

Newsletter Fuzzy Variable Enrichment

Write the welcome email yourself, then have Claude fill three short AI-written blanks per subscriber so the mail merge stops reading like a mail merge.

01 When to use this

Use this when you have a list with real signup data on it and a welcome email you already trust. It is the first move in the copy half of the course. Reach for it the moment you notice your email says the same thing to two thousand people.

02 The prompt

I have a subscriber list at `<SHEET_URL_OR_PATH>`. I want you to enrich every row with fuzzy variables that I will use as merge fields in a welcome email.

Product context: `<PRODUCT_NAME>` is `<ONE_LINE_DESCRIPTION>` for `<AUDIENCE>`. The offer is `<OFFER>`.

Tone for every value you write: casual, direct, first person, no fluff, no em dashes, no marketing language.

Add exactly these three columns, in this exact order, at the end of the row:

1. `shortPlausibleReasonWhySignedUp` - about 10 words. It has to complete this sentence and read naturally inside it: "Extremely excited to have you with us, especially because ____."
2. `shortCustomReasonWhyItsUseful` - about 10 words. It has to complete: "I think `<PRODUCT_NAME>` is going to be useful for you because ____."
3. `paraphrasedChallenge` - about 5 words. It has to complete: "I know a big challenge for you right now is ____."

Rules:

1. Read EVERY row. Do not sample, do not guess, do not extrapolate from the first few rows. If the list is long, work through it in batches until all rows are done.
2. Write each value from that specific person's row. Use their answers, their company, their role. Do not build a lookup table of five stock phrases and reuse them.
3. Vary the sentence structure row to row. Two people who wrote similar answers should still get different phrasing, drawn from the other fields on their row.
4. Do not modify the original sheet. Create a new sheet named `<NEW_SHEET_NAME>` with all original columns plus the three new ones.
5. Where a row has almost no data, write something safely general rather than inventing a fact about that person.

When you are done, show me 5 rendered examples with the full sentence around each variable so I can judge fit, not just content.

03 What it is actually doing

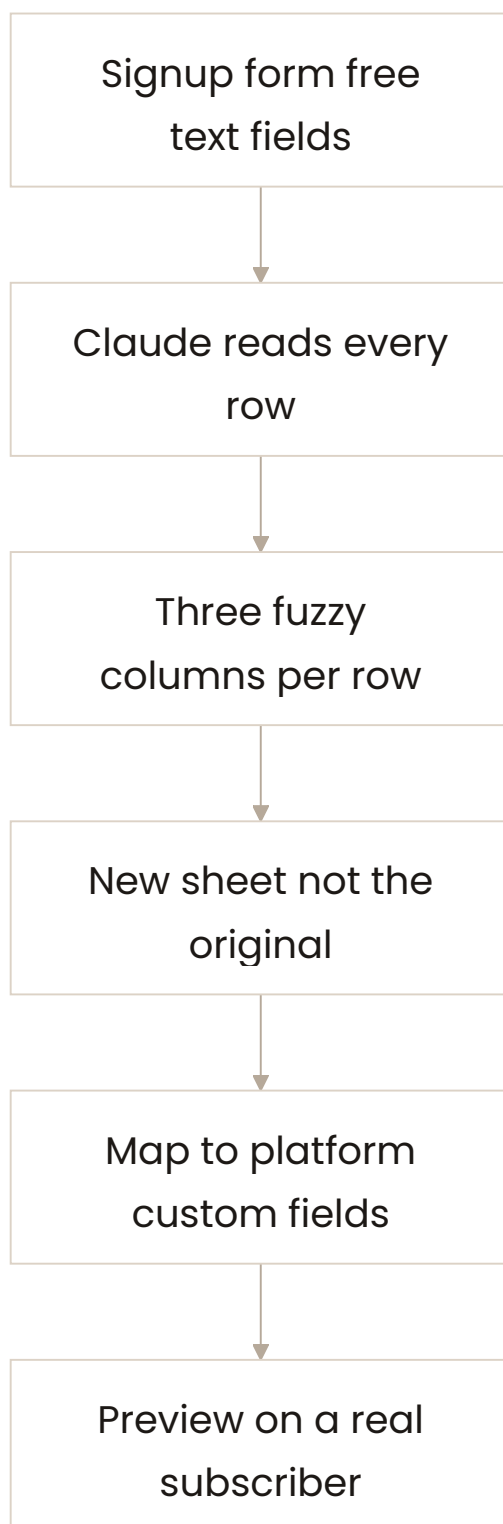
There are two kinds of merge field. A hard variable is copied straight out of a column: first name, company, city. It works, but everyone has seen it a thousand times, so it reads as mail merge because it is mail merge.

