

Owner Rules Written Into a Spreadsheet Roughly Doubled AI Answer Scores on 12 Synthetic Businesses

Zach Kellman

Actual Intelligence Labs

September 2026

Abstract

An AI handed a spreadsheet can read every cell and still get the answer wrong, because the rules for reading the cells live in the owner’s head. Sheet Geek, an open-source tool, asks the owner what the data cannot answer and writes the answers into the workbook as a plain tab of notes. We tested that tab in a study whose plan, tool and analysis were frozen before any test business existed. On twelve synthetic businesses, with an agent playing each owner from a written brief, four models answered the same request with and without the tab, and two blind judges from two model families scored every answer. Over the 73 pairs that survived a pre-registered isolation rule, the open-request score rose from 5.25 to 10.32 of 15. The primary test, a two-sided Wilcoxon signed-rank test over the 12 business-level mean differences, gave +5.05 points (95% bootstrap interval +4.19 to +5.78), positive on 12 of 12 businesses, $p = 0.00049$. Plain questions with a computed key were answered right 60.7% of the time with the tab and 10.5% without. The gain came from Claude Opus 5.5 (+6.98) and GPT-5.6-Luna (+7.96); Claude Haiku 4.5 gained +0.13, and the isolation rule removed 23 of Claude Sonnet 5’s 24 pairs. The claim is narrower than the headline. No arm had the owner’s brief pasted in as plain notes, and only 42.5% of the owner facts came from the tool’s own questions, so the study shows that owner rules in the file help, not that the tool’s interview put them there. Claude Opus 5.5 built and verified the businesses, played the owners, gated the brains, was judge 1 and was an answering model. Brain answers named the notes tab in 20 to 25 of 73 pairs; leaving those out, the gain was +3.90, still 12 of 12.

Code and data. The tool, both pre-registrations, every analysis script and the verification reports: github.com/zachk-alt/sheet-geek.

1 The question

Every spreadsheet has two layers: the cells, and the rules for reading them. The cells travel with the file. The rules usually stay in the owner’s head. Hand the file to a new hire, or to an AI, and the rules stay behind.

Take the trucking fleet in this study. Ask how much the company spent on fuel, and a reader of the cells adds every fuel card line. The owner would not: under the answer key, lines on the cards of units 301 to 306 are left out, because only company units count. Ask who the biggest customer was, and the key treats Kessler Dairy and Northfold Creamery as a single account. Nothing in the workbook states either rule. Without the owner’s rules, the answering models got none of the trucking fleet’s 21 plain-question attempts right. With the rules written into the file, they got 14 right.

A new hire asks. Language models often recognize ambiguity when told to judge it, yet overwhelmingly answer directly instead of asking [10], and data-science agents commit quietly to a plausible reading of an underspecified task [9]. Work on spreadsheets improves how a model reads what is in a sheet [11, 12, 13], but cannot recover a rule that is not in the sheet. The closest precedent, Anthropic’s `data-context-extractor` skill, asks the owner about a warehouse schema,

with no path for a loose workbook [6, 7]. Our prior-art review (September 25, 2026) found no tool that writes an owner’s answers into the workbook itself. That is the result of a search, not a proof.

The pre-registration [1] asks one question, verbatim: “When an AI with no skill and no hint is handed a workbook, does a brain built by the tool (version 0.2, frozen below) make its answers better than the same workbook without a brain, across businesses?” The primary test asks whether the business-level mean difference in open-request score (brain minus no brain) is zero, with a two-sided Wilcoxon signed-rank test at $\alpha = 0.05$ over 12 businesses. Secondary tests apply the same test to plain-question accuracy, owner-catchable mistakes and judge preference. Breakdowns by model, business and judge are described, not tested. With 12 businesses the smallest two-sided p the test can return is $2/2^{12} \approx 0.00049$, reached only when all 12 differences share a sign.

2 The tool

Sheet Geek is an open-source Agent Skill [15], released under Apache-2.0 at version 0.2.0 [5]; it was named spreadsheet-brain when it was tested, and this report studies that release. Its instructions put the division of labor in one line: “Code conducts, you are the voice.” A Python script does every count, every check and every choice of question; the model runs it, asks what it is given, and passes the answers back. The tool reads the whole workbook with code, summarizes it in five lines, and asks four questions, then more in small rounds only when the data or the answers call for them, ten substantive questions at most, each with its reason and, where the data supports one, a recommended answer. A closing question asks “Last one: what would someone new get wrong in this file?” It then writes the brain as a `_brain` tab at the end of the workbook, leaving the data untouched. So far it has been run end to end only in Claude Code on macOS.

Each note in the tab is one plain sentence with its source (`told`, `computed`, `inferred` or `web`) and a date. Every note is a claim, never an instruction: the format has no instruction field. A note records the columns it depends on and a SHA-256 fingerprint of their values, so when the data changes an owner’s note becomes `may-be-outdated` until a person confirms it. Because hidden text in spreadsheets has already been used against AI assistants [18, 19, 8], the security policy assumes every cell is hostile: the tab is spliced into the file without re-saving anything else, and the reader strips invisible Unicode, ignores formulas and flags notes that read like commands. The policy is explicit: “Prompt-level defenses reduce risk; they do not remove it.”

How version 0.2 was developed. Study 1 (Section 8) showed that version 0.1 rarely asked for the owner’s key rules: on seven development businesses it captured a per-business mean of 21.3% of 96 owner facts (pooled 18.8%, lowest business 0%), where a fact counts only if two checkers from different model families (Claude Opus 5.5 and GPT-6-Astra) both find it stated in full, from a question about that fact’s own subject. Version 0.2 followed a ranked plan of 21 changes with acceptance bars set in advance (a per-business capture mean of at least 60%, pooled at least 55%, no business under 35%, at most 10 substantive questions and 15 prompts), under one rule: repair the general mechanism, never add a rule aimed at one development fact. The frozen practice run captured 73.7% per business, 65.6% pooled and 37.0% for the lowest business. Development capture is optimistic by construction, because the fixes were made while looking at those seven files.

3 Design

The design is paired. A *trial* is one business, one answering model, one repetition and one condition: without the brain (the original workbook) or with it (the same workbook plus the `_brain` tab). A

Designed and run by an AI agent (Claude Opus 5.5 in Claude Code) under Zach Kellman's direction. Times are local, 2026-09-29.



Figure 1: How the experiment was run, from plan to independent verification. D1 to D4 are the deviations in Section 6. The only difference between the two arms of a trial is the `_brain` tab; there was no third arm with the owner's brief pasted in as plain notes.

pair is the two trials of one business, model and repetition. The plan called for 12 businesses, 4 answering models and 2 repetitions: 96 pairs and 192 trials. Figure 1 shows the whole run, with local times on September 29, 2026.

