Working with AI, Data and Pattern

Janet Huang

2023.05.11

Outline

- Session I: Working with Data
  - What is data, personal data?
  - “Dear Data” exercise
  - “Data Digitalization” exercise
- Break
- Session II: Working with AI, Data and Pattern
  - OpenAI api on Data Foundry
  - Starboard as data pattern exploration tool
  - “Working with AI” exercise

Working with Data

personal data and your story

What is data?
What can be data?
And what data can be?

Data is not just numbers

because numbers are always a placeholder for something else

What is data?

What is Data?

“Data are factual information (such as measurements or statistics) used as a basic for reasoning, discussion, or calculation.”
(from Merriam-Webster)

“Data are units of information that are collected through observation.
In a more technical sense, data are a set of values of qualitative or quantitative variables about one or more persons or objects.”
(from Wikipedia)

Data is not just numbers

Data is about people;
Data is about story;
Data is the way we perceive the world.

because numbers are always a placeholder for something else

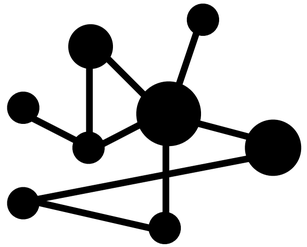
Data can be represented as many forms

images

texts

```
10100000111001
0101010100101001 01
1000111111111110100
0100100010011000101
0100011001010101 0
```

numbers



network

any forms

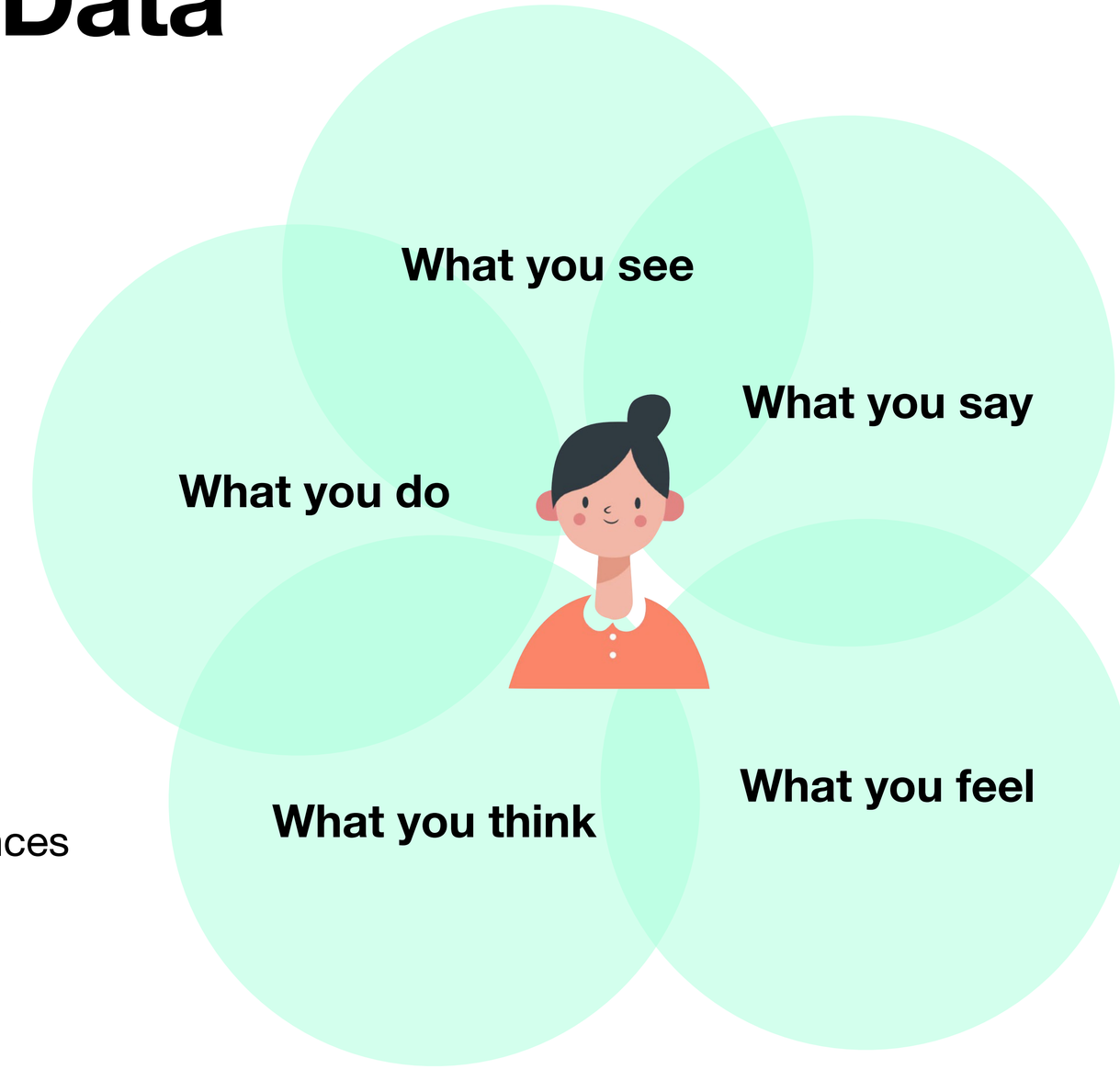
Personal Data

Daily photos

Todo list

Posts on social media

Smartphone app usage



Personality & Traits

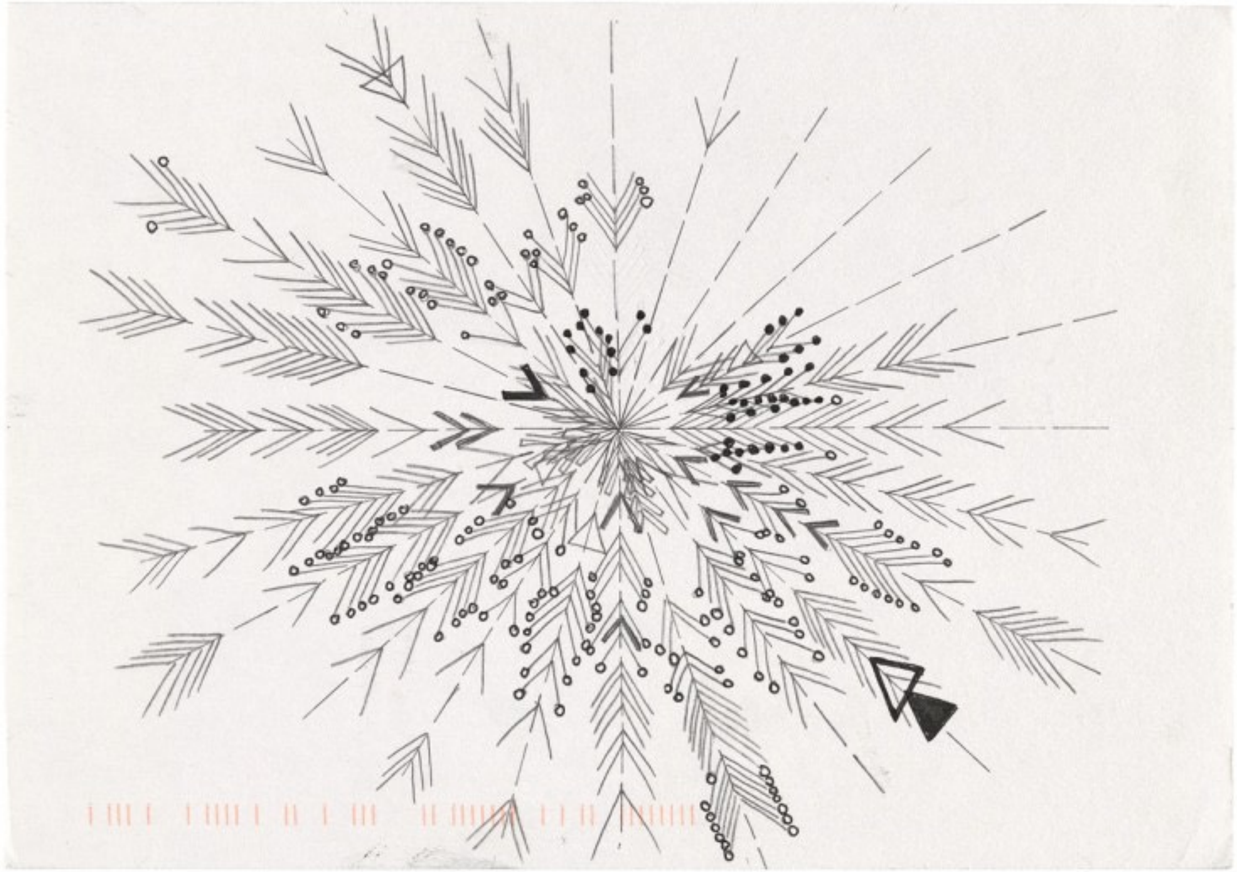
Interests & Preferences

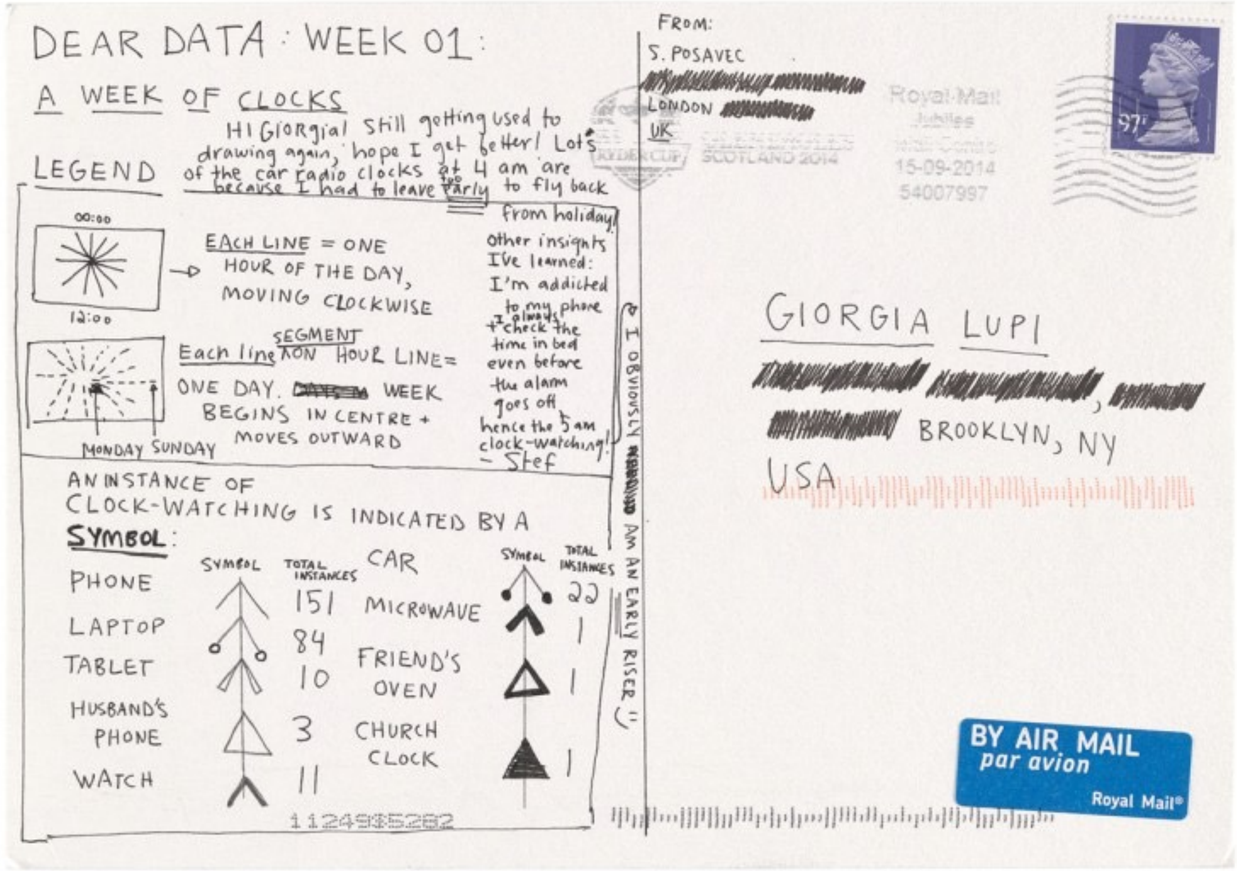
Emotion & Desires

Values & Beliefs


Dear Data

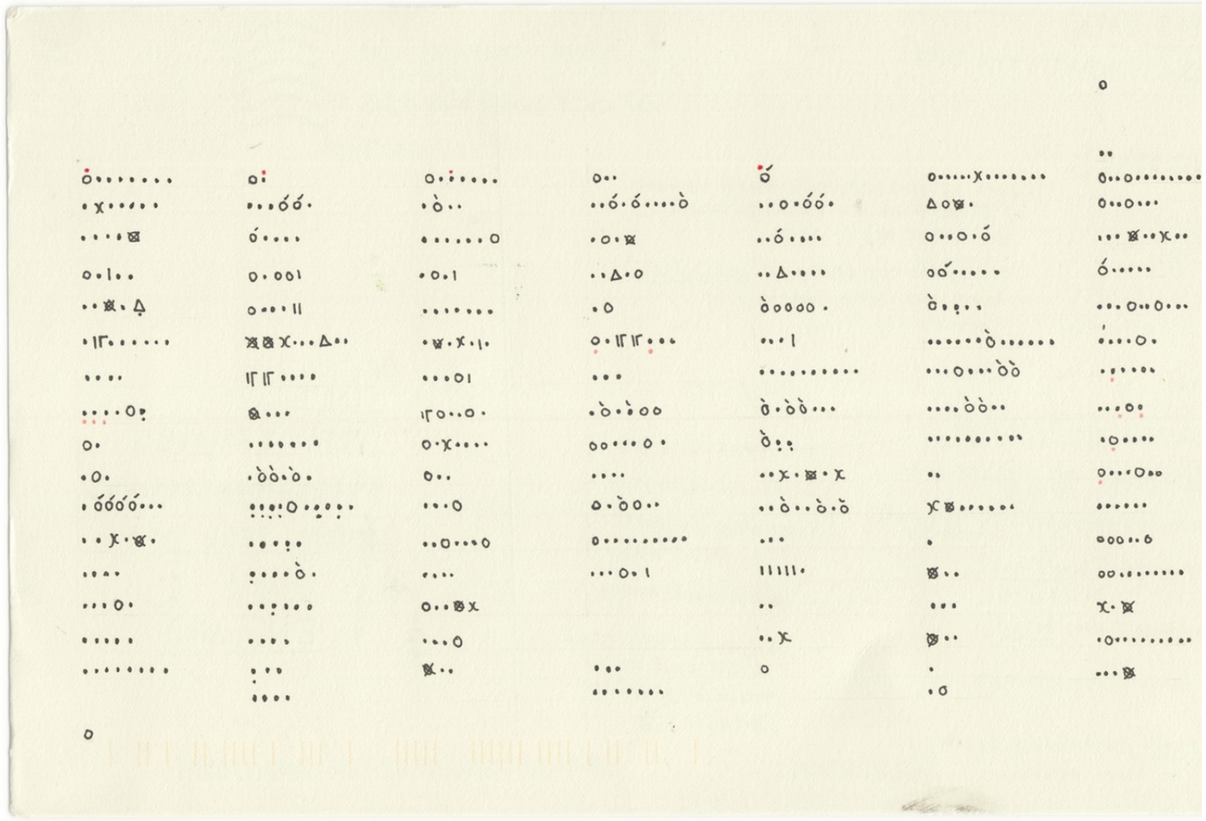
Week 1: A Week of Clocks (Stefanie)

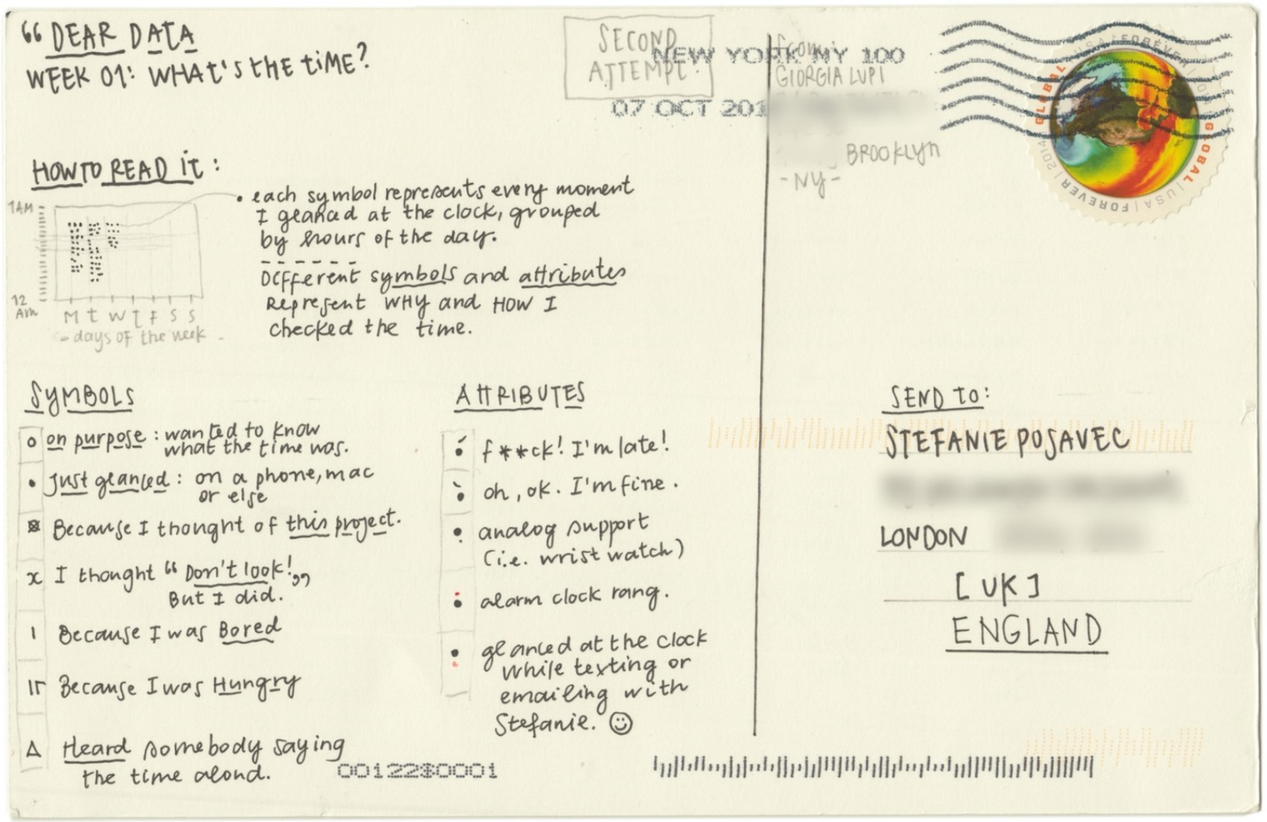




http://www.dear-data.com

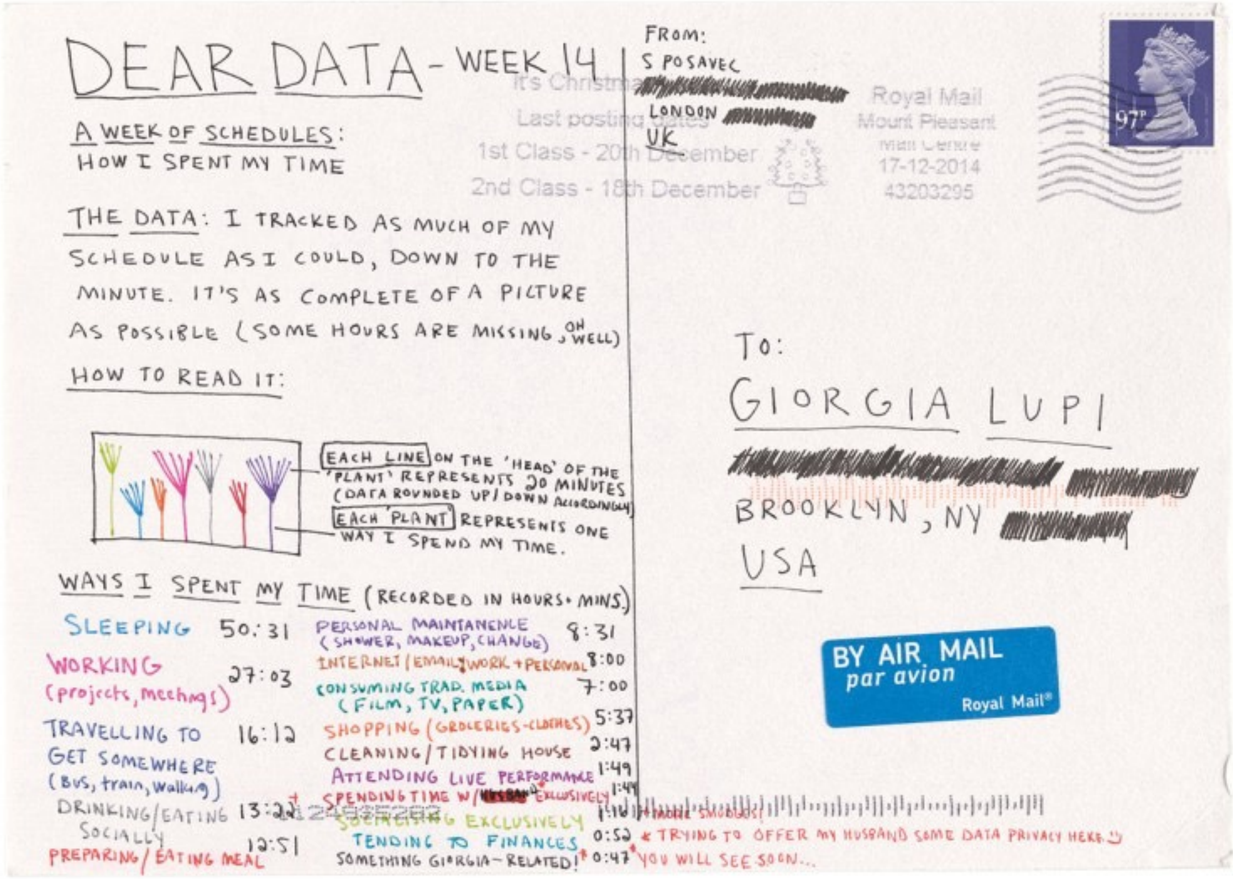
Week 1: A Week of Clocks (Giorgia)





