Applied Computational Mathematics Lab

EPIDEMIC MODELING · MODULE 7

Data-driven models

From observation models to machine-learning forecasts



6 EPISODES

1. Counts are not cases
2. Seasonality and reporting artefacts
3. ARIMA, SARIMA and state-space models
4. Nowcasting under reporting delay
5. Machine learning and deep learning forecasts
6. Large language models in epidemiology

Based on Chapter 7 of T. Lazebnik, Epidemic Modeling: Principles, Algorithms, and Applications (Academic Press, 2026)

acml.teddylazebnik.com

BEFORE WE START

Warm-up: true or false?

CHECK YOURSELF

(a) A dashboard's daily count is the number of new infections that day.

(b) Poisson noise is a safe default for daily case counts.

(c) If reports jump every Monday, transmission must rise over the weekend.

(d) The last few days on a dashboard usually look lower than they will finally turn out.

HOW TO USE THIS

Guess now; the module settles each one. Answers at the end.

WHY THIS MODULE

Until now the models produced the epidemic. Here the data come first: what counts really measure, how to forecast them, how to see the present despite late reports, and what machine learning and language models add.

BEFORE WE START

This module's map

PART 1

Reading the data

The observation model: noise, zeros, the weekly cycle and holidays.

PART 2

Forecasts and nowcasts

ARIMA, the rolling-origin backtest, state space models, reporting delays and nowcasting.

PART 3

Learning machines

Machine learning and deep learning forecasts, and language models as careful readers.

GOALS

By the end you should be able to describe how counts relate to infections, remove calendar artefacts, run an honest backtest, nowcast under reporting delay, and judge where machine learning and language models help.

ONE RULE

Every number in this deck comes from one **simulated surveillance system** with a known truth: we can check each method against what really happened. All values are made up to show mechanisms.

Applied Computational Mathematics Lab

PART 1

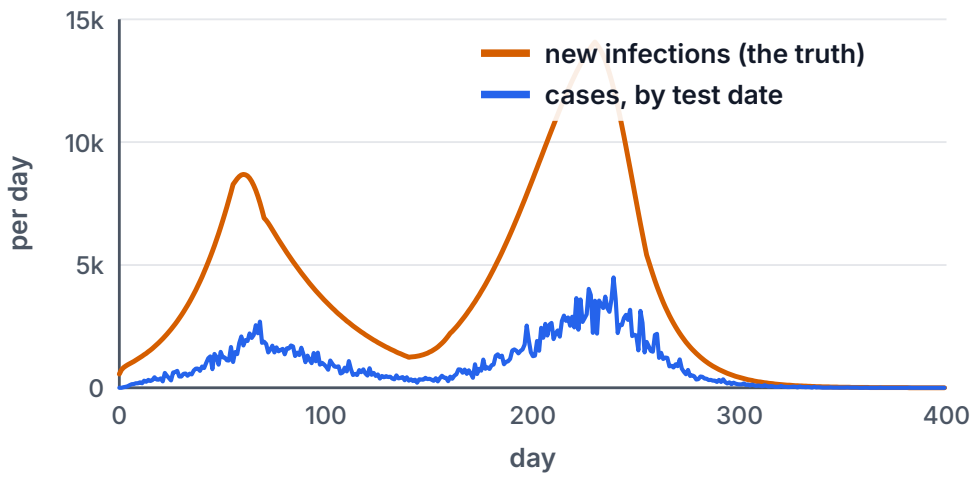
Reading the data

What a daily count really measures.



PART 1 · OBSERVATION

A surveillance system with a known truth



Our simulation: 1,476,599 infections and 360,085 cases in one year; two waves.

OUR SIMULATED SYSTEM (MADE UP)

population	5,000,000
transmission	an SEIR town; R(t) moves between 0.85 and 1.55
infections found	25% become cases
infection to test	gamma delay, mean 6 days
noise	negative binomial, k = 30
test to report	mean 2.8 days; 4.5 on test dates 200–239
reporting offices	Saturday 75%, Sunday 55%, rest on Monday; two holidays

IN PLAIN WORDS

Cases are a filtered, delayed and noisy view of infections. The observation model describes that filter.

