© 2026 Andy Yaro



Outbreak Detective: Working Backwards

Yiguedan Philippe Andy Yaro

View GitHub Repository: github.com/andyyaro/outbreak-detective

Let's connect! Email: yyaro@gmu.edu

LinkedIn: linkedin.com/in/andy-yaro



I built Outbreak Detective, a free, interactive modeling lab for high-school data science. A student is handed an outbreak as a chart of counts and works backwards to the hidden rates that produced it: first by hand, then by racing three methods.

Seven cases run entirely in the browser, with no logins and nothing to install, so it works on a school Chromebook. One case is live: a measles outbreak that is still going on.

The methods come from the research of Dr. Padmanabhan Seshaiyer and colleagues; this is an independent classroom tool, not an official product of those authors. I am Yiguedan Philippe Andy Yaro. The code is open source under Apache-2.0, live at outbreak.andyyaro.com.

A. Why Outbreak Detective Exists



Every outbreak model has knobs: how fast a disease spreads, how long people stay sick. Run it forwards and it draws a curve.

Real life runs the other way. You have the counts and need the knobs: an inverse problem, and most of what epidemic modelling is.

Outbreak Detective teaches that one skill, first with two sliders, then by racing three methods that do it properly.

1. Six Steps, Every Case

The Briefing
story, data, film

1. The data, where it came from and what it can't tell you, plus a narrated film.

Make a Guess
two knobs, by hand

2. Turn the knobs until the curve meets the dots. The site scores each guess by its error.

The Race
three methods

3. Optimizer, Statistics and Neural Net start from the student's guess and race to the best fit.

The Map
every guess, shaded

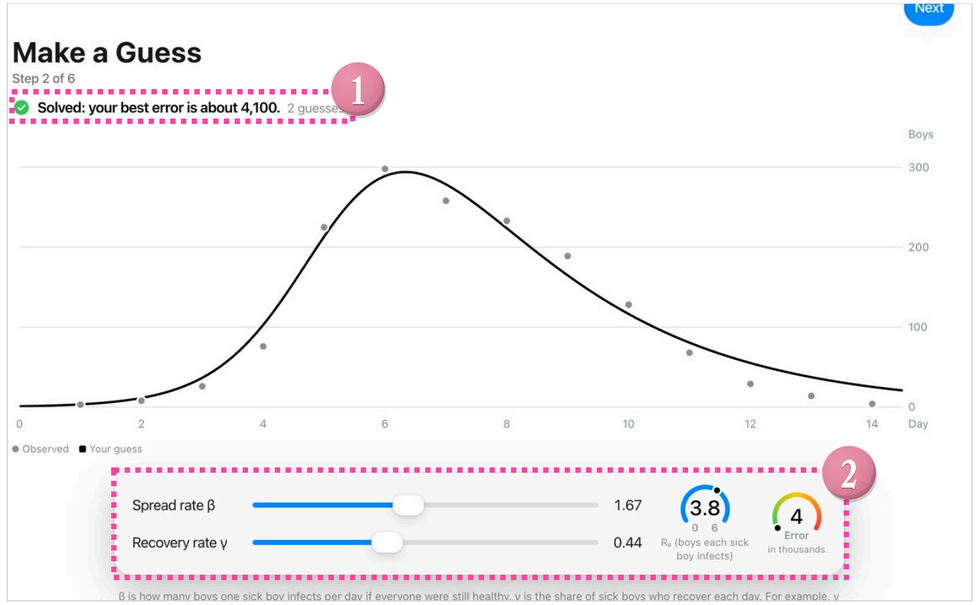
4. Every possible guess, shaded by its error, with each method's path drawn on it.

The Twist
the case's one idea

5. Cut the data early, compare models, add noise, reveal a hidden answer, forecast live, or split two routes.

Verdict
a report to hand in

6. What the methods agree on, how sure they are, and a case report to download.



Make a Guess on case 1, the 1978 boarding-school flu. Screenshot from the repository's README assets, cropped.

1. The error adds up how far the curve misses each day's count, squared, so big misses count extra. The site rounds this guess's error to about 4,100; the optimizer's best is 4,122, so by hand this is about as good as it gets.

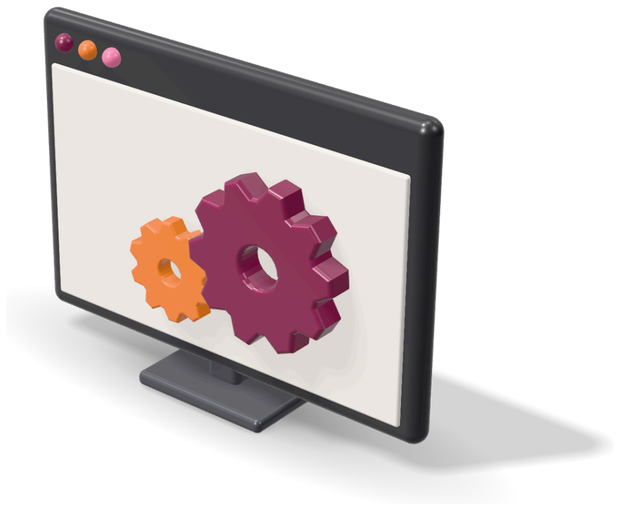
2. Two sliders: β, how many boys one sick boy infects per day, and γ, the share of sick boys who recover each day. At 1.67 and 0.44, $R_0 = \beta \div \gamma$ is about 3.8.

β = spread rate
γ = recovery rate
R_0 = people one case infects at the start

The same six steps run in every case, from the 1978 flu to the live measles case.