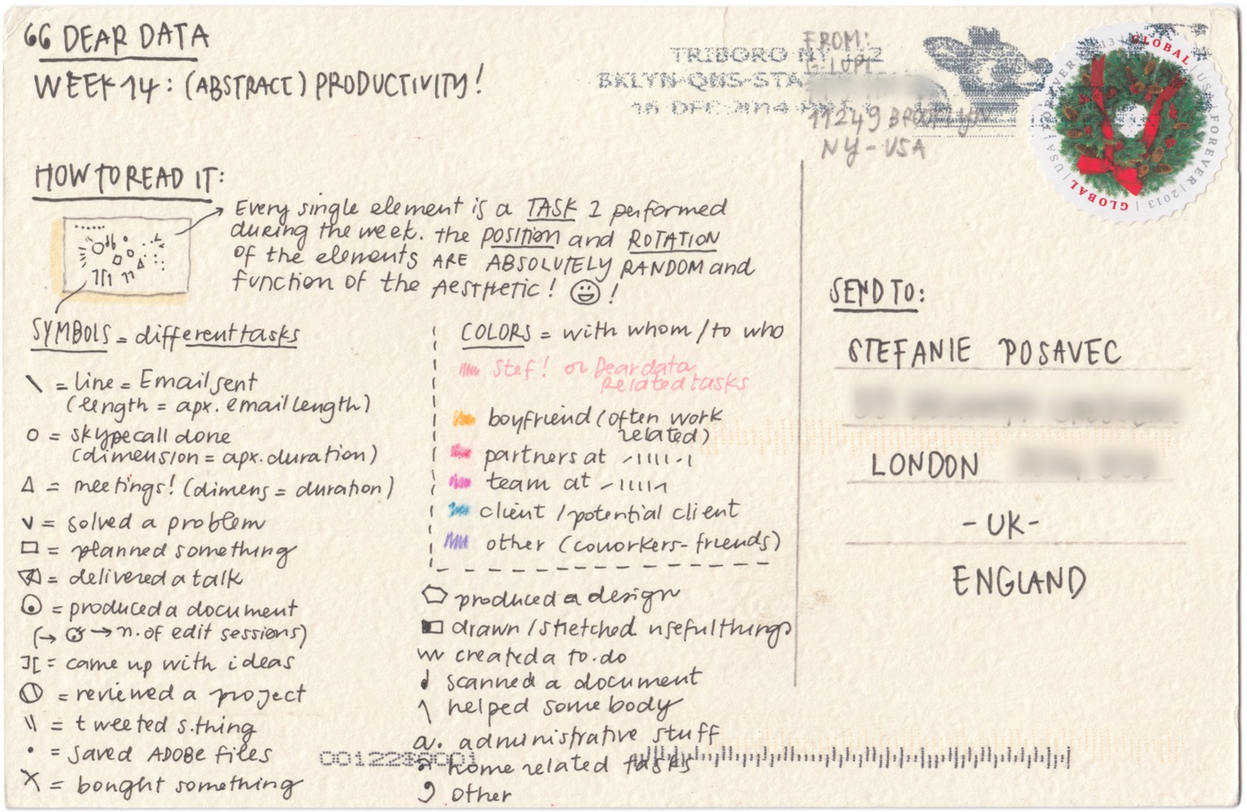
http://www.dear-data.com

Week 14: A Week of Schedules (Stefanie)



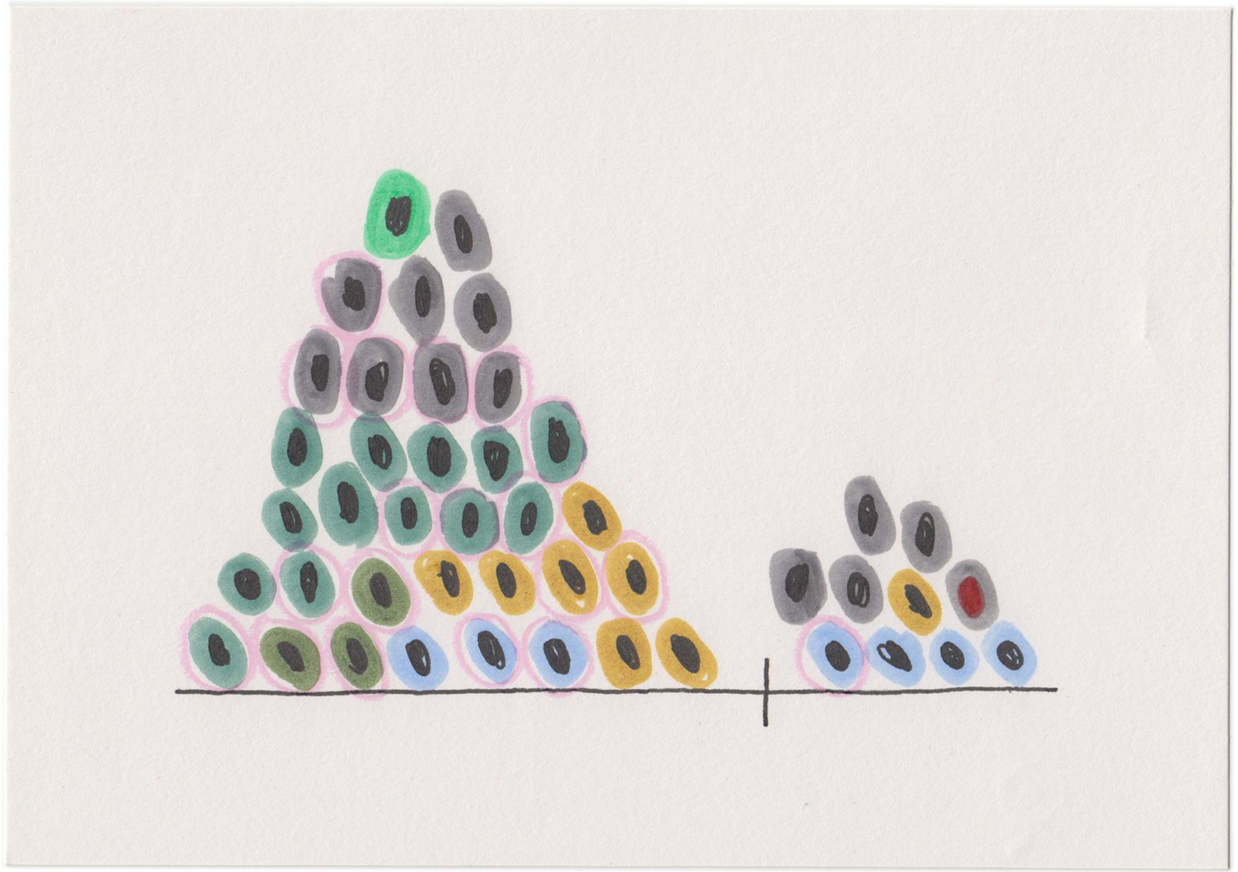
http://www.dear-data.com

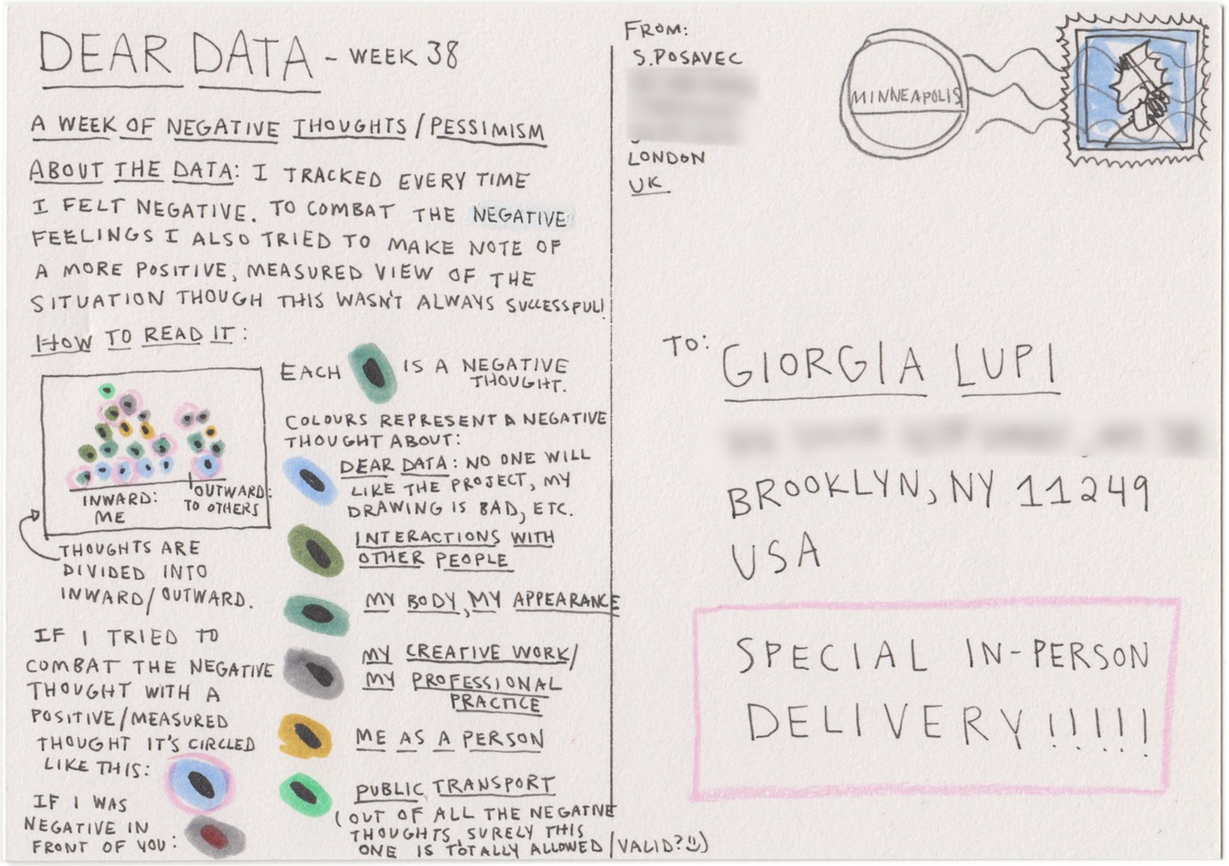
Week 14: A Week of Schedules (Giorgia)



http://www.dear-data.com

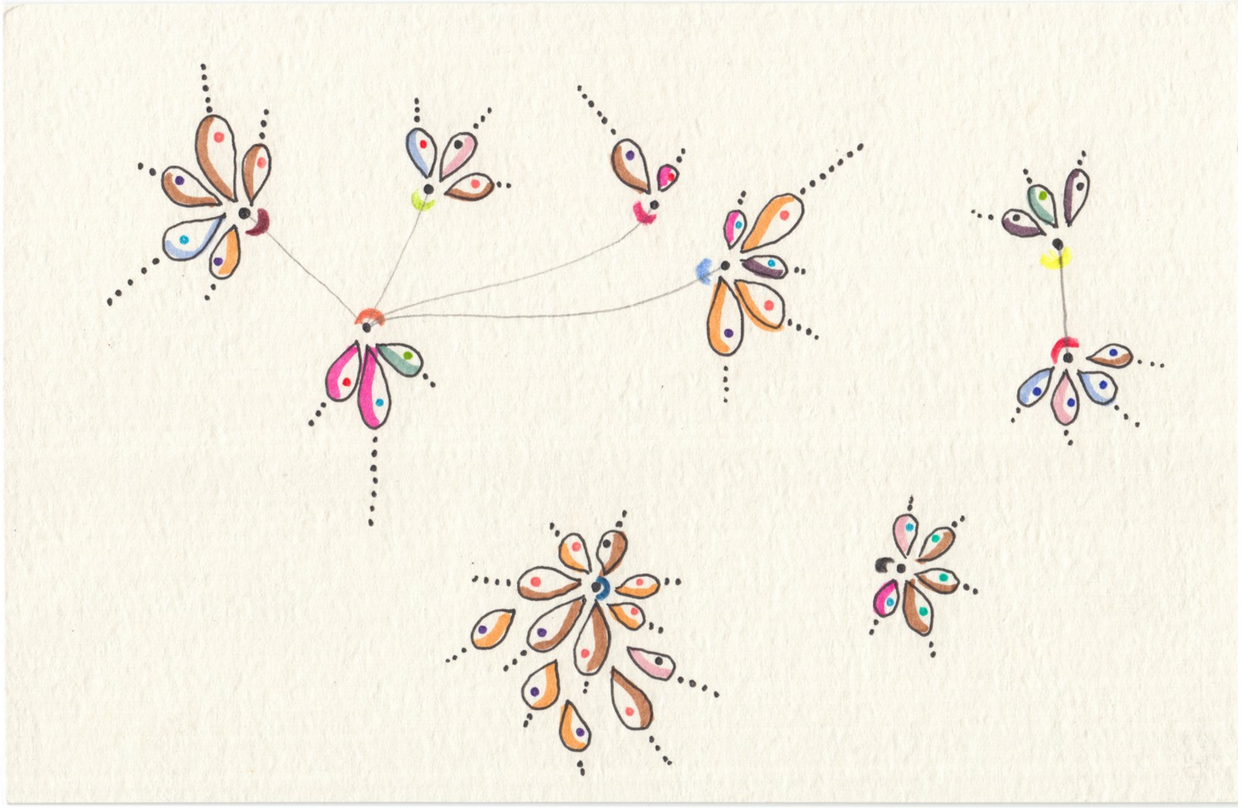
Week 38: A Week of Negative Thoughts (Stefanie)

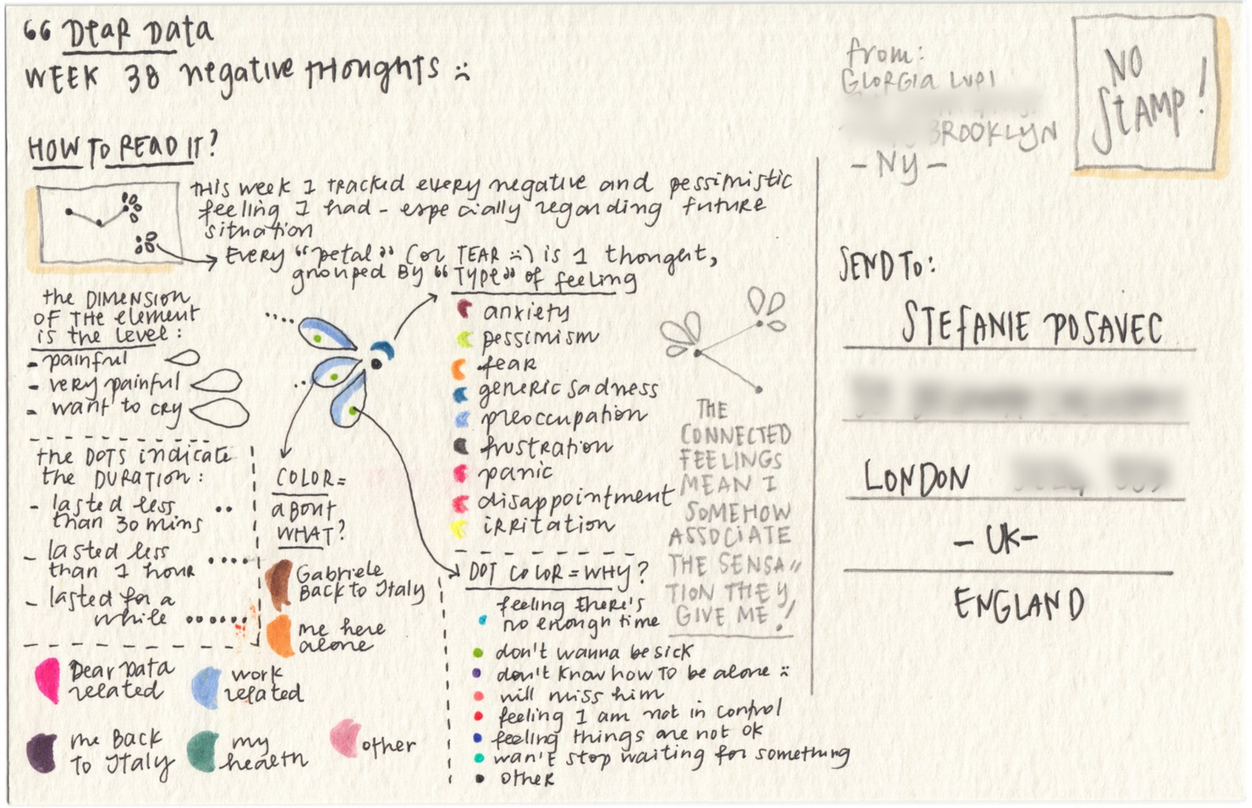




http://www.dear-data.com

Week 38: A Week of Negative Thoughts (Giorgia)





http://www.dear-data.com

“Dear Data” Exercise

- Step 1. Choose a topic: emotions, daily activity, todo-lists, complaints, drinks, etc
- Step 2. Reflect on your experiences last week and record the data for the chosen topic
  - Topics: feelings, activities, time, duration, thoughts, observations, etc.
- Step 3. Define your visual vocabulary
- Step 4. Draw your personal data!!
- Step 5. Share it to your group members

Topics

- to-do lists
- daily activities
- sounds
- books
- our cities
- transportation
- envy
- desires
- emotions
- complaints
- compliments
- thank-yous
- apologies
- food preferences
- drinks
- coffees
- doors/spaces/kitchen
- our time alone
- (…)

“Dear Data” Exercise (30 mins)

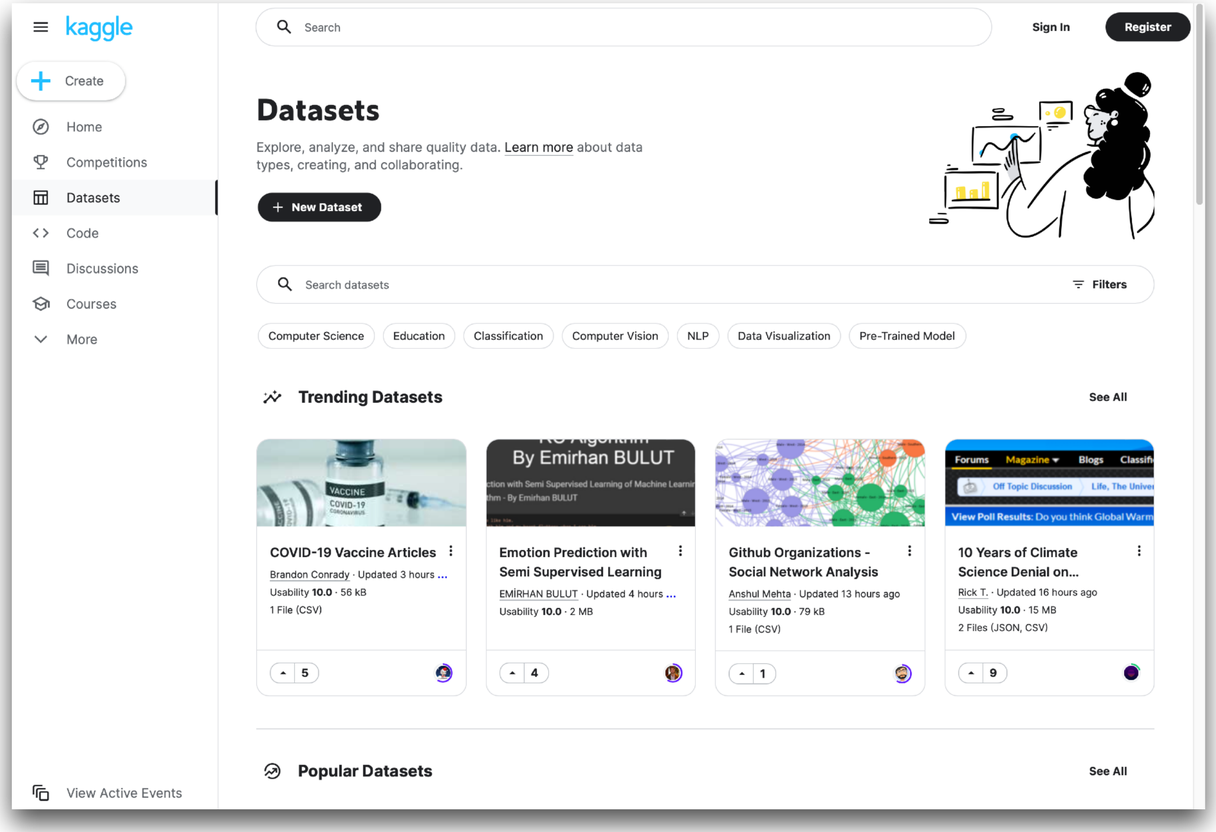
- Step 1. Choose a topic
- Step 2. Reflect on your experiences last week and record the data (e.g., feelings, activities, time, duration, thoughts, observations, etc) for the chosen topic
- Step 3. Define your visual vocabulary
- Step 4. Draw your personal data!!
- Step 5. Share it to your group members

Topics

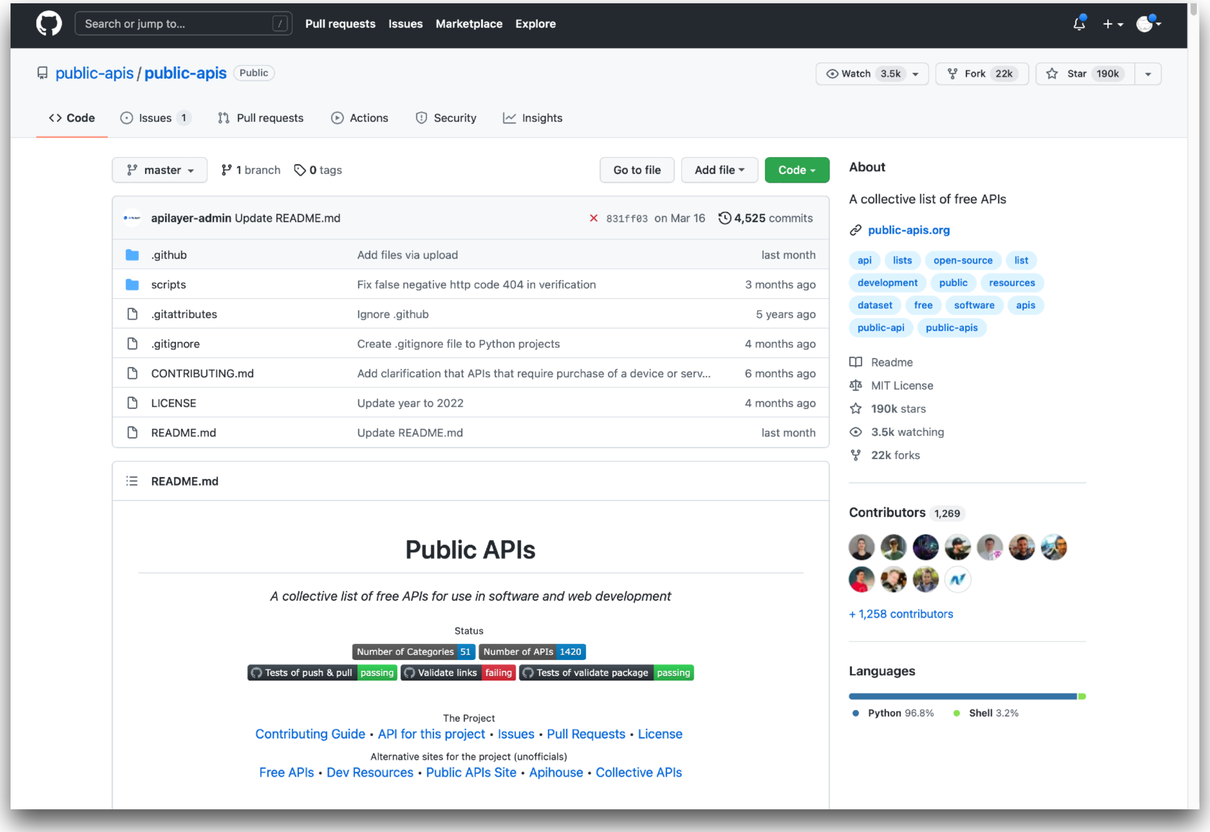
- to-do lists
- daily activities
- sounds
- books
- our cities
- transportation
- envy
- desires
- emotions
- complaints
- compliments
- thank-yous
- apologies
- food preferences
- drinks
- coffees
- doors/spaces/kitchen
- our time alone
- (…)