3.1 Businesses, owners and the gate

Twelve synthetic businesses were built after the freeze, one builder agent each, in twelve lines of work unlike the seven development businesses. Under the builder spec [3], each builder wrote one or two workbooks from a seeded script (main table of 1,000 to 8,000 rows, with real-world mess); at least five *owner facts*, things the owner knows that a new bookkeeper would get wrong and that change correct answers, whose effect the data shows but which nothing in the workbook states; an *owner brief* with every fact in plain words; and a *key*: one open request (a real task, not a quiz) and exactly three plain questions, never naming their rule, each with an answer computed by code under the owner’s rules. The naive answer had to differ from the key by more than 1% or be a different name. Builders were told not to read the lab repository or any other builder’s folder; that was an instruction, not an operating-system block, and the integrity audit found that none of the 329 tool calls made by builders and verifiers touched the repository or another business’s folder. A separate verifier per business recomputed every key answer and searched the files for any statement of an owner fact. All 12 passed on their first verification. The businesses, built between 07:34 and 07:51, after the 07:33 freeze, were a landscaping company, a dental practice, a trucking fleet, a nonprofit, a gym, a music venue’s ticketing, an auto repair shop, a law firm, a home builder, a greenhouse grower, an ad spend and leads log, and a coffee roaster, with 17 workbooks and 80 owner facts in all.

Each brain was built by one owner agent (Claude Opus 5.5) playing the owner, running the full frozen tool on a copy of the workbooks, with only the brief to go on. Its rules, from the harness:

Answer only from the brief and only what is asked. If the brief does not cover it, answer “not sure”. Never invent a fact, number, name or reason.

The twelve brains were built between 07:52 and 07:54, from interviews of 5 to 10 substantive questions (9 to 15 prompts in all), and held 640 owner notes. One behavior matters for reading the result: at the closing question, the owner agents typed out the brief’s rules, mostly nearly word for word, so every owner fact reached the brains whether or not the tool had asked about it (Section 4.6). Before any answer run, every brain passed a gate: a code check that the saved copy holds exactly the source data plus the tab, and two checkers (Claude Opus 5.5) that flagged any note contradicting the brief or crediting the owner with a claim the brief never makes. All 12 brains passed on their first build, and none of the 640 notes was flagged.

3.2 Answer trials and isolation

Four answering models ran cold, with no skill installed: Claude Haiku 4.5, Claude Sonnet 5 and Claude Opus 5.5 through the Claude Code command-line client at its default effort, and OpenAI GPT-5.6-Luna through the Codex command-line client at reasoning effort medium. Trial order was a seeded shuffle, with a unit’s two conditions run back to back in seeded random order. The answer runs took place from 07:59 to 09:58. Every trial of a business received one message, identical across models and conditions, from this template (verbatim from the plan):

I just got {this file / these files} from a colleague: {names}. {open request} Also: (1) {q1}
(2) {q2} (3) {q3} Python 3 with openpyxl is at /usr/bin/python3 if you want to open the files.
Work numbers out from the files; don’t guess. Answer in plain words, under 500 words total.

The message never mentions a notes tab. A trial that reads another trial’s brain, a brief or a key is not a trial, so each trial ran in a new folder holding only its own workbook copies, deleted when it ended, with a scrubbed environment. Claude runs had no settings, instruction files or tool

servers, a sandboxed shell, and operating-system read denial under 94 rules covering the study’s keys, briefs, brains, other trials, the repository and the judges’ folders. GPT-5.6-Luna ran in a sandbox that cannot write; Codex has no operating-system read denial, so every protected place counts as a leak for it. A detector scanned every command for whole-disk or shared-folder searches and paths outside the trial’s folder. **The rule:** a flagged trial’s answer is discarded and the trial is re-run once in a fresh root; a second flag drops the pair, and the drop is reported.

3.3 Judging and analysis

Every pair went to two judges from two model families: judge 1 Claude Opus 5.5 and judge 2 OpenAI GPT-6-Astra. Each judge ran in its own folder holding only the owner’s brief, a reference with the three plain questions and their correct answers, the output schema, and the two answers labeled A and B, with the order set by a seeded coin flip. Judges were told not to open any workbook. One sentence differed from Study 1, added to remove a bias found there: “The workbook may contain a tab of notes written by its owner. An answer that cites such notes is citing a real source in the file, not inventing one; judge what it says against the brief like any other claim.” The full prompt is in Appendix A. Each judge scored every answer on target, right and useful from 0 to 5, graded the plain questions, listed every mistake the owner would catch, flagged any mention of a notes tab, and chose the better answer or a tie. The rubric scores content only: “Length, formatting and tone earn nothing.”

An answer’s *open-request score* is on target + right + useful, 0 to 15, averaged over the two judges. For business b with n_b complete pairs, $d_b = \frac{1}{n_b} \sum_{(m,r)} (S_{bmr}^{\text{brain}} - S_{bmr}^{\text{none}})$, over models m and repetitions r . The primary test is a two-sided Wilcoxon signed-rank test [20] on the 12 values d_b , with a 95% percentile bootstrap interval [22] over businesses (10,000 resamples). Secondary measures are plain-question accuracy (judge 1), mistakes per answer, and judge preference (+1 brain, −1 no brain, 0 tie). Pair-level results (McNemar’s exact test [21], sign tests) are descriptive, because pairs within a business are not independent. A business with fewer than half its pairs complete would be left out of the primary test.

3.4 Integrity, freeze and verification

The tool (46 files, each hashed), 12 procedure files, the analysis script and a tarball of the whole repository were frozen on September 29, 2026 at 07:33, before any business for the test existed, with 1,890 tests passing. The written Freeze section of the plan was added at 07:58:57, after the brains existed and before the first answer trial; the integrity audit found the plan’s body unchanged. The plan also required that the operator, the AI agent running the study, not read the new businesses’ briefs, keys or workbooks before the brains were gated. That rule was broken (deviation 1).

After the analysis, separate passes checked the result [4]. Separate code rebuilt every reported number from the raw files: all 73 rows of `pairs.csv` matched, the Wilcoxon p was confirmed by enumerating all 2^{12} sign patterns, and the bootstrap interval matched within seed noise (+4.17 to +5.79 on the verifier’s seed). GPT-6-Astra, given only the per-pair totals, reproduced all 12 business means, the mean of +5.05 and the exact p . An integrity audit, verdict “confirmed with corrections”, confirmed the freeze hashes, that everything tested was created after the freeze, that no clean trial read another trial’s files, a brief, a key or a judge folder (all 153 clean trials scanned), and that each of the 17 brain workbooks matched its source with only the tab appended.