: The order a student works through

B. Architecture

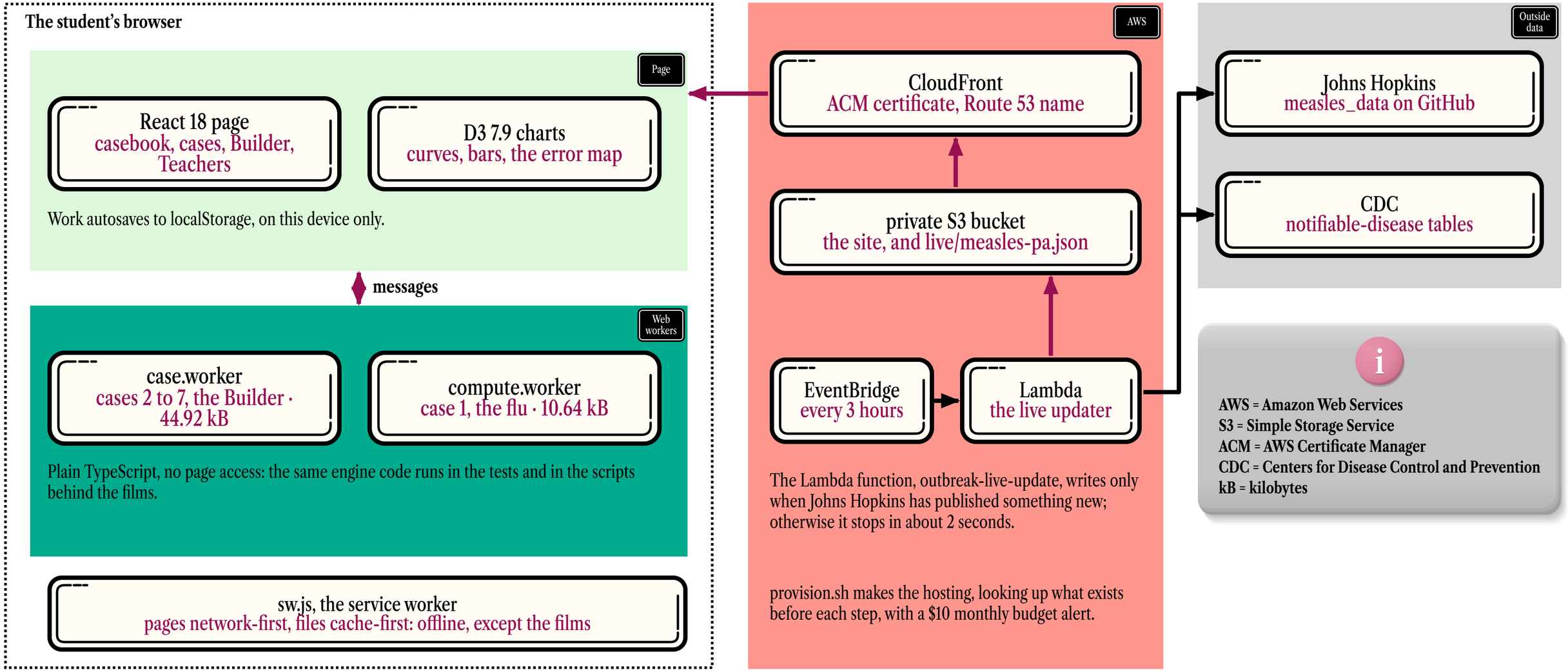


The site is static files. The models and all three methods are plain TypeScript that runs in a web worker, so the page stays responsive while the race runs.

Three runtime dependencies: React, React DOM and D3. The equation solver, the optimizer, the sampler and the neural network are written by hand.

Nothing runs on a server except one scheduled function that refreshes the live case.

1. A Static Site, a Worker Engine, One Scheduled Function



2. Cases 2 to 7 on One General Engine

CaseSpec: What a Case Is

dataset: fixed data, or a seeded generator for The Lab and Mystery

models: the candidates, the first one the default

twist: one of six kinds: cutoff, models, noise, reveal, live, routes

briefing, facts, standards, level and class periods

ModelSpec: What a Model Is

equations, states and parameters, all positive and fitted in log space

rhs: the rates of change, solved by fourth-order Runge–Kutta

observe: each series reads one state linearly, so the neural net can use it

pinn: whether the neural net runs, and why not when it doesn't

caseProtocol: Page and Worker

race: three methods; a new race cancels the old

landscape: a 60 × 45 slice of the error

compare: fit each model and score it fairly

experiments: twenty noisy fits for The Lab

define: compile a Builder model as b:<id>

replies: progress as it runs, then each result

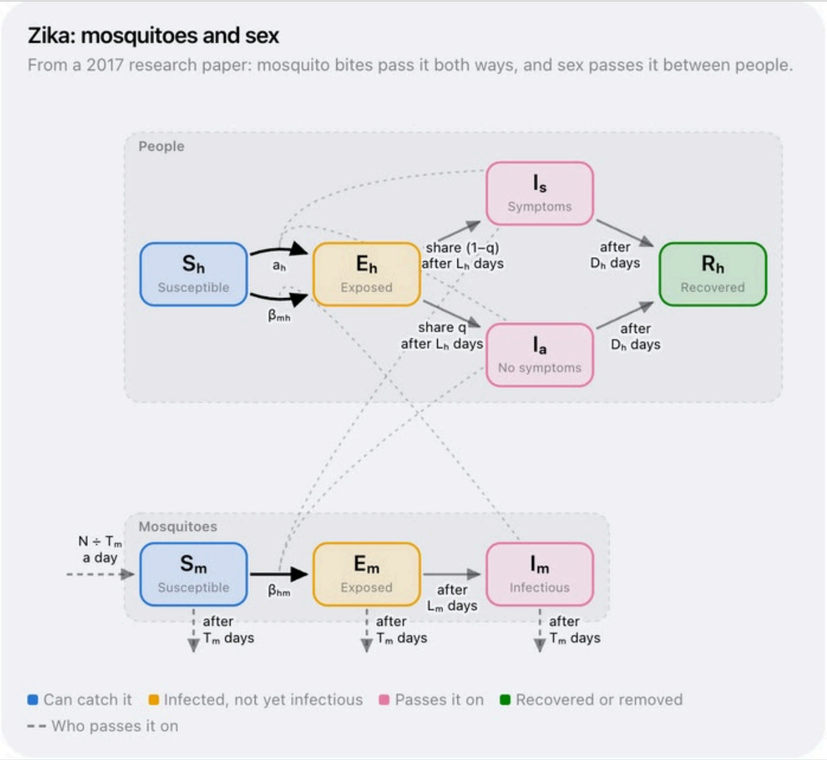
The built-in models, and the cases that use them

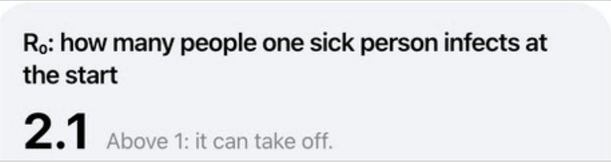
sir	seir	sir-int	seir-cum	seir-pool	lv	rm	b:zika
flu, The Lab, Mystery	Mystery	Mystery	Ebola	measles	hare and lynx	hare and lynx	Zika, a Blueprint

Case 1, the flu, keeps its own page and compute.worker, with simpler SIR-only copies of the three methods. Cases 2 to 7 are data: one general case page renders each from its CaseSpec.

