

Visit bit.ly/ire24-cleaning

Basic concepts

replace and split are
99% of the game

66 years old

66 years old

66

convert to number

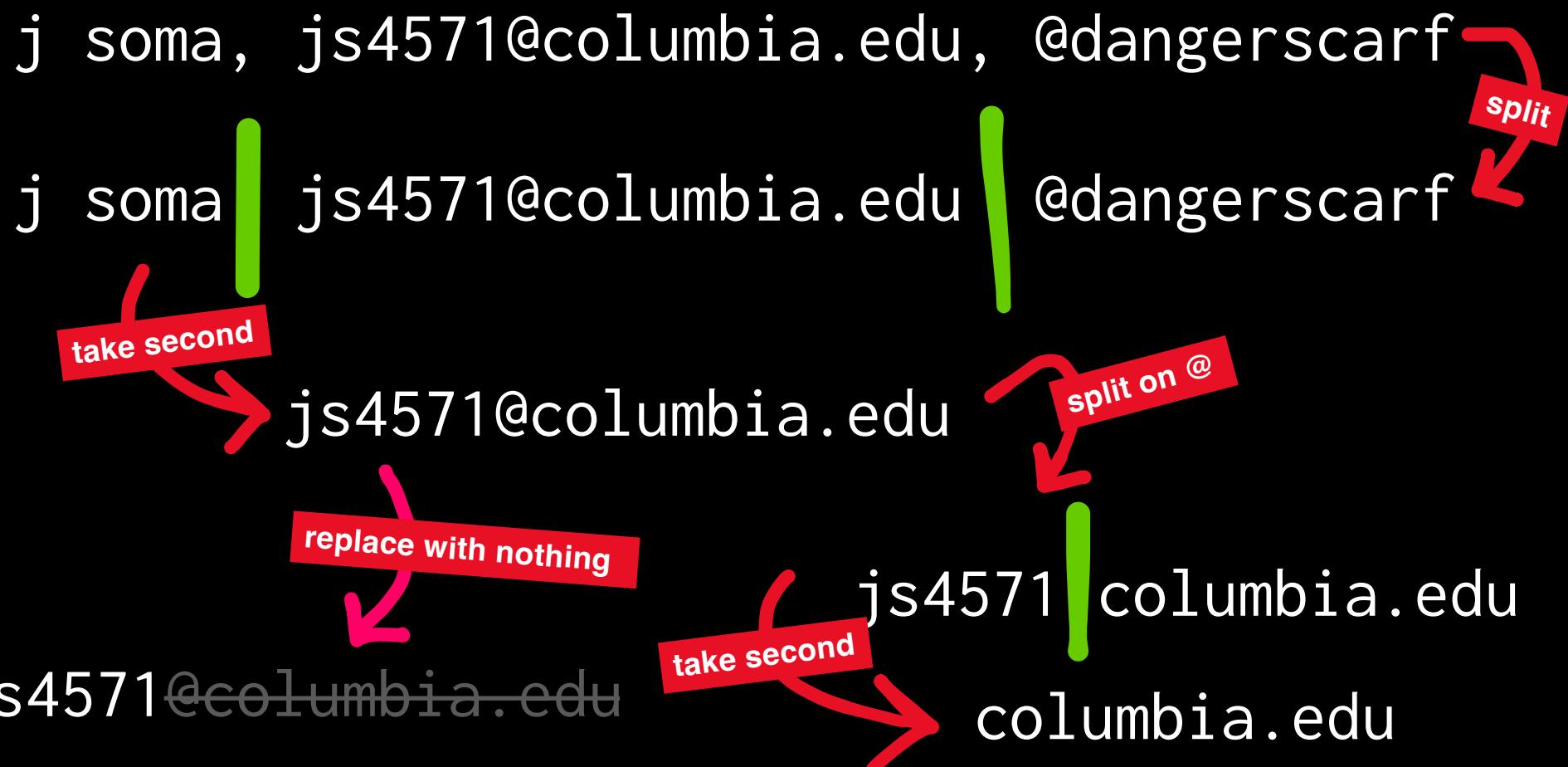
replace “years old”
with... nothing!

j soma, js4571@columbia.edu, @dangerscarf

j soma | js4571@columbia.edu | @dangerscarf

js4571@columbia.edu





ChatGPT

chatgpt.com/c/le21010b-bb12-4ba5-9a33-476e3f00d573

ChatGPT 4o

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a Google Sheet and I want to split it by ", ", take the second part, then replace the "@columbia.edu" with nothing. Just give me the formula.

< 2/2 >

You can use the following Google Sheets formula to achieve this:

plaintext

```
=SUBSTITUTE(INDEX(SPLIT(A1, ", "), 2), "@columbia.edu", "")
```

Copy code

Explanation:

1. `SPLIT(A1, ", ")` splits the text in cell `A1` by `","`
2. `INDEX(SPLIT(A1, ", "), 2)` selects the second part of the split text.
3. `SUBSTITUTE(INDEX(SPLIT(A1, ", "), 2), "@columbia.edu", "")` removes the "@columbia.edu" part from the selected text.

Is this conversation helpful so far?

Message ChatGPT

ChatGPT can make mistakes. Check important info.

ChatGPT

chatgpt.com/c/le21010b-bb12-4ba5-9a33-476e3f00d573

ChatGPT 4o

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a Google Sheet and I want to extract the "@columbia.edu" part. Write me a formula to do this. Only show me the formula.

< 3/3 >

excel

=RIGHT(A1,LEN(A1)-FIND("@",A1)+1)

Copy code

Is this conversation helpful so far?

Message ChatGPT

ChatGPT can make mistakes. Check important info.

Feel free to reach out at 555-555-5555



take the final 12 letters



555-555-5555

Split, Extract, Remove Domain x +

chatgpt.com/c/1e21010b-bb12-4ba5-9a33-476e3f00d573

ChatGPT 4o JS

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a pandas dataframe column named "details" and I want to split it by ", ", take the second part, then replace the "@columbia.edu" with nothing. Try not to use .apply. Don't create the df, just give me the code.