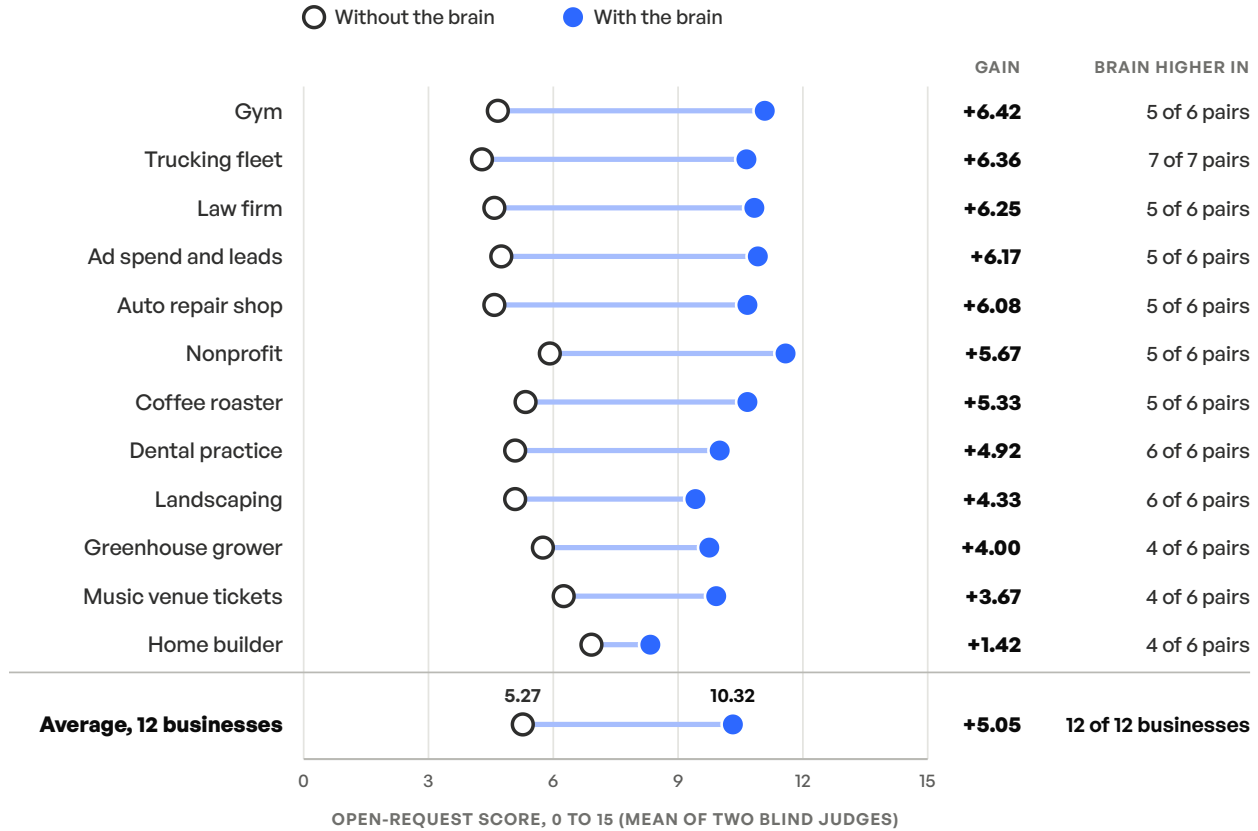


Figure 2: The brain raised the score on all 12 businesses. Open-request score, 0 to 15 (mean of two blind judges), without and with the brain; each row is the mean over that business’s pairs. Business-level gain +5.05 (95% bootstrap CI +4.19 to +5.78), Wilcoxon $p = 0.00049$.

4 Results

Of the 192 planned trials, every Haiku 4.5, Opus 5.5 and GPT-5.6-Luna trial (144) ran clean on its first try. Sonnet 5 searched the whole disk or the shared temporary folder in 43 of its 48 trials; each was re-run once, and 39 were flagged again, so 23 of Sonnet’s 24 pairs were dropped (deviation 3). That left 153 clean trials and 73 complete pairs: 24 each for Haiku 4.5, Opus 5.5 and GPT-5.6-Luna, and one for Sonnet 5. Every business kept at least 6 of its 8 pairs, so none fell under the missing-data rule.

4.1 Primary result

The judged open-request score was 5.25 of 15 without the brain and 10.32 with it, pooled over the 73 pairs. The business-level mean difference was **+5.05 points** (median +5.50), with a 95% bootstrap interval of **+4.19 to +5.78**. The brain answer scored higher on average on **12 of 12 businesses** (Figure 2). The Wilcoxon test gave $W = 0$ and $p = 0.00049$: 2 of the 4,096 possible sign patterns. The pre-registered primary test passes (Table 1). (The figure’s average row is the mean of the 12 business means, 5.27 without and 10.32 with; the pooled means differ slightly because the trucking fleet has 7 pairs and every other business 6.) At the pair level, descriptively only, the brain answer scored higher in 61 of 73 pairs, lower in 9 and tied in 3.

Measure	Without	With	Difference	95% interval	<i>p</i>
Open-request score, 0 to 15 (primary)	5.25	10.32	+5.05	+4.19 to +5.78	0.00049
Plain questions right, judge 1	10.5%	60.7%	+0.50	+0.40 to +0.58	0.00049
Mistakes per answer	7.6	3.7	−3.92	−4.62 to −3.14	0.00049
Judge preference, −1 to +1			+0.68	+0.58 to +0.77	0.00049

Table 1: Primary and secondary results. Differences, intervals and *p* are business level (pre-registered Wilcoxon signed-rank, two-sided, 12 businesses); each favored the brain on 12 of 12 businesses. Without and With are pooled over the 73 pairs. Source: `results.json`.