SIR = susceptible, infectious, recovered
SEIR = adds exposed, not yet infectious
lv, rm = Lotka–Volterra, Rosenzweig–MacArthur

3. The Compartment Builder: Draw a Model, Get R_0





Case 7's Zika model in the Builder, and the R_0 it computes. Screenshots from the repository's README assets, cropped.

boxes, arrows, rates
a Blueprint, plain JSON

→

compileBlueprint
an ordinary ModelSpec

1. A Blueprint holds groups, such as people and mosquitoes, their compartments, and four kinds of flow: infection, move, birth and death.
2. compileBlueprint turns it into an ordinary ModelSpec, so the hand fit, the Optimizer, Statistics and the Map all run on a student's own model.
3. R_0 is the spectral radius of $F \cdot V^{-1}$ at the disease-free state: the next-generation matrix of van den Driessche and Watmough (2002).
4. Before compiling, it explains in plain words what is missing from the drawing.

Checked by Tests

A drawn SIR behaves exactly like the built-in SIR.

The Zika paper's six published R_0 values, 2.02 to 5.33, reproduced to two decimals.

Host to mosquito to host gives the Ross–Macdonald form, $\sqrt{(R_{hv} \cdot R_{vh})}$.

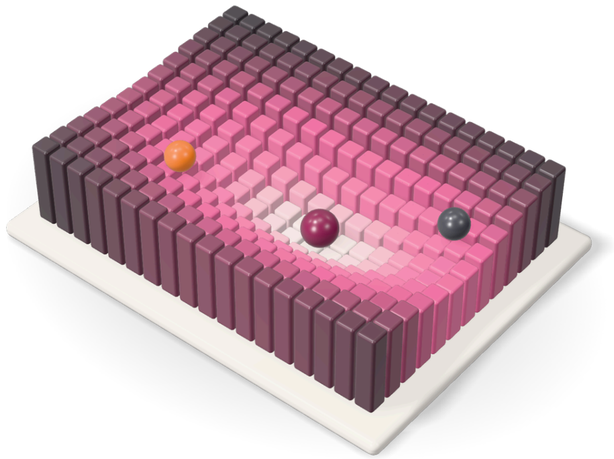
i

R_0 = people one case infects at the start
F = new infections
V = moves out of the infected compartments
JSON = JavaScript Object Notation

i Case 7's model is itself a Blueprint, b:zika, compiled by the same code a student's drawing goes through.

i Drawings save in the browser, like everything else on the site.

C. The Three Methods



Three methods start from the student's guess and race to the best fit, each answering a different question.

The Optimizer walks downhill to the single best fit. Statistics wanders among the good fits and says how sure we can be. The Neural Net learns the curve while held to the model's equations.

They are the three approaches Raissi, Ramezani and Seshaiyer compare in their 2019 paper.

1. Three Methods Race From the Student's Guess

Optimizer: Nelder–Mead

Walks downhill on the error with a few guesses at once: a triangle, for two knobs.

Fits the logarithms of the rates, so rates stay positive.

60 steps for two knobs; 250 with one restart for up to four; 600 with two restarts beyond.

On the Map: a blue path.

Answers: the single best fit.

Statistics: Adaptive Metropolis

Wanders near good fits, keeping better ones more often: a Markov chain Monte Carlo sampler.

In burn-in its steps adapt toward about 25% acceptance, then freeze, so the kept samples are valid (Haario and others, 2001).

2,000 steps with the first 400 discarded, for two knobs; up to 6,000 with 3,000 discarded.

Case 1 uses a simpler random walk whose step adapts toward about 30%.

On the Map: an orange cloud.

Answers: how sure, as a 95% range.

Neural Net: Physics-Informed

Learns a smooth curve through the counts while being held to the model's equations.

The model's rates train alongside the network's weights.

3,000 training passes, 4,000 for SIR. Written by hand, with no machine-learning library.

On the Map: a green path.

Answers: rates that fit the data and obey the equations.

the student's guess
where all three start

→ the race
progress streamed live

→ the Map
every path, one surface

→ Verdict
do they agree?

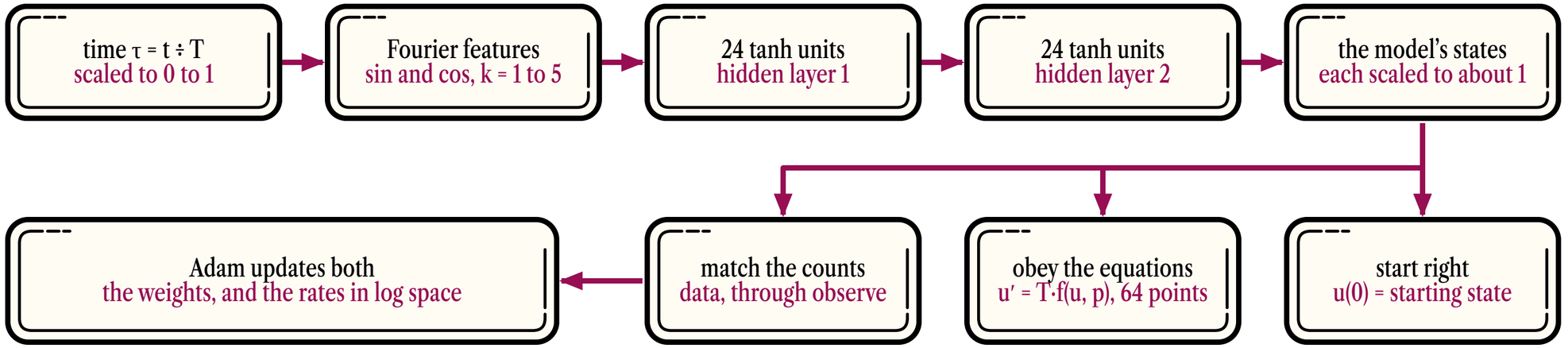
i

SIR = susceptible, infectious, recovered

Every fit is scored on a fixed-step, fourth-order Runge–Kutta solution of the model's equations, so the three errors compare directly. These panels describe the general engine of cases 2 to 7; case 1 runs simpler SIR-only copies.

: A race, in order

2. A Neural Network With No Machine-Learning Library



τ = time scaled by T, the last time used
tanh = hyperbolic tangent
u = the network's states
f, p = the model's equations and rates

Forward-mode dual numbers give each output and its time derivative in one pass. The model's own derivatives come from central finite differences, so any ModelSpec works.

Training schedule for 3,000 passes, the default



1 to 375
data only, rates frozen

376 to 875
physics ramps in, rates frozen

876 to 1,125
ramp ends; rates train

1,126 to 3,000
full weight, everything trains

Warm-up is capped at an eighth of the passes and the ramp at a quarter. SIR: 4,000 passes, 500 data only, no settle, no Fourier features. Learning rate 5×10^{-3} for the weights, decaying to 5×10^{-4}; 64 collocation points, 48 for the predator–prey models.

Case 1, the 1978 flu, runs its own SIR-only copy of this network. There it finds R_0 3.9 against the others' 3.8, with an error of about 7,900 against 4,100: the price of obeying the equations.