Inspired by Big Data

use existing datasets or public api to get big data



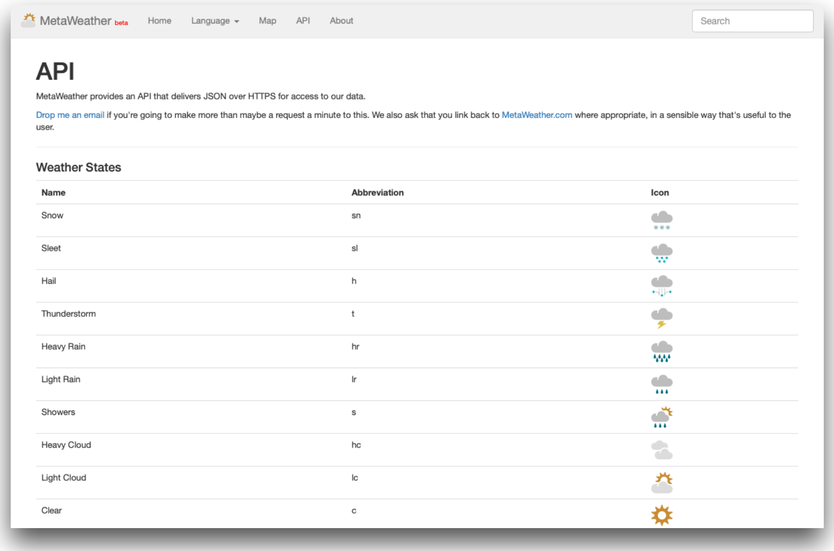
https://www.kaggle.com/datasets



https://github.com/public-apis/public-apis

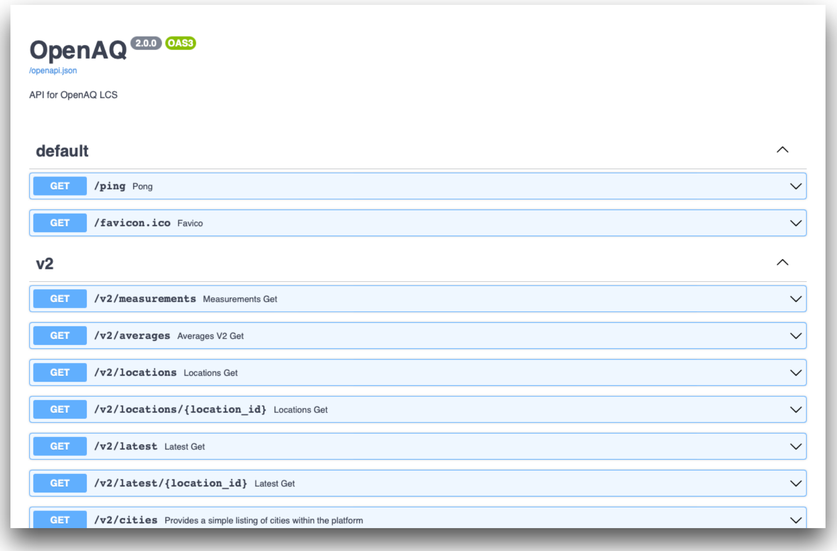
Example Datasets

Weather Data



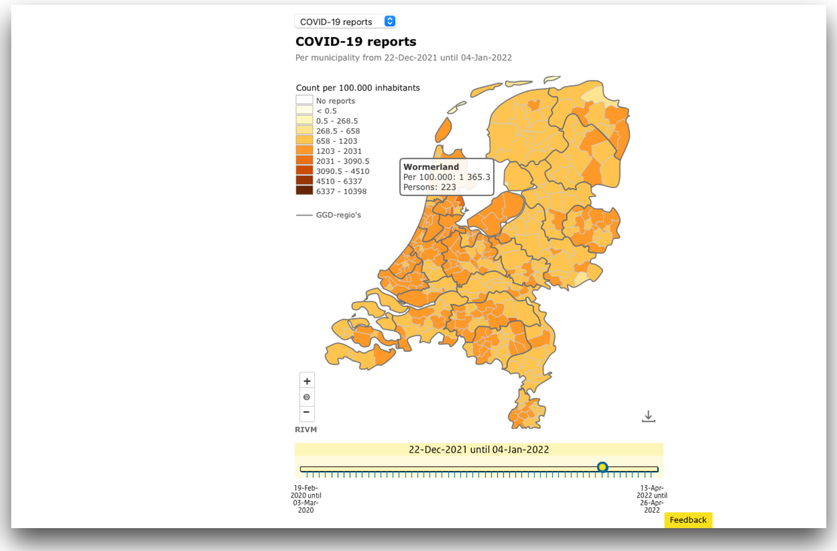
https://www.metaweather.com/api/

Air Quality Data



https://docs.openaq.org

COVID-19 Data



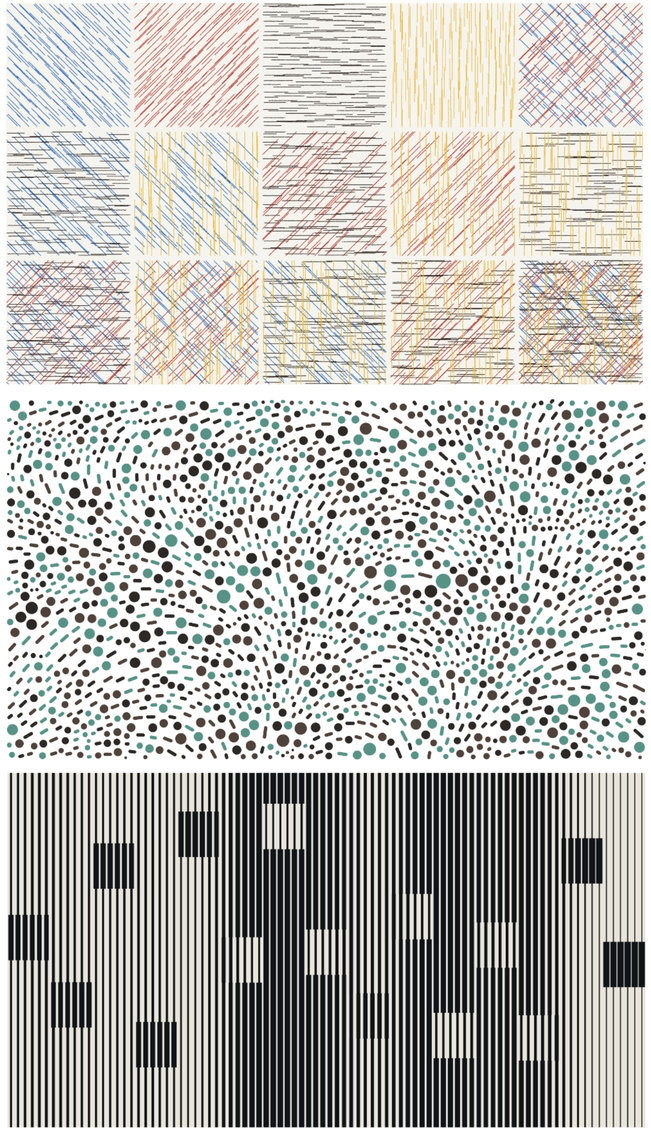
https://data.rivm.nl/covid-19/

dear data

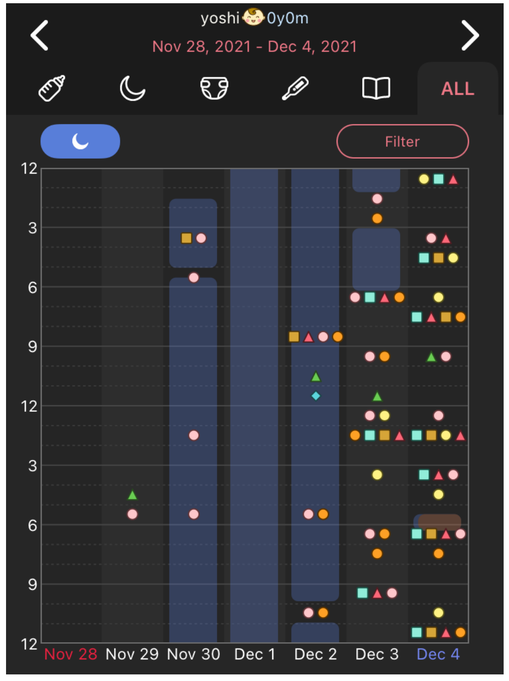
Data Digitalization

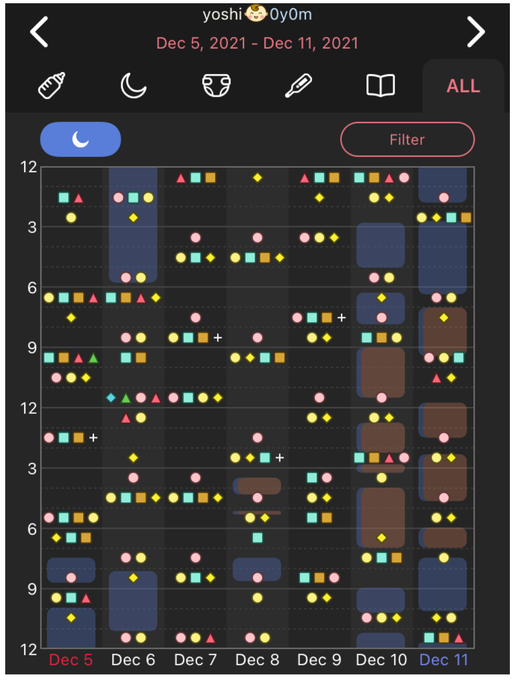
+

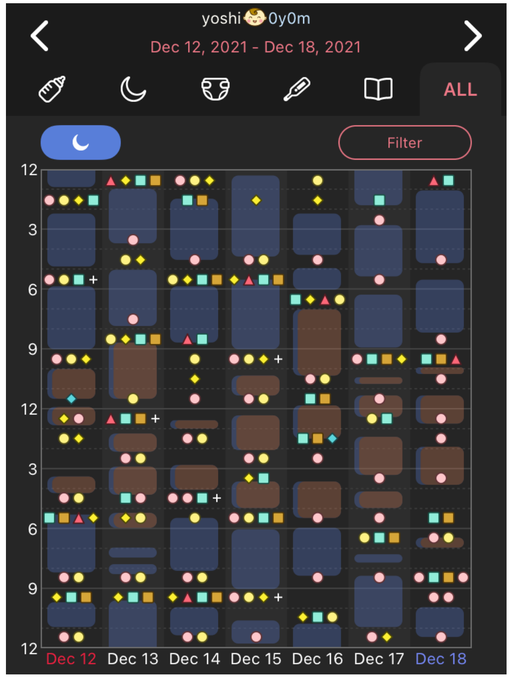
Math, Data, Computing

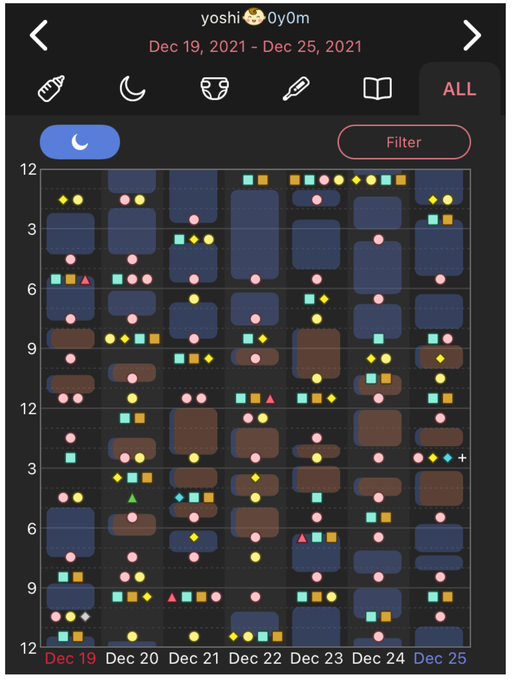


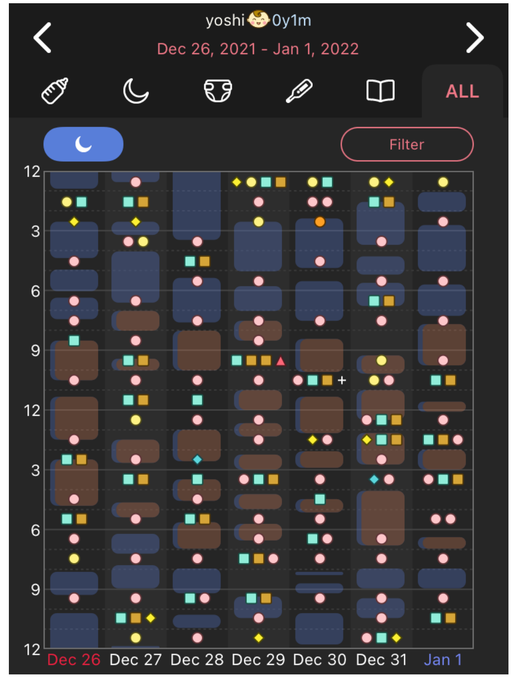
data pattern

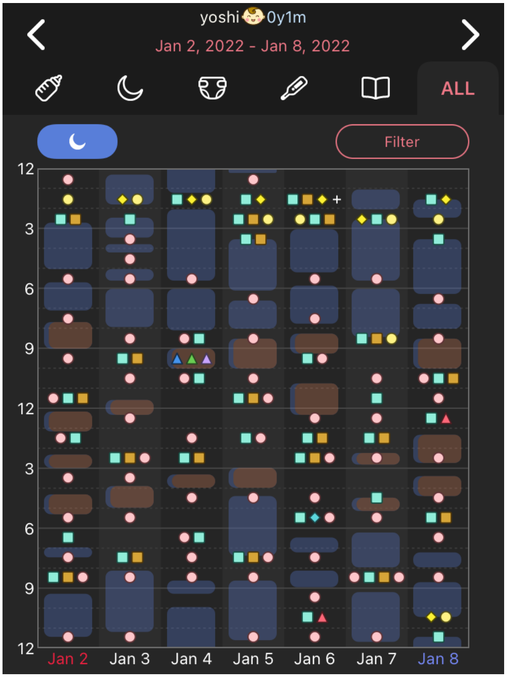

yoshi 0y0m
Nov 28, 2021 - Dec 4, 2021
ALL
Filter
12 3 6 9 12 3 6 9 12
Nov 28 Nov 29 Nov 30 Dec 1 Dec 2 Dec 3 Dec 4

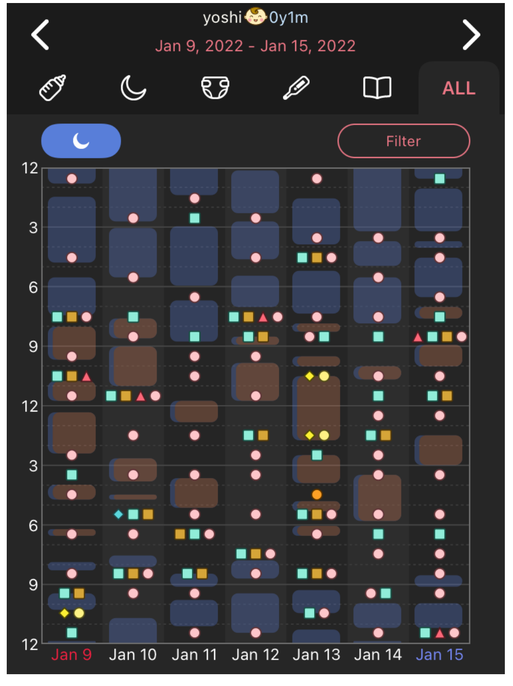

yoshi 0y0m
Dec 5, 2021 - Dec 11, 2021
ALL
Filter
12 3 6 9 12 3 6 9 12
Dec 5 Dec 6 Dec 7 Dec 8 Dec 9 Dec 10 Dec 11

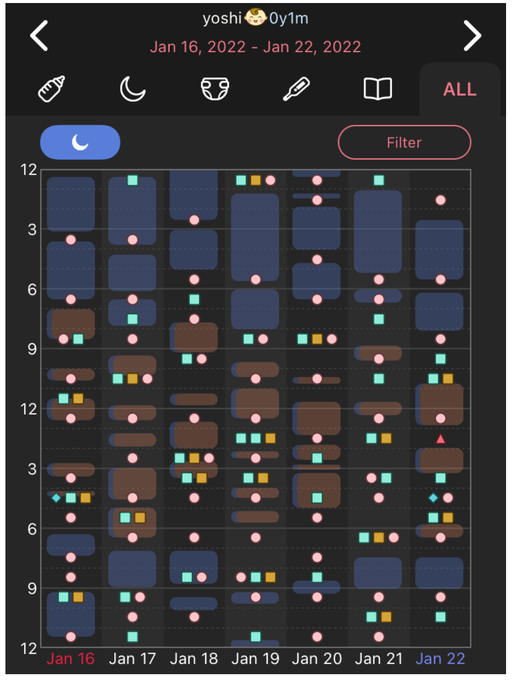

yoshi 0y0m
Dec 12, 2021 - Dec 18, 2021
ALL
Filter
12 3 6 9 12 3 6 9 12
Dec 12 Dec 13 Dec 14 Dec 15 Dec 16 Dec 17 Dec 18

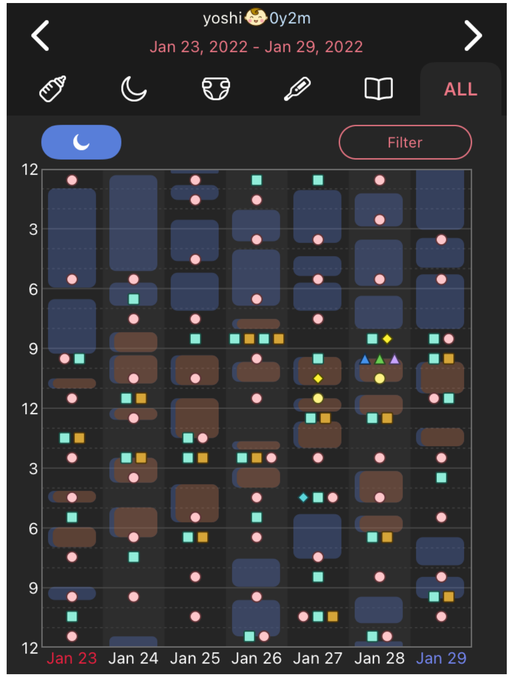

yoshi 0y0m
Dec 19, 2021 - Dec 25, 2021
ALL
Filter
12 3 6 9 12 3 6 9 12
Dec 19 Dec 20 Dec 21 Dec 22 Dec 23 Dec 24 Dec 25

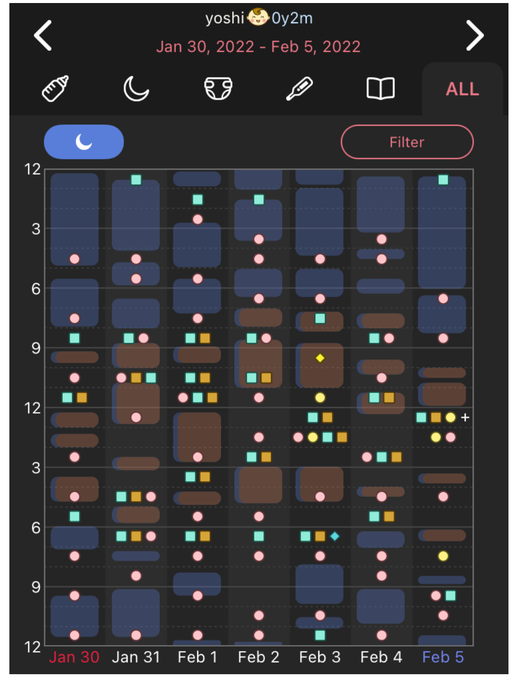

yoshi 0y1m
Dec 26, 2021 - Jan 1, 2022
ALL
Filter
12 3 6 9 12 3 6 9 12
Dec 26 Dec 27 Dec 28 Dec 29 Dec 30 Dec 31 Jan 1


yoshi 0y1m
Jan 2, 2022 - Jan 8, 2022
ALL
Filter
12 3 6 9 12 3 6 9 12
Jan 2 Jan 3 Jan 4 Jan 5 Jan 6 Jan 7 Jan 8


yoshi 0y1m
Jan 9, 2022 - Jan 15, 2022
ALL
Filter
12 3 6 9 12 3 6 9 12
Jan 9 Jan 10 Jan 11 Jan 12 Jan 13 Jan 14 Jan 15

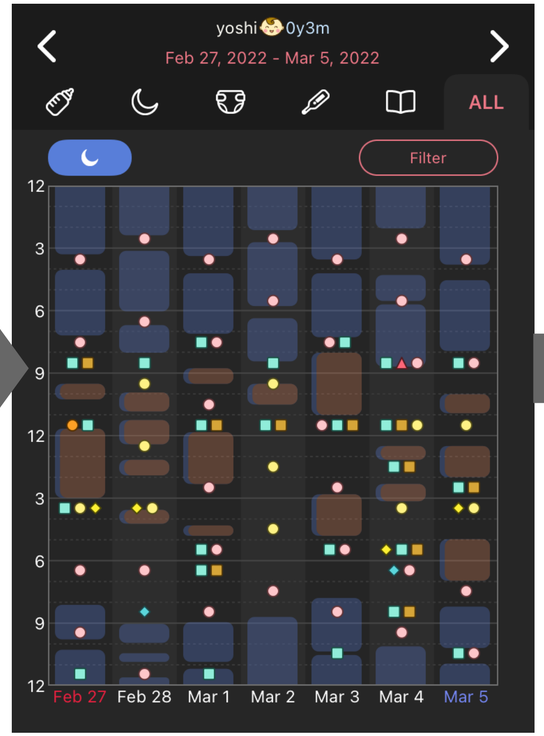

yoshi 0y1m
Jan 16, 2022 - Jan 22, 2022
ALL
Filter
12 3 6 9 12 3 6 9 12
Jan 16 Jan 17 Jan 18 Jan 19 Jan 20 Jan 21 Jan 22


yoshi 0y2m
Jan 23, 2022 - Jan 29, 2022
ALL
Filter
12 3 6 9 12 3 6 9 12
Jan 23 Jan 24 Jan 25 Jan 26 Jan 27 Jan 28 Jan 29


yoshi 0y2m
Jan 30, 2022 - Feb 5, 2022
ALL
Filter
12 3 6 9 12 3 6 9 12
Jan 30 Jan 31 Feb 1 Feb 2 Feb 3 Feb 4 Feb 5

