

Turn customer feedback into improvements you can explain

Revised walkthrough | September 28, 2026

Start with what people said, find the recurring issues, and connect each recommendation to its evidence. This walkthrough uses six fictional workshop comments and a current facts sheet. The result is a feedback analysis you can inspect, rather than a list of plausible business advice.

1. Prepare the feedback so you can trace it

Export reviews, survey responses, support notes or customer emails into a simple table. Keep one response per row, with an ID, the comment, and any useful date or product context. Remove personal details that are unnecessary for the analysis. Keep an untouched copy of the export.

The practice CSV has IDs F01 through F06. Two people were unclear about supplied equipment, two wanted preparation information, and two wanted more practice. Some comments also contain praise. These are overlapping themes, not six neatly separated categories.

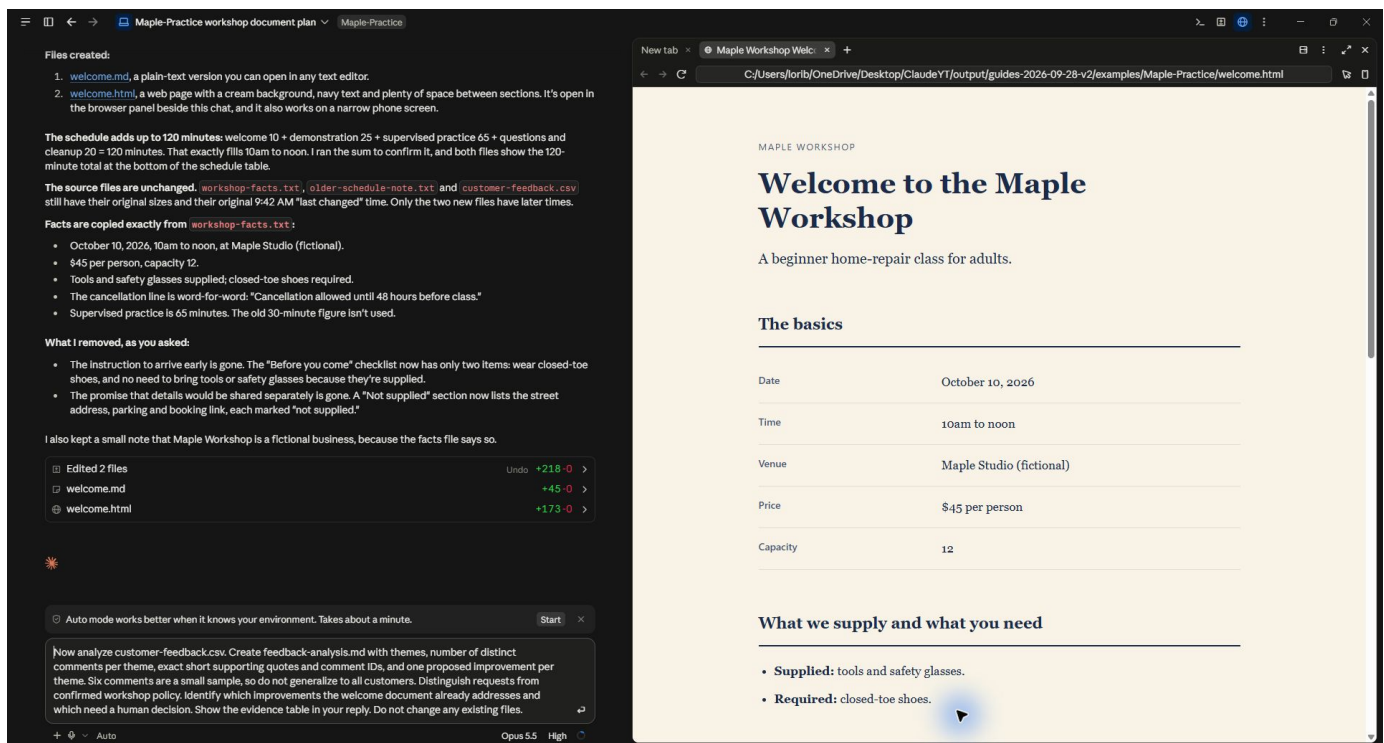
For a real business, include the relevant product version or date when available. A complaint about last year's process may already be fixed. Without that context, an AI can correctly summarize a complaint and still recommend solving an outdated problem.