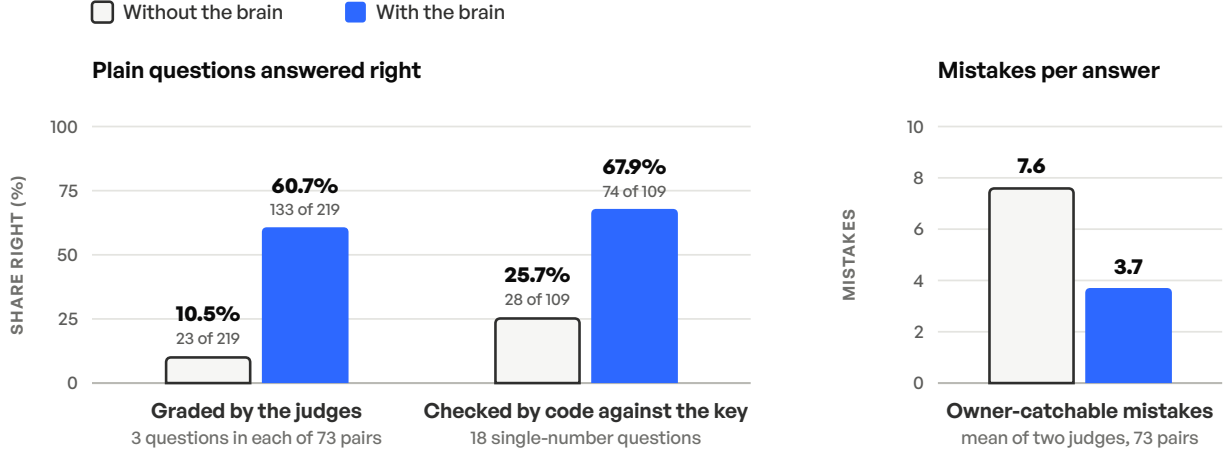


Figure 3: More exact answers, fewer mistakes. Left: plain questions answered right, judged against the key, and by a code check on the 18 single-number questions (not pre-registered, more lenient). Right: owner-catchable mistakes per answer, fewer on all 12 businesses.

4.2 Secondary results

Judged against the key, 23 of 219 plain questions were answered right without the brain (10.5%) and 133 of 219 with it (60.7%), positive on all 12 businesses; at the pair level, 110 questions were right only with the brain and none only without (McNemar exact $p = 1.5 \times 10^{-33}$). The two judges differed on 2 of 438 grades. Owner-catchable mistakes fell from 7.6 to 3.7 per answer, on all 12 businesses. Judges preferred the brain answer on every business: 58 pairs for the brain, 8 for no brain and 7 ties. The two arms wrote the same amount, 356 words per answer without the brain and 359 with it (Figure 3).

4.3 By model and by business

By-model results were planned as described, not tested (Table 2, Figure 4). Claude Opus 5.5 scored higher with the brain in all 24 pairs and, with it, answered all 72 plain questions right (31.9% without). GPT-5.6-Luna was higher in 23 of 24 pairs with one tie. Claude Haiku 4.5 gained +0.13 and answered none of its 72 plain questions right in either condition. It read the notes tab in only 5 of its 24 brain trials, and those five pairs averaged +0.2. A brain helps a model that reads it, in line with Vercel’s finding that a skill carrying documentation was never invoked in 56% of cases [14]. Sonnet 5’s one pair went from 4.50 to 13.00, and one pair says nothing on its own.

Every business gained, and ten of the twelve gained four points or more. The home builder

Model	Pairs	Without	With	Gain	95% interval	Plain right, %	Mistakes
Claude Opus 5.5	24	7.69	14.67	+6.98	+5.96 to +7.81	31.9 to 100	6.5 to 0.4
GPT-5.6-Luna	24	4.63	12.58	+7.96	+6.54 to +9.31	0 to 80.6	7.3 to 1.7
Claude Haiku 4.5	24	3.48	3.60	+0.13	−0.31 to +0.54	0 to 0	9.0 to 9.1
Claude Sonnet 5	1	4.50	13.00	+8.50	none	0 to 100	10.5 to 3.0

Table 2: Results by answering model, described and not tested. Plain right and mistakes go from without to with. GPT-5.6-Luna’s no-brain mean of 4.625 rounds half up to 4.63 (the pre-registration’s results text gives 4.62).

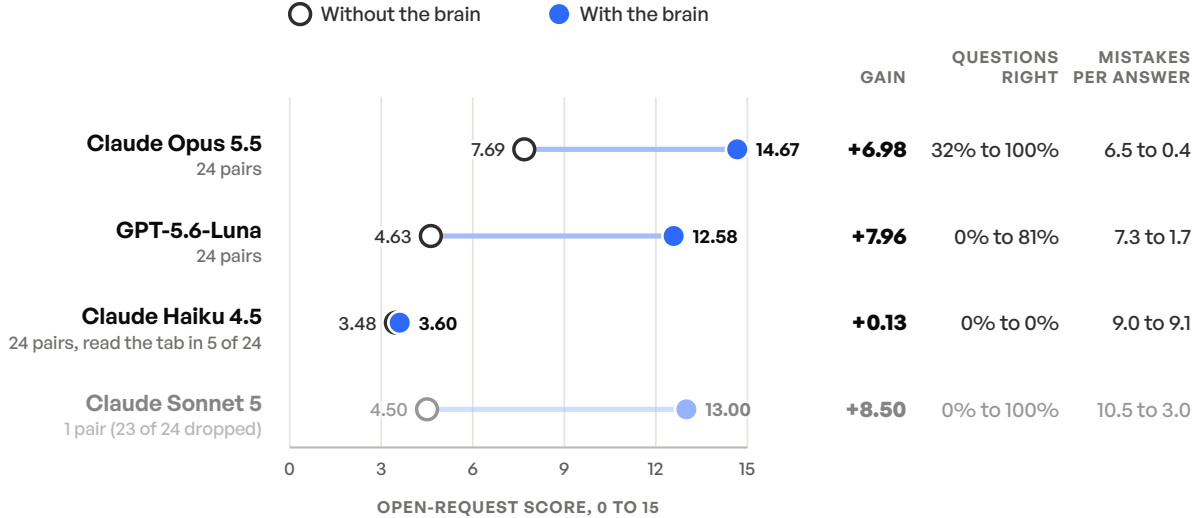


Figure 4: The gain came from the two stronger models. Open-request score by answering model, with plain questions right and mistakes per answer. Sonnet 5’s one remaining pair is shown faded.

gained least, +1.42; the owner agents’ logs record that the frozen tool’s scan read only half of its AP Detail tab, and we did not test whether the two are linked. The trucking fleet went from 0 of 21 plain questions right to 14 of 21, tied with the gym and ad spend (0 of 18 to 12 of 18) for the largest swing.

4.4 Judges and blinding

The two judges ranked answers alike (Spearman 0.91 over 146 answers) and agreed on 436 of 438 plain-question grades, but on the better answer of a pair in only 82% of pairs. Each judge alone passes the primary test: judge 1 (Claude Opus 5.5) gives +5.42 and judge 2 (GPT-6-Astra) +4.68, both 12 of 12, $p = 0.00049$. Judge 1’s larger gain is not specific to Opus 5.5 answers: at the pair level it is larger on GPT-5.6-Luna answers too (+8.46 against +7.46), and it comes mostly from judge 1 scoring no-brain answers lower. Judge 1 also scored Opus 5.5’s brain answers higher (14.96 against 14.38), so self-preference [16, 17] cannot be ruled out; the result holds with judge 2 alone.