Data Digitalization

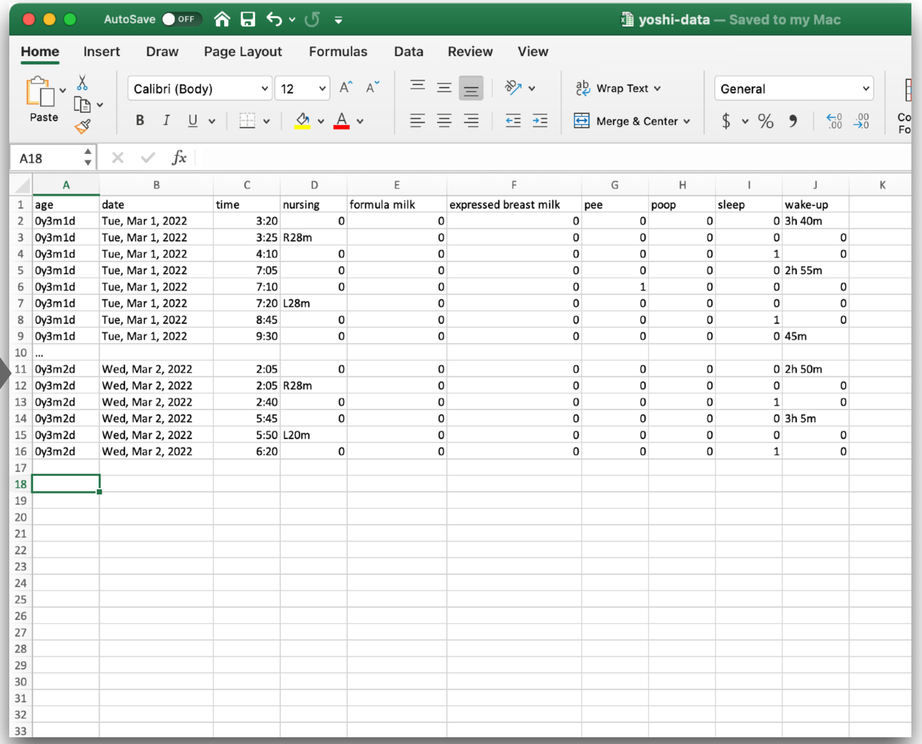
Real-world experience



Data collection

```
----------
Tue, Mar 1, 2022
yoshi (0y3m1d)

03:20 AM   Wake-up (3h 40m)
03:25 AM   Nursing R 28m
04:10 AM   Sleep
07:05 AM   Wake-up (2h 55m)
07:10 AM   Pee
07:20 AM   Nursing L 28m
08:45 AM   Sleep
09:30 AM   Wake-up (0h 45m)
10:30 AM   Nursing L 6m ← R 25m
11:15 AM   Pee
11:15 AM   Poop
11:50 AM   Sleep
02:20 PM   Wake-up (2h 30m)
02:25 PM   Nursing L 30m
04:20 PM   Sleep
04:50 PM   Wake-up (0h 30m)
05:30 PM   Pee
05:35 PM   Nursing R 30m
06:30 PM   Poop
06:30 PM   Pee
08:35 PM   Nursing L 30m
09:00 PM   Sleep
10:55 PM   Wake-up (1h 55m)
11:00 PM   Pee
11:15 PM   Sleep

Total nursing time  L 94m / R 83m
Formula Total: 0 times 0ml
Total sleep         12h40m
Peed                5times
Pooped              2times
```

Intermediate data

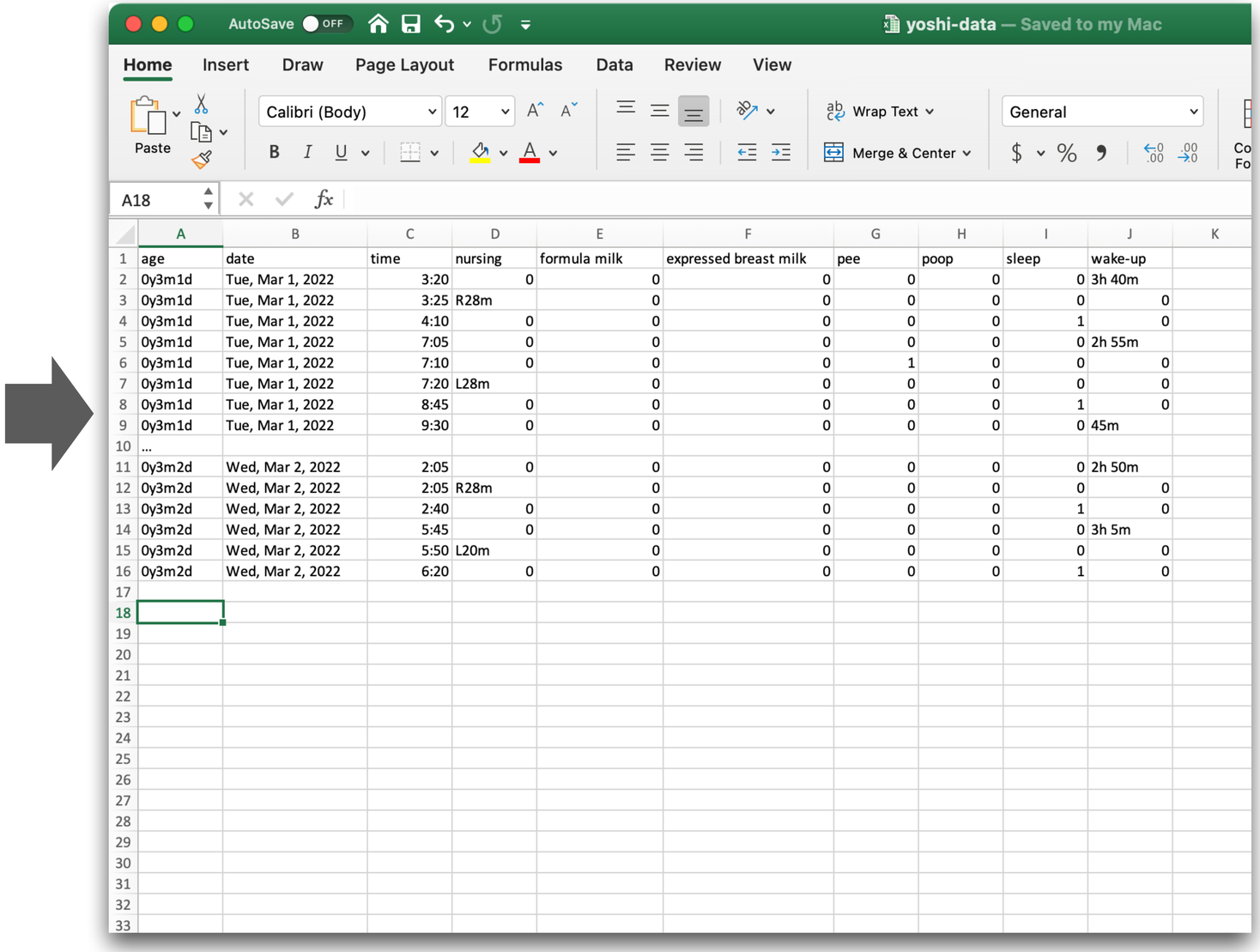


	A	B	C	D	E	F	G	H	I	J	K
1	age	date	time	nursing	formula milk	expressed breast milk	pee	poop	sleep	wake-up	
2	0y3m1d	Tue, Mar 1, 2022	3:20	0	0	0	0	0	0	3h 40m	
3	0y3m1d	Tue, Mar 1, 2022	3:25	R28m	0	0	0	0	0	0	
4	0y3m1d	Tue, Mar 1, 2022	4:10	0	0	0	0	0	1	0	
5	0y3m1d	Tue, Mar 1, 2022	7:05	0	0	0	0	0	0	2h 55m	
6	0y3m1d	Tue, Mar 1, 2022	7:10	0	0	0	1	0	0	0	
7	0y3m1d	Tue, Mar 1, 2022	7:20	L28m	0	0	0	0	0	0	
8	0y3m1d	Tue, Mar 1, 2022	8:45	0	0	0	0	0	1	0	
9	0y3m1d	Tue, Mar 1, 2022	9:30	0	0	0	0	0	0	45m	
10	...										
11	0y3m2d	Wed, Mar 2, 2022	2:05	0	0	0	0	0	0	2h 50m	
12	0y3m2d	Wed, Mar 2, 2022	2:05	R28m	0	0	0	0	0	0	
13	0y3m2d	Wed, Mar 2, 2022	2:40	0	0	0	0	0	1	0	
14	0y3m2d	Wed, Mar 2, 2022	5:45	0	0	0	0	0	0	3h 5m	
15	0y3m2d	Wed, Mar 2, 2022	5:50	L20m	0	0	0	0	0	0	
16	0y3m2d	Wed, Mar 2, 2022	6:20	0	0	0	0	0	1	0	
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											

Machine-readable Data

Machine-Readable Data

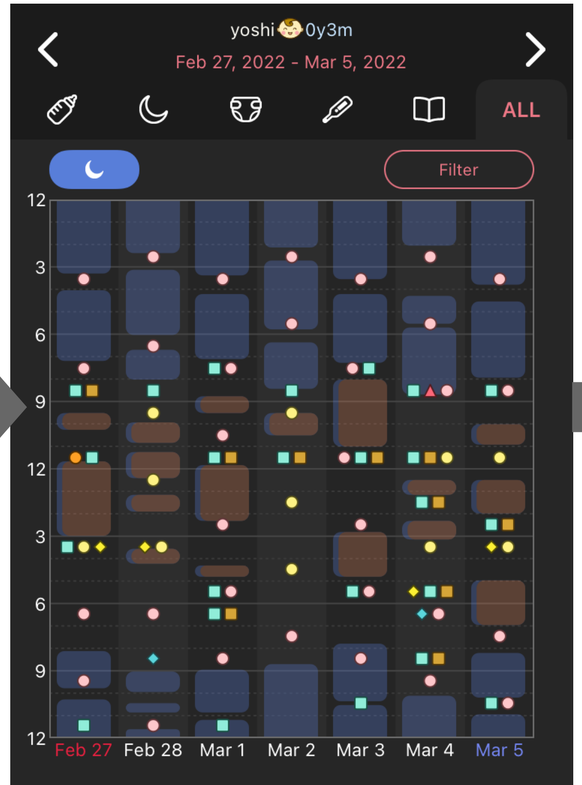
```
----------
Tue, Mar 1, 2022
yoshi (0y3m1d)

03:20 AM   Wake-up (3h 40m)
03:25 AM   Nursing R 28m
04:10 AM   Sleep
07:05 AM   Wake-up (2h 55m)
07:10 AM   Pee
07:20 AM   Nursing L 28m
08:45 AM   Sleep
09:30 AM   Wake-up (0h 45m)
10:30 AM   Nursing L 6m ← R 25m
11:15 AM   Pee
11:15 AM   Poop
11:50 AM   Sleep
02:20 PM   Wake-up (2h 30m)
02:25 PM   Nursing L 30m
04:20 PM   Sleep
04:50 PM   Wake-up (0h 30m)
05:30 PM   Pee
05:35 PM   Nursing R 30m
06:30 PM   Poop
06:30 PM   Pee
08:35 PM   Nursing L 30m
09:00 PM   Sleep
10:55 PM   Wake-up (1h 55m)
11:00 PM   Pee
11:15 PM   Sleep

Total nursing time  L 94m / R 83m
Formula Total: 0 times 0ml
Total sleep         12h40m
Peed                5times
Pooped              2times
```



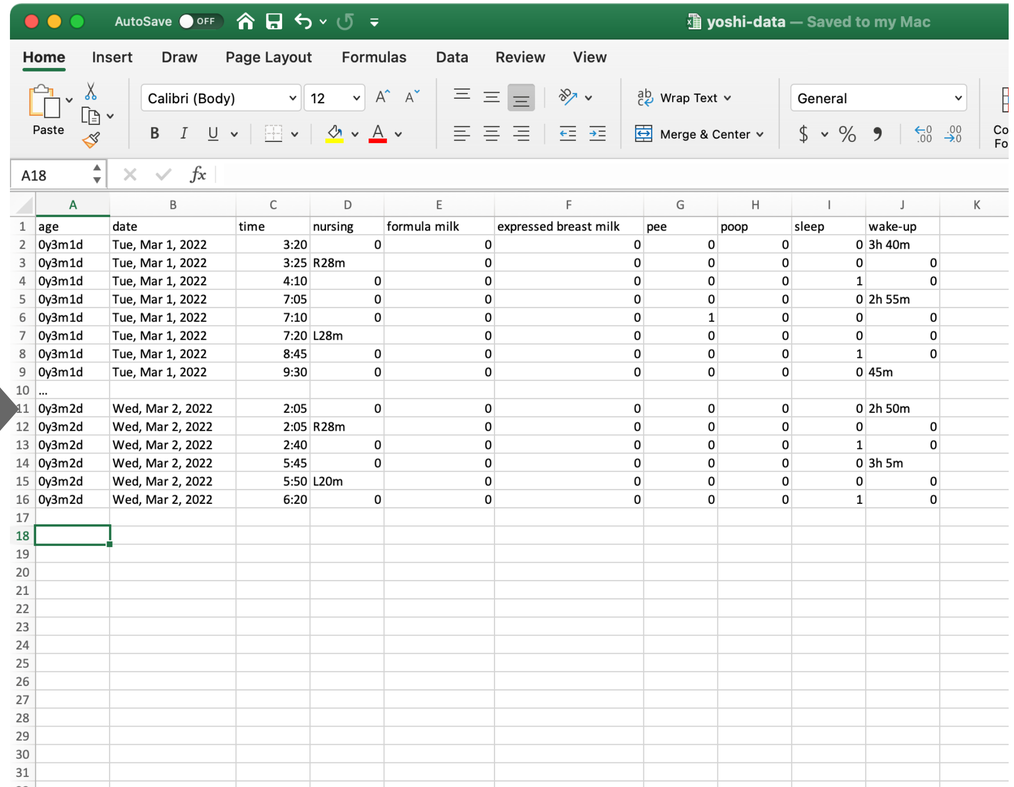
	A	B	C	D	E	F	G	H	I	J	K
1	age	date	time	nursing	formula milk	expressed breast milk	pee	poop	sleep	wake-up	
2	0y3m1d	Tue, Mar 1, 2022	3:20	0	0	0	0	0	0	3h 40m	
3	0y3m1d	Tue, Mar 1, 2022	3:25	R28m	0	0	0	0	0	0	
4	0y3m1d	Tue, Mar 1, 2022	4:10	0	0	0	0	0	1	0	
5	0y3m1d	Tue, Mar 1, 2022	7:05	0	0	0	0	0	0	2h 55m	
6	0y3m1d	Tue, Mar 1, 2022	7:10	0	0	0	1	0	0	0	
7	0y3m1d	Tue, Mar 1, 2022	7:20	L28m	0	0	0	0	0	0	
8	0y3m1d	Tue, Mar 1, 2022	8:45	0	0	0	0	0	1	0	
9	0y3m1d	Tue, Mar 1, 2022	9:30	0	0	0	0	0	0	45m	
10	...										
11	0y3m2d	Wed, Mar 2, 2022	2:05	0	0	0	0	0	0	2h 50m	
12	0y3m2d	Wed, Mar 2, 2022	2:05	R28m	0	0	0	0	0	0	
13	0y3m2d	Wed, Mar 2, 2022	2:40	0	0	0	0	0	1	0	
14	0y3m2d	Wed, Mar 2, 2022	5:45	0	0	0	0	0	0	3h 5m	
15	0y3m2d	Wed, Mar 2, 2022	5:50	L20m	0	0	0	0	0	0	
16	0y3m2d	Wed, Mar 2, 2022	6:20	0	0	0	0	0	1	0	
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											

“Data Digitalization” Exercise (15 mins)

- Step1: Create a spreadsheet and define your columns
- Step2: “Digitalize” your Dear Data




```
----------
Tue, Mar 1, 2022
yoshi (0y3m1d)

03:20 AM   Wake-up (3h 40m)
03:25 AM   Nursing R 28m
04:10 AM   Sleep
07:05 AM   Wake-up (2h 55m)
07:10 AM   Pee
07:20 AM   Nursing L 28m
08:45 AM   Sleep
09:30 AM   Wake-up (0h 45m)
10:30 AM   Nursing L 6m ← R 25m
11:15 AM   Pee
11:15 AM   Poop
11:50 AM   Sleep
02:20 PM   Wake-up (2h 30m)
02:25 PM   Nursing L 30m
04:20 PM   Sleep
04:50 PM   Wake-up (0h 30m)
05:30 PM   Pee
05:35 PM   Nursing R 30m
06:30 PM   Poop
06:30 PM   Pee
08:35 PM   Nursing L 30m
09:00 PM   Sleep
10:55 PM   Wake-up (1h 55m)
11:00 PM   Pee
11:15 PM   Sleep

Total nursing time  L 94m / R 83m
Formula Total: 0 times 0ml
Total sleep         12h40m
Peed                5times
Pooped              2times
```



	A	B	C	D	E	F	G	H	I	J	K
1	age	date	time	nursing	formula milk	expressed breast milk	pee	poop	sleep	wake-up	
2	0y3m1d	Tue, Mar 1, 2022	3:20	0	0	0	0	0	0	3h 40m	
3	0y3m1d	Tue, Mar 1, 2022	3:25	R28m	0	0	0	0	0	0	
4	0y3m1d	Tue, Mar 1, 2022	4:10	0	0	0	0	0	1	0	
5	0y3m1d	Tue, Mar 1, 2022	7:05	0	0	0	0	0	0	2h 55m	
6	0y3m1d	Tue, Mar 1, 2022	7:10	0	0	0	0	1	0	0	0
7	0y3m1d	Tue, Mar 1, 2022	7:20	L28m	0	0	0	0	0	0	0
8	0y3m1d	Tue, Mar 1, 2022	8:45	0	0	0	0	0	1	0	
9	0y3m1d	Tue, Mar 1, 2022	9:30	0	0	0	0	0	0	45m	
10	...										
11	0y3m2d	Wed, Mar 2, 2022	2:05	0	0	0	0	0	0	2h 50m	
12	0y3m2d	Wed, Mar 2, 2022	2:05	R28m	0	0	0	0	0	0	0
13	0y3m2d	Wed, Mar 2, 2022	2:40	0	0	0	0	0	1	0	
14	0y3m2d	Wed, Mar 2, 2022	5:45	0	0	0	0	0	0	3h 5m	
15	0y3m2d	Wed, Mar 2, 2022	5:50	L20m	0	0	0	0	0	0	0
16	0y3m2d	Wed, Mar 2, 2022	6:20	0	0	0	0	0	1	0	
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											

Break

Working with AI, Data and Pattern

prompt-to-code, data pattern exploration

Starboard

The shareable in-browser notebook

Support Markdown, Latex, HTML, CSS, Javascript, and Python

View source

EXAMPLE NOTEBOOKS

- Introduction to Starboard
- Data Visualization in JS
- Python: Pandas and Matplotlib
- Procedural Art

Starboard is an Open Startup.

Starboard is in beta, you can read more about Starboard here.

MARKDOWN

H1 # Introducing Starboard Notebook

Starboard brings cell-by-cell notebooks to the browser, no code is running on the backend here!
It's probably the quickest way to visualize some data with interactivity, do some prototyping, or build a rudimentary dashboard.

H4 **Some features**

- Mix Markdown, LaTeX, HTML, CSS, Javascript, and Python.
- The file format is a plaintext file, which plays nice with version control systems like git.
- Runs entirely in your browser, everything is static: no server, no setup and no build step.
- You can embed a fully functional notebook on your website.

Let's see it in action!

> *Tip: Press the ▶ Play button on the left to run a cell's code.*

</>

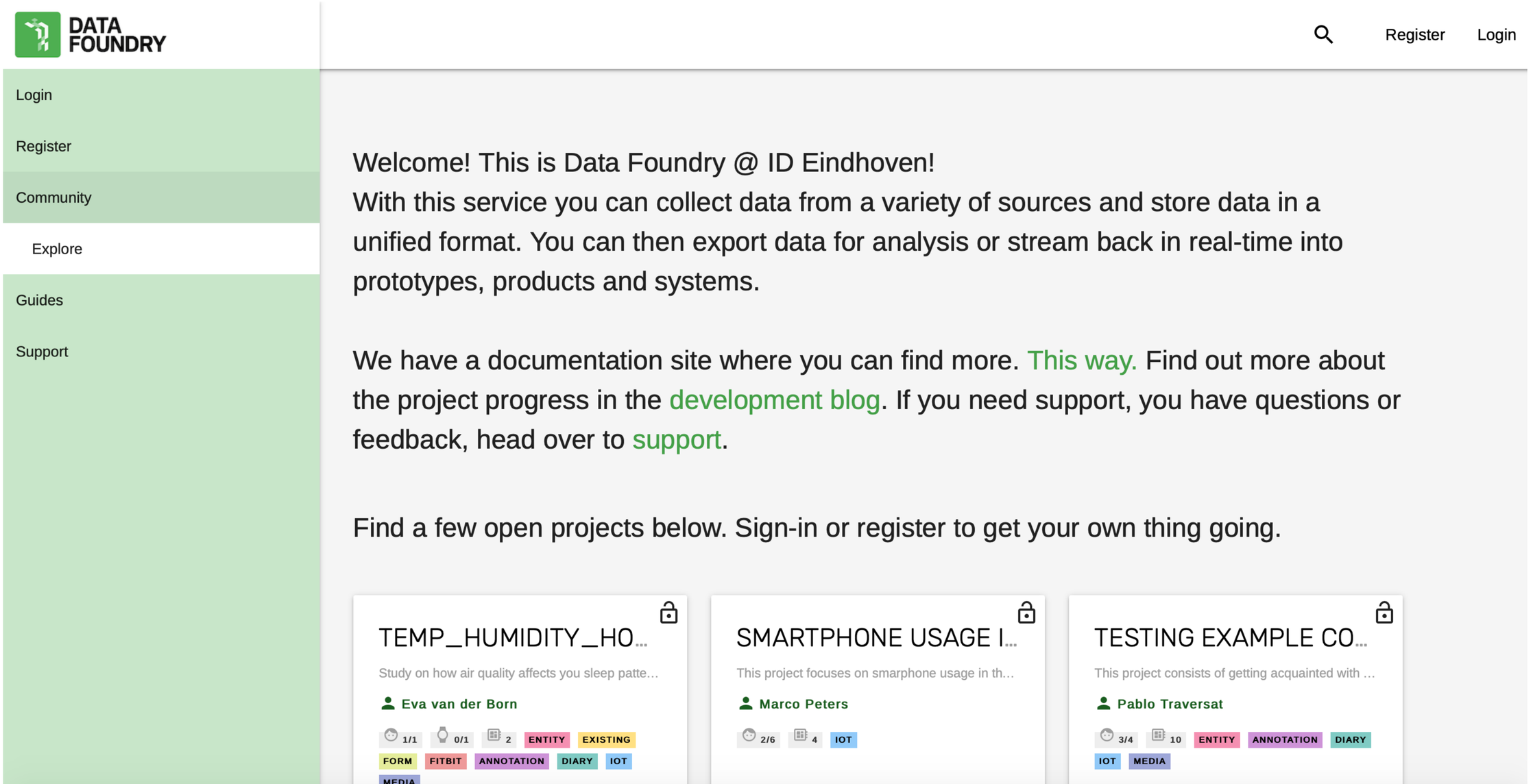
JAVASCRIPT

```
1  // You write vanilla Javascript
2  const greeting = "Hello world!";
3
4  // The last statement in a cell will be displayed if it is not undefined.
5  greeting
```

https://starboard.gg/

https://data.id.tue.nl

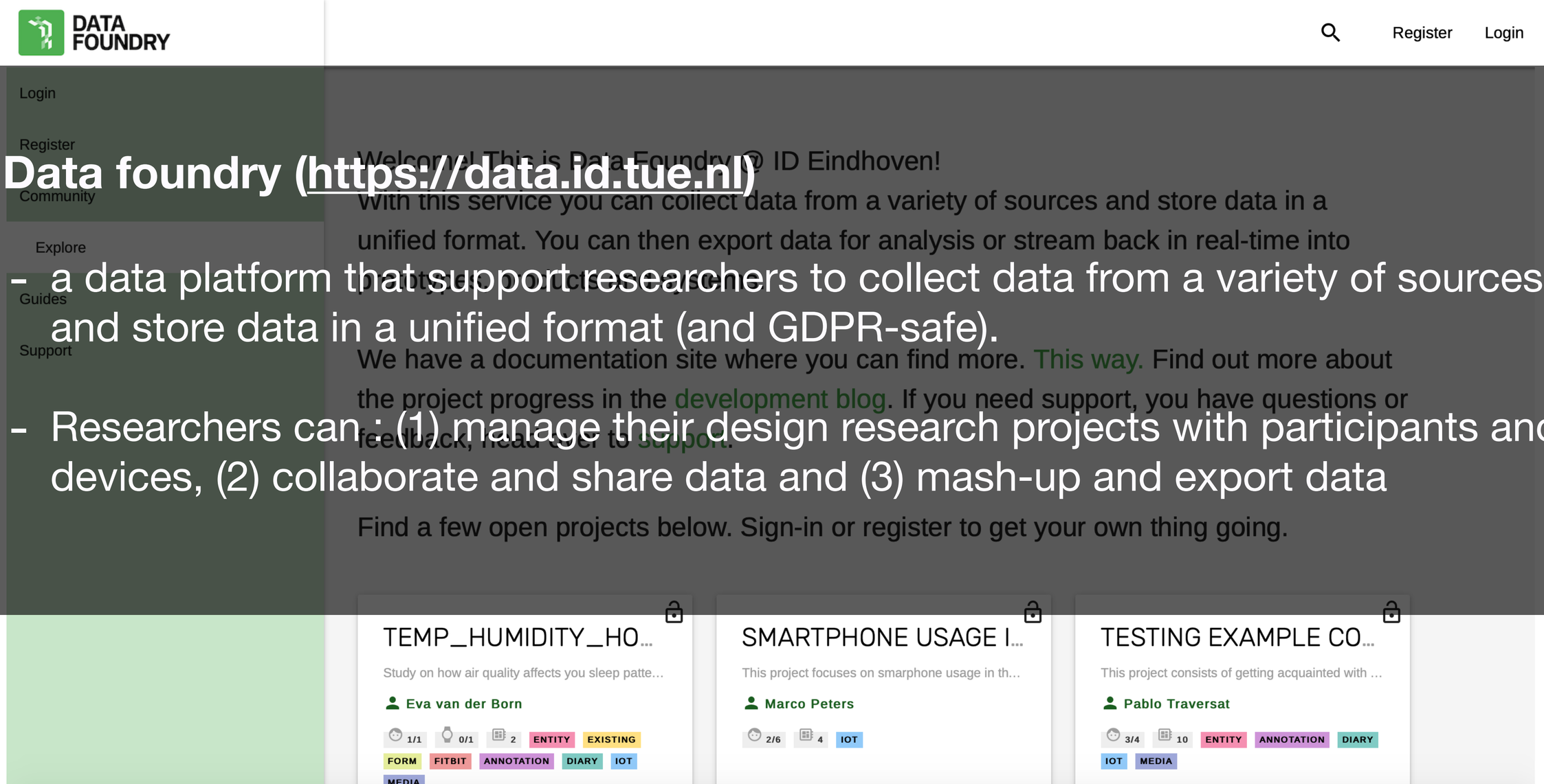
What is Data Foundry?



