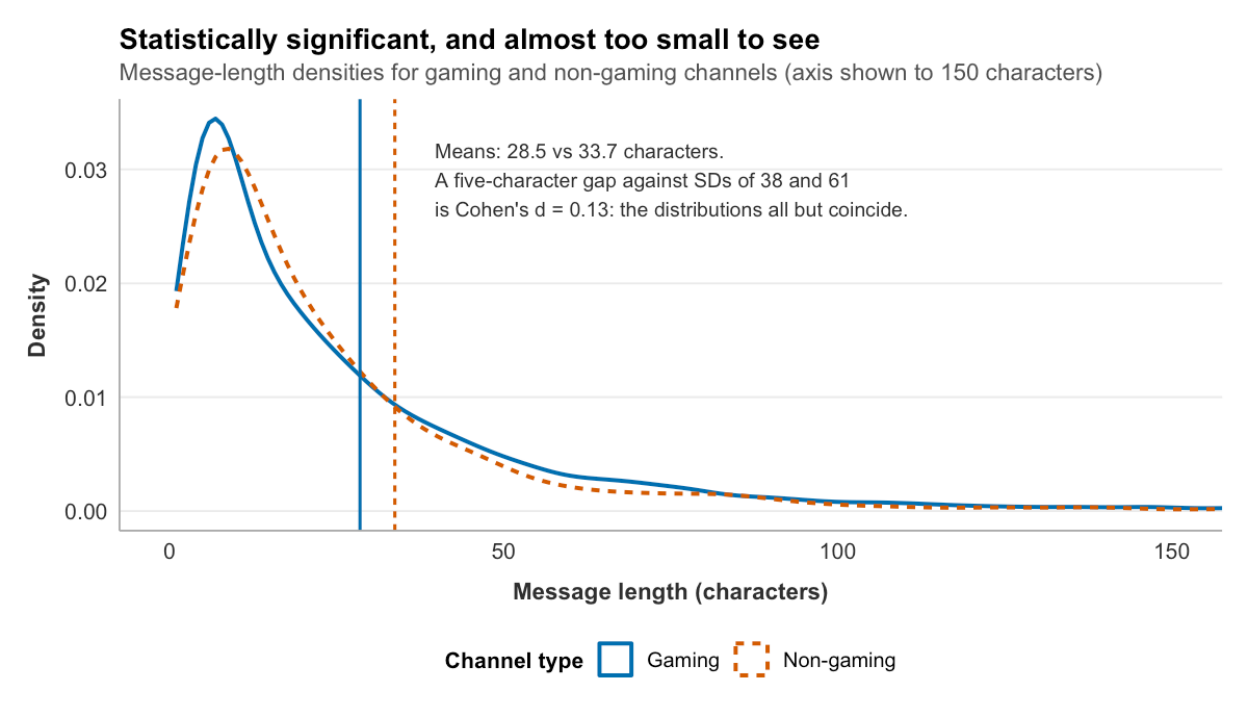
14.5 How big is the difference

For that, the p-value has to hand the question to a different number: an **effect size**.

A p-value answers "is there a difference." An effect size answers "how much of one." The effect size for a gap between two means is **Cohen's d**, and it works by expressing the gap not in raw characters but in units of how much the data naturally vary. The output reports a Cohen's d of −0.13.

To read that number, it needs a yardstick, and the standard one comes from Jacob Cohen (1988), who proposed rough benchmarks: a d around 0.2 is a small effect, around 0.5 a medium one, and around 0.8 a large one. A d of 0.13, smaller than even the "small" benchmark, falls into the range usually called negligible. The difference between gaming and non-gaming message length is real, in the sense that the test ruled out chance, and it is also trivially small.

This is worth dwelling on, because it sounds like a contradiction and is not. The raw gap is about five characters, roughly fifteen percent of the non-gaming average, and fifteen percent does not sound negligible. But Cohen's d measures the gap against the spread of the data, and message lengths vary enormously, with standard deviations of thirty-eight and sixty-one characters. A five-character difference set against that much variation is a ripple. The percentage sounds substantial; the effect size, which is the honest comparison, is not.



That leaves one question: how did a negligible difference come back statistically significant at all? The answer is the sample size. The test ran on 34,762 messages, and with a sample that large, a hypothesis test can detect a difference far too small to care about. This is the rule to carry forward: a large enough sample will make almost any difference, however tiny, statistically significant. Significance and size are different questions, and big samples pull them apart.

Power and the strength of a claim. Sample size cuts the other way too. A significant result from a well-powered study (at least 80%) is stronger evidence than the same result from an underpowered one: underpowered studies that reach significance do so partly through sampling variation, which inflates the effect-size estimate. Report the achieved power of your study alongside the *p*-value. `run_t_test()` returns Cohen's *d* with a confidence interval, which lets a reader judge how precisely the effect is pinned down, independently of the significance test.