PART 1 · OBSERVATION

The observation model

$$\mu_t = \rho \sum_{s \geq 0} g_s \, I_{t-s}, \qquad C_t \sim \mathrm{NegBin}(\mu_t, k), \qquad \mathrm{Var}(C_t) = \mu_t + \frac{\mu_t^2}{k}$$

- I_t: new infections on day t. ρ: the share that becomes a case (ascertainment, here 25%).
- g_s: the chance that a case is tested s days after infection (here a gamma distribution, mean 6 days).
- C_t: cases with test date t. k: the dispersion. Large k means close to Poisson; small k, much more spread.
- Reports by report date add one more layer: a reporting delay and the reporting calendar.

IN PLAIN WORDS

A model of the epidemic says what happens to infections. The observation model says what that looks like in the data. Fitting either one without the other confuses the two.

WHY IT MATTERS

A mechanistic model (Modules 2–6) predicts the infections; the observation model turns them into expected counts and their noise. Together they give a likelihood for the data, used for fitting in Module 8.

SCALE OF THE NOISE

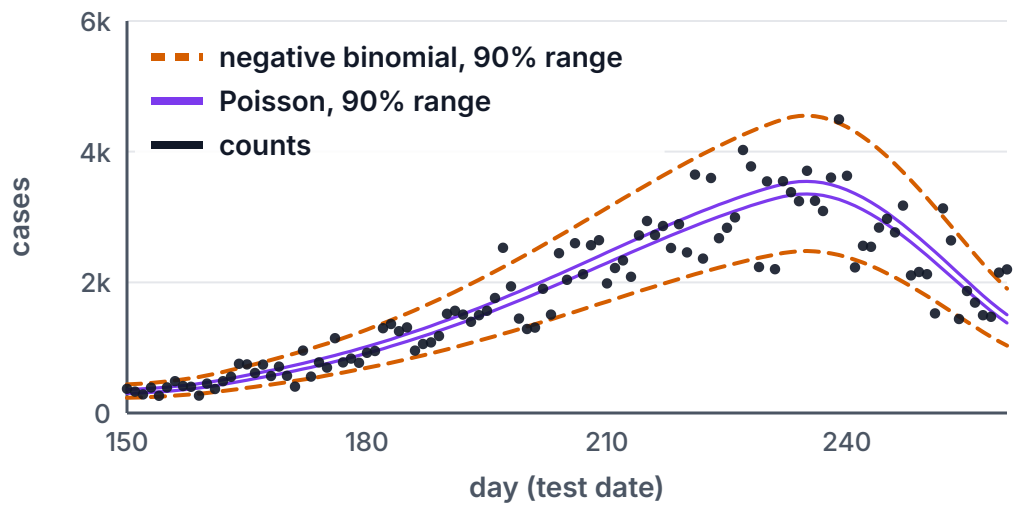
One day with 3,342 expected cases (day 230): Poisson spread is about ±58, negative binomial with k = 30 about ±613 (one standard deviation).

TRY IT

If only 10% of infections were found instead of 25%, which parts of the observation model change, and how would the counts look?

PART 1 · NOISE

Poisson or negative binomial?



Our counts on days 150–260 with each model's 90% range around the true expected count.