Each model has its own tuned settings, and whether the net runs on it at all was set from the tests.

Where It Sits Out, and Says Why

SIR with an intervention: the cut in spread is too sharp for a smooth curve to bend.
SEIR with an unknown pool (measles): in tests it guessed about 1.5 times the pool the others found, and fit worse.
Builder models, case 7's included: not yet tested with the network.

3. The Race and the Map, on Case 1





1 Optimizer and Statistics land on R_0 3.8, error about 4,100. The Neural Net lands on 3.9, error about 7,900: it gives up some fit to obey the equations.

2 Only Statistics says how sure: 95% sure R_0 is between 3.5 and 4.0.

3 On the Map, every path from the student's guess ends in the same light valley, near β 1.67 and γ 0.44.

Screenshots of case 1, the 1978 boarding-school flu, from the repository's README assets, cropped. The test core-engine pins the optimizer's fit: $\beta \approx 1.669$, $\gamma \approx 0.443$, error $\approx 4{,}122$.

D. Seven Cases, Seven Ideas



Each case adds one new idea to the same six steps.

Five use real, sourced data. The Lab and the Mystery Outbreak are simulated from a seed, so a whole class can work the same outbreak.

The twists run from cutting the data short, to choosing between models, to a forecast scored as the real numbers arrive.

1. Seven Cases, Seven Ideas

#	Case	Data	Model	The twist	Level · periods
1	Boarding-School Flu, 1978	763 boys; daily counts read off the BMJ's graph	SIR	The Twist: cut the data off early and the answer grows less certain	Start here · 2
2	Hare & Lynx, 1900–1920	Hudson's Bay Company pelt records	Lotka–Volterra; Rosenzweig–MacArthur	Which Model? More knobs fit the past, not always the future	Intermediate · 2
3	The Lab	made by the student, from a seed	SIR	Twenty Experiments: noise decides how close the methods get	Start here · 1–2
4	Ebola, West Africa 2014	WHO weekly totals for Sierra Leone	SEIR with case counting	The Forecast: an early projection overshoots by far	Advanced · 2
5	Mystery Outbreak	simulated, with a hidden answer	SIR; SEIR; SIR with an intervention	The Reveal: commit to a model and an answer, then see the truth	Advanced · 1–2
6	Measles, Pennsylvania 2026	Johns Hopkins weekly counts, live	SEIR with an unknown pool	The Forecast: score it as the real numbers arrive	Intermediate · 1–2
7	Zika, French Polynesia 2013	sentinel doctors' reports from Moorea	a Blueprint of people and mosquitoes	Which Route? Two routes of spread draw the same curve	Advanced · 2

DS.7
judging how reliable source data is

DS.9
choosing models, predicting, assessing uncertainty

DS.10
summarizing and interpreting visualizations

DS.11
models, and how well they match the data

Virginia Data Science Standards of Learning, as each case spec lists them; cases 1, 2, 4, 6 and 7 cover all four. A comment in the code says to recheck the codes before printing. Periods are 50-minute class periods.

2. More Knobs, and an Early Forecast

Hare & Lynx: Which Model?

Model	Knobs	Fit error	Fair score	Forecast error
Lotka–Volterra	6	502.7	100.22	121.1
Rosenzweig–MacArthur	8	504.6	107.53	128.8

Fit 1900–1914, forecast 1915–1920. Two more knobs fit no better, score worse once charged for them, and forecast worse.

All 42 numbers match across three published reproductions. They count pelts traded, not animals.

A lower fair score is better: the error, plus a charge for every knob.

Kept as reported: Sierra Leone's total falls from 5,235 in week 21 to 4,759 in week 22, because WHO changed its data sources. Both cases' figures are pinned by the tests core-hare and core-ebola.

Ebola: The Forecast

Fitted to the first 20 weeks, to 19 October 2014: R_0 1.40, cases doubling about every 4.6 weeks. Total cases, carried forward:

Week	Projected	Reported
24 · 16 Nov 2014	6,958	6,073
33 · 18 Jan 2015	27,509	10,306
56 · 28 Jun 2015	721,691	13,115

Close for a month, then 55 times too high: a constant-rate model can't see people change what they do.

CDC's September 2014 projection: about 550,000 cases in Liberia and Sierra Leone by 20 January 2015, if nothing changed. The whole epidemic there: 24,802.



Fair score = AICc, corrected Akaike information criterion
WHO = World Health Organization
CDC = Centers for Disease Control and Prevention

3. Zika: Two Routes, One Curve

Fitted six times, with the sexual rate held at each value; the mosquito count, the starting infections and the reporting share fitted each time

Sexual rate, per day	R_0 from each route alone: ■ mosquitoes ■ sex	R_0 in all	Fit error
0.001	2.13 0.01	2.13	654 (+3%)
0.05	2.02 0.25	2.14	640 (+1%)
0.1	1.90 0.50	2.16	633 (best)
0.2	1.62 1.00	2.19	640 (+1%)
0.3	1.27 1.50	2.22	673 (+6%)
0.4	0.69 2.00	2.22	687 (+8%)

Every version fits the 25 weekly counts to within 9% of the best, and R_0 stays between 2.1 and 2.2. Only the split moves: from these counts, the two routes cannot be told apart.

The Antibodies Disagree

49% of people across French Polynesia had antibodies after the outbreak (Aubry and others, 2017).

The model, in which everyone mixes and meets the same mosquitoes, says about 95% were infected.

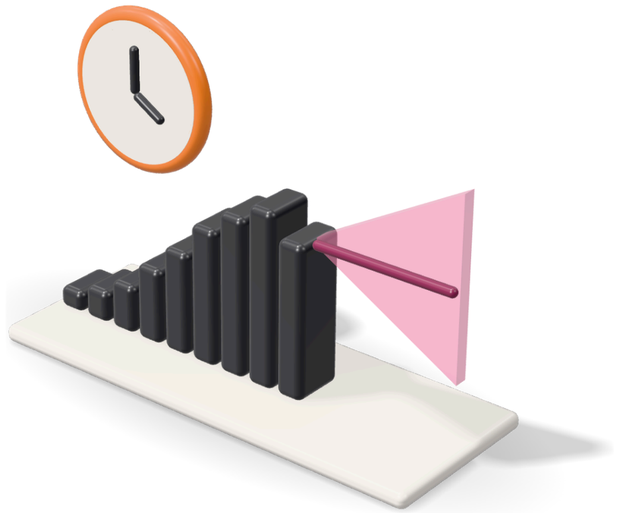
The starting sexual rate, 0.02 a day, is a teaching choice, not a published value; the README and SOURCES.md say so.