14.6 Seeing the difference

The two facts, a real gap and a negligible one, are easier to hold together as a picture. Plotting each group's mean with its ninety-five percent confidence interval, a band that reflects how precisely the mean is estimated, puts both on the same axis:

```
means_ci <- analysis %>%
  filter(!is.na(is_gaming)) %>%
  group_by(is_gaming) %>%
  summarise(
    mean = mean(message_length, na.rm = TRUE),
    se   = sd(message_length, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  ) %>%
  mutate(
    lower = mean - 1.96 * se,
    upper = mean + 1.96 * se,
    group = if_else(is_gaming, "Gaming", "Non-gaming")
  )

ggplot(means_ci, aes(group, mean, colour = group)) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.14, linewidth = 0.9) +
  geom_point(size = 3.4) +
  v2v::scale_colour_v2v() +
  scale_y_continuous(limits = c(0, NA)) +
  labs(x = NULL, y = "Mean message length (characters)") +
  v2v::theme_v2v() +
  theme(legend.position = "none")
```

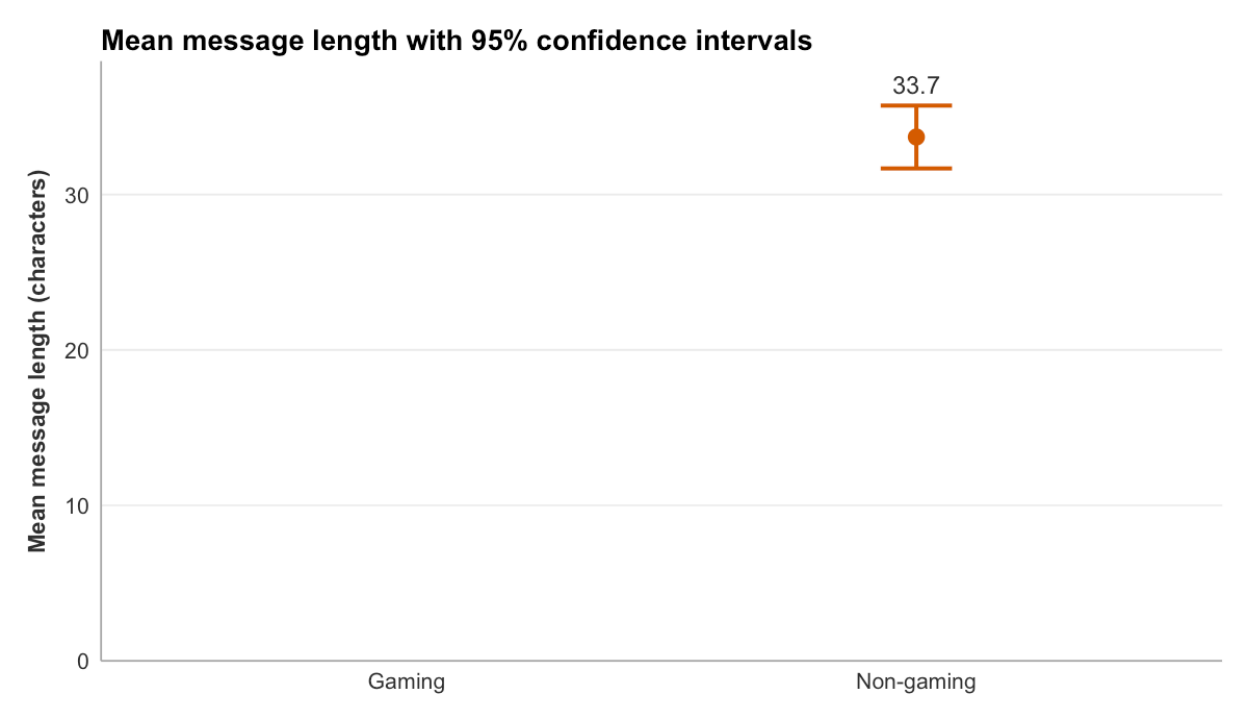


Figure 1: Mean chat-message length for non-gaming and gaming channels, each with a ninety-five percent confidence interval, on an axis that starts at zero.

The intervals are so short they nearly vanish behind the points, a direct consequence of the large sample: with tens of thousands of messages, each group's mean is pinned down precisely. That the two intervals

do not overlap is the visual signature of a statistically significant difference. And yet the two points sit close together near the top of the axis, both far above zero, so the eye reads them as almost the same height. That is the negligible effect size made visible. The gap is real enough that the intervals clear each other, and small enough that against the scale of the data it barely shows.

14.7 Statistical significance is not practical importance

Putting the two numbers together is the call the chapter is named for, and it is a more careful call than "significant or not."