< 6/8 >

Certainly! Here's the code to achieve that:

```
python
# Split the 'details' column by ", " and take the second part
df['email'] = df['details'].str.split(', ').str[1]

# Replace "@columbia.edu" with nothing
df['email'] = df['email'].str.replace('@columbia.edu', '', regex=False)
```

This will split the column and then replace the domain part as required.

Message ChatGPT

ChatGPT can make mistakes. Check important info.

Split, Extract, Remove Domain x +

chatgpt.com/c/1e21010b-bb12-4ba5-9a33-476e3f00d573

ChatGPT 4o JS

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a pandas dataframe column named "details" and I want to extract the "@columbia.edu" part. Try not to use .apply. Don't create the df, just give me the code.

< 8/8 >

You can use the `str.extract` method to extract the "@columbia.edu" part from the "details" column. Here's the code to do that:

```
python
df['extracted'] = df['details'].str.extract(r'(\b@\w+\.\edu\b)')
```

Copy code

Is this conversation helpful so far? ⌂ ⌂ X

Message ChatGPT

ChatGPT can make mistakes. Check important info. ?

Regular expressions

regex is the other
other 1%

66 years old



(66) years old

(.*) years old

I am 66 years old



I am (66) years old



I am (.*) years old

Contact me at 555-555-5555

Contact me at (.*)

but maybe not!

Contact me at 555-555-5555 or
js4571@columbia.edu

Contact me at (\d\d\d-\d\d\d-\d\d\d\d)

just numbers

Contact me at 555-555-5555

(\d\d\d-\d\d\d-\d\d\d\d)

maybe we don't
need an anchor?

Contact me at 5555555555

(\d\d\d-?\d\d\d-?\d\d\d\d)

maybe hyphens
are optional?

Contact me at 555+555+5555

(\d\d\d.?\d\d\d.?\d\d\d\d)

any character is fine!
(and also optional)

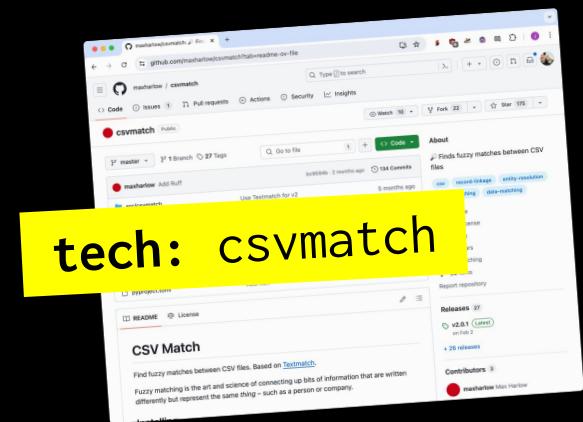
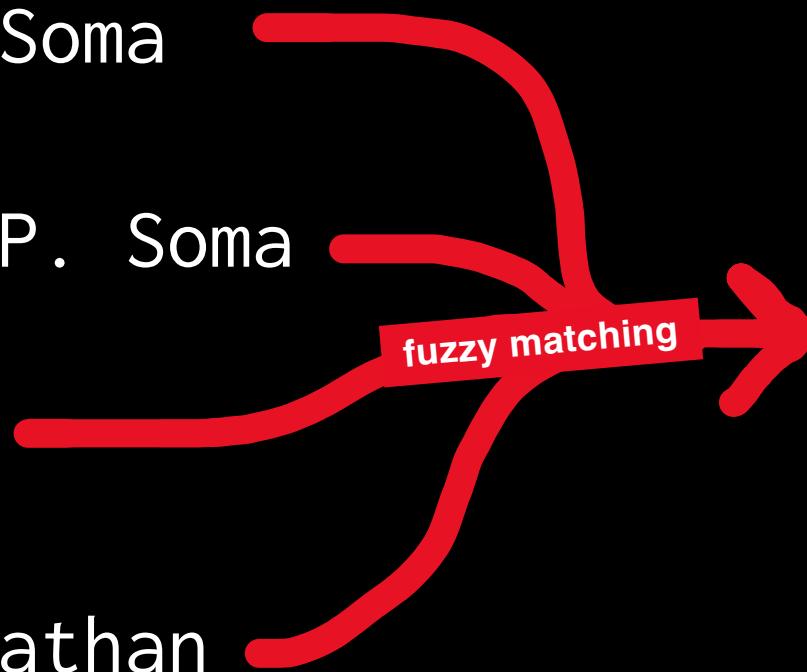
Fuzzy matching