The model is Padmanabhan, Seshaiyer and Castillo-Chavez (2017), with bed nets and spraying switched off.

Computed for this deck with the project's engine, the way the case's Which Route? step does it: tools/zika_routes.ts.

R_0 = people one case infects at the start

E. The Live Case

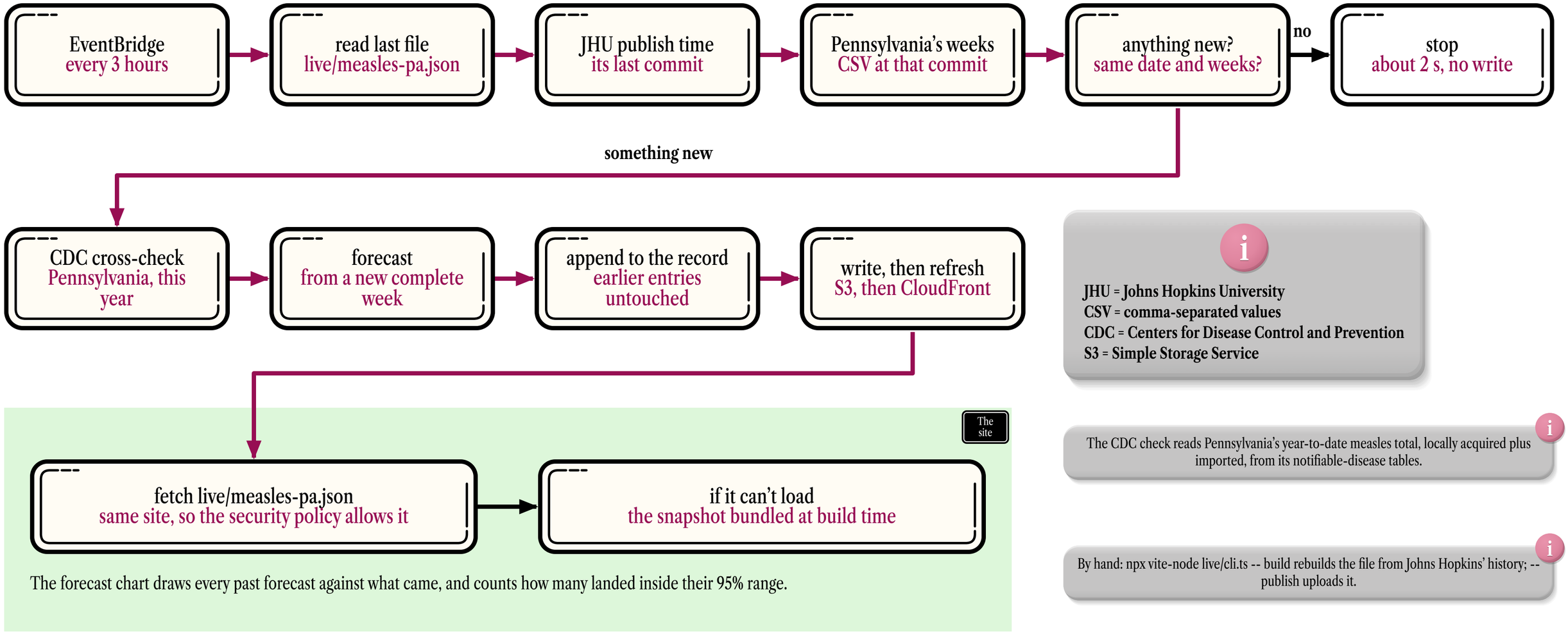


Case 6 is an outbreak that is still going on: measles in Pennsylvania, 2026, from the Johns Hopkins weekly counts.

Every 3 hours a scheduled function checks for a new release. When there is one, it rebuilds the case's data and adds a forecast to a public record; it never rewrites the earlier entries.

The chart shows every past forecast against what came, and keeps score.

1. The Updater, Step by Step



JHU = Johns Hopkins University
CSV = comma-separated values
CDC = Centers for Disease Control and Prevention
S3 = Simple Storage Service

The CDC check reads Pennsylvania's year-to-date measles total, locally acquired plus imported, from its notifiable-disease tables.

By hand: npx vite-node live/cli.ts -- build rebuilds the file from Johns Hopkins' history; -- publish uploads it.

: In order : No change, or the site's fallback

2. Forecasting an Outbreak That Hasn't Ended

fit
SEIR, unknown pool

Fit every complete week from two starting pools, 3,000 and 30,000 people, and keep the better fit.

sample
Statistics, 4,000 steps

Run Statistics from that fit, discarding the first 1,500 steps. About 400 of the rest run four weeks ahead.

scatter
in proportion, ±30%

Give each week's count the scatter the data have shown, as a share of its size, so it can't go below zero.

range
2.5th to 97.5th

The middle 95% of the results is the week's 95% range; the median is the forecast.

seed
the date it is made

Random numbers are seeded by the day, so the same data on the same day give the same forecast.

i

SEIR = susceptible, exposed, infectious, recovered

24 of 26 inside the range
92% of checkable weekly forecasts

Keeping Score

Each forecast looks four weeks ahead, and each of those weeks is scored once its count is complete.

A week that had not ended when Johns Hopkins published is still being counted.

A good 95% range misses about 1 week in 20. This record misses 2 in 26.