The finding is this. Gaming and non-gaming channels do differ in their mean chat-message length; the difference is statistically significant, unlikely to be an accident of sampling. And the difference is negligible in size, about five characters, a word's worth, dwarfed by how much message lengths vary in the first place. Both halves are true, and a report that gave only the first half would mislead. A reader who heard "gaming and non-gaming chat differ significantly, $p < .001$" and nothing else would picture a real divide between two kinds of community. The effect size corrects the picture: there is a divide, and it is almost too small to see.

This is the difference between **statistical significance** and **practical significance**, and it is perhaps the single most important habit this book asks for. Statistical significance asks whether an effect is distinguishable from zero. Practical significance asks whether the effect is large enough to matter. They are not the same question, large samples make them diverge, and a finding is only honestly reported when both are on the page.

In practice that means a result is written up with the full set of numbers, in the compact form journals expect:

> Gaming channels produced shorter chat messages on average than non-gaming channels (28.49 vs. 33.70 characters), Welch's $t(3768.7) = -4.94$, $p < .001$, Cohen's $d = -0.13$.

And then, because a string of statistics is not an interpretation, it is written up a second time in a sentence a non-statistician can read: gaming and non-gaming channels do differ in how long their chat messages run, but the difference is so small that for most purposes the two are better thought of as alike than as different. That sentence is the call. It reports the real difference, refuses to inflate it, and tells the reader what the study actually found.

The pre-registration disclosure. A pre-registered study names it in the write-up: "This study's hypotheses, measures, and analysis plan were pre-registered on OSF prior to data collection at [URL]." Anything you ran beyond that plan is labelled exploratory and read with corresponding caution. For your own study, draft the disclosure you would put in the methods section, and list any deviations from the plan and how you would report them.

14.8 More than two groups: ANOVA

The t-test settled a question with exactly two sides: gaming versus non-gaming. Many questions have more. Suppose the study asked not whether gaming and non-gaming chat differ, but whether chat length differs across the specific games being played. Keep each channel's game category, the modal category built in Chapter 11, alongside `is_gaming`, restrict the data to its four largest categories, and four means appear rather than two:

```
game category       n     mean length
Fortnite          4000       33.23
Hearthstone       3004       36.82
Just Chatting     2429       39.03
League of Legends 4000       22.63
```

A t-test cannot settle this, because a t-test compares two means and there are four. Comparing them two at a time, Fortnite against Hearthstone, Hearthstone against Just Chatting, and so on, is worse than useless: it takes six separate tests, each carrying its own chance of a false positive, and those chances pile up. The test built for the whole comparison at once is the **analysis of variance**, or **ANOVA**. It asks a single question of all the groups together: is any of these means different enough from the rest that chance is an unconvincing explanation? `aov()` fits it and `summary()` reports it:

```
four_games <- analysis %>%
  filter(game %in% c("Fortnite", "Hearthstone",
                     "Just Chatting", "League of Legends"))

summary(aov(message_length ~ game, data = four_games))
```

```
              Df    Sum Sq  Mean Sq  F value  Pr(>F)
game           3    547281   182427    92.26  < 2e-16 ***
Residuals  13429  26552407     1977
```

The logic is the t-test's logic, widened. Where the t-test produced a `t`, ANOVA produces an **F**, and F compares two kinds of variation: the variation *between* the group means against the variation *within* the groups. If the four categories truly shared a mean, the spread between them would be no larger than the ordinary noise inside each, and F would sit near one. Here F is 92.26, far above one, and the p-value is minuscule: the categories do not all share a mean. League of Legends chat, at 22.63 characters, runs visibly shorter than Just Chatting at 39.03.

And, exactly as with the t-test, significance is not size. The effect size for ANOVA is **eta-squared** (η^2), the share of the total variation in message length that the game category accounts for. Here it is 0.02: category explains about two percent of the variation in how long messages run. The categories differ, the difference is real, and it is small, the same honest pair of facts the t-test forced.

14.9 From difference to model: regression

The t-test and ANOVA both compare group means. A third tool asks a subtly different question: not "do the groups differ" but "can one variable predict another." That tool is **linear regression**, and its reach is wide enough to contain the other two.

Return to the original two-group question, but fit it as a regression, an `lm()`, a linear model of message length on gaming status:

```
summary(lm(message_length ~ is_gaming, data = analysis))
```

```
              Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)     33.70       0.70      48.08   < .001
is_gamingTRUE   -5.21       0.74      -7.06   < .001
```

Read the two numbers. The intercept, 33.70, is the mean message length of the reference group, non-gaming channels. The coefficient on `is_gamingTRUE`, −5.21, is exactly the gap the t-test found: gaming channels average 5.21 characters fewer. The regression has re-derived the t-test's result and cast it in a new form, a baseline plus an adjustment. (Its `t` of −7.06 differs a little from the Welch test's −4.94 because a plain regression pools the two groups' spread rather than letting them differ; the coefficient, the difference itself, is identical.)