2. Give the AI the feedback and the current context

In the demonstrated route, Claude Code has access to a dedicated practice folder containing the CSV and current workshop facts. You can also attach these files to ChatGPT Work or Claude. Use the file route available in your account; the analytical request is the same.

Ask for themes, counts, original evidence and recommendations separately:

Analyze customer-feedback.csv. Create feedback-analysis.md with themes, number of distinct comments per theme, exact short supporting quotes and comment IDs, and one proposed improvement per theme. Six comments are a small sample, so do not generalize to all customers. Distinguish requests from confirmed workshop policy. Identify which improvements the welcome document already addresses and which need a human decision. Show the evidence table in your reply. Do not change any existing files.

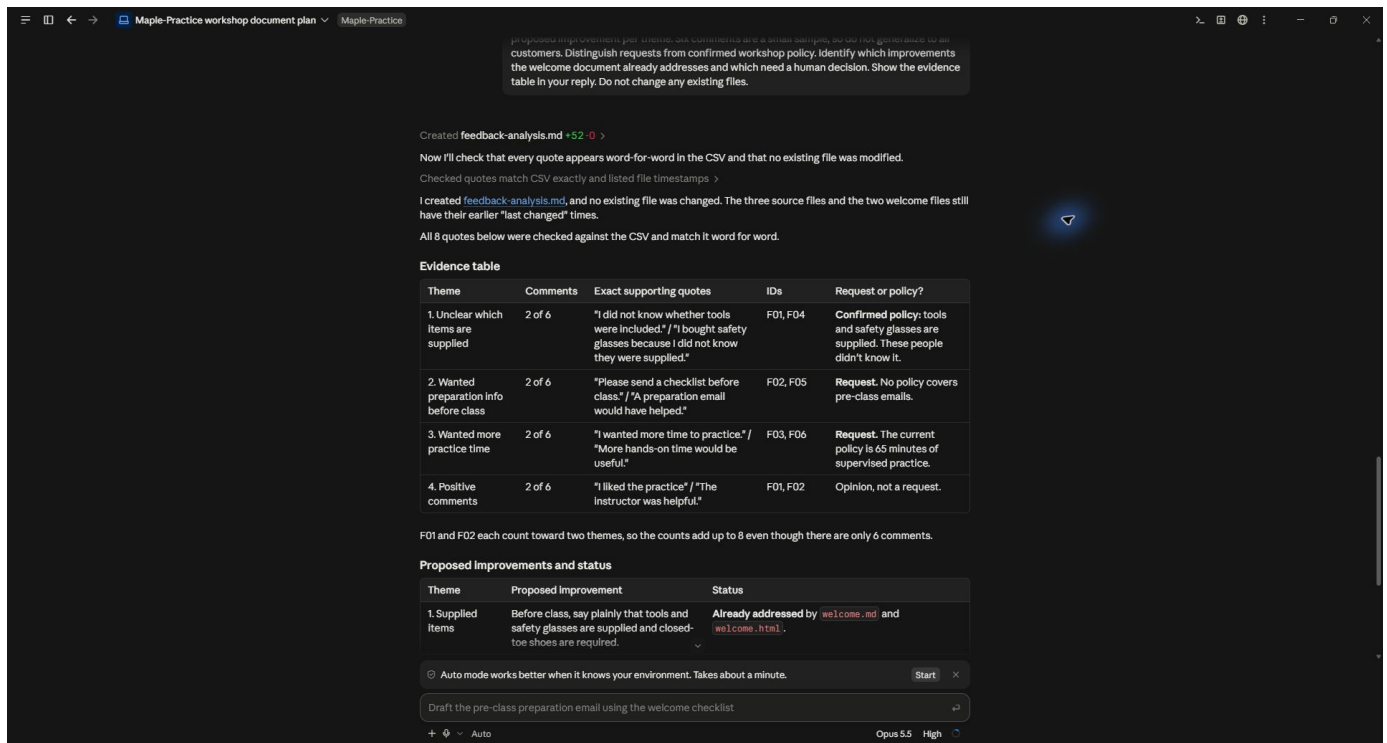


Feedback analysis prompt in Claude Code. [Open full-resolution screenshot.](#)

The request names the actual file, the evidence required, and the decision the analysis should support.