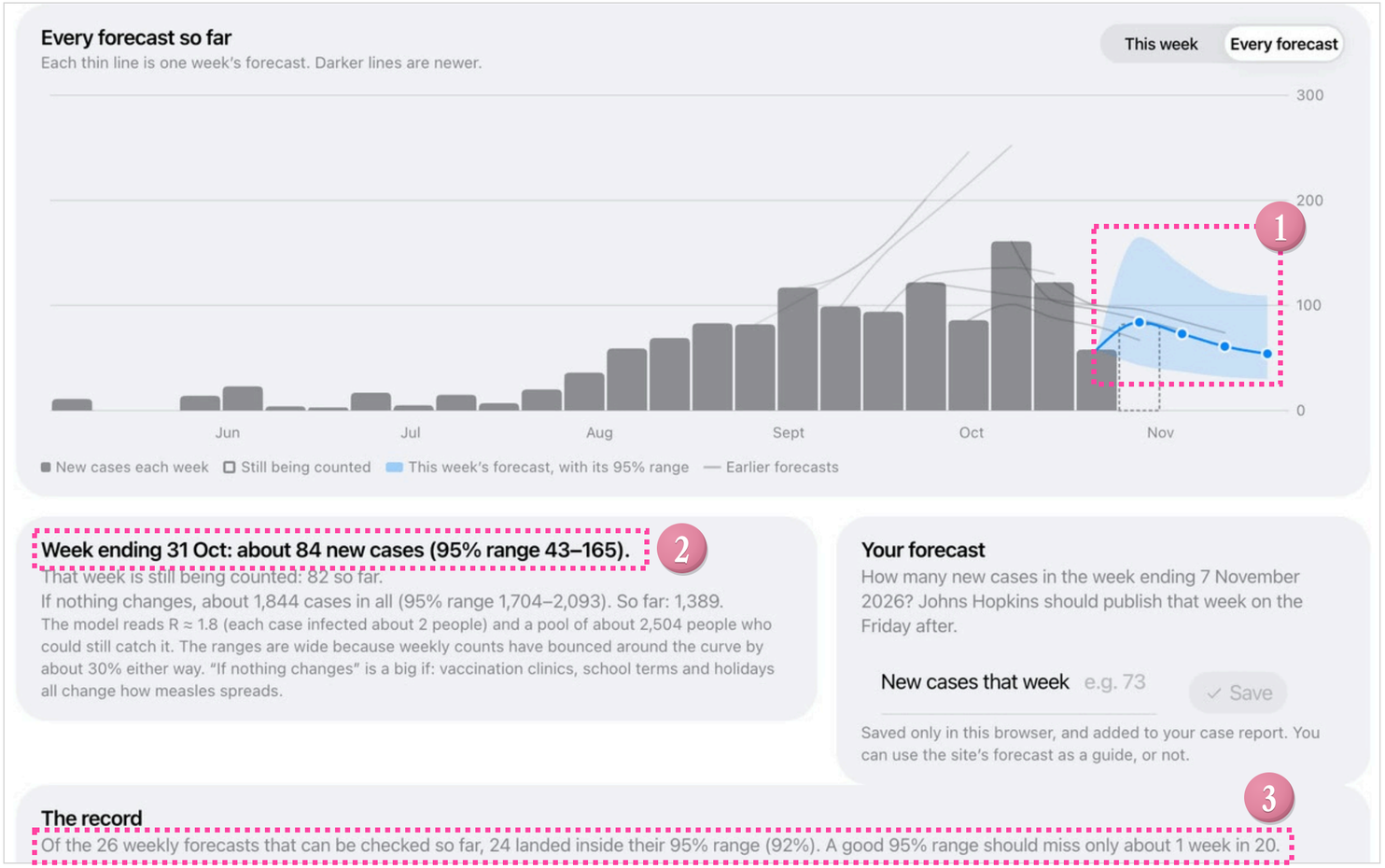
A Choice Made on 19 October

Constant scatter caught 5 of the first 18 checkable weeks; proportional scatter caught 16 of 18.

Forecasts dated before 19 October were rebuilt with that choice, so they are not a pure test. Later ones are.

: Each forecast, in order

3. The Live Forecast, Reproduced




```
$ npx vite-node tools/live_repro.ts
reconstructed forecasts rebuilt: 7; identical to the shipped snapshot: true   4
JHU published 2026-10-30T20:16:25Z; 26 weeks; CDC check: 1335 cases in 2026 to MMWR week 42
new forecast (data through 2026-10-24): week ending 2026-10-31 about 86, 95% range 43 to 152
forecast weeks scored so far: 24 of 26 inside their 95% range   4
```

1. This week's forecast, with its 95% range, runs on from the last complete week. Thin grey lines are earlier forecasts.

2. The site's forecast for the week ending 31 October: about 84, range 43 to 165. That week was still being counted.

3. The record: 24 of the 26 checkable weekly forecasts landed inside their 95% range.

4. Rebuilt for this deck from Johns Hopkins' own history: the 7 reconstructed forecasts match the shipped ones exactly, and the record is again 24 of 26.

Today's rerun forecasts about 86, range 43 to 152: the seed is the date, and the date differs. Screenshot from the README assets, cropped; terminal from tools/live_repro.ts, run on 2 November 2026.

F. Data You Can Check



Every number in the case data was copied from a source listed in SOURCES.md, with the checks made on it. None was typed from memory.

Where the record is messy, the site says so instead of smoothing it over: the 1978 flu counts are read off a graph, and an Ebola total falls one week because WHO changed its sources.

1. Provenance, Case by Case

Case	Source	How it was checked	Kept visible
Flu, 1978	BMJ 1978;1(6112):587, which prints no table, only a graph	The 14 daily values are readings off that graph, as reproduced by Murray and others; they agree with the scan to within a few boys.	The site calls them readings, not reported numbers.
Hare and lynx, 1900–1920	Hudson's Bay Company pelts, as compiled by Elton and Nicholson, and MacLulich	All 42 numbers identical in three reproductions: Howard's 2009 notes, Stan's example models, Hsu, Liu and Magal (2020).	Pelts traded, not animals counted; the 1937 and 1942 tables were not obtained.
Ebola, Sierra Leone, 2014–15	WHO totals: Rivers' digitised series to December 2014, then the Humanitarian Data Exchange	Spot values match WHO bulletins; the two sources agree wherever both cover August 2014 to January 2015.	Week 22 falls below week 21; kept, with WHO's explanation.
Measles, Pennsylvania 2026	Johns Hopkins Measles Tracking Team, CC BY 4.0, weekly by rash onset	Cross-checked against CDC's notifiable-disease totals on every update; revisions were at most 2 cases a week from 11 September to 16 October.	The newest week is always marked as still being counted.
Zika, Moorea, 2013–14	Champagne and others, eLife 2016, supplementary file 3; the same counts as Kucharski and others (2016)	25 weekly counts, 462 cases; a test reproduces the model paper's R_0 values.	The starting sexual rate is a teaching choice; the README and SOURCES.md say so.

retrieved 18 October 2026
every fixed dataset

none typed from memory
copied from a listed source

a licence per dataset
WHO data used non-commercially

BMJ = British Medical Journal
WHO = World Health Organization
CDC = Centers for Disease Control and Prevention
CC BY = Creative Commons Attribution