Jonathan Soma

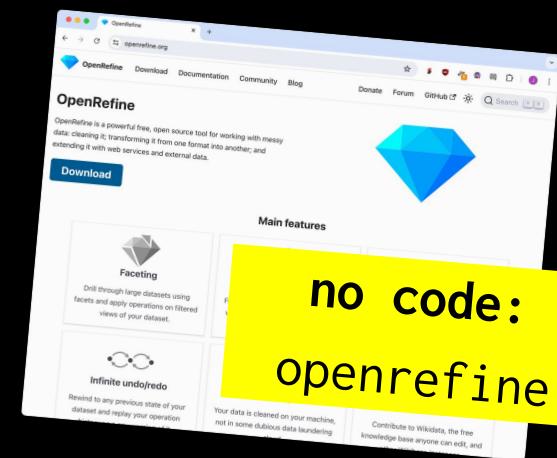
Jonathan P. Soma

Jon Soma

Soma, Jonathan



Jonathan Soma



Addresses and maps



Plain address

Geocoding



Latitude/Longitude

Reverse geocoding

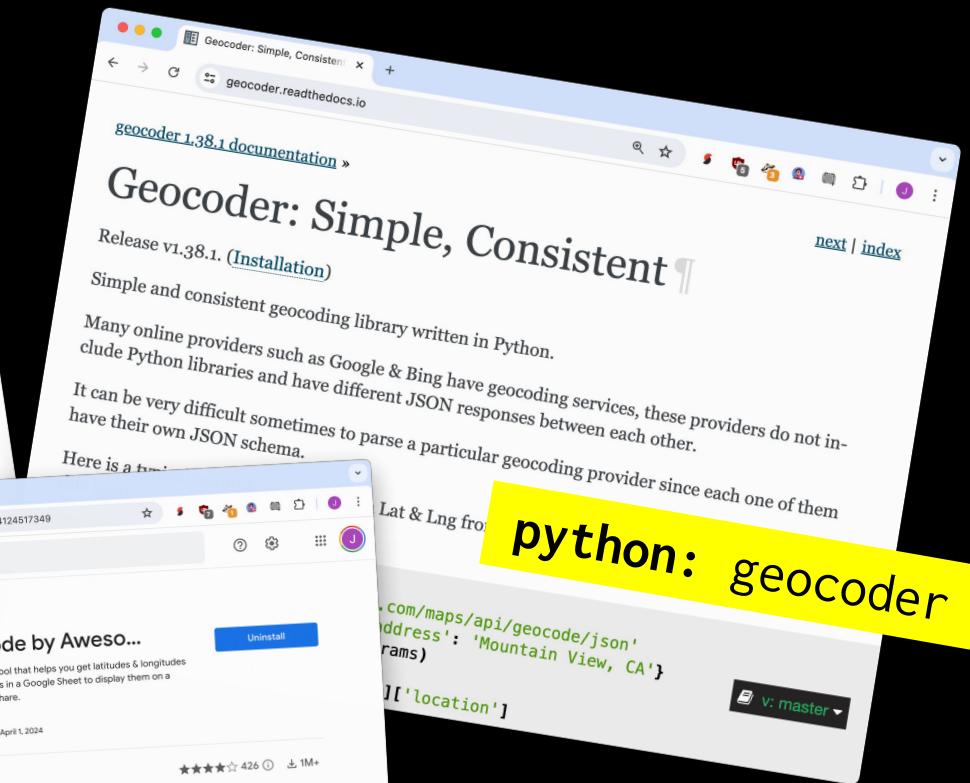
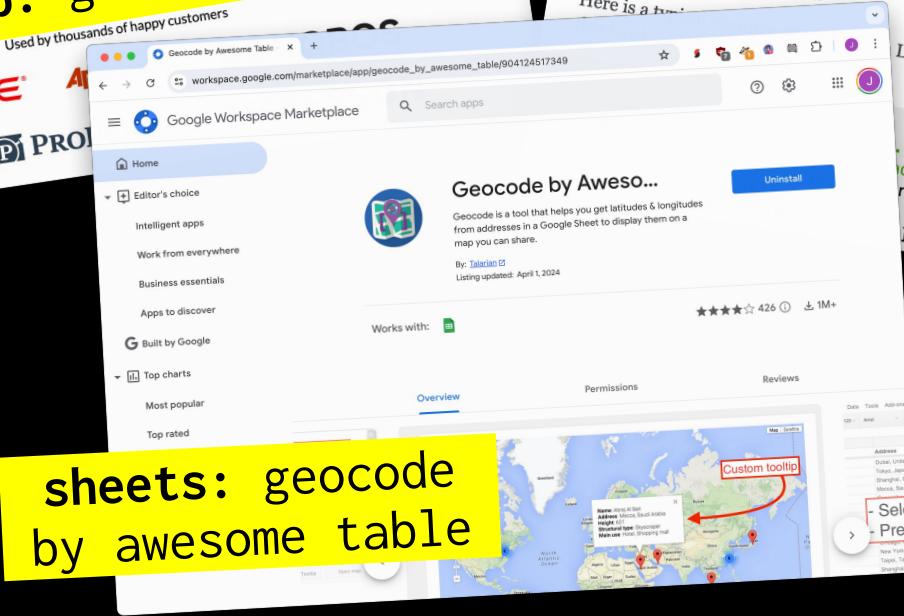
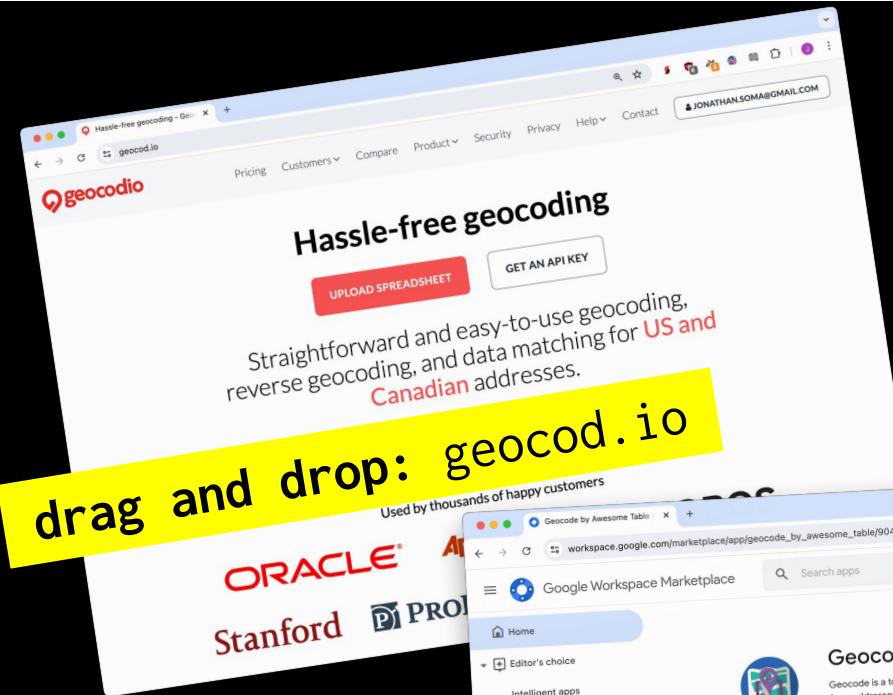
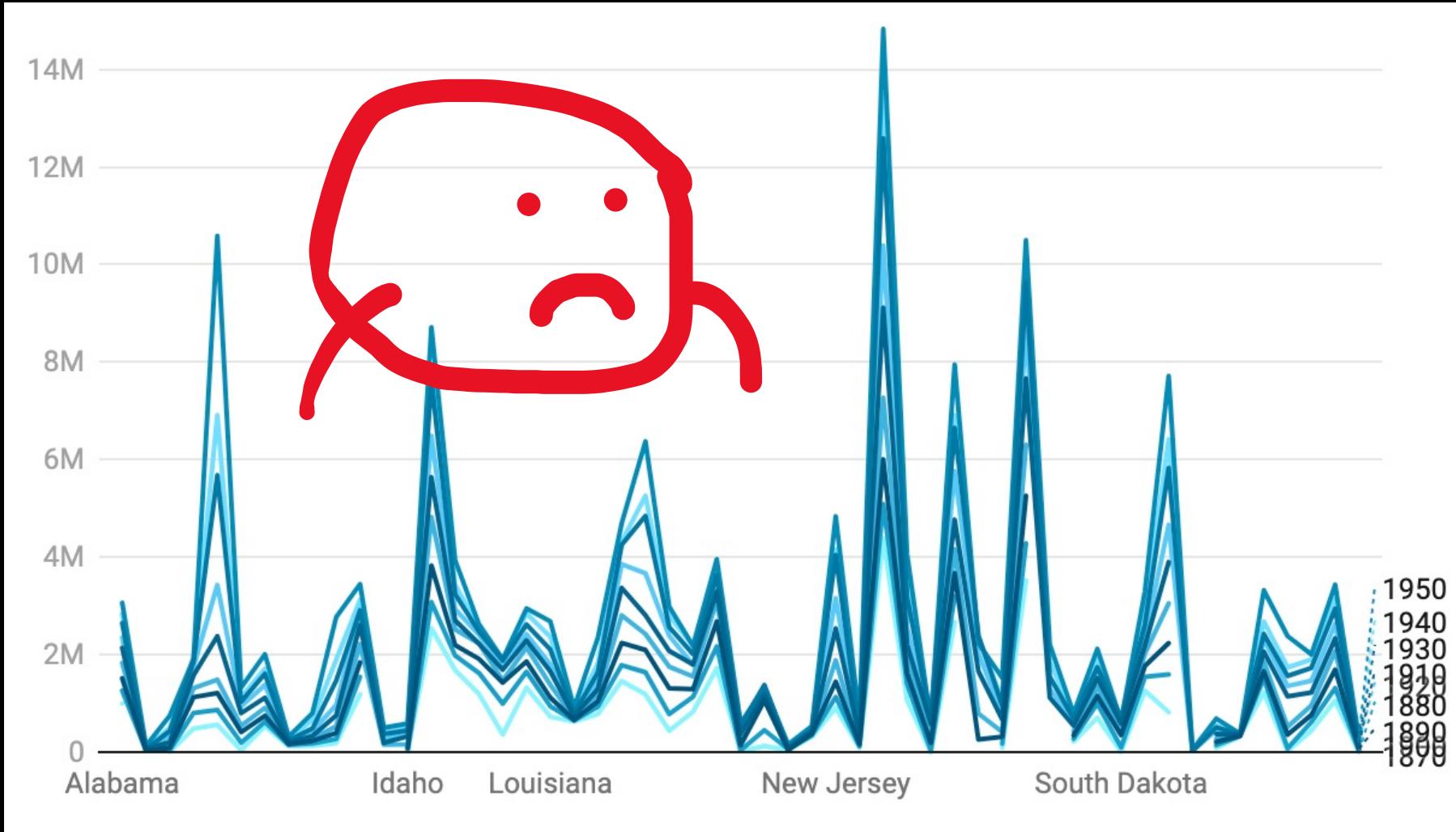