3. Check the themes against the comments

The response grouped equipment clarity under F01 and F04. One comment said, “I did not know whether tools were included.” Another described buying safety glasses because the attendee did not know they were supplied. Those comments support an improvement to pre-class communication. They do not support changing the equipment policy, which already says tools and glasses are provided.



Evidence table with counts, quotes and feedback IDs. [Open full-resolution screenshot.](#)

Each theme has two of six comments. F01 and F02 also contain positive feedback, so theme counts overlap.

Open the CSV and verify each quoted passage and ID. Count distinct responses within a theme, rather than counting every sentence as a different customer. If the AI reports percentages, ask it to show the numerator and denominator. With this sample, “two of six comments” communicates the evidence more clearly than an impressive-looking percentage.

A useful correction request is: “F01 contains both praise and a complaint. Keep both, count the response once within each theme, and explain that categories overlap.” Never force mixed comments into a single sentiment label just to make the chart look tidy.

4. Separate the finding from the decision

The actual analysis proposed clearer equipment instructions and a preparation checklist. It also suggested reviewing practice time. That last item needs more context: the current schedule already has 65 minutes of practice, while an older note says 30. The comments have no dates, so we cannot tell which schedule the attendees experienced.

		they were supplied.	didn't know it.	
2. Wanted preparation info before class	2 of 6	"Please send a checklist before class." / "A preparation email would have helped"	F02, F05	Request. No policy covers pre-class emails.
3. Wanted more practice time	2 of 6	"I wanted more time to practice." / "More hands-on time would be useful."	F03, F06	Request. The current policy is 65 minutes of supervised practice.
4. Positive comments	2 of 6	"I liked the practice" / "The instructor was helpful."	F01, F02	Opinion, not a request.

F01 and F02 each count toward two themes, so the counts add up to 8 even though there are only 6 comments.

Proposed Improvements and status

Theme	Proposed Improvement	Status
1. Supplied Items	Before class, say plainly that tools and safety glasses are supplied and closed-toe shoes are required.	Already addressed by <code>welcome.md</code> and <code>welcome.html</code> .
2. Preparation Info	Send attendees a short checklist before class.	Partly addressed: the welcome document has a "Before you come" checklist. You need to decide whether to email it, when, and who sends it.
3. Practice time	Review whether 65 minutes of practice is enough.	You need to decide. The welcome document shows the current 65 minutes; it doesn't change them.
4. Positive comments	Keep instructor-supervised practice if the schedule changes.	No action needed.

Limits of this sample

- With only 6 comments, "2 of 6" means two people said this. It doesn't show how customers in general feel.
- The comments have no dates or session details. So we can't tell whether the people asking for more practice had 65 minutes or the shorter 30 minutes from the older draft. A longer schedule might already fix what they asked for.
- Any longer practice time would have to fit within 10am to noon, or the class times would need to change.

Auto mode works better when it knows your environment. Takes about a minute. Start ×

Draft the pre-class preparation email using the welcome checklist ↗

+ 🔍 Auto Opus 5.5 High

Improvements separated from current policy and unresolved decisions. [Open full-resolution screenshot.](#)

The output distinguishes changes already covered by the welcome document from decisions the organizer still needs to make.

For your own analysis, add an action table with the recommendation, supporting IDs, expected benefit, effort, owner and open question. Have the AI leave unknown owner or effort fields unfilled instead of guessing. Prioritize problems you can substantiate and act on.

Turn the verified findings into an action list. Separate quick communication fixes from changes that need more research. Do not treat the number of mentions as proof of severity. Show what evidence would change each recommendation.

5. Apply one improvement and review it

Choose a small, specific change first. For Maple Workshop, the welcome document can state that tools and safety glasses are supplied and that attendees should wear closed-toe shoes. That directly addresses a documented confusion without inventing a new policy.

If you want the AI to revise a document, identify the exact file and ask for a new version. Compare it with the current policy. Feedback should inform the rewrite; it should not silently become the source of truth for dates, prices or promises.

After the change has been used, gather new feedback using the same question and preserve dates. That makes a later comparison more meaningful. AI can help organize the evidence, but six comments cannot establish a business-wide trend or prove that a change caused an improvement.

Practice files and demonstrated scope

Use [the six-comment CSV](#), [current workshop facts](#), and [the actual analysis](#). All data is fictional. The screenshots show a real Claude Code run on September 28, 2026. The example demonstrates grouping, evidence tracing and recommendations, not a customer survey or measured business outcome.