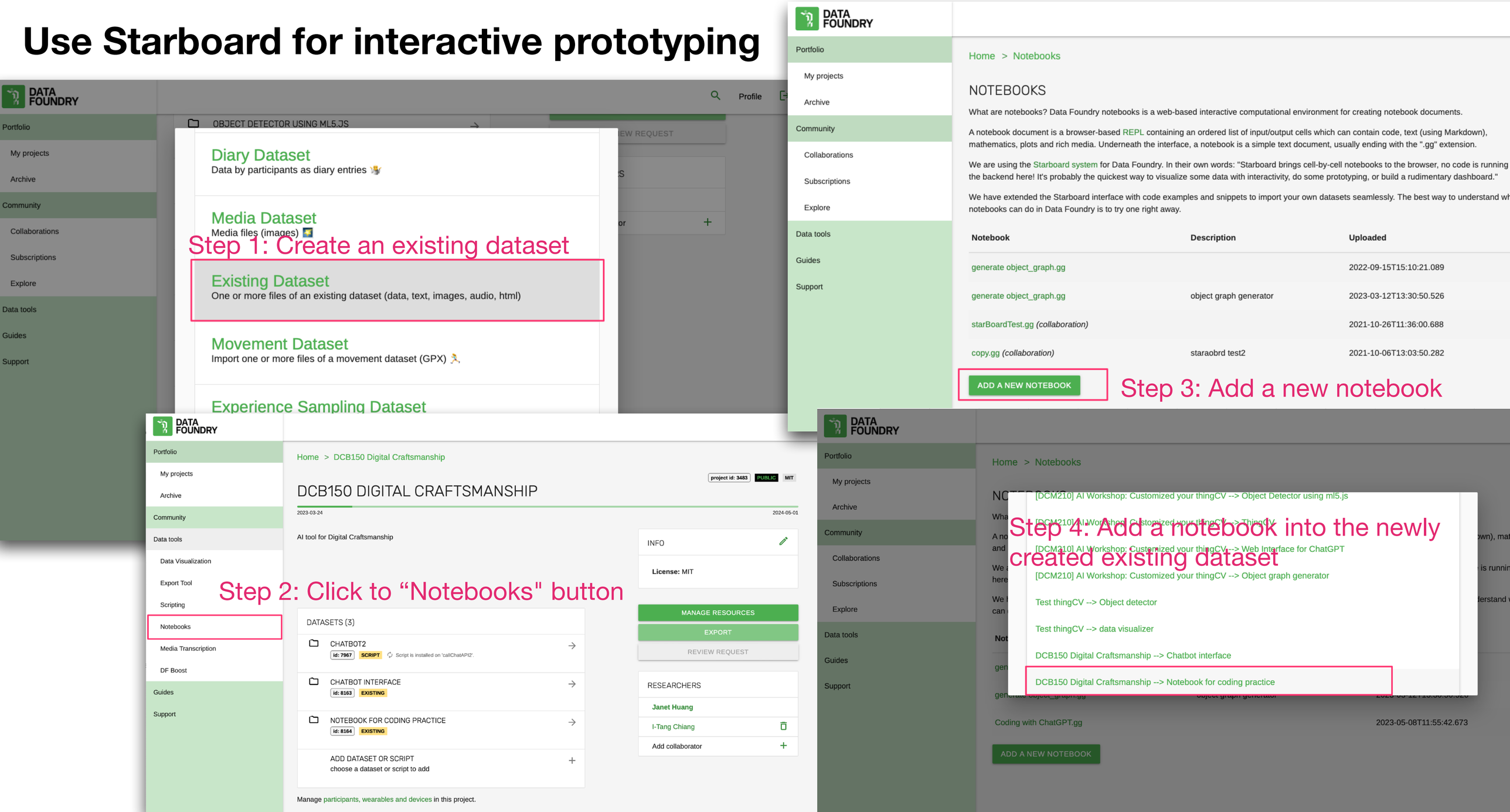
What is Data Foundry?

Data foundry (https://data.id.tue.nl)

- a data platform that support researchers to collect data from a variety of sources and store data in a unified format (and GDPR-safe).
- Researchers can : (1) manage their design research projects with participants and devices, (2) collaborate and share data and (3) mash-up and export data



Use Starboard for interactive prototyping



DATA FOUNDRY

Profile

- Portfolio
  - My projects
  - Archive
- Community
  - Collaborations
  - Subscriptions
  - Explore
- Data tools
- Guides
- Support

Home > DCB150 Digital Craftsmanship

DCB150 DIGITAL CRAFTSMANSHIP

project id: 3483 PUBLIC MIT

2023-03-24 — 2024-05-01

AI tool for Digital Craftsmanship

INFO

License: MIT

MANAGE RESOURCES

EXPORT

REVIEW REQUEST

DATASETS (3)

- CHATBOT2 — id: 7967 SCRIPT — Script is installed on 'callChatAPI2'.
- CHATBOT INTERFACE — id: 8163 EXISTING
- NOTEBOOK FOR CODING PRACTICE — id: 8164 EXISTING

ADD DATASET OR SCRIPT
choose a dataset or script to add

RESEARCHERS

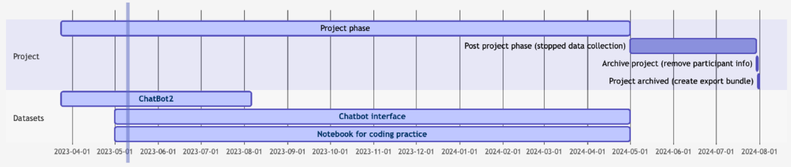
Janet Huang

I-Tang Chiang

Add collaborator

Manage participants, wearables and devices in this project.

TIMELINE | STUDY MANAGEMENT | RESEARCHERS AND PARTICIPANTS | CONNECTIONS | LAB NOTES



Home > DCB150 Digital Craftsmanship > Notebook for coding practice

NOTEBOOK FOR CODING PRACTICE

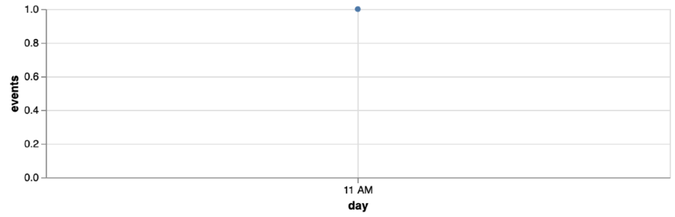
id: 8164 EXISTING PUBLIC MIT

2023-05-01 — 2024-05-01

Starboard notebook for coding practice

INFO

License: MIT



VIEW DATA

DOWNLOAD

UPLOAD FILE(S)

DATASET FILES

File name	Description	Uploaded	
Coding with ChatGPT.gg		May 08 at 11:55	download | edit | delete

CONFIGURATION

CSV/JSON TOKEN LINK | WEB-ACCESS | OOCSI STREAM

p5.js

Hello!

Search p5js.org

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else! p5.js is free and open-source because we believe software, and the tools to learn it, should be accessible to everyone.

Using the metaphor of a sketch, p5.js has a full set of drawing functionality. However, you're not limited to your drawing canvas. You can think of your whole browser page as your sketch, including HTML5 objects for text, input, video, webcam, and sound.

Join the p5.js Discord!

Start creating with the p5 Editor!

Community

We are a community of, and in solidarity with, people from every gender identity and expression, sexual orientation, race, ethnicity, language

https://p5js.org
https://editor.p5js.org

Pattern exploration with AI

<< back to dataset page

Notebook: Coding with ChatGPT.gg

Edit your notebook below, don't forget to save your changes. You can access your datasets with a bit of JavaScript or Python, depending on the notebook cell.

Save notebook

MARKDOWN

Session 0: Setup for using ChatGPT

Step 1: define request function for further POST call

In starboard, you can only use Pyodide to get data from the url. But, it is currently impossible to use the requests since sockets are not available in Pyodide. To make POST calls, you can use Web APIs in pyodide.

PYTHON

```python
# code is borrowed from https://docs.pyscript.net/latest/guides/http-requests.html
from pyodide.http import pyfetch, FetchResponse
from typing import Optional, Any

async def request(url: str, method: str = "GET", body: Optional[str] = None,
                  headers: Optional[dict[str, str]] = None, **fetch_kwargs: Any) -> FetchResponse:
    """
    Async request function. Pass in Method and make sure to await!
    Parameters:
        url: str = URL to make request to
        method: str = {"GET", "POST", "PUT", "DELETE"} from `JavaScript` global fetch())
        body: str = body as json string. Example, body=json.dumps(my_dict)
        headers: dict[str, str] = header as dict, will be converted to string...
            Example, headers=json.dumps({"Content-Type": "application/json"})
        fetch_kwargs: Any = any other keyword arguments to pass to `pyfetch` (will be passed to `fetch`)
    Return:
        response: pyodide.http.FetchResponse = use with .status or await.json(), etc.
    """
    kwargs = {"method": method, "mode": "cors"}  # CORS: https://en.wikipedia.org/wiki/Cross-origin_resource_shari
    if body and method not in ["GET", "HEAD"]:
        kwargs["body"] = body
    if headers:
        kwargs["headers"] = headers
    kwargs.update(fetch_kwargs)

    response = await pyfetch(url, **kwargs)
    return response
```

```
Python initialization complete
```

MARKDOWN

Step 2: Get your personal information from DF project for enabling OpenAI api

Copy the following information from your project and paste it to the following code.
(1) api_key
(2) project_id

PYTHON

```python
#api_key = '<api_key>'  #use your api_key
#project_id = "<project_id>" #use your project_id
api_key = 'df-YitjZDNYa3ZiRjg2TklHZTVkb2N5V0RJc3I0dEoyUk1DalBTUlBzTFN1az0='
project_id = '3483'
headers = {'Content-Type': 'application/json'}

# system_role = f"""
# You are a new media artist who uses code and technology to create patterns. You are good at programming language
# """

# setup a system role
```

Prompt-to-Code

Use starboard to explore pattern generation through ChatGPT

specify the role of ChatGPT and constraint the AI model's response

System prompt

> You are a new media artist who uses code and technology to create patterns. You are good at using p5.js to generate patterns.

user's input/question/instruction for a specific task

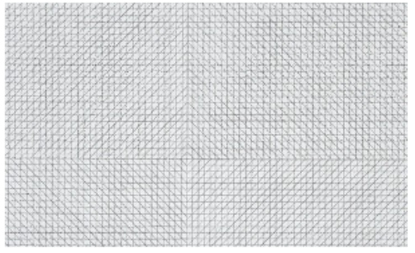
User prompt

specific instruction

> Write p5.js sketch that that draws the following:
>
> "Vertical lines, not straight, not touching, covering the wall evenly."

pattern description

possible by Agnes Gund.

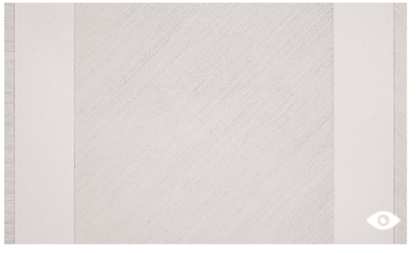


Wall Drawing 11

A wall divided horizontally and vertically into four equal parts. Within each part, three of the four kinds of lines are... Read More

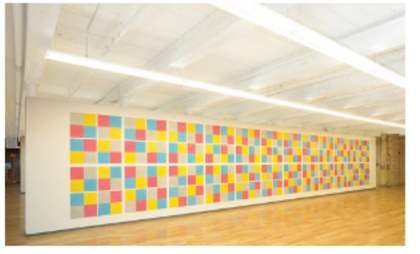
Wall Drawing 16

Bands of lines 12 inches (30 cm) wide, in three directions (vertical, horizontal, diagonal right) intersecting. Septembe... Read More



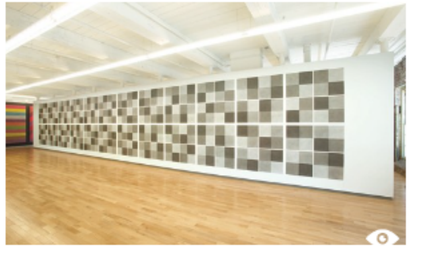
Wall Drawing 17

Four-part drawing with a different line direction in each part. September 1969 Black pencil Albright Knox Art Gallery, B... Read More



Wall Drawing 413

Drawing Series IV (A) with color ink washes. (24 drawings.) March 1984 Color ink wash LeWitt Collection, Chester, Connec... Read More

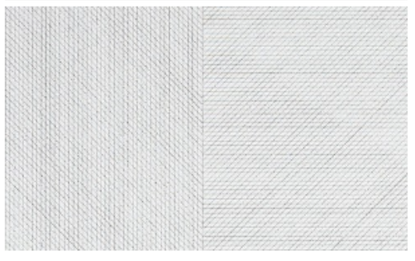


Wall Drawing 414

Drawing Series IV (A) with India ink washes. (24 Drawings.) March 1984 India ink wash LeWitt Collection, Chester, Connec... Read More

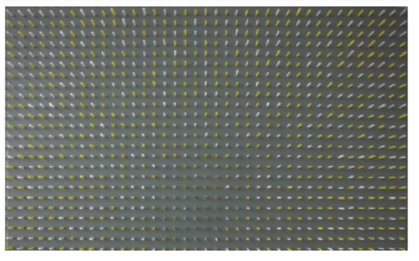
Wall Drawing 415D

Double Drawing. Right: Isometric Figure (Cube) with progressively darker graduations of gray on each of three planes; Le... Read More



Wall Drawing 19

A wall divided vertically into six equal parts, with two of the four kinds of line directions superimposed in each part.... Read More



Wall Drawing 38

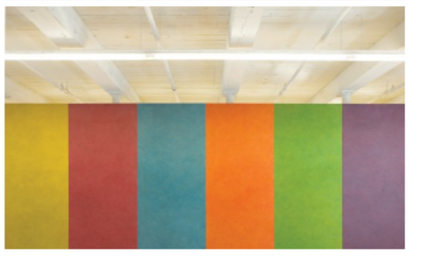
Tissue paper cut into 1½-inch (4 cm) squares and inserted into holes in the gray pegboard walls. All holes in the walls... Read More

Wall Drawing 46

Vertical lines, not straight, not touching, covering the wall evenly. May 1970 Black pencil LeWitt Collection, Chester, ... Read More

Wall Drawing 419

The wall is bordered and divided horizontally and vertically into four equal parts with a 6-inch (15 cm) black ink band.... Read More



Wall Drawing 422

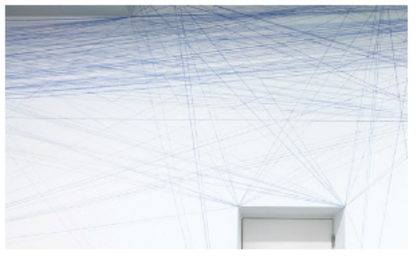
The room (or wall) is divided vertically into fifteen parts. All one-, two-, three-, and four-part combinations of four ... Read More

Wall Drawing 439

Asymmetrical pyramid with color ink washes superimposed. May 1985 Color ink wash Cuomo Collection First installation Sal... Read More

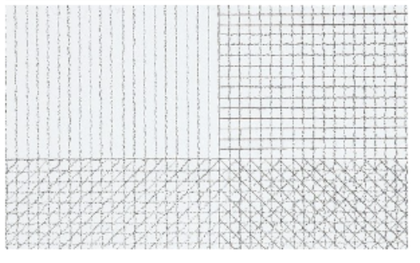
Wall Drawing 47

A wall divided into fifteen equal parts, each with a different line direction, and all combinations. June 1970 Black pen... Read More



Wall Drawing 51

All architectural points connected by straight lines. June 1970 Blue snap lines LeWitt Collection, Chester, Connecticut ... Read More



Wall Drawing 56

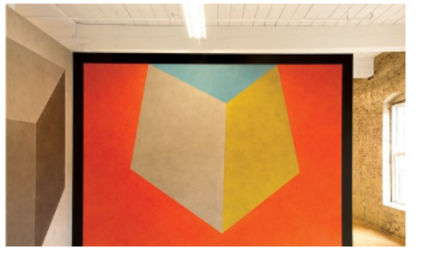
A square is divided horizontally and vertically into four equal parts, each with lines in four directions superimposed p... Read More

Wall Drawing 462

On four walls, one room, arcs 4 inches (10 cm) wide, from the midpoints of four sides, drawn with alternating bands of g... Read More

Wall Drawing 527

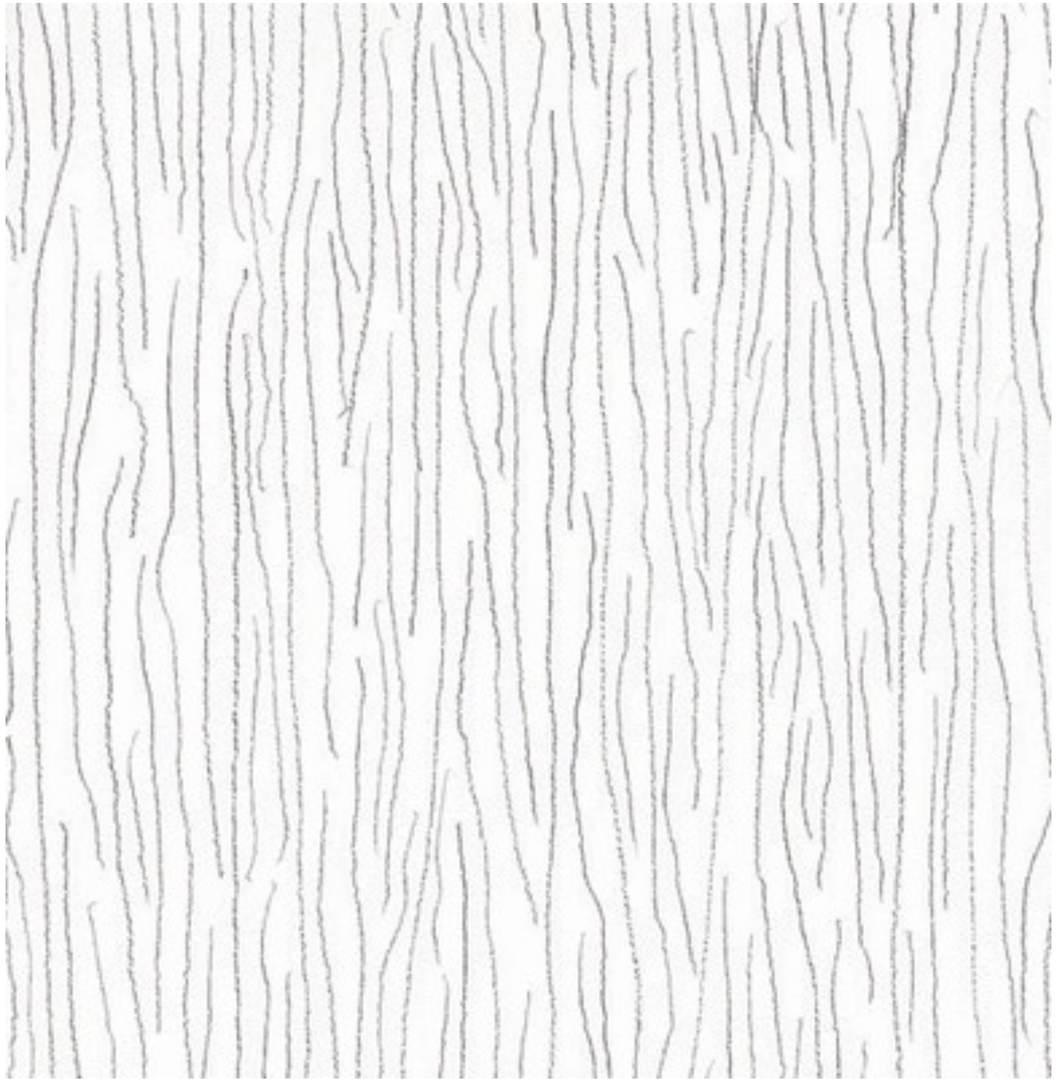
Two flat-topped pyramids with color ink washes superimposed. April 1987 Color ink wash Courtesy of the Estate of Sol LeW... Read More



Wall Drawing 552D

Tilted forms with color ink washes superimposed. December 1987 Color ink wash Courtesy of the Estate of Sol LeWitt First... Read More

Wall Drawing 46
Sol LeWitt



Sol LeWitt
"Wall Drawing 46," May 1970

Vertical lines, not straight, not touching, covering the wall evenly.