Answer order was balanced (the brain answer was A in 38 pairs and B in 35), with no position bias. The answers themselves were not blind. Brain answers named the notes tab in 20 to 25 of the 73 pairs, depending on who counted: 20 by the integrity audit’s text search, 24 by judge 1’s flag, 25 by either judge’s. So a judge could tell the condition in about a third of pairs. The key-graded plain questions are less exposed to this than the open-request score, and they point the same way.

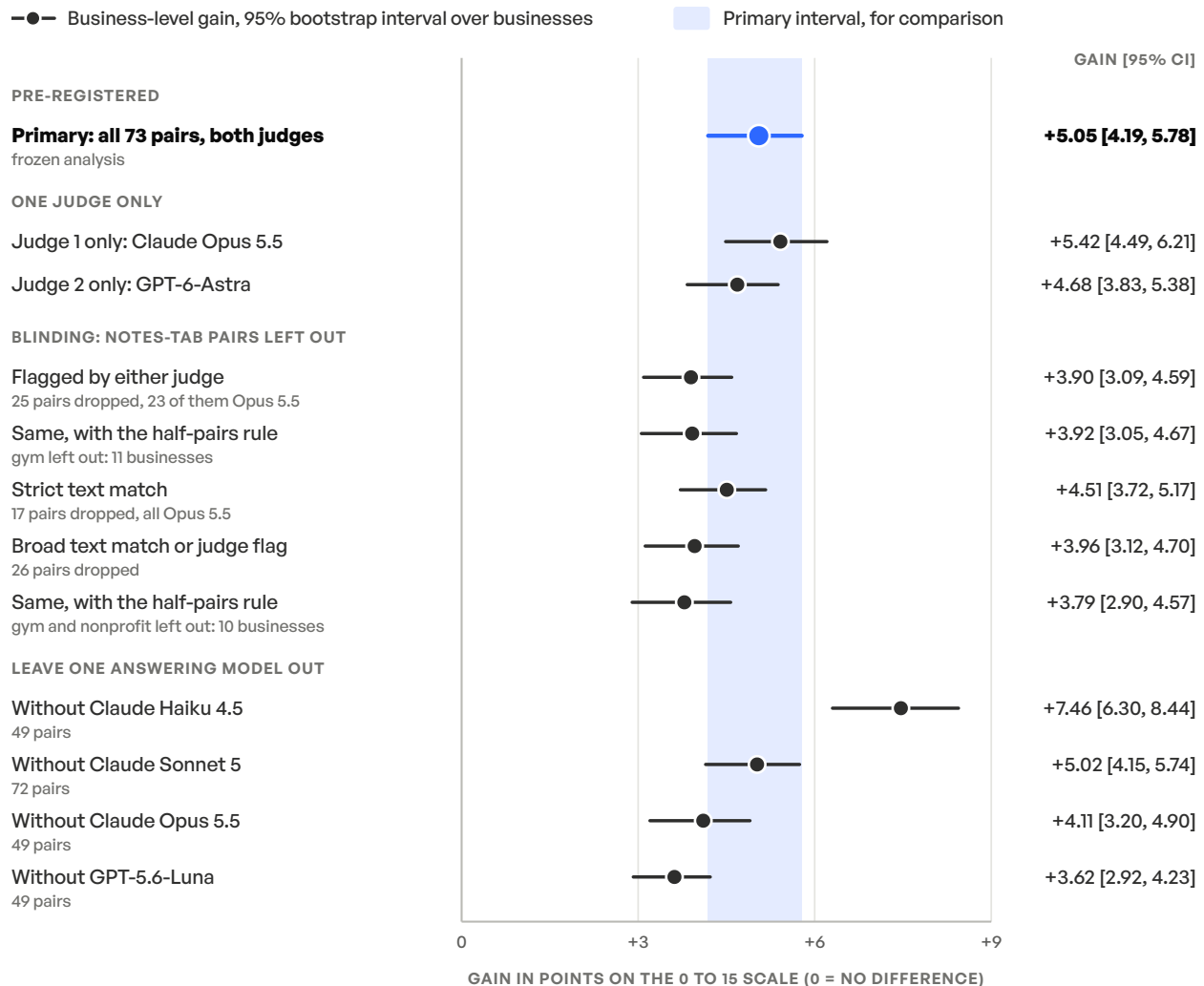


Figure 5: The gain held in every sensitivity check (none pre-registered). Intervals: 95% bootstrap over business means, 10,000 resamples. The checks use the verifier’s resampling (seed 12345), so the single-judge intervals differ slightly from the frozen analysis (+4.51 to +6.19 and +3.86 to +5.37).

4.5 Robustness

The independent verifier ran the checks in Figure 5. None was pre-registered, and every check kept the brain higher on every business included. Dropping the 25 pairs in which either judge flagged the notes tab gives +3.90, 12 of 12, $p = 0.00049$; but 23 of the 25 were Opus 5.5 pairs, so this check mostly tests GPT-5.6-Luna and Haiku 4.5. Applying the plan’s half-pairs rule to it drops the gym and gives +3.92 over 11 businesses ($p = 0.00098$, the smallest eleven can give). Leaving each model out in turn gives +3.62 (without GPT-5.6-Luna) to +7.46 (without Haiku 4.5), and +4.11 without Opus 5.5. A code check on the 18 single-number plain questions, which counts a hit when the answer contains a number within 1% of the key, gave 67.9% with the brain and 25.7% without (business level $p = 0.0033$). It is lenient: all 21 of its disagreements with the judges were code hits the judges had marked wrong, and none went the other way.