A fuzzy variable is different. Claude reads the whole row and writes a short phrase custom to that one person. Then that phrase gets dropped into the middle of a sentence you wrote. The reader never sees a slot. They just see a sentence that happens to know something about them.

The trick that makes it work is the variable name. You write it in camelCase as an instruction, like `shortPlausibleReasonWhySignedUp`. The name is the prompt for that slot. Claude reads the name and knows what to write, which means you do not need a separate instruction block for each one.

The part people get backwards: the human writes the whole email and AI fills a few blanks. Not the other way around. If you let AI write the entire email, you throw away your proven copy, your offer, and your voice, and you get worse results. You are renting the machine for the one job it is better at, which is reading two thousand rows without getting bored.

04 How it flows



05 Step by step

01 Collect real signup data

Your form needs to ask something worth personalizing on. Nick used first name, company, why they signed up, biggest challenge, and favorite tool. Free-text answers are the fuel.

02 Write the email yourself first

Draft the full welcome email in your own voice. Decide where the blanks go before you decide what fills them. The email is yours, the blanks are Claude's.

03 Name the slots as instructions

Use camelCase names that describe exactly what goes in the hole. `shortPlausibleReasonWhySignedUp` tells Claude the length, the content, and the framing all at once.

04 Forbid sampling out loud

Models will read ten rows and generalize to save effort. Say explicitly that every row must be read and processed. Ask it to work in batches for a long list.

05 Forbid the lookup table

Tell it to vary sentence structure row to row and to pull from other fields on the row. Without this, two thousand subscribers get five sentences between them.

06 Write to a new sheet

Never let it edit the source. Ask for a new sheet with the original columns plus the three new ones appended in a stated order. Order matters when you map fields later.

07

Judge the rendered sentence

Ask for examples shown inside the full surrounding sentence. A value can be a good sentence on its own and still be broken in the socket it has to fill.

06 Words explained

Hard variable

A merge field copied word for word out of a column, like first name or city. Fast, but obviously a merge field.

Fuzzy variable

A short phrase the AI writes from the whole row, built to sit inside a sentence you wrote so the reader cannot spot the seam.

camelCase

Writing a multi-word name with no spaces and each new word capitalized, like shortPlausibleReasonWhySignedUp. It keeps the name valid as a field name.

Merge field

A placeholder in an email that the sending platform swaps for real data before it goes out.

Row context

Everything you know about one person on one line of the sheet. The fuzzy variable is written from all of it, not just one column.

07 Watch out for

- If you do not say read every row, the model reads a sample and invents the rest with confidence.
-

- Without a vary-the-structure instruction the whole list collapses into four or five repeated sentences.
- Letting Claude edit the source sheet means one bad run destroys your list. Always output to a new sheet.
- A value that reads fine alone can break the sentence it lands in. Judge it rendered, never bare.
- Rows with thin data tempt the model to invent facts about the person. Tell it to stay general instead.

08 How it connects

Fuzzy Variable Tightening

The first pass here always comes back too long and sometimes structurally broken, and tightening is the correction pass that fixes it.

Daily Enrichment Skill via API

This note is the one-off sheet job, and the daily skill is the same work running against the live platform every 24 hours.

Cold Email Personalization

Identical machine, different intake. Swap opt-in rows for scraped rows and the technique carries over unchanged.

Fuzzy Variables for Personalized Copy

The framework note that holds the full rule set behind this prompt.

Credit. Framework and approach by Nick Saraev, from the newsletter personalization build in his Zero to One course on Claude Code for marketing.

Written up for the Kape vault by BrewedOps.