Black pencil

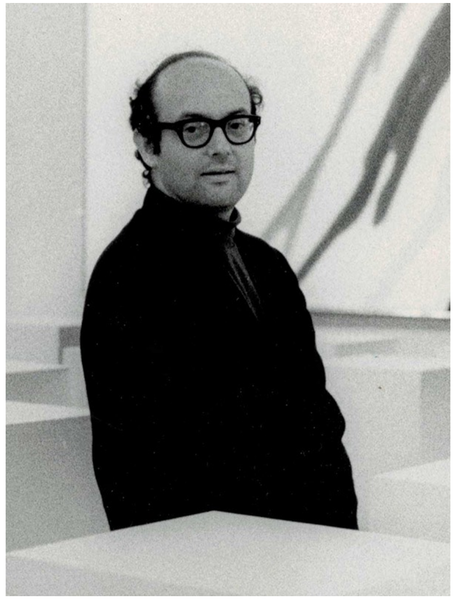


Image source 1: https://massmoca.org/event/walldrawing46/
Image source 2: https://www.ragoarts.com/auctions/2021/06/prints-multiples/544

Step 2: Call DF's OpenAI api to generating patterns through p5.js code

PYTHON

```python
import json

body = json.dumps(data)
new_post = await request(f"https://data.id.tue.nl/api/vendor/openai/"+project_id, body=body, method="POST", header
results = await new_post.json()
print(f"POST request=> status:{new_post.status}, json:{results}")
print("-------------------------------------------------")

#print out the important information: cost and output
print("cost:%d" % results["cost"])
print(results["content"])
```

```
POST request=> status:200, json:{'role': 'assistant', 'content': "Here's a p5.js sketch that draws
g, covering the wall evenly:\n\n```javascript\nfunction setup() {\n  createCanvas(windowWidth, wind
draw() {\n  stroke(255);\n  strokeWeight(2);\n  for (let x = 0; x < windowWidth; x += 10) {\n    fo
let randomOffset = random(-5, 5);\n      line(x + randomOffset, y, x + randomOffset, y + 10);\n
that fills the entire window, sets the background color to black, and then draws vertical lines usi
xels wide with a white stroke color. The x and y values are incremented by 10 in each loop to ensu
domOffset` variable adds a random value between -5 and 5 to the x coordinate of each line, making
'finishReason': 'stop', 'cost': 301}
cost:301
Here's a p5.js sketch that draws vertical lines, not straight and not touching, covering the wall e
""
```javascript
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(0);
}
""
function draw() {
  stroke(255);
  strokeWeight(2);
  for (let x = 0; x < windowWidth; x += 10) {
    for (let y = 0; y < windowHeight; y += 10) {
      let randomOffset = random(-5, 5);
      line(x + randomOffset, y, x + randomOffset, y + 10);
    }
  }
}
```
""
This code creates a canvas that fills the entire window, sets the background color to black, and th
ps. Each line is drawn 2 pixels wide with a white stroke color. The x and y values are incremented
are evenly spaced. The `randomOffset` variable adds a random value between -5 and 5 to the x coordi
raight and not touching.
```

copy and paste

p5* File Edit Sketch Help English Hello, janetyc!

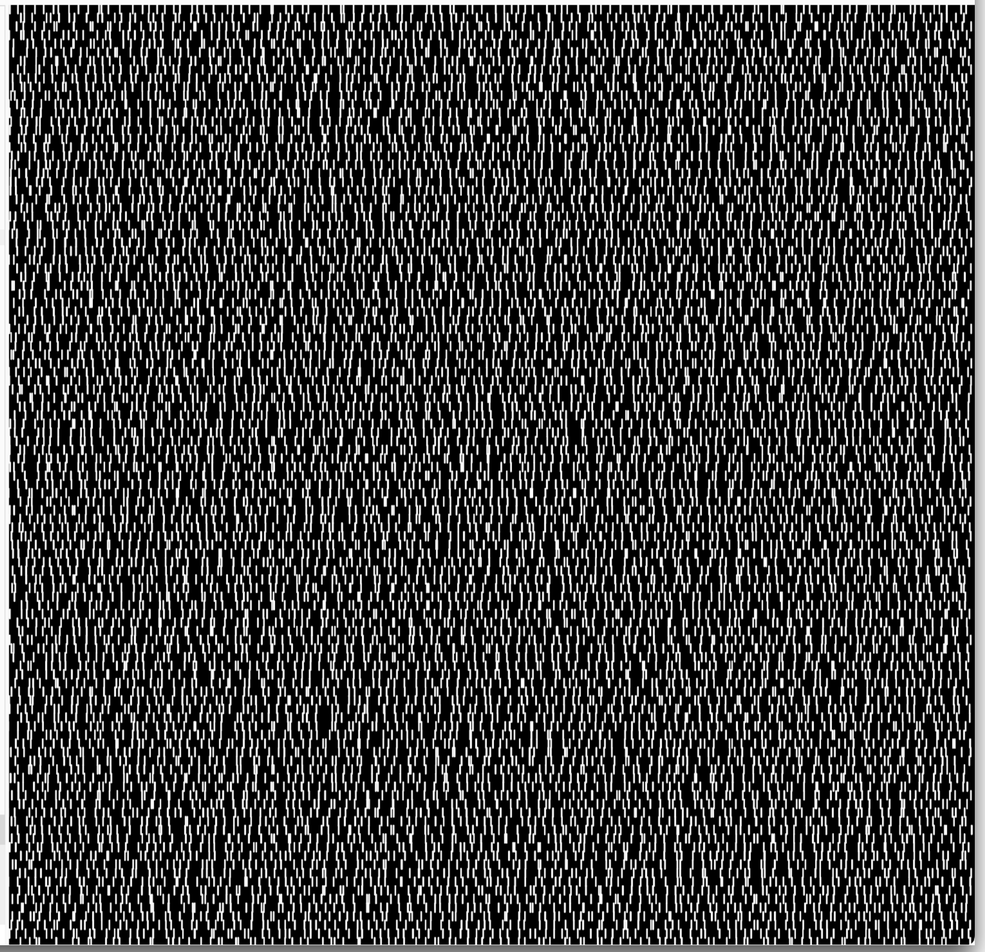
Auto-refresh Materialistic collision

sketch.js

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(0);
  noLoop();
}

function draw() {
  stroke(255);
  strokeWeight(2);
  for (let x = 0; x < windowWidth; x += 10) {
    for (let y = 0; y < windowHeight; y += 10) {
      let randomOffset = random(-5, 5);
      line(x + randomOffset, y, x + randomOffset, y + 10);
    }
  }
}
```

Preview



Console Clear

p5* File Edit Sketch Help English Hello, janetyc!

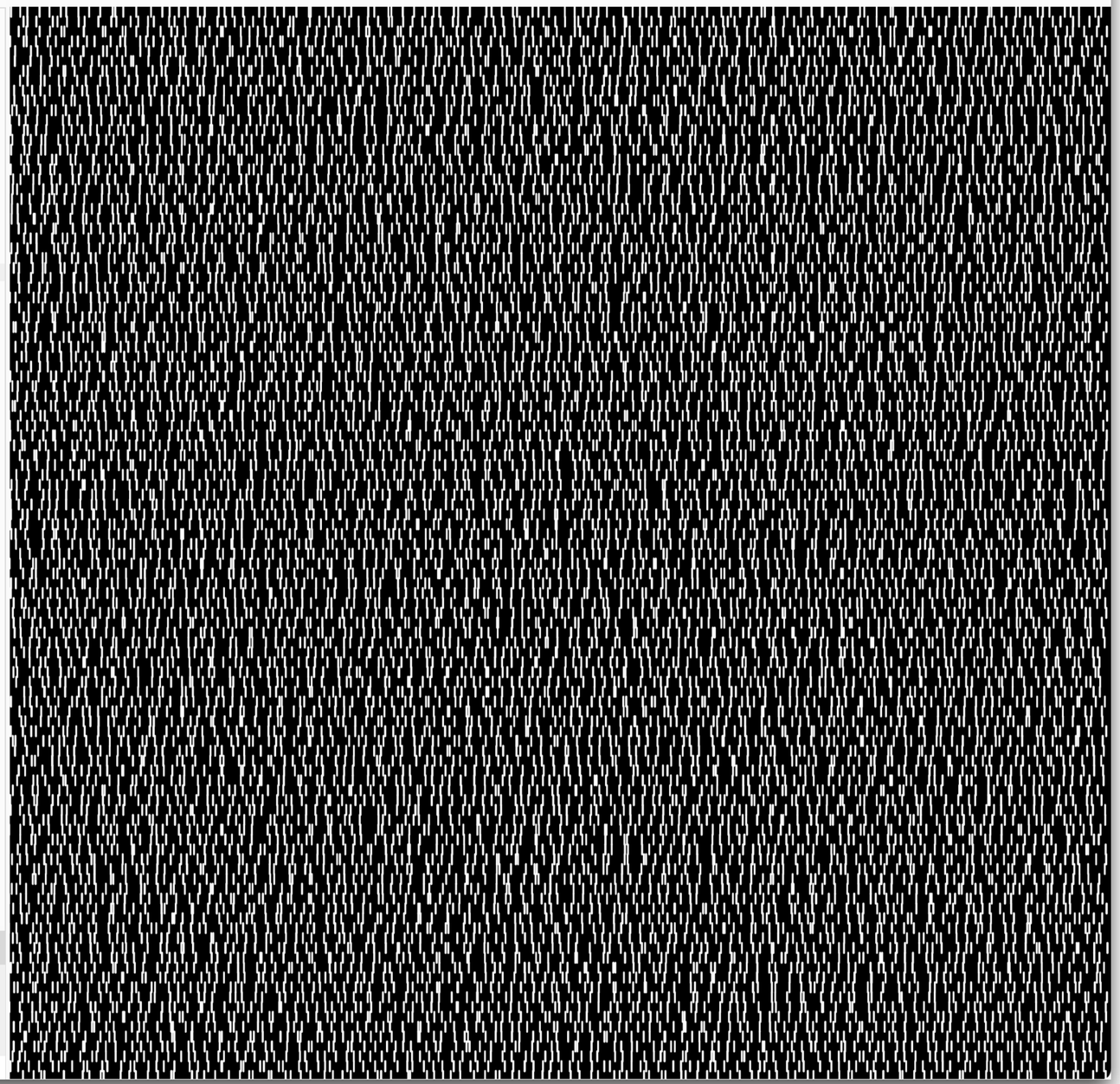
Auto-refresh Materialistic collision

sketch.js

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(0);
  noLoop();
}

function draw() {
  stroke(255);
  strokeWeight(2);
  for (let x = 0; x < windowWidth; x += 10) {
    for (let y = 0; y < windowHeight; y += 10) {
      let randomOffset = random(-5, 5);
      line(x + randomOffset, y, x + randomOffset, y + 10);
    }
  }
}
```

Preview



Console Clear

p5* File Edit Sketch Help English Hello, janetyc!

Auto-refresh Materialistic collision by janetyc

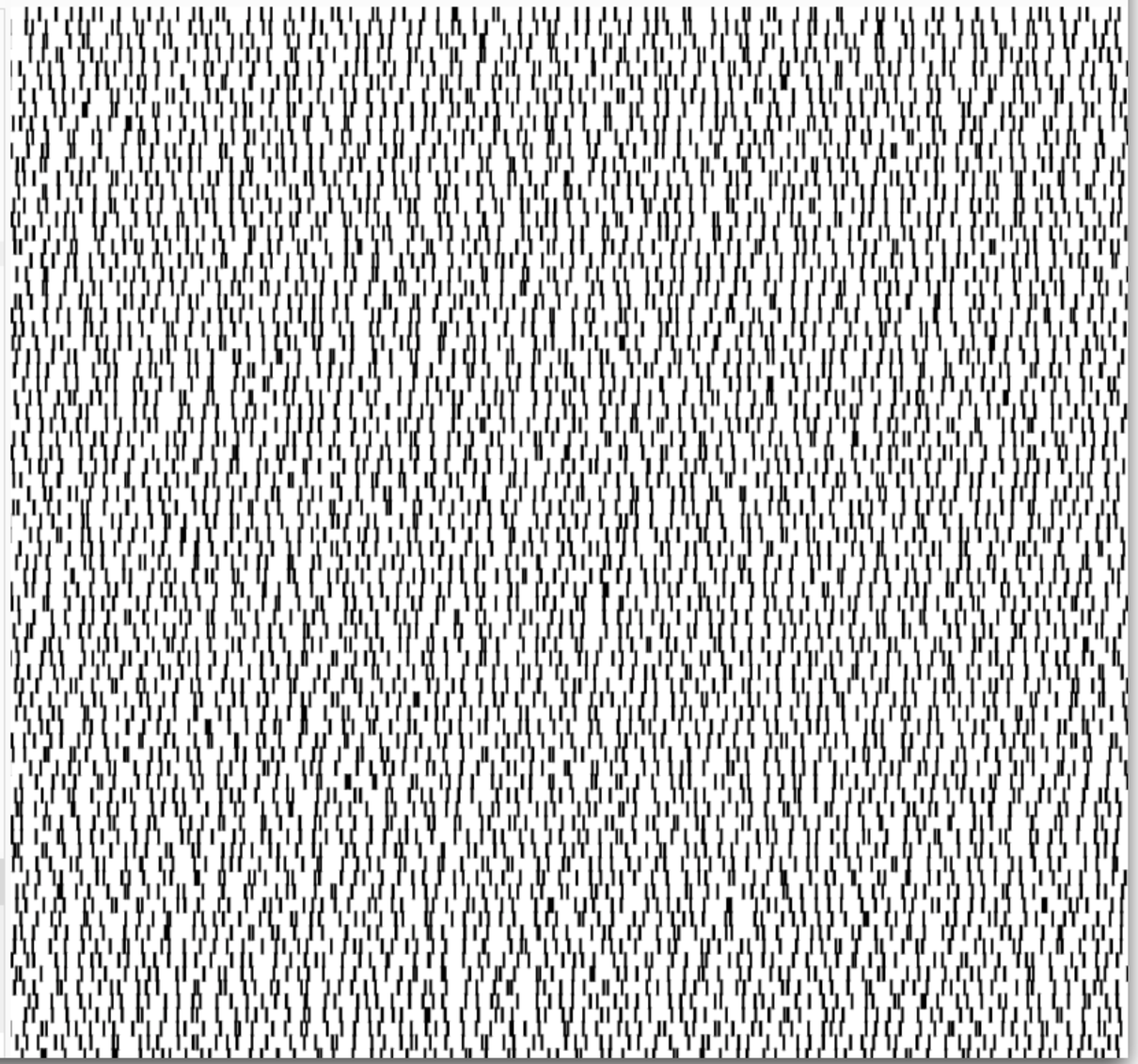
sketch.js Saved: 25 seconds ago

```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(255);
  noLoop();
}

function draw() {
  stroke(0);
  strokeWeight(2);
  for (let x = 0; x < windowWidth; x += 10) {
    for (let y = 0; y < windowHeight; y += 10) {
      let randomOffset = random(-5, 5);
      line(x + randomOffset, y, x + randomOffset, y + 10);
    }
  }
}
```

Console Clear

Preview



p5* File Edit Sketch Help English Hello, janetyc!

Auto-refresh Materialistic collision

sketch.js

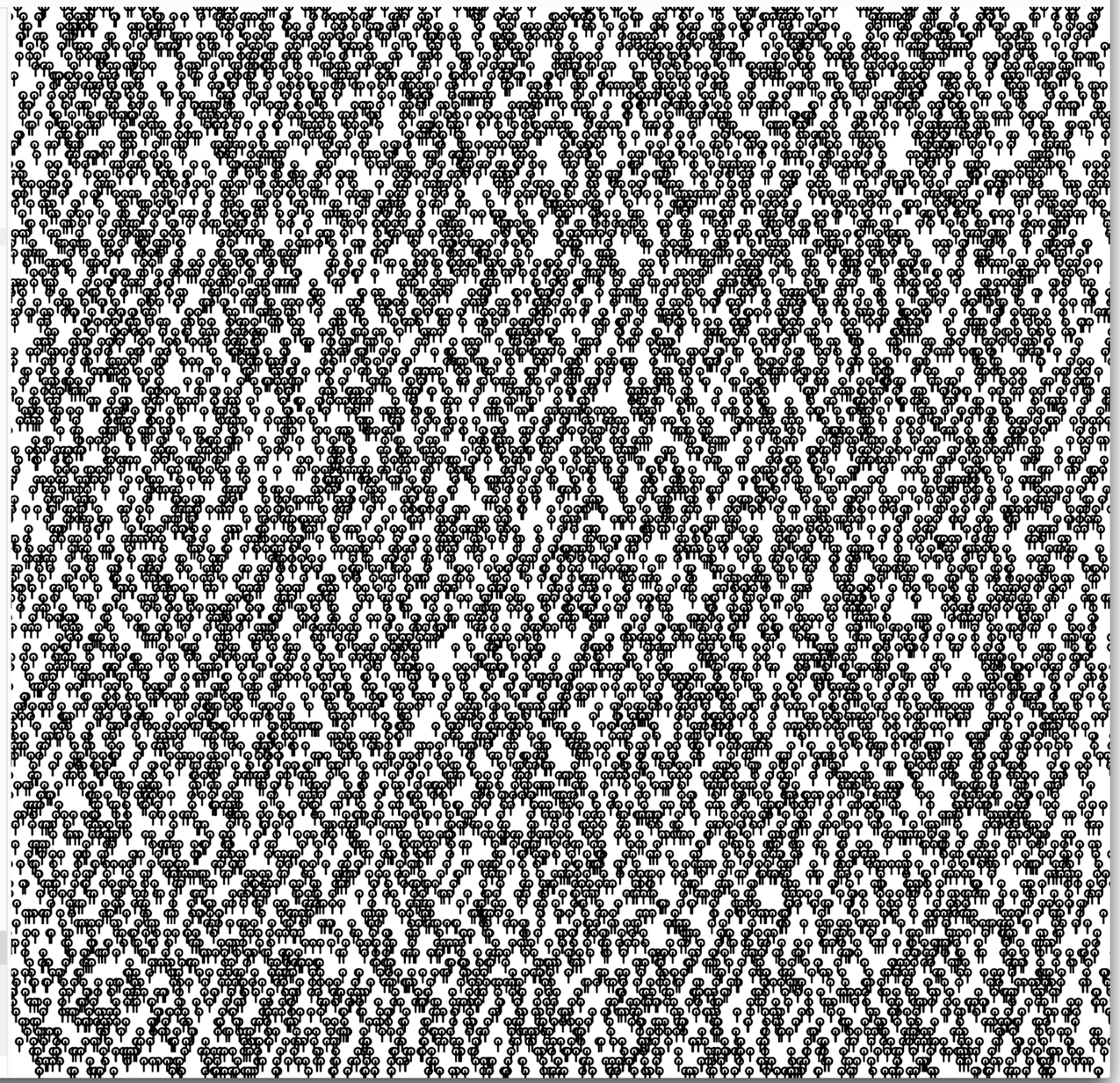
```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(255);
  noLoop();
}

function draw() {
  stroke(0);
  strokeWeight(2);
  for (let x = 0; x < windowWidth; x += 10) {
    for (let y = 0; y < windowHeight; y += 10) {
      let randomOffset = random(-50, 50);

      circle(x + randomOffset, y, 7);
      line(x + randomOffset, y, x + randomOffset, y + 10);
    }
  }
}
```

Console Clear

Preview



p5* File Edit Sketch Help English Hello, janetyc!

Auto-refresh Materialistic collision

sketch.js

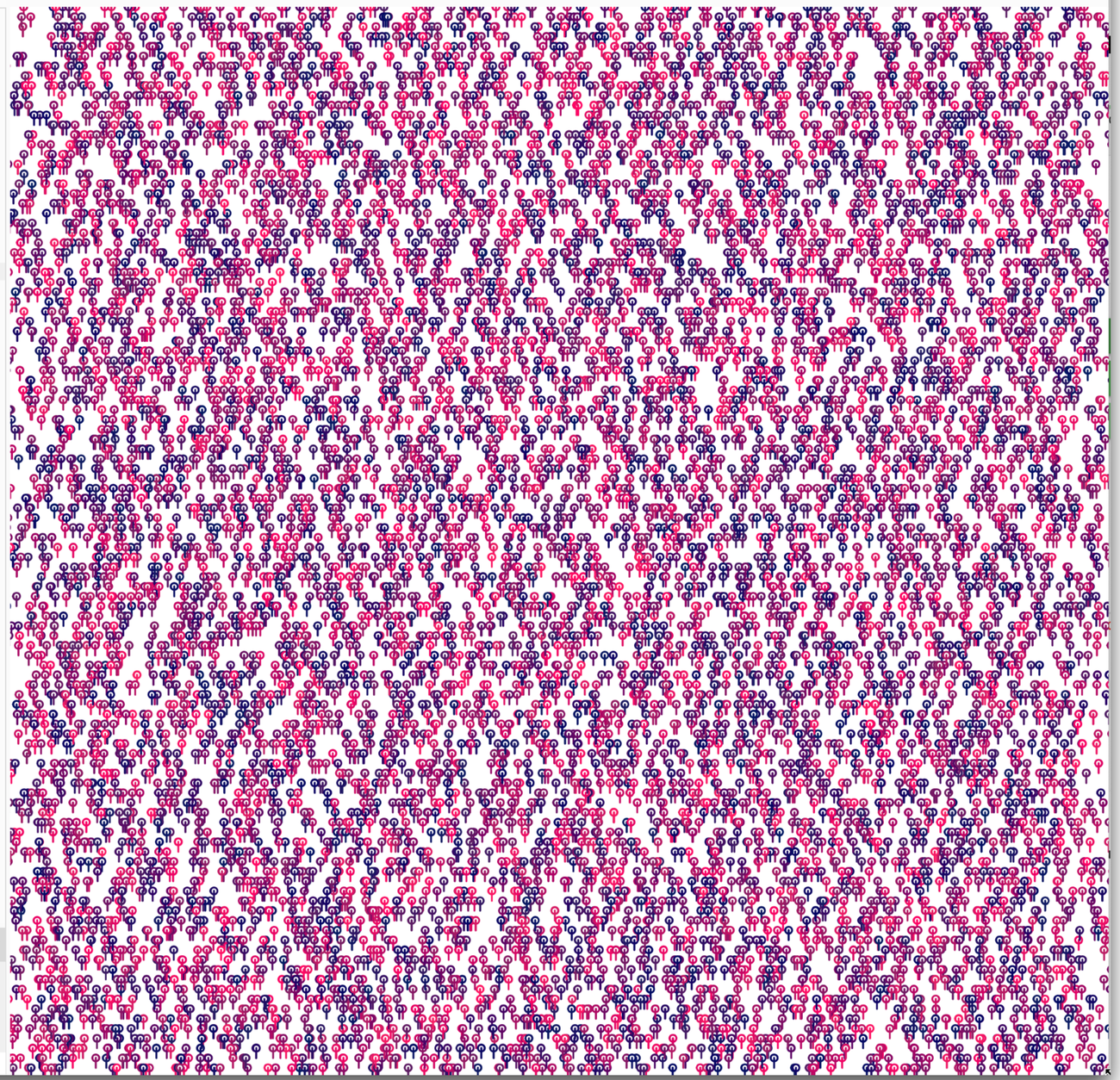
```
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(255);
  noLoop();
}

function draw() {

  strokeWeight(2);
  for (let x = 0; x < windowWidth; x += 10) {
    for (let y = 0; y < windowHeight; y += 10) {
      let randomOffset = random(-50, 50);
      stroke(random(255), random(30), 100);
      circle(x + randomOffset, y, 7);
      line(x + randomOffset, y, x + randomOffset, y + 10);
    }
  }
}
```