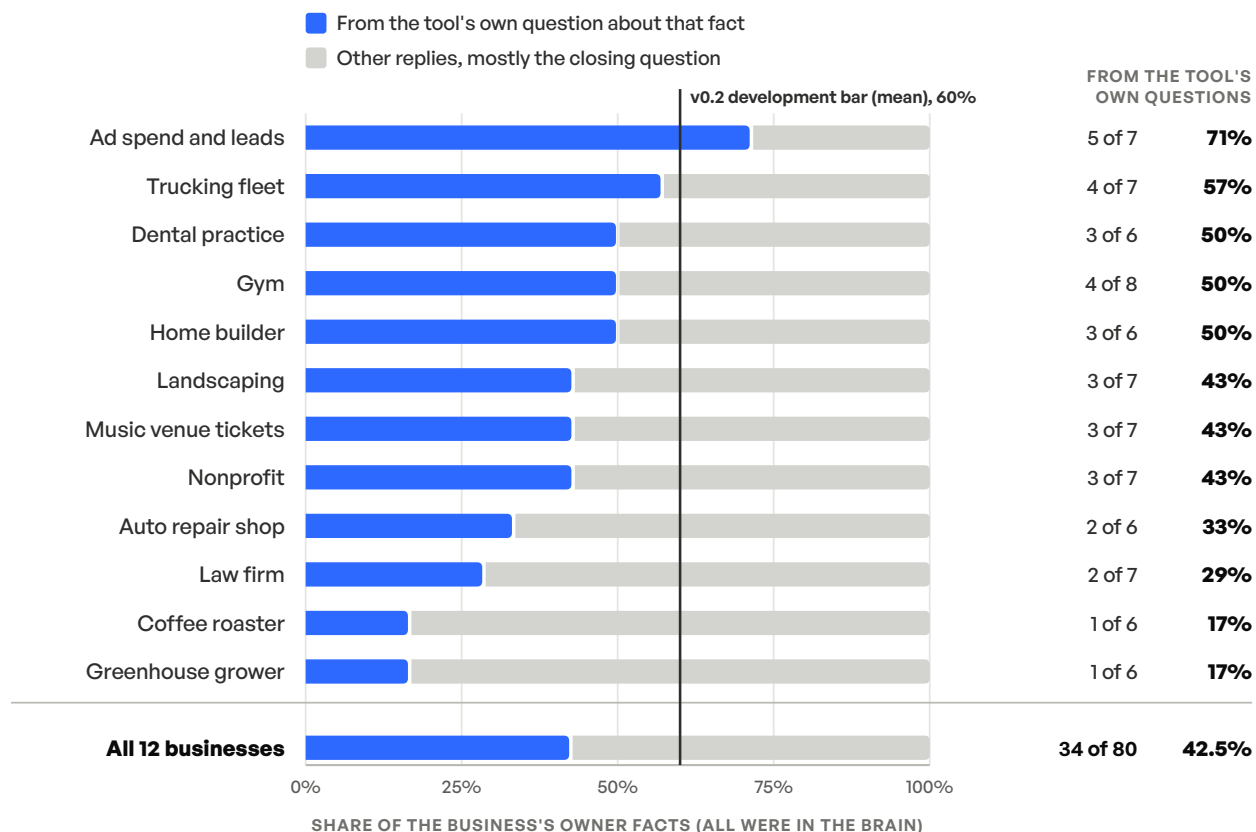


Figure 6: Only 34 of 80 owner facts came from the tool's own questions. The rest came mostly from the owner agents' reply to the closing question. Reported, not tested.

4.6 Where the owner facts came from

All 80 owner facts from the twelve briefs ended up in the brains, stated in full. But only 34 of the 80 (42.5% pooled; per-business mean 41.9%; lowest 16.7%) came from a question the tool asked about that fact (Figure 6). The integrity audit found the rest came mostly from the owner agents' reply to the closing question, where they pasted their brief nearly word for word. No note simply states a key answer, but the brains carry the briefs' rules almost verbatim. Capture on these new businesses fell short of all three development bars, as the plan warned it might. It was reported, not tested.

5 A worked example: the runway question

Here the same question is asked of the same workbook with and without a brain. These runs are not from Study 2 and were not judged; they show what the difference looks like, not that it exists. They use the finance model, one of the seven development files, so the brain shows the tool at its best, and each run had the same isolation as the studies. The prompt asked "What's our runway?", to be answered from the file in under 150 words. Both answers are Claude Opus 5.5, quoted word for word, with markdown rendered as plain text and cuts marked [...].

Without the brain. Your runway runs out in May 2027, about 8 months from now, unless the company raises money. [...] The catch: the model assumes a \$15M equity raise in March 2027.

It’s a typed-in number in the Cash Flow sheet, not something calculated. [...] Your colleague’s version looks safe, but only if that raise really happens. Ask whether it’s committed or just planned. [...]

With the brain. Our runway is about 8 months, and the money runs out in May 2027. That assumes the Series B doesn’t close. [...] The model shows cash never dropping below \$930K, but that’s only because it counts a \$15M Series B in March 2027. The owner’s notes in the file (a notes tab named `_brain`) say that round “is not closed yet,” so I left it out. [...]

All three Opus 5.5 runs without the brain found the May 2027 run-out and flagged the raise as unconfirmed; all three with the brain found the same date and cited the owner’s note. The arithmetic did not change. What changed is where the answer ends: without the brain, the AI has to tell the reader to go ask; with it, the file already says. The runs are in `evals/film-chat-runs/`.

6 Deviations from the plan

The plan lists four deviations, labeled D1 to D4 in Figure 1. Each is quoted from the pre-registration, followed by what it changes.

D1: the operator saw the owner interview logs before the gate.

“The brain-build workflow returns each owner agent’s question-and-reply log to the operator’s session. All 12 logs, which repeat owner facts from the briefs, came back that way; the completion notice displayed part of the landscaping business’s log. (This deviation first said only the landscaping log; the integrity audit corrected it.) The tool was frozen before any business existed, the gate and everything after it run by script and report only pass or fail, and nothing was changed because of it. All 12 businesses passed verification on their first build and all 12 brains passed the gate on their first build.”

The operator blinding rule was broken. Every step after the brain builds was scripted and frozen, so we see no path from the exposure to the scores; but the rule did not hold, and its first report understated how far it failed.

D2: the gate’s wiring script was written after the freeze.

“`confirm_gate.sh` (07:52) chains the frozen code check and the frozen two-checker gate and passes a business only when both pass, as the plan says; it was not in the frozen procedure hashes. It is kept with the results.”

Nothing measurable changes: the audit found its logic matches the plan. It is still an unfingerprinted file in the pipeline.

D3: Sonnet 5 is almost absent from the analysis.

“Under the frozen isolation rule (unchanged from the first test), Claude Sonnet 5 searched the whole disk or the shared temp folder for its files in 43 of its 48 trials; each was re-run once and 39 were flagged again, so 23 of its 24 pairs were dropped. The plan’s 96 pairs became 73: Haiku 4.5, Opus 5.5 and GPT-5.6-Luna 24 each, Sonnet 5 one. Every business kept at least 6 of its 8 pairs, so none was left out of the primary test. This is the rule applied as written, reported here because it changes what the result covers: three answering models, not four.”

The audit confirmed that all 39 flagged trials ran a real whole-disk or shared-temp search on both attempts.

D4: detector limits found by the audit.

“The isolation check does not expand `$TMPDIR`, catches `/private/tmp` only as a search’s first root, and by design does not flag Claude trials that search the home folder, which the OS denial blocks. Five Haiku trials and the one analyzed Sonnet pair ran such searches; every read outside the trial’s own folder came back denied, and the audit found no trial that read another trial’s files, a brief, a key or a judge folder. Dropping the Sonnet pair would move trucking from +6.36 to +6.00. Judge 1 (Claude) leaves no tool trace, so its reads rely on the OS denial; judge 2’s logs show only its four files read.”