Table formatting



[Get the data](#) • Created with [Datawrapper](#)



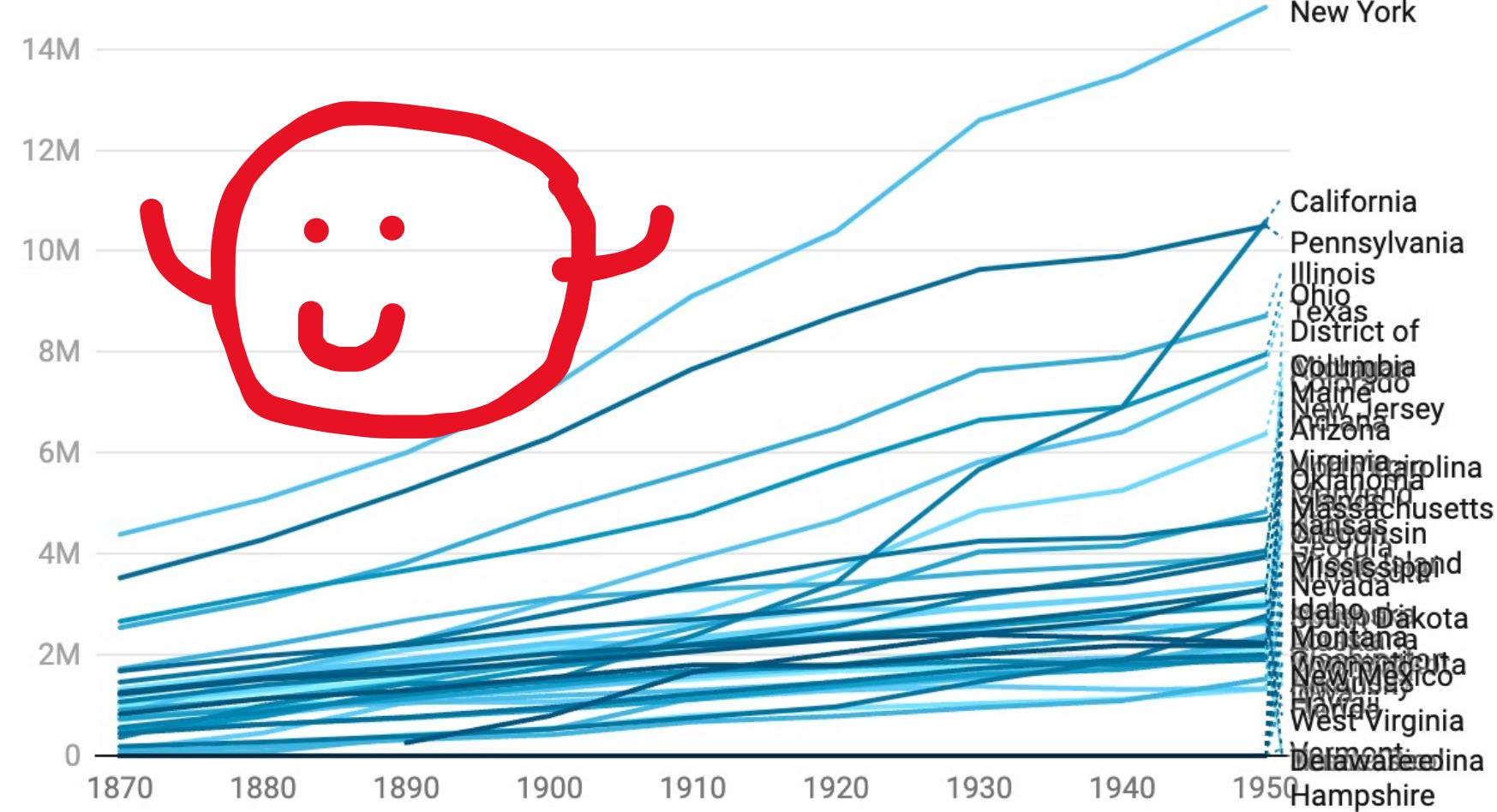
The diagram illustrates the transpose operation on a table. A red arrow points from the original table (top) to the transposed table (bottom), labeled "transpose (swap rows/columns)".