That the regression reproduces the t-test is not a coincidence. A t-test is a regression with a single two-level predictor; an ANOVA is a regression with a single many-level predictor. All three belong to one framework, the **general linear model**, and once a study is written as a regression it can grow in ways a t-test cannot: a second predictor added, a continuous predictor rather than a categorical one, several influences weighed at once. This chapter's study needs only the two-group comparison, so the t-test is enough. But regression is where inference stops being a menu of separate tests and becomes one extensible way of modeling how one thing depends on another. Its effect size here tells the same story a third time: the model's R^2, the share of variance it explains, is about 0.001, so gaming status accounts for almost none of the variation in message length.

The assumptions underneath `aov()` and `lm()`. Running the model and reading the output is the first step; the numbers earn trust only once their assumptions are checked. ANOVA and ordinary regression assume the groups share a common variance (homoscedasticity), and Chapter 12 showed the chat groups plainly do not, which is why the two-group test used Welch's correction, and why `oneway.test()` (Welch's ANOVA) is the safer choice over the pooled `aov()` above. A significant ANOVA says only that *some* pair of means differs, not which; naming the pairs takes post-hoc comparisons (Tukey's HSD, `TukeyHSD()`) that correct for the multiple looks. A regression's credibility rests on its residuals: `plot(model)` draws the diagnostic quartet: residuals-versus-fitted for linearity and equal spread, a Q–Q plot for normal residuals, and leverage for influential points. And whether an added predictor earns its place is a question of model comparison, adjusted R^2 and AIC or a nested *F*-test, not raw R^2, which never falls when a predictor is added. For your own design, name the test it calls for, the assumption most at risk in your data, and the diagnostic or robust alternative you would use to handle it.

14.10 How much the finding depends on one decision

A test tells you whether a gap is larger than sampling noise. It cannot tell you whether the gap survives a different but equally defensible definition of the groups being compared. Here it does not, and that turns out to matter more than the test.

`is_gaming` was built as a property of the **channel**: each channel's modal category across the week, applied to every message that channel sent. The same data supports a second operationalization. Label each **message** by the category the channel was actually streaming when it was sent, read from the nearest earlier stream snapshot.

How the label is defined	gaming	non-gaming	gap
the channel's modal category	28.49 (n = 31,305)	33.70 (n = 3,457)	+5.21
the category on screen at the time	29.11 (n = 30,707)	28.13 (n = 4,044)	−0.98

Under the second definition the gap reverses direction, falls to about one character, and is no longer statistically significant: Welch's $t(4993.6) = 1.34$, $p = .18$.

Neither calculation is wrong, and they are not in competition. They answer different questions, and the distance between them is concentrated in one fact worth seeing plainly: **the non-gaming group is five channels.** `bobross`, `hitch`, `xqcow`, `tjsmith` and `darksydephil`, three of them at the corpus's 1,000-message cap, 3,457 messages between them. `xqcow` alone is 29 percent of the group, and that channel is filed under Just Chatting for most of the week while spending a good deal of it playing games. Under the channel-level label every one of its messages is non-gaming. Under the message-level label most of them are not.

Two things follow, and both belong in the write-up.

The **unit of analysis** problem is doing real work here. The label varies across 48 channels, five of which are non-gaming, while the test is computed over 34,762 messages. The p-value answers "could 34,762 independent draws have produced this gap by chance". The question that matters is "could five channels", and the answer to that one is much less comfortable.

And the **finding has to be stated at the level it was actually measured**. Not "gaming chat is shorter", but "chat on the five predominantly non-gaming channels in this sample is longer on average, by an amount too small to matter, and the difference does not survive relabelling messages by what was on screen at the time".

That is a duller sentence, and it is the one the data supports.

14.11 Looking ahead

The chat study now has all of its pieces. It has a question and a theory, a prospectus, a codebook tested for reliability, a sample, a clean dataset, three figures, and a tested finding interpreted honestly. What it does not yet have is a form. The pieces are scattered across scripts and outputs, and a study that cannot be read as a single coherent document is not yet finished. Chapter 14 is where it becomes one. It is the publication chapter, where everything built since Chapter 9 is assembled into a single reproducible report and put where other people can read it.

14.12 References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates.

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

14.12.1 Graduate readings

Lakens, D., Adolfi, F. G., Albers, C. J., Anvari, F., Apps, M. A. J., Argamon, S. E., … Zwaan, R. A. (2018). Justify your alpha. *Nature Human Behaviour*, *2*, 168–171. https://doi.org/10.1038/s41562-018-0311-x

15 Chapter 14: The one-click report

Listen in Dr. Leith's voice

The chat study is, in almost every sense, done. Across the chapters since the prospectus it has acquired everything an empirical study needs. It has a question and a theory: do gaming and non-gaming Twitch channels differ in how their audiences talk, and what would a uses-and-gratifications account predict. It has a codebook, tested for reliability. It has a sample, drawn with a defensible procedure. It has a clean dataset, a set of figures, and a tested finding interpreted with care. Every hard part is finished.