No reported number changes; judge 1’s isolation rests on the operating system’s denial, not on a log.

Two further disclosures. The written freeze record came after the brains (Section 3.4). And the plan’s results text says 0 of 73 no-brain answers named a notes tab, while `results.json` records 1, a tab the answer created itself.

7 Limitations

- **No arm with the brief pasted in as plain notes.** The study shows that owner rules written into the file help. It does not show that the tool’s interview put them there, or that its format beats pasted text.
- **The owner agents pasted their briefs.** Only 42.5% of owner facts came from the tool’s targeted questions, so the brains hold more owner knowledge than a busy real owner might give.
- **The businesses were built to need the owner.** Every plain question depends on an owner fact, which makes the possible gain large by construction. On a real file the gain could be smaller.
- **One model in many roles.** Claude Opus 5.5 built and verified the businesses, played the owners, checked the gate and capture, was judge 1, was an answering model, and designed and ran the study. A shared bias would not show up as disagreement anywhere. The result holds with judge 2 alone and without Opus 5.5 answering (+4.11), but no part of the study was free of Opus 5.5.
- **Judges were told the workbook may hold an owner’s notes tab,** which may have made them more willing to credit an answer that leans on it.
- **The blinding leaked** in 20 to 25 of 73 pairs; the check that drops them was post hoc and mostly tests the other models.
- **Every judge is a model.** No person graded an answer, and the judges agreed on the winner in only 82% of pairs.
- **Coverage.** Sonnet 5 kept 1 of 24 pairs; Haiku 4.5 gained nothing; only two command-line clients with a Python shell were tested, not chat apps; and businesses, owners and judges are all synthetic.
- **Procedure.** The frozen tool’s faults stayed in; deviations 1 to 4 stand; the written freeze record came after the brains; judge 1’s reads rest on the operating system’s denial, not a log.

- **Statistics.** The smallest possible p says all twelve businesses moved the same way, not how far; that is what the interval is for. By-model and by-business results were not tested, and pair-level tests are descriptive. The sensitivity checks and the mechanical grader were added after the analysis. With the brain, 39.3% of plain questions were still answered wrong, and the mean score was 10.32 of 15.
- **Who ran it.** The lab built the tool, ran both studies and wrote this report, and the verification was run inside the lab by AI models, not by an outside party.

8 Study 1 in brief

Study 1 [2] tested version 0.1 on four held-out synthetic businesses, with the same four answering models and the same two judge families. Its deviations are listed in its plan; among them, the held-out businesses were built before the freeze, a seven-line writer fix was made after the freeze because the frozen tool could not write a tab into those workbooks, and isolation took five rounds to get right. On 31 pairs the open-request score went from 5.06 to 7.00 (brain higher in 25 of 31 pairs; Wilcoxon $p = 8.7 \times 10^{-5}$), and plain questions right from 5 of 93 to 23 of 93. But the test counted 31 pairs as if they were independent, while the brain varied across only four businesses. At the business level the effect was 3 of 4 positive, and a t -test on the four business means gave $p = 0.11$. Study 1 showed the brain helped on those businesses, not on a new business in general, and it exposed version 0.1, which never asked about most of the owners' key rules. Study 2 answered with a business-level test on 12 businesses built after the freeze, and version 0.2.

9 Conclusion

The claim that survives is narrow, and it is solid inside its limits. On twelve synthetic businesses nobody had seen, built after the tool and the analysis were frozen, handing an AI the same workbook with the owner's rules written into it roughly doubled the judged quality of its answer (5.25 to 10.32 of 15) and raised exact answers from about 1 in 10 to about 6 in 10. The pre-registered test passed on all twelve businesses, the numbers were reproduced by separate code and by a second model family, and the gain held in every check we could run after the fact.

It held for the two stronger models tested. The smallest model gained nothing, mostly because it rarely opened the tab, and one of the four planned models is missing. And the study cannot tell whose work the gain is, because no arm tested the brief pasted in as plain notes. So the defensible thesis is this: *owner rules written into the file help models that read them*. The next study needs real owners, a brief-as-notes arm, human graders on a sample, and a model outside the one that built everything else.

Reproducibility and data. The tool, both pre-registrations, the results of both studies, their verification reports and every analysis script are public under Apache-2.0 at <https://github.com/zachk-alt/sheet-geek>, release v0.2.0 (commit 8cff9f5) [5]: the Study 2 plan with its freeze, deviations and results (evals/PREREG-confirmatory.md); the tool and procedure hashes (evals/FREEZE-confirmatory-manifest.sha256, whose own sha256 is 4c64d4e11e2221c0bc2715599ae7d6b9f951b6f5c58dea432edebb66741811e0, and evals/FREEZE-confirmatory-procedures.sha256); Study 1's plan and results (evals/PREREG-2026-09-26.md, evals/rigor-2026-09-26/); the runway runs (evals/film-chat-runs/); and, in evals/rigor-confirmatory/, the builder spec, businesses.json (open requests, plain questions, keys and recipes), pairs.csv (one row per analyzed pair), results.json (the frozen analysis output), the capture and gate results, the frozen scripts in tools/, and the recomputation, second-family transcript and audit reports in verify/. From a clone at v0.2.0, Python 3 and SciPy recompute the business-level mean

(+5.0506), median (+5.5) and p (0.00048828125) from `pairs.csv`. Not public: the raw trial and verdict folders that `analyze.py` and the verifier’s `verify.py` read, and every Study 2 trial record, answer, verdict, judge order, workbook, brief, key, brain, gate and capture record, kept in an evidence archive outside the repository (sha256 0b24ffddacec9158e4380d9881677de7ff6216aa77d9e8dad2e1592c06c1bc7b4); Study 1’s archive is fingerprinted in its plan (sha256 6428ff3c21547c33da2f6fadbabc61ccf5295afda828f96e94dfe15b09d27858). Rerunning the whole study needs the Claude Code and Codex command-line clients, the same models, and new businesses built to the builder spec; the frozen scripts carry machine-specific paths. The six figures are drawn by one script that stops if any plotted number disagrees with `results.json` or `verify/mine.json`. This report accompanies the article at actualintelligence.ai/research/sheet-geek-experiment. All businesses, people and numbers in the study are synthetic; no client data was used.