Original Table (Top):

	A	B	C	D	E	F	G	H	I	J	
1	Name	1870	1880	1890	1900	1910	1920	1930	1940	1950	
2	Alabama	996,992	1,262,505	1,513,401	1,828,697	2,138,093	2,348,174	2,646,248	2,832,961	3,061,743	
3	Alaska	—	33,426	32.052	63,592	64,356	55,036	59,278	72,524	128,643	
4	Arizona	9,658	40,440	88.243	122,931	204,354	334,162	435,573	499,261	749,587	
5	Arkansas	484,471	560,247	802,525	1,311,564	1,574,449	1,752,204	1,854,482	1,949,387	1,909,511	
6	California	560,247	622,700	746,258	864,694	924,322	2,377,549	3,426,861	5,677,251	6,907,387	10,586,223
7	Colorado	537,454	622,700	746,258	864,694	924,322	1,380,631	1,606,903	1,709,242	2,007,280	
8	Connecticut	125,015	146,608	168,493	184,735	202,322	223,003	238,380	266,505	318,085	
9	Delaware	—	278,718	331,069	437,571	486,869	663,091	802,178			
10	District of Columbia	131.7	187,74	177.624	269,49	230.392	391,42	278.718	528,54	331.069	752,61

Transposed Table (Bottom):

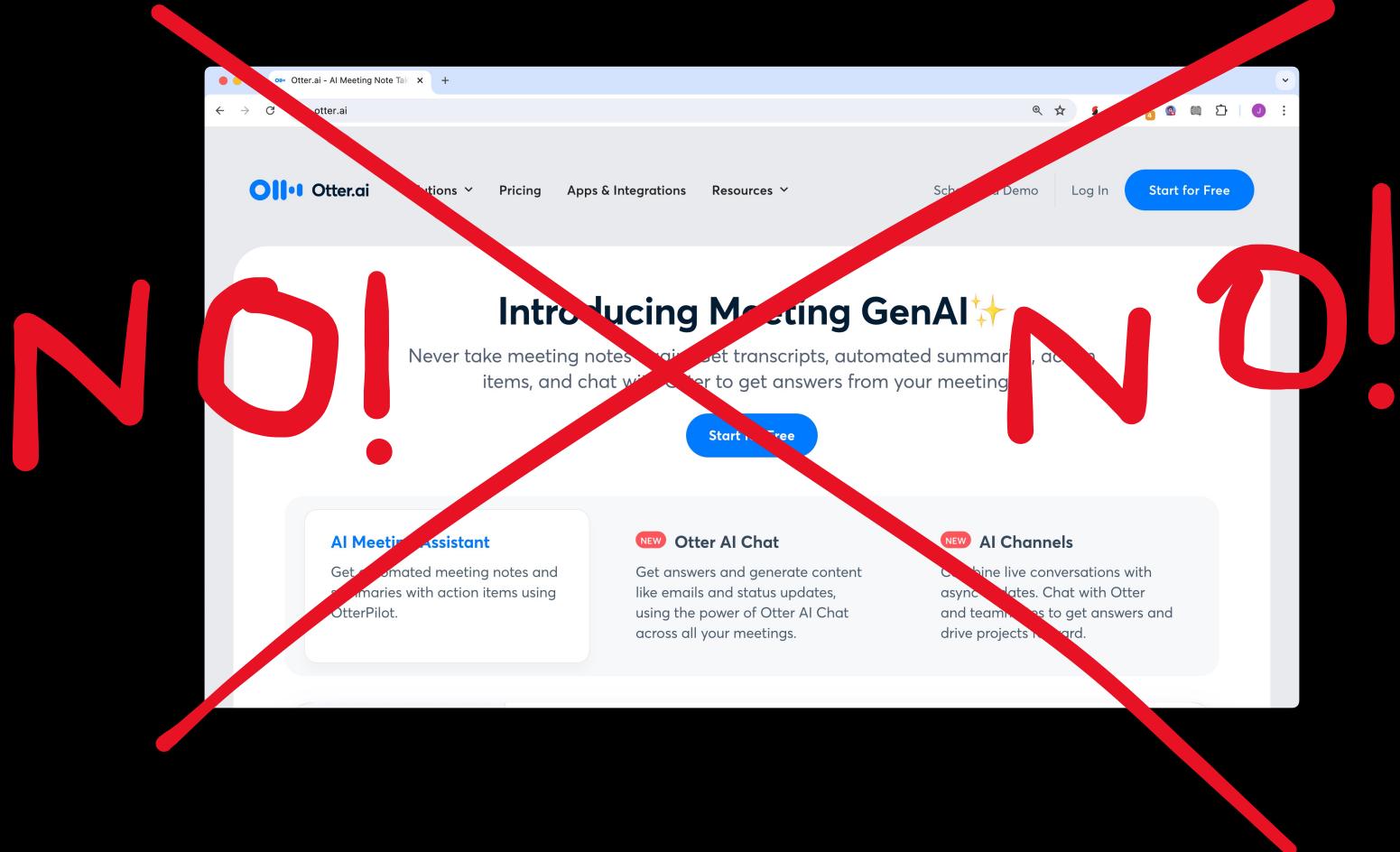
	A	B	C	D	E	F	G	H	I	J	K
1	Name	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut	Delaware	District of Columbia	Florida
2	1870	996,992	—	9.658	484,471	560,247	39,864	537,454	125.015	131.7	187,74
3	1880	1,262,505	33.426	40.44	802,525	864,694	194,327	622,700	146.608	177.624	269,49
4	1890	1,513,401	32.052	88.243	1,128,211	1,213,398	413,249	746,258	168.493	230.392	391,42
5	1900	1,828,697	63.592	122.931	1,311,564	1,485,053	539,700	908,420	184.735	278.718	528,54
6	1910	2,138,093	64.356	204.354	1,574,449	2,377,549	799,024	1,114,756	202.322	331.069	752,61
7	1920	2,348,174	55.036	334.162	1,752,204	3,426,861	939,629	1,380,631	223.003	437.571	968,47
8	1930	2,646,248	59.278	435.573	1,854,482	5,677,251	1,035,791	1,606,903	238.38	486.869	1,468,21
9	1940	2,832,961	72.524	499.261	1,949,387	6,907,387	1,123,296	1,709,242	266.505	663.091	1,897,41
10	1950	3,061,743	128.643	749.587	1,909,511	10,586,223	1,325,089	2,007,280	318.085	802.178	2,771,30