And yet the study is not finished, because all of that lives in pieces. The codebook is one document. The wrangling happened in an R script. The figures are a folder of images. The test result scrolled past in a console. A study scattered across a dozen files is not yet something another person can read, check, or build on. It is raw material for a study, not the study itself.

This last chapter gives the work a form. It assembles every piece into one report, makes that report reproducible, gathers it with the study's other documents into a single site, and publishes that site where anyone can find it. The chapter's title names the goal. By the time the study has been built the right way, producing the finished, public report is close to a single action: one click.

Two additional deliverables. Beyond the reproducible portfolio, the graduate study produces two further deliverables: a *white paper* structured for academic publication and a *conference poster* formatted for a communication research conference.

15.1 A report that rebuilds itself

Begin with the report, and with the ordinary way of writing one. Run the analysis in R. Copy the numbers into a word processor. Take screenshots of the figures and paste them in. Write the prose around them. This works, and it is quietly fragile. The instant the analysis changes, and analyses always change, a coding rule corrected, a few hundred messages added, every pasted number and every pasted figure is silently out of date, and nothing in the document says so. The write-up and the analysis drift apart, and only luck reveals it.

The alternative is to stop copying and put the analysis inside the document. A **Quarto document**, saved with a `.qmd` extension, is a plain-text file that interleaves ordinary prose with **code chunks**, blocks of live R code. When the document is rendered, Quarto runs every chunk, captures what it produces, the tables, the figures, the test statistics, and places that output into the finished document where the chunk sat. Numbers can also be computed inline, mid-sentence:

```
The analysis dataset held `r nrow(analysis)` coded messages, of which
`r sum(!is.na(analysis$is_gaming))` carried a gaming-status label.
```

In the rendered report those backtick expressions become the actual counts. Nothing is typed by hand, so nothing can fall out of date. A figure works the same way. The histogram from Chapter 12 is not pasted in; it is produced by a chunk that lives in the report:

````
````markdown
```{r}
#| label: fig-msglen
#| echo: false
#| fig-cap: "Message length by channel type."

ggplot(msglen, aes(message_length, fill = is_gaming)) +
  geom_histogram(binwidth = 5) +
  v2v::theme_v2v()
```
````
````

This is **reproducible publication**, and it has two payoffs that matter. The report cannot drift from the analysis, because the report is the analysis: re-render it and every number updates itself. And the report can be checked, because anyone holding the source and the data can render it and get exactly the same results, which is the standard a scientific claim is supposed to meet (Peng, 2011). The idea is older than Quarto, older than R. Donald Knuth named it **literate programming** (Knuth, 1984): the conviction that runnable code and the prose explaining it belong together in one document rather than apart in two. Quarto is a modern descendant of that idea, and the "one click" of this chapter's title is its render button. One action, and the document rebuilds itself from source.

15.2 The shape of a report: IMRaD

A document that rebuilds itself still needs a structure, and empirical research has a standard one, settled enough to have an acronym: **IMRaD**, for Introduction, Methods, Results, and Discussion. Four sections, in that fixed order, and the order is itself a small argument, carrying the reader from why the study was done to what it found to what that means.

The white paper. The white paper follows IMRaD format (Introduction, Methods, Results, Discussion), uses APA 7th edition throughout, reports effect sizes and power statistics, and includes a pre-registration disclosure statement in the methods section. Its discussion situates the findings within the theoretical literature from Chapter 5. For your own study, convert the key results and discussion into IMRaD structure, and ask what is lost in the move and what must be added to meet white paper standards.

The **Introduction** states the question and why it is worth asking. For the chat study this is the work of the prospectus from Chapter 6: the question of whether gaming and non-gaming Twitch chat differ in message length, and the uses-and-gratifications lens that framed talk on a stream as behavior an audience chooses for its own reasons.

The **Methods** section says exactly what was done, in enough detail that another researcher could repeat it. This is the home of the codebook from Chapter 8, the sampling procedure and the inter-coder reliability check from Chapter 10, and the wrangling steps from Chapter 11. Methods is the section that earns the reader's trust, and it is the section a reproducible document supports best, because the code that did the work is right there to be read.