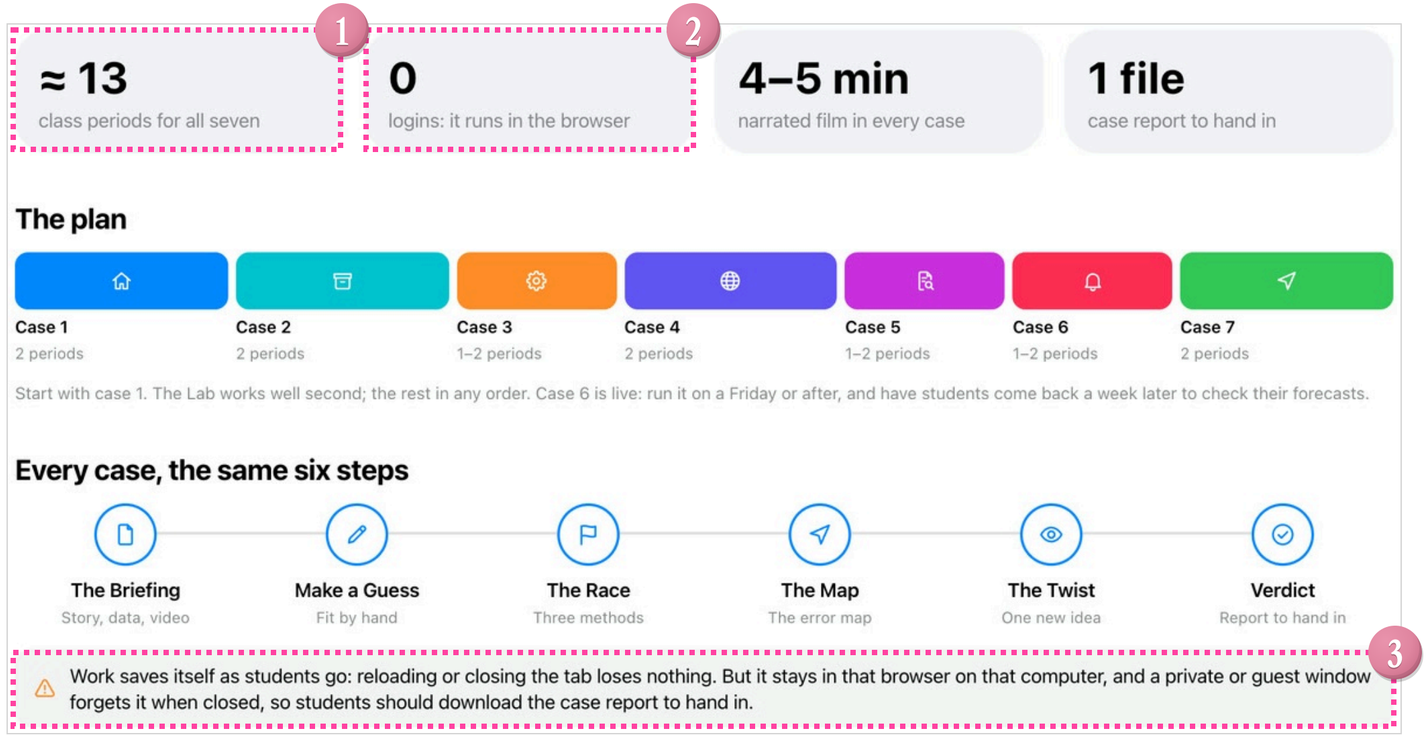
G. Built for a Classroom



It has to work on a school Chromebook, in a 50-minute period, with no accounts. So everything runs in the browser, nothing is collected, and work saves itself on the device.

Every case has a narrated film of about four minutes, animated in code. Teachers get class periods, expected answers and a grading guide.

1. Periods, Privacy and Offline



1. About 13 class periods for all seven cases: 2, 2, 1–2, 2, 1–2, 1–2 and 2.
2. No logins and nothing to install, so it works on a school Chromebook.
3. Work autosaves on the device. A private window forgets it, so students download the case report.

The Teachers page, from the repository's README assets, cropped.

Private by Construction

No accounts, and no analytics or tracking scripts.

The security policy lets pages connect only to the site itself: connect-src 'self'.

Saved work and Builder drawings stay in the browser's localStorage.

There is nowhere to send work: the case report is a file the student downloads.

A service worker caches the site, so after a first visit it works offline, except the films.

Virginia Standards Covered

DS.7 judging how reliable source data is

DS.9 choosing models, predicting, assessing uncertainty

DS.10 summarizing and interpreting visualizations

DS.11 models, and how well they match the data

2. Films Made in Code

narration.py
the lines, as data

→ Kokoro-82M voice
af_heart, speed 0.97

→ Whisper check
clips read back

→ manim scenes
the engine's numbers

→ ffmpeg
join, subtitles, poster

→ a versioned file
never replaced

1. The lines, scene order and titles, read as data without running the file.
2. A local text-to-speech voice. A scene that feels rushed gets more time on screen, never a slower voice.
3. Every clip is transcribed back to catch mispronunciations before a film is published.
4. 3Blue1Brown's manim. Every number comes from the site's own engine, run with vite-node.
5. Joins the scenes, adds subtitles taken from the narration, and picks a poster frame.
6. A changed film gets a new name, -v2 or -v3, because the site caches video forever.

The Eight Films

1 · Boarding-School Flu	5:06
2 · Hare & Lynx	3:56
3 · The Lab	3:55
4 · Ebola	4:01
5 · Mystery Outbreak	3:57
6 · Measles	4:03
7 · Zika	4:14
Teach a fly	5:06

Analogies Are Drawn, Not Just Said

Three to five everyday analogies per film: a rumour through a school, a wobbling bathroom scale, a hiker finding a valley floor in fog.

Each is animated, then morphs into the real chart or equation, so the viewer sees they are the same thing.

intro film, 1 minute; share film, 64 seconds
made with HyperFrames from real recordings of the site

: How a film is made, in order

H. Proof and Limits



Every figure in this deck was checked against the repository on 2 November 2026. The type check, the 80 tests and the production build were re-run for it, and the live record was rebuilt from Johns Hopkins' own history.

The last slides say what passed, what needed explaining, and what the site cannot tell you.

1. Type Check, Tests and Build, Re-run for This Deck

```
$ npm run typecheck
> outbreak-detective@0.1.0 typecheck
> tsc --noEmit

$ CI=true npx vitest run
[ the tests' own log lines left out ]
 ✓ tests/blueprint.test.ts  (15 tests) 525ms
 ✓ tests/core-engine.test.ts  (17 tests) 3831ms
 ✓ tests/core-pinn.test.ts  (17 tests) 83526ms
 ✓ tests/live-measles.test.ts  (10 tests) 2433ms
 ✓ tests/pinn.test.ts  (6 tests) 67429ms
 ✓ tests/fit.test.ts  (5 tests) 338ms
 ✓ tests/core-hare.test.ts  (4 tests) 754ms
 ✓ tests/engine.test.ts  (4 tests) 100ms
 ✓ tests/core-ebola.test.ts  (2 tests) 4ms
 Test Files  9 passed (9)
      Tests  80 passed (80)

$ npx vitest run
[ the same 80 tests without CI=true; all other lines left out ]
 FAIL  tests/pinn.test.ts > trainSirPinn > recovers known rates from a synthetic SIR outbreak
AssertionError: expected 46748.286088 to be less than 15000
 Test Files  1 failed | 8 passed (9)
      Tests  1 failed | 79 passed (80)
```

The one miss is a speed budget, not a wrong answer. The neural-net recovery test allows 15 s on a laptop and 60 s on shared CI runners; on this 2-core container it took 46.7 s. GitHub Actions sets CI=true, so its runs use the 60 s budget.