[Get the data](#) • Created with Datawrapper



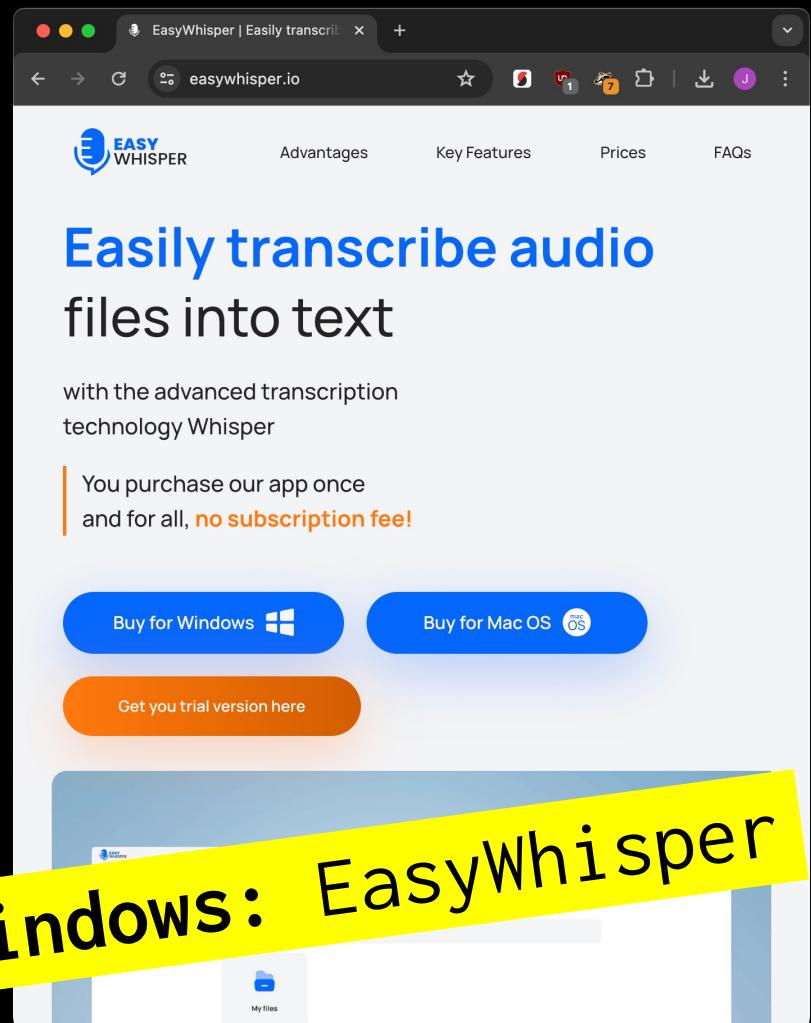
Transcribing audio



magic technology: OpenAI's Whisper

The screenshot shows the GitHub repository page for `openai/whisper`. The repository has 63 pull requests and 7.4k forks. A yellow callout box highlights the title "magic technology: OpenAI's Whisper". The repository page includes a table comparing different model sizes based on size, parameters, English-only model variants, multilingual model variants, required VRAM, and relative speed.

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	<code>tiny.en</code>	<code>tiny</code>	~1 GB	~32x
base	74 M	<code>base.en</code>	<code>base</code>	~1 GB	~16x
small	244 M	<code>small.en</code>	<code>small</code>	~2 GB	~6x
medium	769 M	<code>medium.en</code>	<code>medium</code>	~5 GB	~2x
large	1550 M	N/A	<code>large</code>	~10 GB	1x



Practical AI for Investigative Journalism

youtube.com/playlist?list=PLewNEVDy7gq1_GPUaL0OQ31QsiHP5ncAQ

YouTube

Search

Home

Shorts

Subscriptions

You >

Your channel

History

Playlists

Your videos

Watch later

Liked videos

Your clips

Subscriptions

Freya Holmér

Home RenoVisio...