Console Clear

Preview

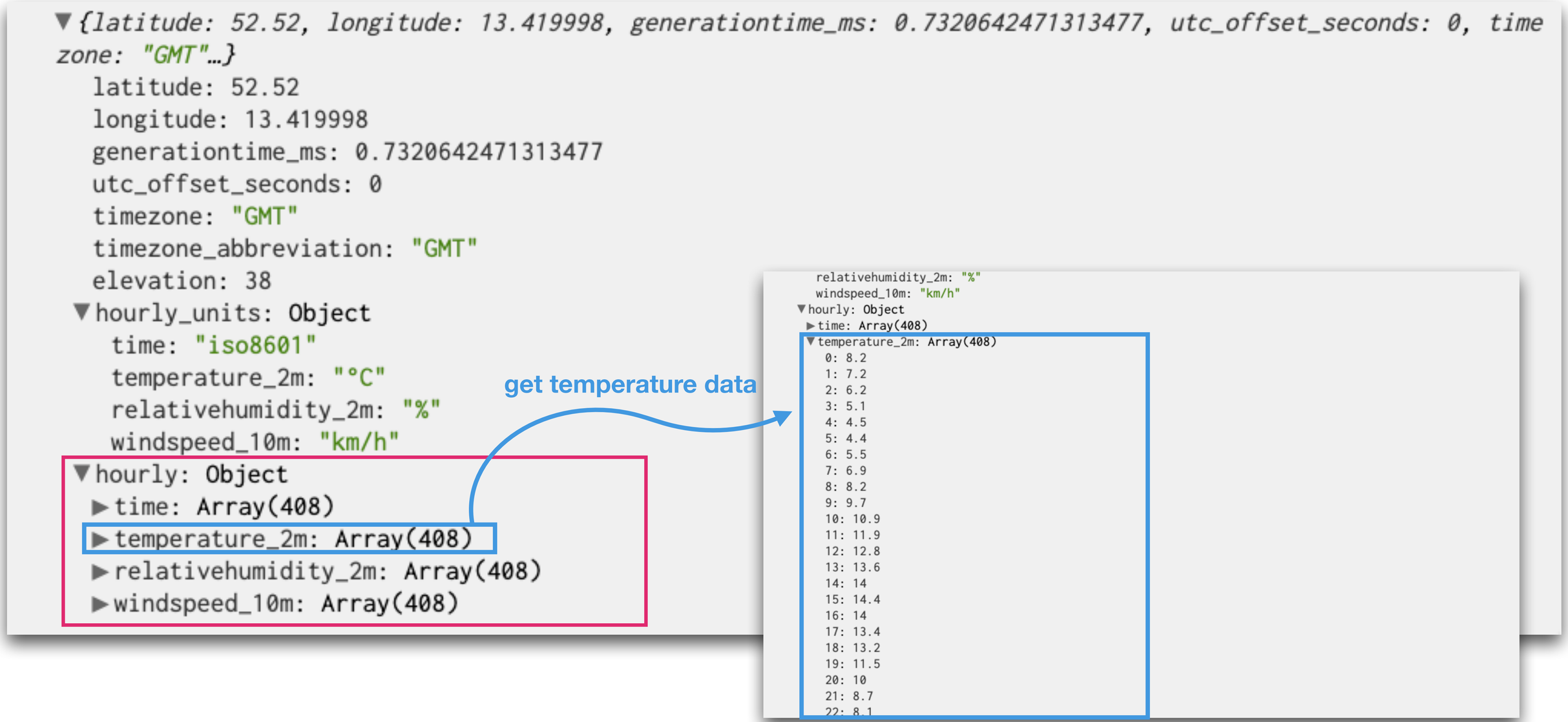


Add data into pattern

```
(base) ~ ▷ curl "https://archive-api.open-meteo.com/v1/era5?latitude=52.52&longitude=13.41&start_date=2021-01-01&end_date=2021-12-31&hourly=temperature_2m"
{"latitude":52.5,"longitude":13.400009,"generationtime_ms":0.39899349212646484,"utc_offset_seconds":0,"timezone":"GMT","timezone_abbreviation":"GMT","elevation":38.0,"hourly_units":{"time":"iso8601","temperature_2m":"°C"},"hourly":{"time":["2021-01-01T00:00","2021-01-01T01:00","2021-01-01T02:00","2021-01-01T03:00","2021-01-01T04:00","2021-01-01T05:00","2021-01-01T06:00","2021-01-01T07:00","2021-01-01T08:00","2021-01-01T09:00","2021-01-01T10:00","2021-01-01T11:00","2021-01-01T12:00","2021-01-01T13:00","2021-01-01T14:00","2021-01-01T15:00","2021-01-01T16:00","2021-01-01T17:00","2021-01-01T18:00","2021-01-01T19:00","2021-01-01T20:00","2021-01-01T21:00","2021-01-01T22:00","2021-01-01T23:00","2021-01-02T00:00","2021-01-02T01:00","2021-01-02T02:00","2021-01-02T03:00","2021-01-02T04:00","2021-01-02T05:00","2021-01-02T06:00","2021-01-02T07:00","2021-01-02T08:00","2021-01-02T09:00","2021-01-02T10:00","2021-01-02T11:00","2021-01-02T12:00","2021-01-02T13:00","2021-01-02T14:00","2021-01-02T15:00","2021-01-02T16:00","2021-01-02T17:00","2021-01-02T18:00","2021-01-02T19:00","2021-01-02T20:00","2021-01-02T21:00","2021-01-02T22:00","2021-01-02T23:00","2021-01-03T00:00","2021-01-03T01:00","2021-01-03T02:00","2021-01-03T03:00","2021-01-03T04:00","2021-01-03T05:00","2021-01-03T06:00","2021-01-03T07:00","2021-01-03T08:00","2021-01-03T09:00","2021-01-03T10:00","2021-01-03T11:00","2021-01-03T12:00","2021-01-03T13:00","2021-01-03T14:00","2021-01-03T15:00","2021-01-03T16:00","2021-01-03T17:00","2021-01-03T18:00","2021-01-03T19:00","2021-01-03T20:00","2021-01-03T21:00","2021-01-03T22:00","2021-01-03T23:00","2021-01-04T00:00","2021-01-04T01:00","2021-01-04T02:00","2021-01-04T03:00","2021-01-04T04:00","2021-01-04T05:00","2021-01-04T06:00","2021-01-04T07:00","2021-01-04T08:00","2021-01-04T09:00","2021-01-04T10:00","2021-01-04T11:00","2021-01-04T12:00","2021-01-04T13:00","2021-01-04T14:00","2021-01-04T15:00","2021-01-04T16:00","2021-01-04T17:00","2021-01-04T18:00","2021-01-04T19:00","2021-01-04T20:00","2021-01-04T21:00","2021-01-04T22:00","2021-01-04T23:00","2021-01-05T00:00","2021-01-05T01:00","2021-01-05T02:00","2021-01-05T03:00","2021-01-05T04:00","2021-01-05T05:00","2021-01-05T06:00","2021-01-05T07:00","2021-01-05T08:00","2021-01-05T09:00","2021-01-05T10:00","2021-01-05T11:00","2021-01-05T12:00","2021-01-05T13:00","2021-01-05T14:00","2021-01-05T15:00","2021-01-05T16:00","2021-01-05T17:00","2021-01-05T18:00","2021-01-05T19:00","2021-01-05T20:00","2021-01-05T21:00","2021-01-05T22:00","2021-01-05T23:00","2021-01-06T00:00","2021-01-06T01:00","2021-01-06T02:00","2021-01-06T03:00","2021-01-06T04:00","2021-01-06T05:00","2021-01-06T06:00","2021-01-06T07:00","2021-01-06T08:00","2021-01-06T09:00","2021-01-06T10:00","2021-01-06T11:00","2021-01-06T12:00","2021-01-06T13:00","2021-01-06T14:00","2021-01-06T15:00","2021-01-06T16:00","2021-01-06T17:00","2021-01-06T18:00","2021-01-06T19:00","2021-01-06T20:00","2021-01-06T21:00","2021-01-06T22:00","2021-01-06T23:00","2021-01-07T00:00","2021-01-07T01:00","2021-01-07T02:00","2021-01-07T03:00","2021-01-07T04:00","2021-01-07T05:00","2021-01-07T06:00","2021-01-07T07:00","2021-01-07T08:00","2021-01-07T09:00","2021-01-07T10:00","2021-01-07T11:00","2021-01-07T12:00","2021-01-07T13:00","2021-01-07T14:00","2021-01-07T15:00","2021-01-07T16:00","2021-01-07T17:00","2021-01-07T18:00","2021-01-07T19:00","2021-01-07T20:00","2021-01-07T21:00","2021-01-07T22:00","2021-01-07T23:00","2021-01-08T00:00","2021-01-08T01:00","2021-01-08T02:00","2021-01-08T03:00","2021-01-08T04:00","2021-01-08T05:00","2021-01-08T06:00","2021-01-08T07:00","2021-01-08T08:00","2021-01-08T09:00","2021-01-08T10:00","2021-01-08T11:00","2021-01-08T12:00","2021-01-08T13:00","2021-01-08T14:00","2021-01-08T15:00","2021-01-08T16:00","2021-01-08T17:00","2021-01-08T18:00","2021-01-08T19:00","2021-01-08T20:00","2021-01-08T21:00","2021-01-08T22:00","2021-01-08T23:00","2021-01-09T00:00","2021-01-09T01:00","2021-01-09T02:00","2021-01-09T03:00","2021-01-09T04:00","2021-01-09T05:00","2021-01-09T06:00","2021-01-09T07:00","2021-01-09T08:00","2021-01-09T09:00","2021-01-09T10:00","2021-01-09T11:00","2021-01-09T12:00","2021-01-09T13:00","2021-01-09T14:00","2021-01-09T15:00","2021-01-09T16:00","2021-01-09T17:00","2021-01-09T18:00","2021-01-09T19:00","2021-01-09T20:00","2021-01-09T21:00","2021-01-09T22:00","2021-01-09T23:00","2021-01-10T00:00","2021-01-10T01:00","2021-01-10T02:00","2021-01-10T03:00","2021-01-10T04:00","2021-01-10T05:00","2021-01-10T06:00","2021-01-10T07:00","2021-01-10T08:00","2021-01-10T09:00","2021-01-10T10:00","2021-01-10T11:00","2021-01-10T12:00","2021-01-10T13:00","2021-01-10T14:00","2021-01-10T15:00","2021-01-10T16:00","2021-01-10T17:00","2021-01-10T18:00","2021-01-10T19:00","2021-01-10T20:00","2021-01-10T21:00","2021-01-10T22:00","2021-0
```

https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&past_days=10&hourly=temperature_2m,relativehumidity_2m,windspeed_10m

json data structure



p5* File Edit Sketch Help English Hello, janetyc!

Auto-refresh Materialistic collision by janetyc

Sketch Files

- era5.json
- forecast.json
- index.html
- sketch.js
- style.css

sketch.js• Saved: 25 seconds ago

```
function setup() {
   createCanvas(windowWidth, windowHeight);
   background(255);
   stroke(0);
   strokeWeight(2)

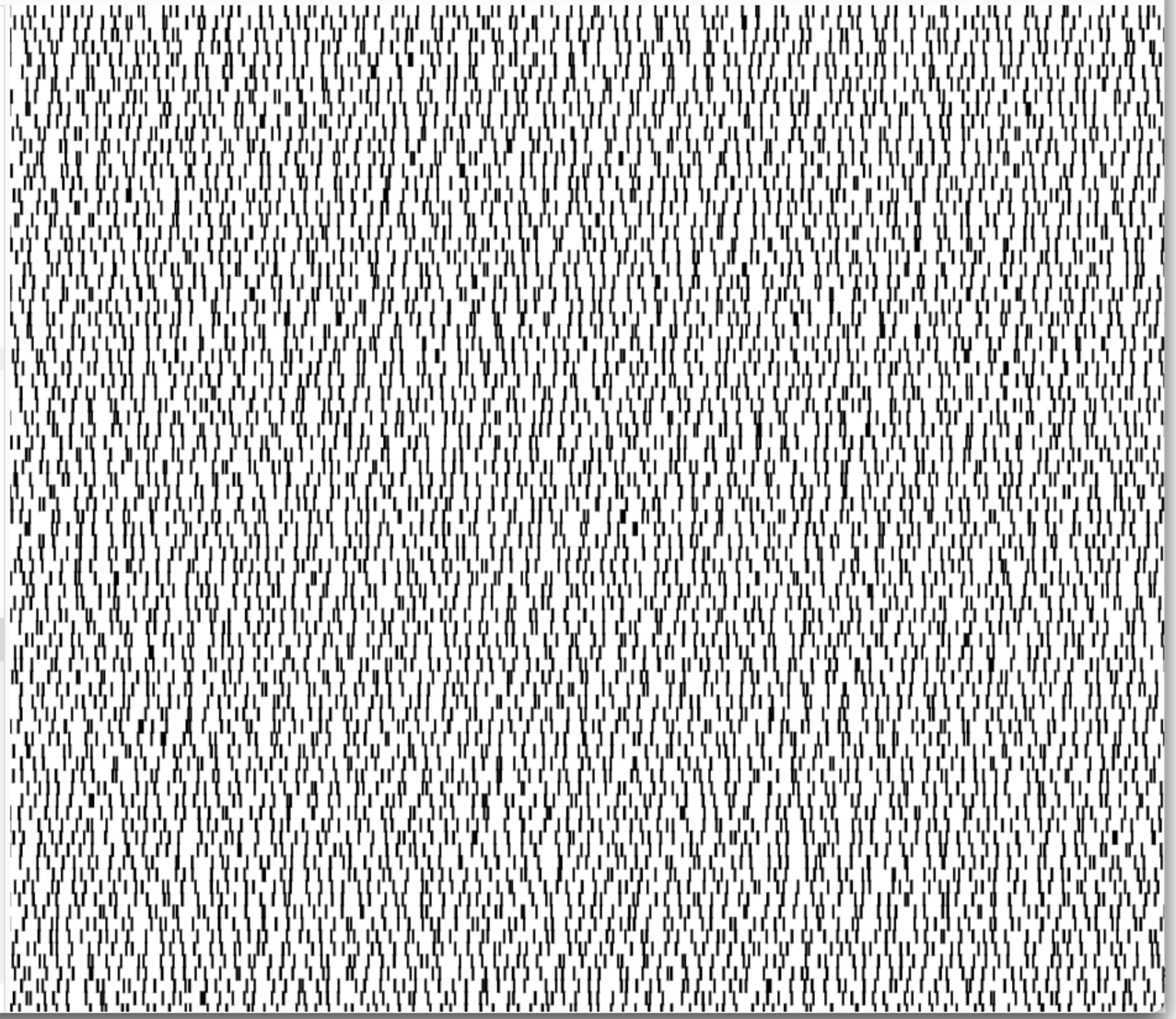
   for (let x = 0; x < windowWidth; x += 10) {
     for (let y = 0; y < windowHeight; y += 10) {
       let randomOffset = random(-5, 5);
       line(x + randomOffset, y, x + randomOffset, y + 10);
     }
   }

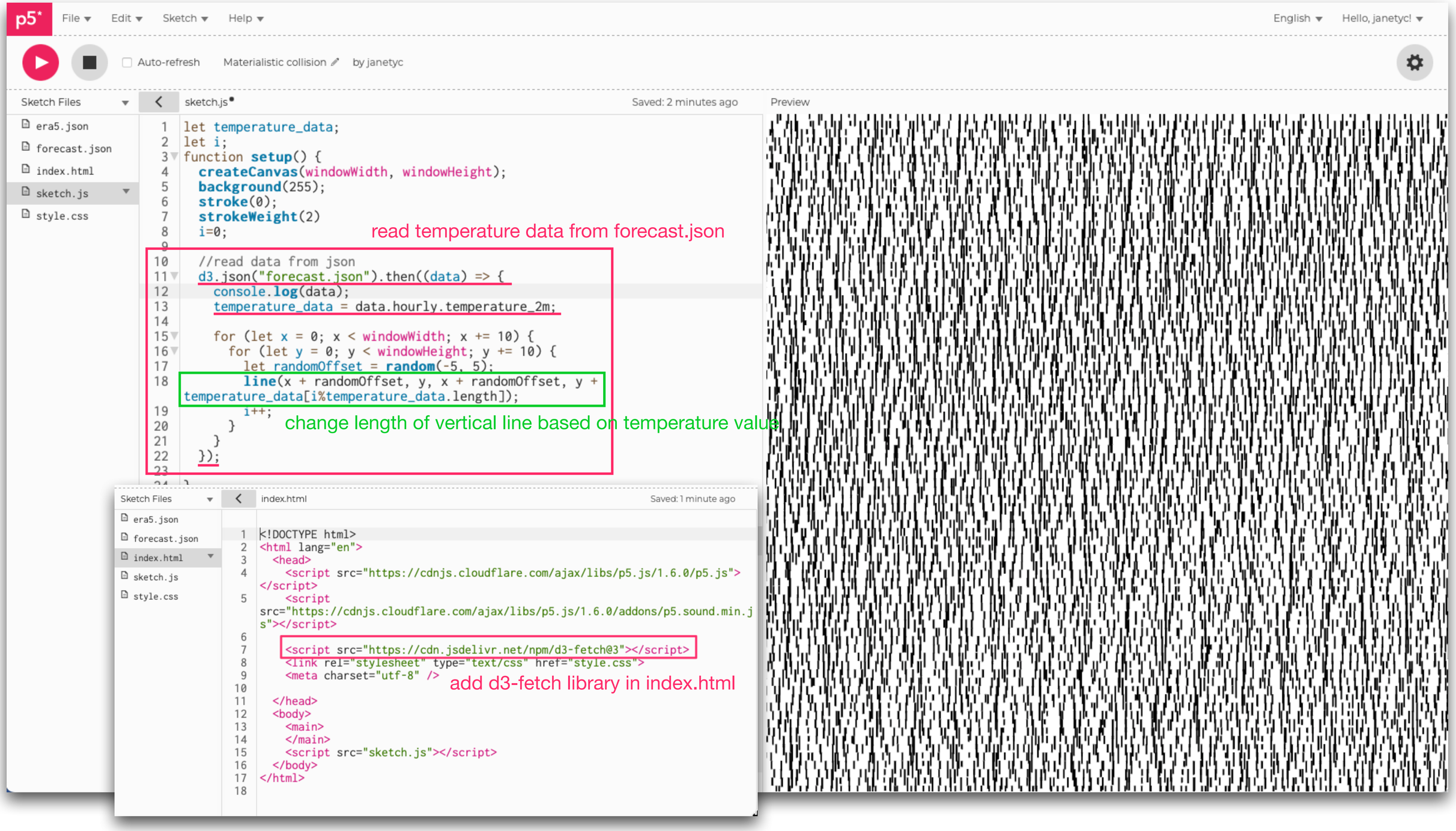
}

function draw() {
}
```

Console Clear

Preview




p5*
File
Edit
Sketch
Help
English
Hello, janetyc!
Auto-refresh
Materialistic collision
by janetyc
Sketch Files
era5.json
forecast.json
index.html
sketch.js
style.css
sketch.js
Saved: 2 minutes ago
Preview
let temperature_data;
let i;
function setup() {
createCanvas(windowWidth, windowHeight);
background(255);
stroke(0);
strokeWeight(2)
i=0;
read temperature data from forecast.json
//read data from json
d3.json("forecast.json").then((data) => {
console.log(data);
temperature_data = data.hourly.temperature_2m;
for (let x = 0; x < windowWidth; x += 10) {
for (let y = 0; y < windowHeight; y += 10) {
let randomOffset = random(-5, 5);
line(x + randomOffset, y, x + randomOffset, y + temperature_data[i%temperature_data.length]);
i++;
}
}
});
change length of vertical line based on temperature value
Sketch Files
era5.json
forecast.json
index.html
sketch.js
style.css
index.html
Saved: 1 minute ago
<!DOCTYPE html>
<html lang="en">
<head>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.6.0/p5.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.6.0/addons/p5.sound.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/d3-fetch@3"></script>
<link rel="stylesheet" type="text/css" href="style.css">
<meta charset="utf-8" />
add d3-fetch library in index.html
</head>
<body>
<main>
</main>
<script src="sketch.js"></script>
</body>
</html>

p5* File Edit Sketch Help English Hello, janetyc!