Methods section requirements. A graduate methods section includes three elements: (1) a data provenance statement (Chapter 9); (2) an intercoder reliability report, including the κ value, training protocol, and threshold applied (Chapter 8); and (3) the pre-registration disclosure with OSF URL (Chapter 5). Each element is a transparency marker that signals to reviewers that the study meets graduate-level evidentiary

standards. The pre-registration disclosure in particular protects a distinction that preregistration exists to make legible:

> "Preregistration distinguishes analyses and outcomes that result from predictions from those that result from postdictions."
>
> Nosek et al. (2018, p. 2600)

The **Results** section reports what was found, plainly and without interpretation. The three figures from Chapter 12 and the t-test from Chapter 13 belong here. Results states the facts: the distributions, the group means, the test statistic with its degrees of freedom, its p-value, and its effect size. It does not yet say what they mean.

The **Discussion** says what the results mean. This is where the interpretation from Chapter 13 belongs: that the difference between the two kinds of channel is statistically real but negligible in size, that gaming and non-gaming chat are better thought of as alike than as different in message length, and what that answer implies for the uses-and-gratifications question raised in the Introduction. Discussion is also where an honest study states its limits.

A short **abstract** summarizing all four sections usually sits in front, and references close the document. The V2V Hub ships an `imrad-template.qmd` with this skeleton already in place. The structure is not bureaucracy. It is a contract with the reader: anyone who has read empirical work knows it holds these parts in this order, and knows exactly where to look for the question, the procedure, the finding, and the meaning.

15.3 One site, many pages

The report is not the only document the study produced. There is also the codebook, the record of how the sample was drawn, the log of wrangling decisions. In a finished study these are not crushed into one enormous file. They become separate pages of one **site**.

Chapter 9 built the scaffold for this with `v2v::new_portfolio()`, which created a project folder already organized as a Quarto website. The piece that turns a folder of `.qmd` files into a navigable site is a configuration file named `_quarto.yml`. It names the site and lists its pages and the navigation that links them:

```yaml
project:
  type: website

website:
  title: "Twitch Chat Study"
  navbar:
    left:
      - text: "Report"
        href: index.qmd
      - text: "Codebook"
        href: codebook.qmd
      - text: "Figures"
        href: figures.qmd
```

The IMRaD report is one page, the codebook another, a figure gallery another, each its own `.qmd`, all rendered together and linked by a shared navigation bar. Running `quarto render` on the project builds every page at once and writes out a complete website, ready to publish. This is the portfolio synthesis the study has been moving toward since Chapter 9: not a pile of files, but one coherent site that holds the entire study and lets a reader move through it.

15.4 Going live

A website sitting in a folder on a laptop is still private. Publishing it means placing it somewhere with a public address, and the standard free option for a Quarto site is **GitHub Pages**, which serves a website directly from a GitHub repository.

Publishing has a checklist, and the checklist is exactly where studies fail in their final week. `v2v::deploy_portfolio()` runs the checklist and then publishes:

```
v2v::deploy_portfolio()
```

```
Pre-flight checks
  renv library in sync ......... ok
  no uncommitted changes ....... ok
  _quarto.yml valid ............ ok

Rendering 4 pages ... done
Publishing to GitHub Pages ...

Live at: https://username.github.io/twitch-chat-study/
```

Before it publishes anything, the function runs three **pre-flight checks**. It confirms that renv, the tool from Chapter 9 that records the project's exact package versions, is in sync, so the published site was built with the versions the source declares. It confirms there are no uncommitted Git changes, so the site that goes live matches the source that is actually saved. And it confirms that `_quarto.yml` is valid, so the render will not fail partway through. Only when all three pass does it run the publish step itself.

The checks are not red tape. Each one heads off a specific and demoralizing failure: a site that renders on its author's machine and nowhere else, a published page that does not match the work that was actually done, a deployment that collapses at eleven at night the evening a portfolio is due. The function exists so that the last step of a long study is the calm one. What it produces is a URL. The chat study is now a public, reproducible website, one that anyone can read and anyone can rebuild from its source.

The conference poster. A standard academic communication conference poster is 36 × 24 inches, landscape orientation, structured as: title/authors/institution; abstract (~100 words); background and hypotheses; condensed methods; key results with one or two `theme_v2v()` figures; discussion and implications; QR code linking to the OSF pre-registration and/or Quarto portfolio. The poster is your 90-second pitch. Design the layout of your own 36 × 24 poster, and decide which single figure from your analysis best communicates your main finding in 30 seconds.

15.5 The reflection

One short piece remains, and the portfolio is not complete without it: a one-paragraph **reflection**. It is not part of IMRaD and it is not a result. It is the researcher's own brief, candid account of what the study taught them and what a second attempt would do differently.

A good reflection is specific. It names what turned out harder than expected, where a coding rule proved ambiguous in practice, what the dataset could not answer because of how it was collected, what a future version of the study would change. It resists the two easy failures, the bland report that everything went well and the anxious confession that everything went wrong. A reflection is not an apology. It is evidence of judgment: a researcher who can say precisely where their study is weak is a researcher who genuinely understands it. It is kept to a paragraph on purpose. One honest paragraph is worth more than a page of hedging.