1kb construction

Javier Mercedes

Bill McClintock

Dearest friend, I daresay I have not partaken of food in ages. I'm positively famished

Practical AI for Investigative Journalism

Jonathan Soma

6 videos Public

A six session series held in April 2024 about real-life use cases for journalism in (mostly investigative) jour ...more

Play all

Sort

Dearest friend, I daresay I have not partaken of food in ages. I'm positively famished 2:09:33

Sorting documents (Practical AI for Investigative Journalism, Session 1)

Jonathan Soma • 1K views • Streamed 2 months ago

Large language models don't understand facts or concepts, they only know statistical probability 2:14:32

Structured, validated data from LLMs (Practical AI for Investigative Journalism, Session 2)

Jonathan Soma • 716 views • Streamed 2 months ago

Evaluating Verifiability in Generative Search Engines 1:45:26

Why generative AI is a dead end for responsible journalism (Practical AI for Journalism, Session 3)

Jonathan Soma • 572 views • Streamed 1 month ago

AI, Hugging Face and non-chatbot models (Practical AI for Journalism, Session 4)

Jonathan Soma • 810 views • Streamed 1 month ago

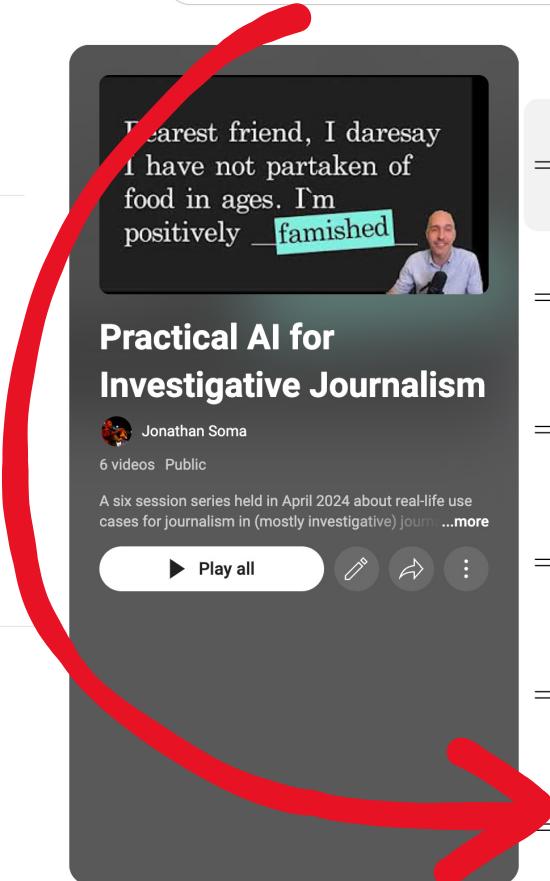
Context window 2:01:02

Local models/private AI (Practical AI for Investigative Journalism, Session 5)

Jonathan Soma • 444 views • Streamed 1 month ago

Transcription and audio models (Practical AI for Investigative Journalism, Session 6)

Jonathan Soma • 325 views • Streamed 1 month ago



***Generating structured data
with the awful power of LLMs***

FROM: Mulberry Peppertown
(mulbs@example.com)

When I pick up the cans of beans they are all so light! At first I thought they were empty, but it turns out they are just futuristic beans that are not heavy like the old style beans I was used to. It is incredible.