Auto-refresh Materialistic collision by janetyc

Sketch Files

- era5.json
- forecast.json
- index.html
- sketch.js
- style.css

sketch.js Saved: just now

```
let temperature_data;
let humidity_data;
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(255);
  strokeWeight(2)
  i=0;

  //read data from json
  d3.json("forecast.json").then((data) => {
    console.log(data);
    temperature_data = data.hourly.temperature_2m;
    humidity_data = data.hourly.relativehumidity_2m;

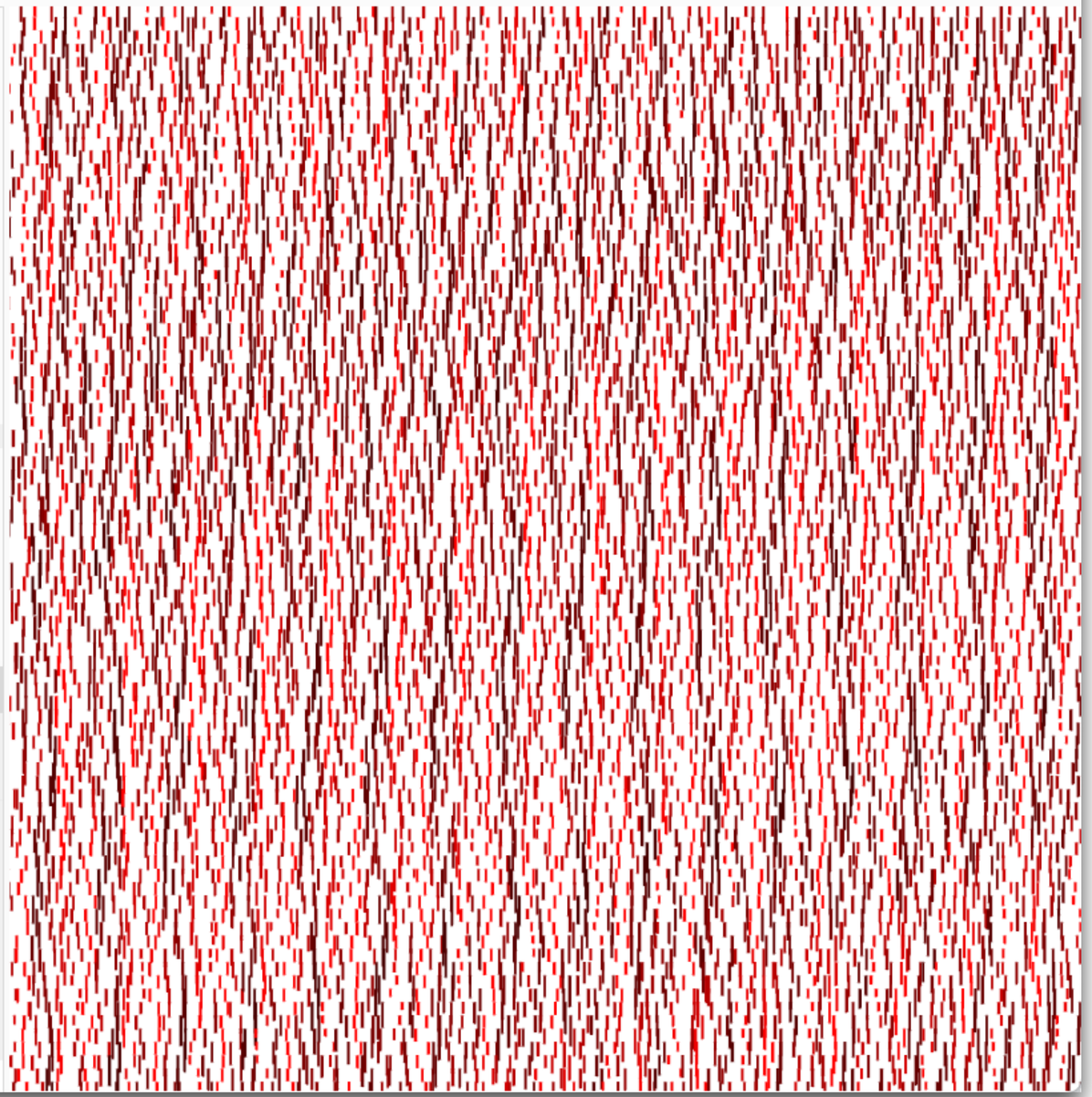
    for (let x = 0; x < windowWidth; x += 10) {
      for (let y = 0; y < windowHeight; y += 10) {
        let randomOffset = random(-5, 5);
        stroke(humidity_data[i%humidity_data.length]*3, 0, 0);
        line(x + randomOffset, y, x + randomOffset, y +
temperature_data[i%temperature_data.length]);
        i++;
      }
    }
  });

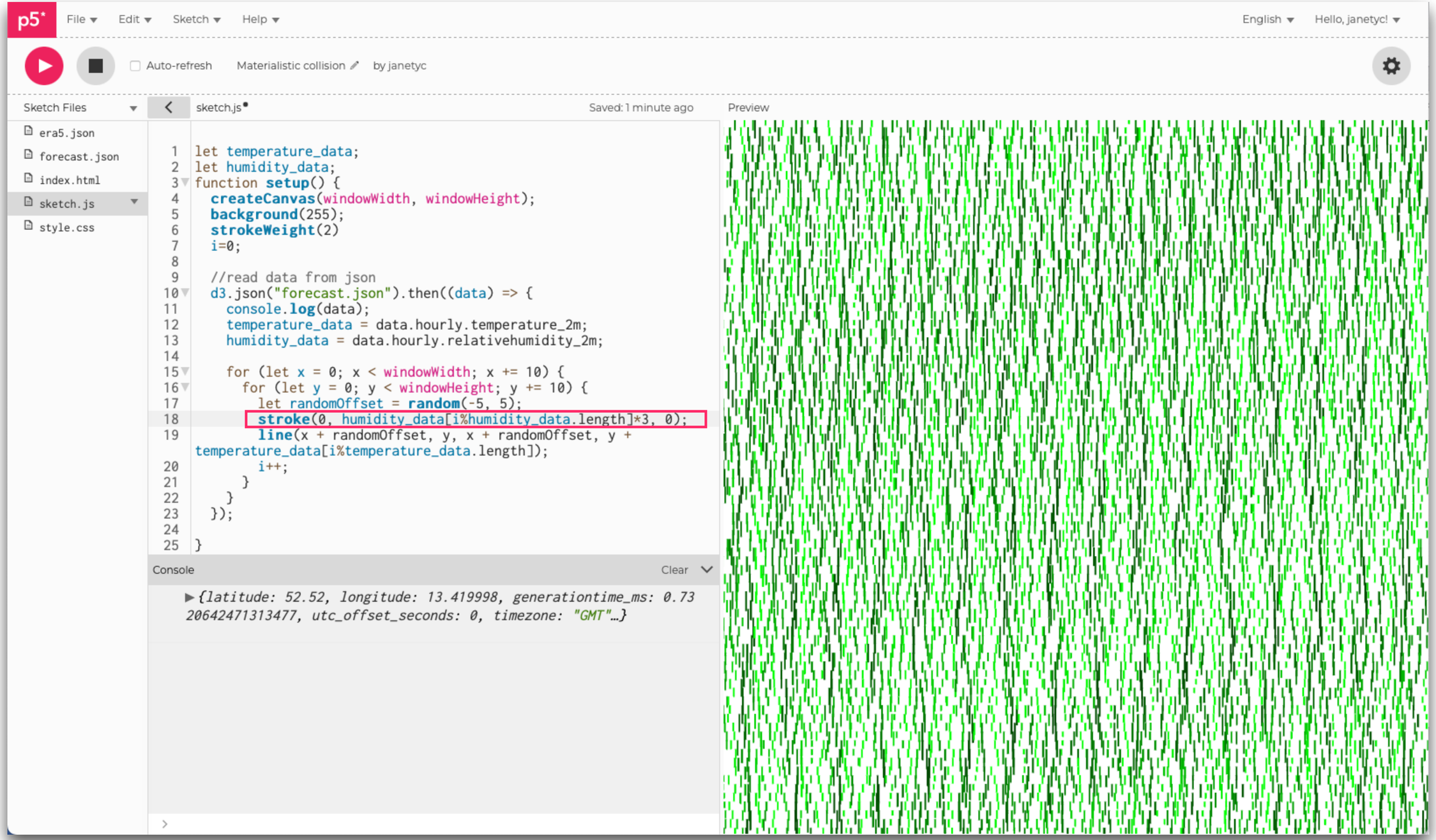
}
```

Console Clear

▶{latitude: 52.52, longitude: 13.419998, generationtime_ms: 0.73
20642471313477, utc_offset_seconds: 0, timezone: "GMT"…}

Preview




p5*
File
Edit
Sketch
Help
English
Hello, janetyc!
Auto-refresh
Materialistic collision
by janetyc
Sketch Files
era5.json
forecast.json
index.html
sketch.js
style.css
sketch.js
Saved: 1 minute ago
Preview
let temperature_data;
let humidity_data;
function setup() {
createCanvas(windowWidth, windowHeight);
background(255);
strokeWeight(2)
i=0;
//read data from json
d3.json("forecast.json").then((data) => {
console.log(data);
temperature_data = data.hourly.temperature_2m;
humidity_data = data.hourly.relativehumidity_2m;
for (let x = 0; x < windowWidth; x += 10) {
for (let y = 0; y < windowHeight; y += 10) {
let randomOffset = random(-5, 5);
stroke(0, humidity_data[i%humidity_data.length]*3, 0);
line(x + randomOffset, y, x + randomOffset, y +
temperature_data[i%temperature_data.length]);
i++;
}
}
});
}
Console
Clear
▶{latitude: 52.52, longitude: 13.419998, generationtime_ms: 0.73
20642471313477, utc_offset_seconds: 0, timezone: "GMT"…}

p5* File Edit Sketch Help English Hello, janetyc!

Auto-refresh Materialistic collision by janetyc

Sketch Files

- era5.json
- forecast.json
- index.html
- sketch.js
- style.css

sketch.js Saved: 15 seconds ago

```
let temperature_data;
let humidity_data;
function setup() {
  createCanvas(windowWidth, windowHeight);
  background(255);
  strokeWeight(2)
  i=0;

  //read data from json
  d3.json("era5.json").then((data) => {
    console.log(data);
    temperature_data = data.hourly.temperature_2m;
    humidity_data = data.hourly.relativehumidity_2m;

    for (let x = 0; x < windowWidth; x += 10) {
      for (let y = 0; y < windowHeight; y += 10) {
        let randomOffset = random(-5, 5);
        stroke(0, humidity_data[i%humidity_data.length]*3,
0);
        line(x + randomOffset, y, x + randomOffset, y +
temperature_data[i%temperature_data.length]);
        i++;
      }
    }
  });

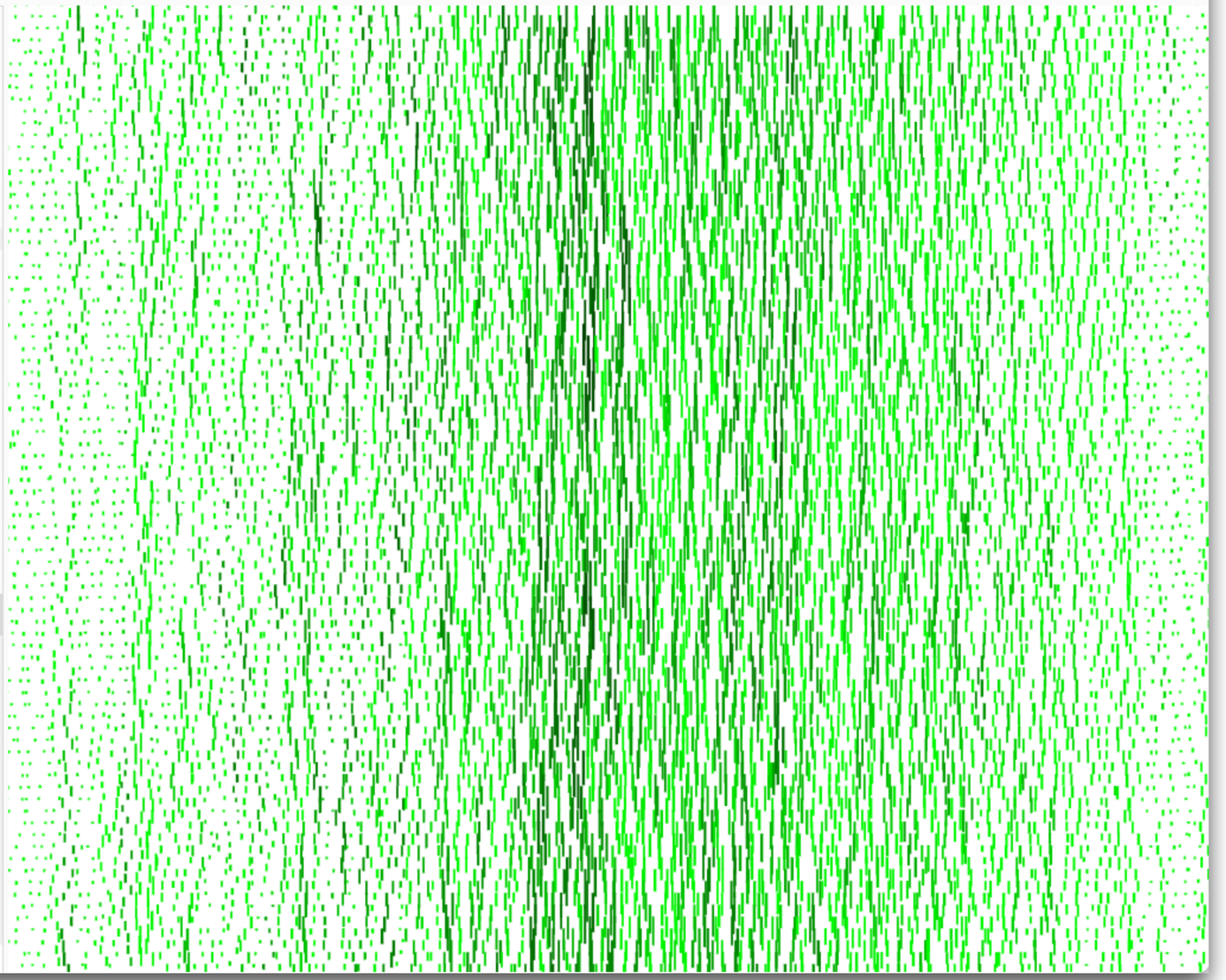
```

change weather data from 10 days to 1 year

Console Clear

▶ {latitude: 52.5, longitude: 13.400009, generationtime_ms: 0.800013542175293, utc_offset_seconds: 0, timezone: "GMT"…}

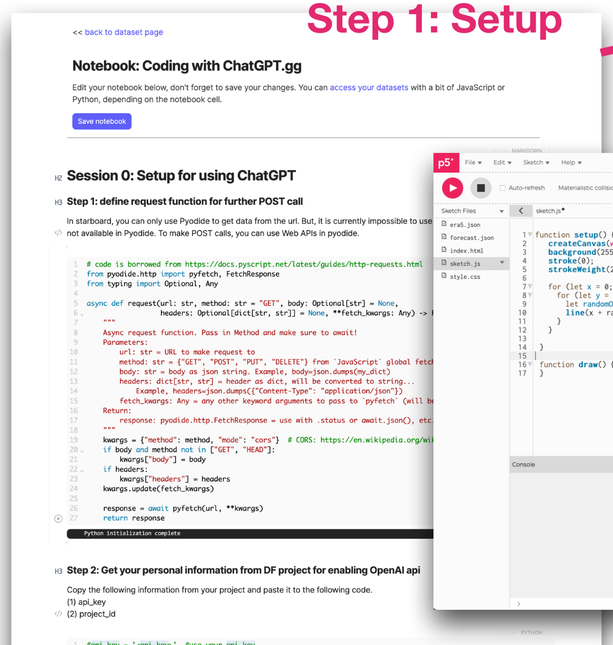
Preview

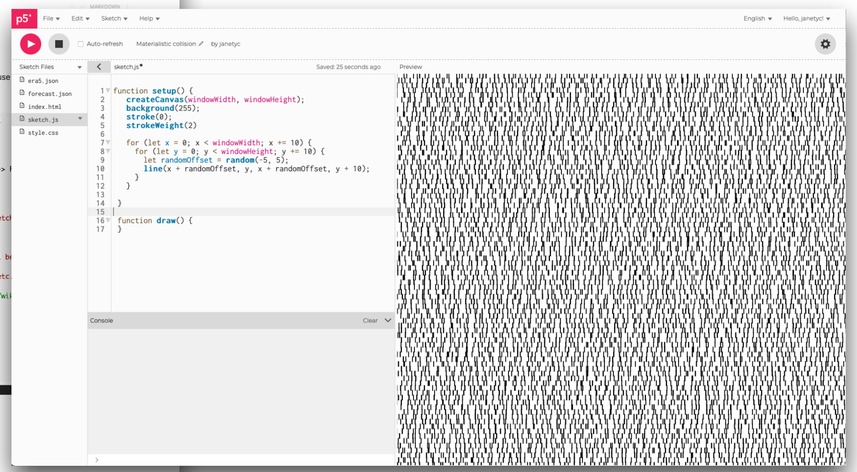


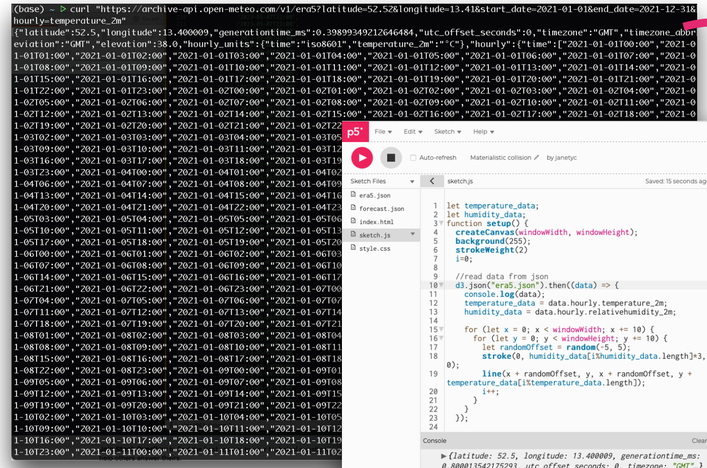
https://editor.p5js.org/janetyc/sketches/6PrSbFtCp

“Working with AI, data and patterns” Exercise (30 mins)

- Step1: Create DF project to upload the starboard notebook
- Step2: Write a prompt to describe you “dear data” pattern and generate patterns through code
- Step3: Read weather data and generate data pattern







Data Foundry x OpenAI API

Hey, you are interested in using GPT and soon chatGPT models in your design prototyping? Right on!

Thanks already for filling this form. We will credit your API key with extra credits as a small 🙏.

Hi, Janet. When you submit this form, the owner will see your name and email address.

* Required

1. Have you used GPT models or chatGPT before? *

- ◯ Yes
- ◯ No
- ◯ No idea

2. Have you used Data Foundry before in your design (research) projects? *

- ◯ Yes
- ◯ No
- ◯ No idea

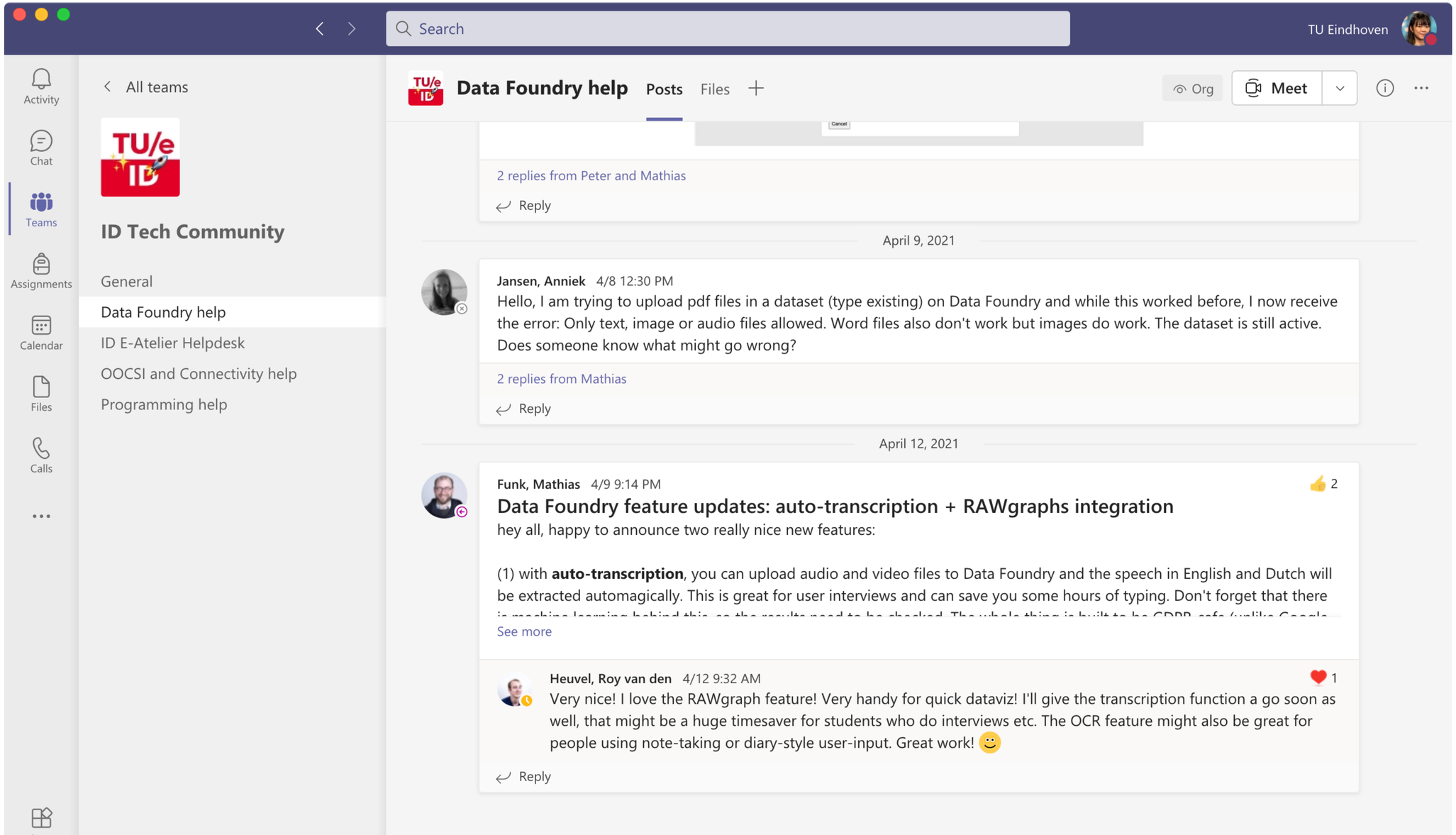
Submit

Need more extra API credits?

https://forms.office.com/e/entYh9Pjs6

Technical support from TU/e ID

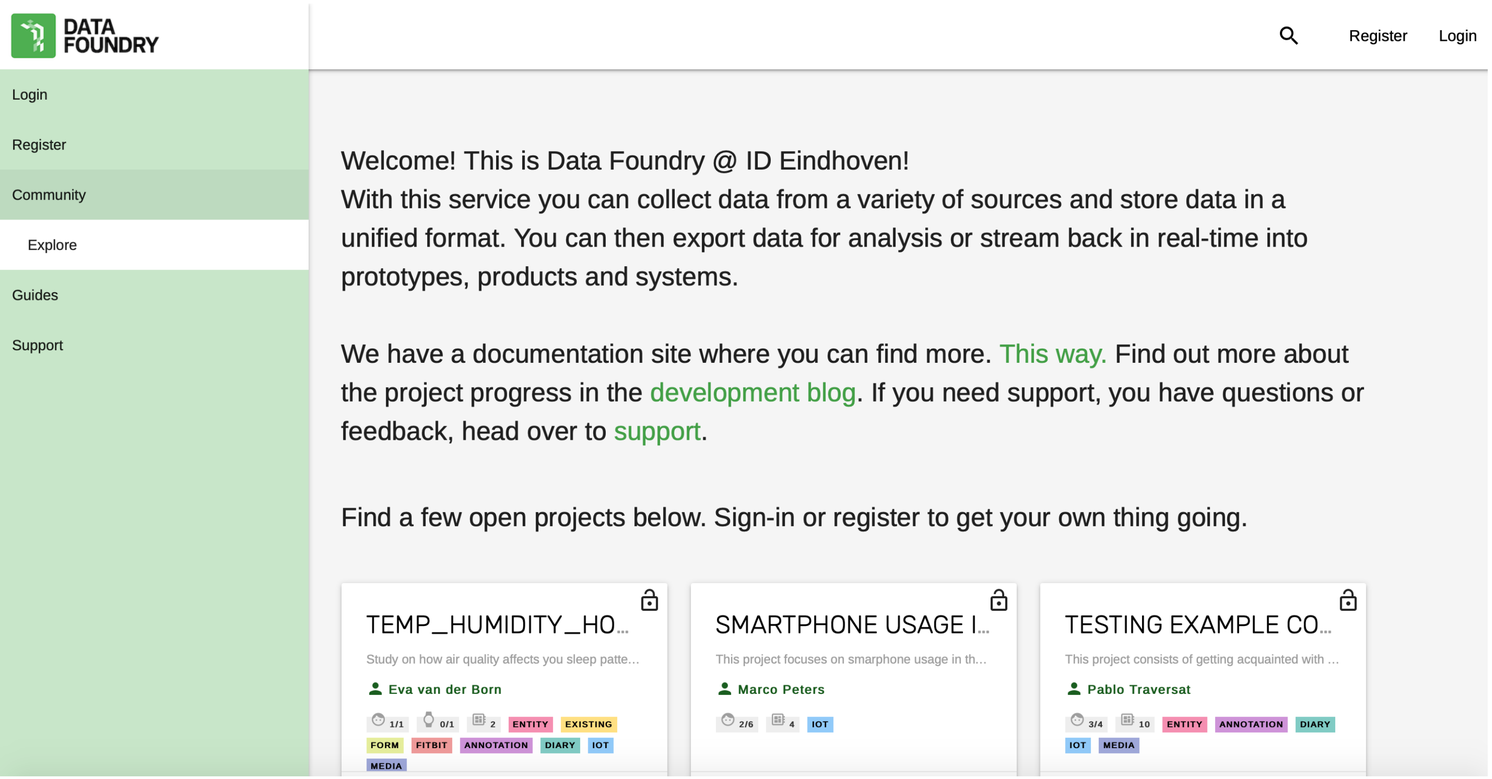
TU/e ID Tech Support [Link]



Want to know more?

- You can use the DF to collect your movement data via mobile phones, collect sensor data via smart things, track your web activities, and so on.
- Please check the use cases in the Data foundry website (https://data.id.tue.nl/documentation/use-cases)

TU/e ID Tech Support



Any problems related to Data Foundry or other technical issues, please contact Eden Chiang (@d.search lab) i.chiang@tue.nl

Recap

- What is data?
- Data as a creative material
- “Dear Data” exercise
- “Data Digitalization” exercise
- “Working with AI” exercise
- Your task: working with AI to explore data pattern
  - Step 1: collect your data (small data or big data)
  - Step 2: sketching/designing your data
  - Step 3: data digitalization
  - Step 4: working with AI, data and pattern

Q&A

Thank you!