15.6 Looking ahead

This is the last chapter, and looking ahead now means looking past the book.

The chat study is finished. A question became a design. A design became data. Data became figures and a test. The whole of it became a reproducible site with a public address, a study another person can read, check, and extend. The dataset along the way happened to be Twitch chat, but nothing in the path depended on that. The sequence is the same for any dataset a reader will ever meet: a question, a theory, a codebook, a sample, a reliability check, a clean dataset, an honest figure, a careful test, and a reproducible report.

That sequence is what the book's title has been describing all along. **Vibes** are where a study starts, a hunch, an interesting feeling that some corner of the world works a particular way. **Variables** are what discipline turns the vibe into, something defined precisely enough to measure, and measured carefully enough to test. The distance between the two is the whole of research methods, and a reader who has now walked it once, from a vague curiosity about Twitch chat to a published finding about it, can walk it again with a dataset of their own. The tools will keep changing. The path from vibes to variables does not.

15.7 References

Knuth, D. E. (1984). Literate programming. *The Computer Journal*, *27*(2), 97-111. https://doi.org/10.1093/comjnl/27.2.97

Peng, R. D. (2011). Reproducible research in computational science. *Science*, *334*(6060), 1226-1227. https://doi.org/10.1126/science.1213847

15.7.1 Graduate readings

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

Appendices

16 Data Dictionary

The Twitch Dataset

Every example, exercise, and walkthrough in *Vibes to Variables* derives from a single dataset: a 2018-11-18 to 2018-11-24 snapshot of public Twitch IRC chat and concurrent stream metadata. One corpus, six days, two tables. Students learn one community deeply rather than four shallowly.

The full dump contains 21,964,296 chat messages and 590,876 stream snapshots across 1,695 channels and 457 games. For the book, that has been distilled to a 50-channel working corpus shipped with the `v2v` R package. Eight of those channels are **anchor channels**, referenced repeatedly in the textbook's worked examples. The other 42 are a stratified random sample by chat-volume decile from the broader population of 1,690 joinable channels, drawn so that Chapter 10's stratified-sampling lesson works against a real sample frame rather than a curated friends list.

16.1 Loading the data

```
library(v2v)

# Direct (data objects exposed by the package)
data(twitch_chat_sample,    package = "v2v")
data(twitch_streams_sample, package = "v2v")

# Or via the loader functions introduced in Chapter 9
chat    <- v2v::twitch_chat()
streams <- v2v::twitch_streams()
```

Both forms return tibble-compatible data frames. The loader functions accept optional filter arguments (`channels`, `games`, `n`) for the worked examples in later chapters.

16.2 `twitch_chat_sample`: chat-message rows

Column	R class	Range or values	Notes
`id`	integer	primary key	Surrogate identifier from the source database
`channel`	character	50 distinct values	**In the raw dump, prefixed with `#`** (IRC convention). The shipped fixture has the prefix stripped so `channel` joins cleanly to `twitch_streams_sample`. Recovering the IRC representation is a Chapter 11 exercise

Column	R class	Range or values	Notes
`sender`	character	tens of thousands distinct	Pseudonymous Twitch username; public on the IRC feed; chosen by the user
`message`	character	1 to 509 chars	UTF-8; includes Twitch emote tokens (`LULW`, `OMEGALUL`, `monkaS`, `Pog`, `Kappa`, etc.) as first-class lexical items
`date`	integer (bigint)	1542578125327 to 1543083667610	**Unix epoch in milliseconds**, not seconds. Convert with `as.POSIXct(date / 1000, origin = "1970-01-01", tz = "UTC")`. This is the headline Chapter 11 conversion lesson

Approximately 35,000 rows total. Per-channel sample size is `min(1000, available)`: most channels contribute the full 1,000 messages, but 19 of the 50 contribute fewer because they had fewer than 1,000 messages during the collection window. This per-channel imbalance is itself part of the population structure students should see.

16.3 The 8 anchor channels

These channels appear in named-vignette worked examples throughout the textbook:

Channel	Top game in the window	Why this channel earns named-anchor status
`xqcow`	Just Chatting	Top chat volume in the entire corpus; variety streamer (Just Chatting is a non-game category)
`forsen`	The Elder Scrolls V: Skyrim	RPG; concentrated repeat-audience (the lowest sender-to-message ratio in the anchor set)
`sodapoppin`	Hand Simulator	Variety streamer with a long-tail game; 22,919 average viewers despite the obscure category
`asmongold`	World of Warcraft	MMO; broad-audience streamer

Channel	Top game in the window	Why this channel earns named-anchor status
`loltyler1`	League of Legends	MOBA; peak audience in the focal set (78,340 concurrent viewers)
`disguisedtoast`	Pokemon Let's Go	Pokemon Let's Go launched November 16, 2018, two days before the collection window opened
`giantwaffle`	Rocket League	Esports-adjacent; highest snapshot density among the top-chat streamers
`bobross`	Art	Non-gaming contrast. The late painter's PBS series, on a continuous Twitch loop since November 2015

16.4 The 42 stratified additions

The remaining 42 channels are drawn by stratified random sample: the 1,682 non-anchor channels are sorted by chat volume during the collection window, partitioned into 10 deciles, and 4 channels are drawn at random from each decile (plus 2 random fills). Seed: 20261113.