Mulberry Peppertown

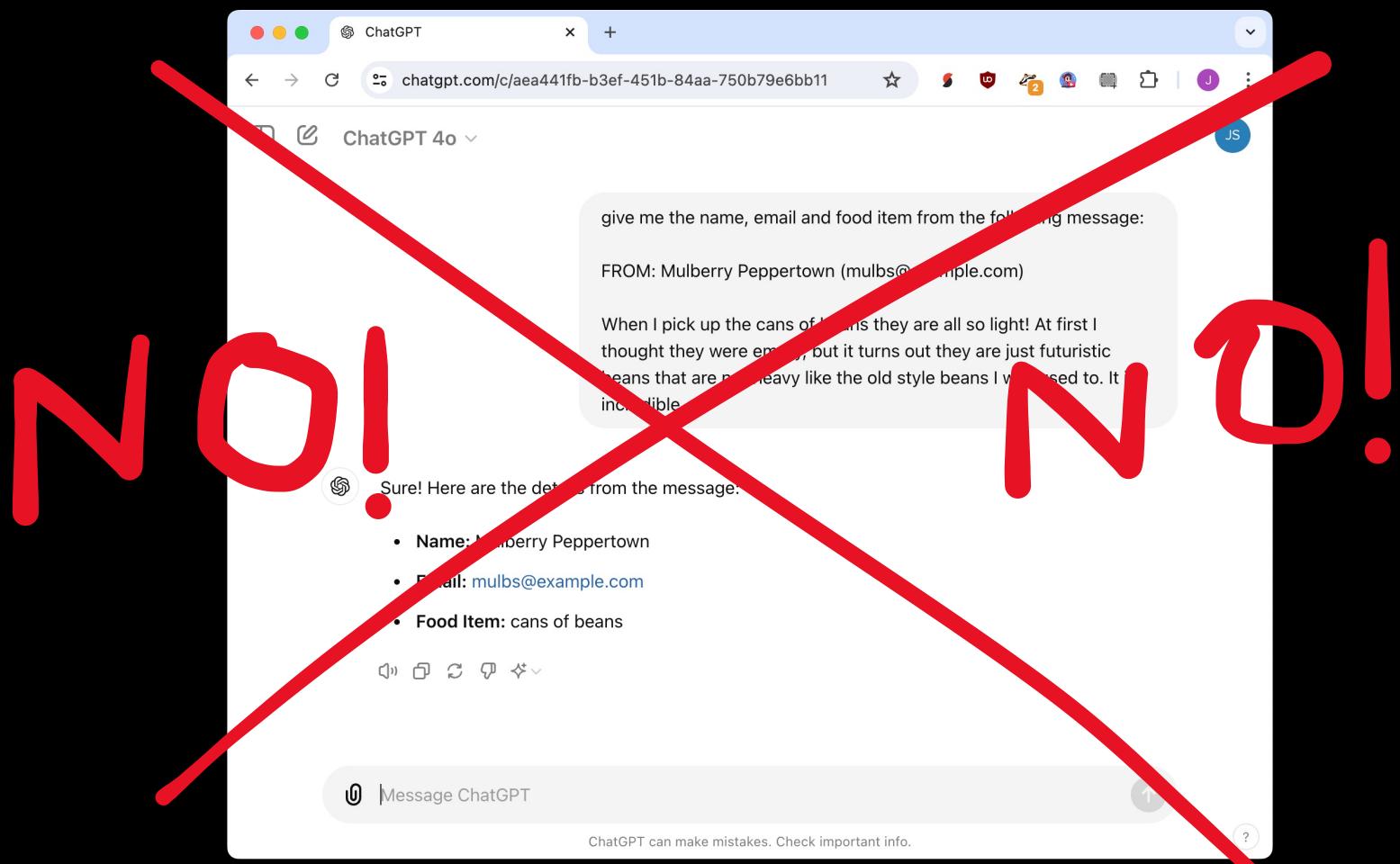
name

mulbs@example.com

email

beans

food item



email	name	email	product
FROM: Mulberry Peppertown (mulbs@example.com)			

When I pick up

=CLAUDEEXTRACT("grocery product, mention all if there are multiple", A2)

incredible.

I am irate about
shopping at your
broccoli mailin

Jackary Balon
jackary.balone

Google Sheets

Google Workspace Marketplace

CLAUDEEXTRACT("name", A2)

Claude for Sheets

Uninstall

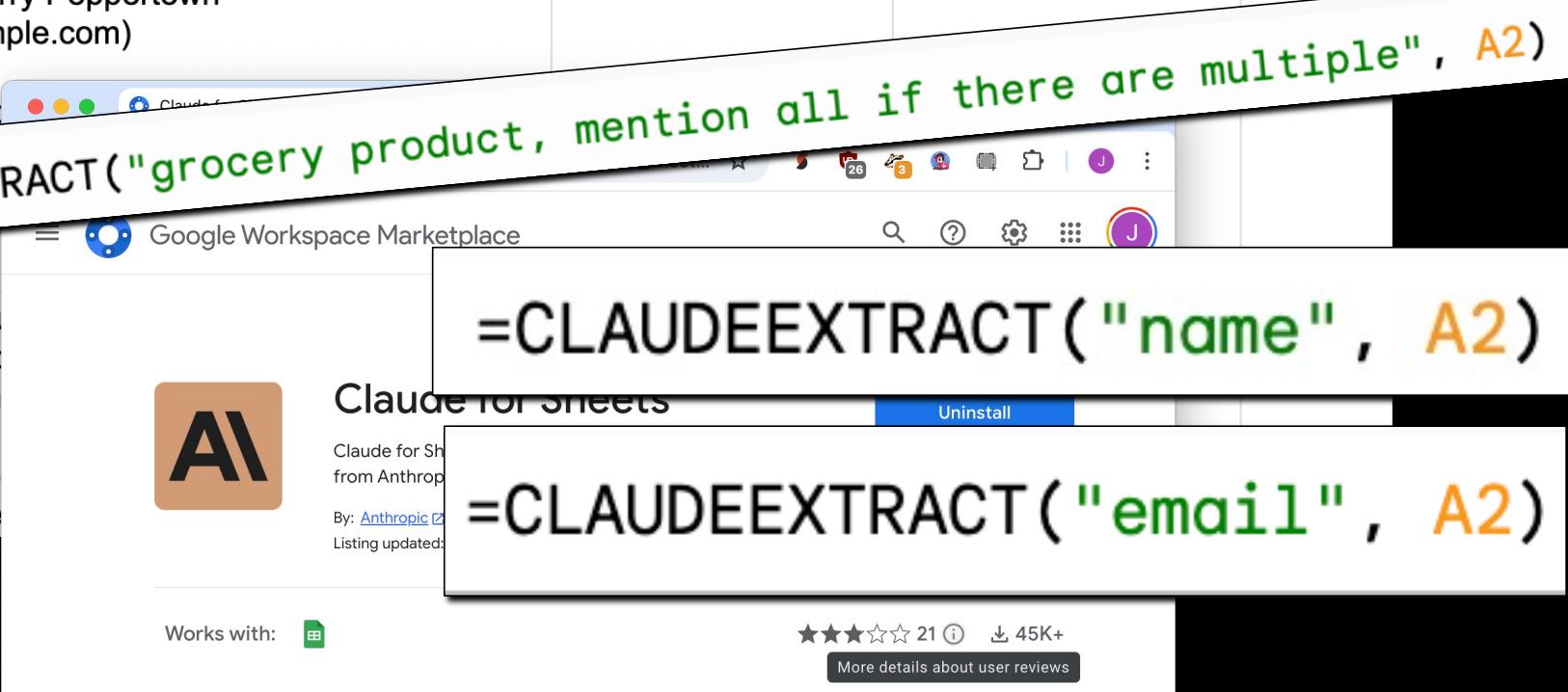
CLAUDEEXTRACT("email", A2)

Works with:

★★★★★ 21

45K+

More details about user reviews



Sheets: Claude for Sheets

Python: Instructor

```
class Comment(BaseModel):
    name: str = Field(description="Person who submitted the comment")
    email: Optional[str] = Field(description="Email address of commenter")
    food_item: str = Field(description="Food item the comment is about")
    emotion: Literal["positive", "negative", "uncertain"]
```

```
comment = """
FROM: Mulberry Peppertown, mulberry (at) example.co

When I pick up the cans of beans they are all so light.
first I thought they were empty, but it turns out to be futuristic beans that are not heavy like the old stuff I was used to. It is incredible.
"""

result = client.chat.completions.create(
    response_model=Comment,
    messages=[{"role": "user", "content": comment}],
```

```
{
    'name': 'Mulberry Peppertown',
    'email': 'mulberry@example.com',
    'food_item': 'cans of beans',
    'emotion': 'positive'
}
```

Visit bit.ly/ire24-cleaning