Conflict of interest. Actual Intelligence Labs, an AI research and implementation lab in Sarasota, Florida, built Sheet Geek and released it for free under Apache-2.0. The lab designed and ran both studies and wrote this report. The lab builds custom AI systems for clients, so a favorable result could benefit it. No outside party reviewed the study before publication; the verification in Section 3.4 was run inside the lab by AI models. To offset the conflict, the plans were frozen before any test business existed, every deviation is reported, and the plans, scripts, per-pair results and verification reports are public.

AI use. This experiment was designed and run by an AI agent, Claude Opus 5.5 operating in Claude Code, under Zach Kellman’s direction. That agent is the “operator” named in the pre-registration. AI models also filled every other role in the study: Claude Opus 5.5 built and verified the businesses, played the owners, gated the brains, checked fact capture with GPT-6-Astra, answered as one of four models, and judged as judge 1; GPT-6-Astra was judge 2 and reproduced the primary result; Claude Haiku 4.5, Claude Sonnet 5 and GPT-5.6-Luna were the other answering models. The sources do not record which model ran the independent recomputation and the integrity audit. This report was drafted by Claude Opus 5.5 in Claude Code from the study files listed above, and every number in it was checked against those files.

References

- [1] Actual Intelligence Labs. *Pre-registered confirmatory test: does a spreadsheet brain help on businesses nobody has seen?* Drafted September 26, 2026; frozen September 29, 2026. `evals/PREREG-confirmatory.md` in [5].
- [2] Actual Intelligence Labs. *Pre-registered test: does a spreadsheet brain make an AI better at a workbook?* September 26, 2026. `evals/PREREG-2026-09-26.md` in [5].
- [3] Actual Intelligence Labs. *Builder spec: confirmatory test businesses.* `evals/rigor-confirmatory/BUILDER-SPEC.md` in [5].
- [4] Actual Intelligence Labs. *Independent verification and integrity audit of the confirmatory test.* `evals/rigor-confirmatory/verify/verification_reports.json` in [5].
- [5] Actual Intelligence Labs. *Sheet Geek* (released as `spreadsheet-brain`), release v0.2.0. <https://github.com/zachk-alt/sheet-geek>
- [6] Anthropic. *data-context-extractor* skill, knowledge-work-plugins repository. <https://github.com/anthropics/knowledge-work-plugins/blob/main/data/skills/data-context-extractor/SKILL.md>
- [7] J. Cherry. *How Anthropic enables self-service data analytics with Claude*. Claude blog, June 3, 2026. <https://claude.com/blog/how-anthropic-enables-self-service-data-analytics-with-claude>
- [8] Anthropic. *Use Claude for Excel*. Claude documentation. <https://claude.com/docs/office-agents/excel>
- [9] J. L. Stoisser et al. *Ambig-DS: A Benchmark for Task-Framing Ambiguity in Data-Science Agents*, 2026. <https://arxiv.org/abs/2605.09698>
- [10] J. Su and C. Cardie. *Knowing but Not Showing: LLMs Recognize Ambiguity but Rarely Ask Clarifying Questions*, 2026. <https://arxiv.org/abs/2605.25284>
- [11] Z. Ma et al. *SpreadsheetBench: Towards Challenging Real World Spreadsheet Manipulation*, 2024. <https://arxiv.org/abs/2406.14991>
- [12] H. Dong et al. *SpreadsheetLLM: Encoding Spreadsheets for Large Language Models*, 2024. <https://arxiv.org/abs/2407.09025>

- [13] Z. Wang et al. *SheetBrain: A Neuro-Symbolic Agent for Accurate Reasoning over Complex and Large Spreadsheets*, 2025. <https://arxiv.org/abs/2510.19247>
- [14] Vercel. *AGENTS.md outperforms skills in our agent evals*. Vercel blog. <https://vercel.com/blog/agents-md-outperforms-skills-in-our-agent-evals>
- [15] Agent Skills. *Specification*. <https://agentskills.io/specification>
- [16] L. Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*, 2023. <https://arxiv.org/abs/2306.05685>
- [17] K. Wataoka et al. *Self-Preference Bias in LLM-as-a-Judge*, 2024. <https://arxiv.org/abs/2410.21819>
- [18] PromptArmor. *CellShock: Claude AI is Excel-lent at Stealing Data*. <https://www.promptarmor.com/resources/cellshock-claude-ai-is-excel-lent-at-stealing-data>
- [19] PromptArmor. *ChatGPT for Google Sheets Exfiltrates Workbooks*. <https://www.promptarmor.com/resources/gpt-for-google-sheets-data-exfiltration>
- [20] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80 to 83, 1945. <https://doi.org/10.2307/3001968>
- [21] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153 to 157, 1947. <https://doi.org/10.1007/BF02295996>
- [22] B. Efron. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7(1), 1 to 26, 1979. <https://doi.org/10.1214/aos/1176344552>

A The judge prompt and grading rules, verbatim

Each judge received this prompt (from `tools/run_judges.py`), with the files named in it staged in its own folder.

You are a strict, blind judge in a controlled experiment. Two AI assistants were each handed the same business workbook(s) and the same message from the owner. Grade both answers against what the owner actually knows and wants.

Files in this folder:

- `brief.txt`: the owner's own knowledge of the business and its files (file names in it may differ from the ones in the request; they are the same workbooks).
- `reference.json`: the owner's full message, the three plain questions with their correct answers and how each is computed, the grading rules, and for some businesses known issues in the data.
- `answer_A.md` and `answer_B.md`: the two answers.

Read these four files and nothing else. Do not open any workbook: judge from the brief and the reference.

The workbook may contain a tab of notes written by its owner. An answer that cites such notes is citing a real source in the file, not inventing one; judge what it says against the brief like any other claim.

For each answer (A and B):

- `plain_correct`: exactly three booleans, one per plain question in order, graded by the rules in `reference.json`.
 - `on_target` (integer 0 to 5): does it address what the owner asked, for the purpose they gave?
 - `right` (integer 0 to 5): are its claims and numbers correct given the brief and the reference? Penalize numbers that break the owner's rules (for example counting what the owner leaves out) and confident claims the brief contradicts.
 - `useful` (integer 0 to 5): would the owner act on it? Specific, prioritized, with the numbers they need.
 - `mistakes`: every mistake the owner would catch, one short line each (empty list if none).
 - `cites_notes_tab`: true if the answer mentions a notes, brain or documentation tab in the file. Then better: "A", "B" or "tie", and why in at most two sentences.
- Judge content only. Length, formatting and tone earn nothing.

The grading rules in each reference, verbatim:

A number is correct within 1% of the key or within the key's rounding. A named answer must match. An answer that computes a different thing than the key's question (for example a total that includes rows the owner leaves out) is wrong. A question with several parts is correct only if every part is. A question asking for a list is correct if it names every item in the key; extra items do not make it wrong. An answer that gives no answer to a question is wrong for that question.