The resulting 42 channels span:

- The full chat-volume distribution, from channels with over 100,000 messages in the window down to channels with single-digit message counts.
- Additional non-gaming categories beyond Bob Ross: comedy (RiffTrax), sports (NFL Redzone), variety streaming.
- Mid-tier gaming streamers across MOBA, FPS, RPG, sports, and party-game categories.
- Smaller communities and niche channels that round out what the long tail of Twitch actually looked like in late 2018.

A full per-channel breakdown lives in the V2V data report.

16.5 `twitch_streams_sample`: stream-snapshot rows

Column	R class	Range or values	Notes
`id`	integer	primary key	Surrogate identifier
`channel`	character	50 distinct values	No `#` prefix; matches the normalized `chat_log.channel` in `twitch_chat_sample`
`title`	character	up to 128 chars; some NULL	Current stream title at snapshot time

Column	R class	Range or values	Notes
`game`	character	25+ distinct; some NULL	Current Twitch category. Includes both gaming categories and non-gaming categories (Just Chatting, Art, Music & Performing Arts, Sports & Fitness, Talk Shows & Podcasts)
`viewers`	integer	0 to ~78,000	Concurrent viewer count, as reported by Twitch at snapshot time. Not de-botted
`date`	integer (bigint)	1542578200214 to 1543083621279	Unix epoch in milliseconds; same conversion rule as `twitch_chat_sample.date`

Approximately 32,000 rows. This is the **full** focal-50 stream coverage (no sampling); the underlying snapshot count is small enough to bundle in entirety.

16.6 Joining the tables

A typical analysis joins each chat message to the nearest-prior stream snapshot on the same channel. The pattern Chapter 11 develops:

```
library(dplyr)
library(lubridate)

chat <- twitch_chat_sample %>%
  mutate(ts = as.POSIXct(date / 1000, origin = "1970-01-01", tz = "UTC"))

streams <- twitch_streams_sample %>%
  mutate(ts = as.POSIXct(date / 1000, origin = "1970-01-01", tz = "UTC"))

joined <- chat %>%
  arrange(channel, ts) %>%
  left_join(
    streams %>% arrange(channel, ts) %>% select(channel, ts, game, viewers),
    by = join_by(channel, closest(ts >= ts))
  )
```

On the full focal corpus, this join produces a 99.95% match rate. The 0.05% unmatched chat messages are messages whose timestamp precedes the first stream_log snapshot for their channel during the collection window: a Chapter 11 discussion of `left_join` vs `inner_join` semantics.

16.7 What this dataset cannot answer

A six-day snapshot supports a narrow class of research questions well, and rules others out:

- **Supported:** within-stream dynamics (chat density vs viewer trajectory), across-channel comparisons (how does Bob Ross's audience differ from xqcow's), category-level audience structure (gaming vs non-gaming), codebook design and reliability work on real chat text, sampling design and stratification, the long-tail Twitch ecosystem.
- **Not supported:** longitudinal cohort analysis, seasonality, holiday effects, long-term audience evolution, post-collection Twitch policy changes.

Chapter 6 (The Prospectus) teaches students to align research question scope with data scope. This dataset is the standing demonstration of why that alignment matters.

16.8 Ethics framing in brief

Twitch IRC chat is public by default and visible to every concurrent viewer in real time. Usernames are pseudonymous and self-chosen. The dataset contains no IP addresses, no email addresses, no Twitch internal user IDs, no payment data, no subscription status, and no whisper or private-message content. Three published manuscripts have already analyzed adjacent slices of this data infrastructure under an exempt-by-design IRB determination based on the public-channel basis. Chapter 3 walks the full ethics determination from Belmont Report principles through SIUE IRB tiers, using this dataset as the worked example.

16.9 Full data report and provenance

The detailed data report (maintained in the SIM Lab research tree at `D:/hub/academic/research/sim-lab/v2v/data/report/v2v-data-report.md`) documents the full extraction pipeline, top-line statistics on the full 22-million-row corpus, the 50-channel sampling design with seed and per-channel breakdown, chapter-keyed analytical results that match every number quoted in chapter prose, and the population vs. sample distinctions students should learn to track. Students who want to know "where did this number come from" should be pointed to that report.