Run on 2 November 2026 from the uploaded copy of the repository, after npm ci, with Node 22; the project's CI uses Node 24. The tests' own log lines are left out of the pane.

This Run

Type check	passes
Tests, with CI=true	80 of 80
Tests, without it	79 of 80
Modules built	689
Two workers	10.64 and 44.92 kB
Runtime dependencies	3

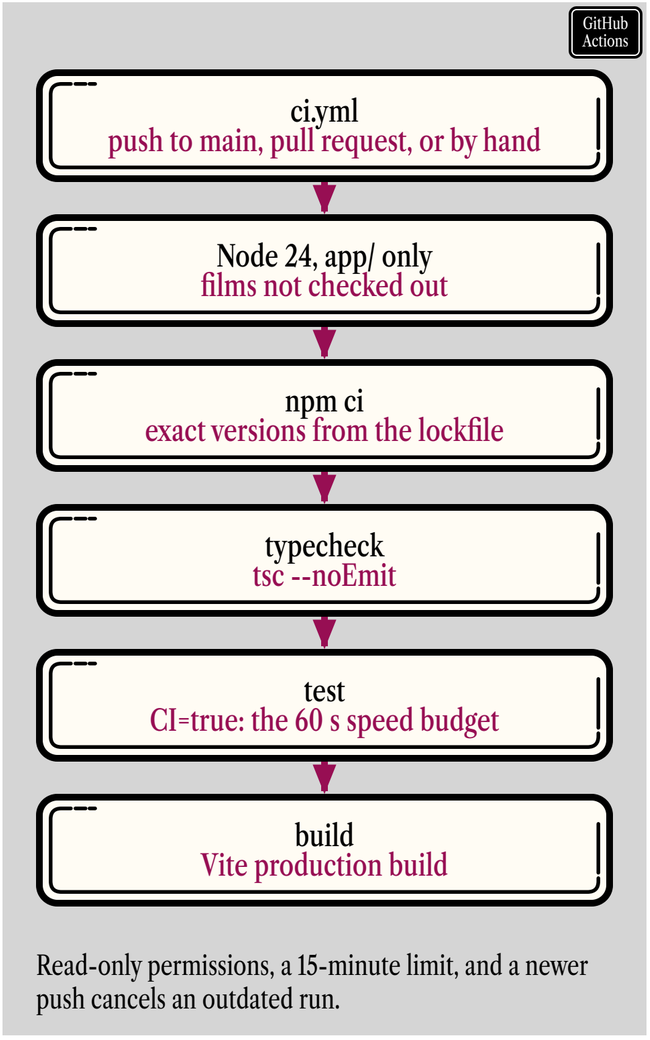
```
$ npm run build
> outbreak-detective@0.1.0 build
> tsc --noEmit && vite build
✓ 689 modules transformed.
dist/index.html                    4.54 kB │ gzip:   1.65 kB
dist/assets/compute.worker-BBEBy02T.js   10.64 kB
dist/assets/case.worker-B7Y7HWUR.js      44.92 kB
dist/assets/fly-CN7lwnIY.webp     154.46 kB
dist/assets/App-D3rUYVEX.css       32.03 kB │ gzip:   6.81 kB
dist/assets/index-BaD8IAKd.js     145.43 kB │ gzip:  47.12 kB │ map:   351.76 kB
dist/assets/App-CF80eBep.js       412.87 kB │ gzip: 141.06 kB │ map: 1,303.33 kB
✓ built in 3.73s
```

Build: 689 modules; the two web workers are 10.64 and 44.92 kB.

2. What the 80 Tests Pin Down

Test file	Tests	Time	One thing it pins down
core-engine	17	3.8 s	the optimizer's $\beta \approx 1.669$, $\gamma \approx 0.443$, error ≈ 4,122; Statistics' 95% intervals cover the truth about 95% of the time, with constant noise
core-pinn	17	83.5 s	gradients match finite differences; on the flu the neural net lands near β 1.67, γ 0.44
blueprint	15	0.5 s	a drawn SIR equals the built-in one; the Zika paper's six R_0 values
live-measles	10	2.4 s	a week still being counted is never scored; nothing new means nothing written; the log is never rewritten
pinn	6	67.4 s	the flu's neural net recovers known rates, inside a speed budget
fit	5	0.3 s	Nelder–Mead recovers made-up rates within 2%; an early cutoff widens the R_0 range
core-hare	4	0.8 s	a ten-year cycle; the fair comparison of the two predator–prey models
engine	4	0.1 s	S + I + R stays constant; a step of 0.1 day agrees with 0.001 to 0.5% of the peak
core-ebola	2	4 ms	doubling about every 4.6 weeks; the projection overshoots what happened

Counts and times from this run, with CI=true on a 2-core container: the two neural-net files take most of it. Each example is taken from the test names.



Read-only permissions, a 15-minute limit, and a newer push cancels an outdated run.

CI = continuous integration
SIR = susceptible, infectious, recovered

: .github/workflows/ci.yml : Its steps, in order

3. What It Can and Cannot Tell You

It Can Show You

How fitting works: from the counts back to the rates, by hand and three ways.

How sure a fit is, and how that shrinks when the data stop early.

Why a model that fits the past can still forecast badly.

Where every number in the case data came from.

It Cannot Tell You

What to do about a real outbreak: the models are deliberately simple, for learning.

Which route spread Zika on Moorea: these counts cannot separate them.

Final measles numbers: the newest week is always incomplete, and earlier weeks can change.

That the early forecast record is a pure test: forecasts before 19 October were rebuilt.

Found While Checking

If the updater can't look up when Johns Hopkins published, as when GitHub limits unauthenticated calls, it uses the current time. That can count the newest, still-counting week as complete. Rebuilding the file here hit exactly that: the record read 27 of 29 until the real date was supplied, then 24 of 26.

The updater never removes an entry, so such a forecast would stay, and would block the right one for that week. While the lookup fails, it also rewrites the file every 3 hours instead of stopping.

The hand-run build, live/cli.ts -- build, rebuilds the record only to 19 October plus today: publishing it would drop the live forecasts made in between.

The fly pages linked from the site are a separate project, with their own deploy and source; this deck does not cover them. They pair a housefly's body with a fruit fly's brain wiring, and say so.

The methods come from the research of Dr. Padmanabhan Seshaiyer and colleagues. This is an independent classroom tool, not an official product of those authors.