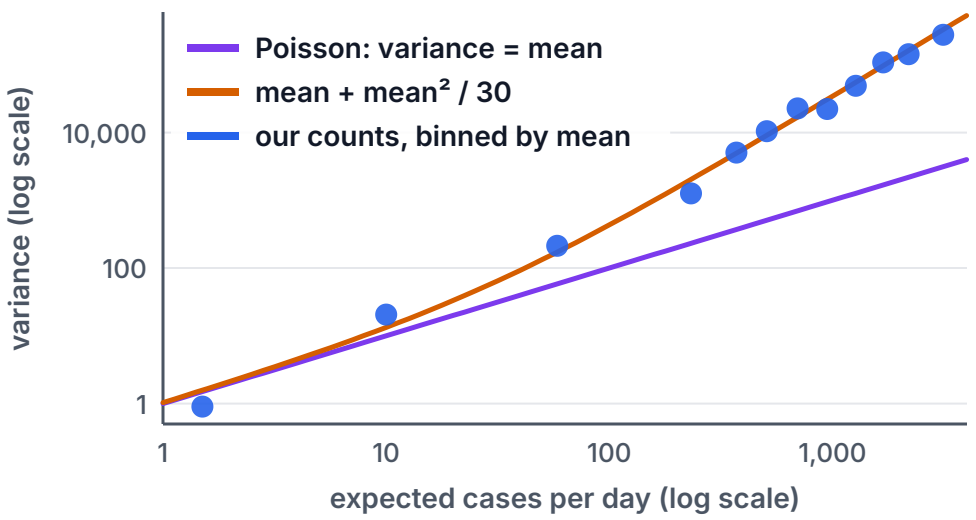
EXPECTED 2,000 CASES	90% RANGE
Poisson	1,927 to 2,074
negative binomial, k = 30	1,435 to 2,641

DAYS THE COUNT FELL INSIDE THE 90% RANGE	SHARE
Poisson ranges	40%
negative binomial ranges	90%

Our check over 398 days with at least half a case expected.

- Clusters, testing campaigns and batched results make real counts spread more than Poisson.
- So a jump of 200 cases on a day with 2,000 expected can be pure noise. Choose the observation model before reading meaning into the data.

Finding k, and why the log scale



Squared distance of each count from its true expected value, averaged in 12 bins of the expected value.

- Poisson is the straight line (variance = mean). Our counts bend upward. A method-of-moments fit of variance − mean = mean² / k gives k = 35; the true value is 30.
- In real data the true mean is unknown: use a smooth trend as the mean, or fit k with the rest of the model by maximum likelihood.
- On the log scale the noise is about $\sqrt{1/\mu + 1/k}$: for large counts it levels off near $1/\sqrt{k}$, about 18% here. A 10% wobble looks the same at 100 cases as at 10,000.

TRY IT

A count has mean 400 and k = 30. Compare its standard deviation with that of a Poisson count with the same mean.

PART 1 · NOISE

Zeros, and the full observation model

QUIET DAYS (EXPECTED COUNT BELOW 1)	
days	14
observed share of zero days	50.0%
Poisson predicts	51.7%
negative binomial predicts	52.1%

- A zero can mean no cases, a closed lab, or a missing report. These need different treatment.
- Before reaching for a zero-inflated model (an extra chance of a structural zero), check whether small means already explain the zeros. Here they do.
- Missing reports are not zeros: record them as missing, and let the model skip them (state space models do this naturally, Part 2).

LAYER 1

Ascertainment

What share of infections become cases; it changes with testing policy.

LAYER 2

Delays

Infection to test, test to report.

LAYER 3

Calendar

Weekdays, weekends and holidays in the reporting.

LAYER 4

Noise

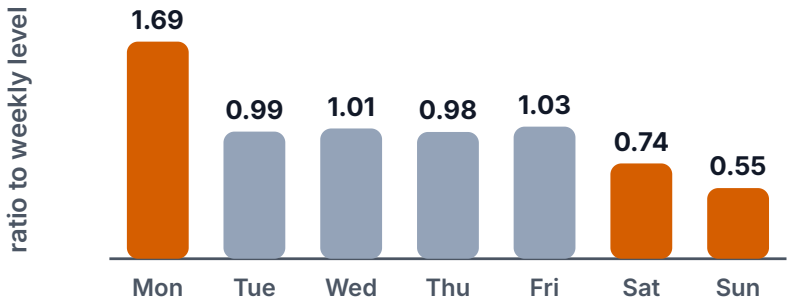
Poisson or negative binomial around the expected count.

TRY IT

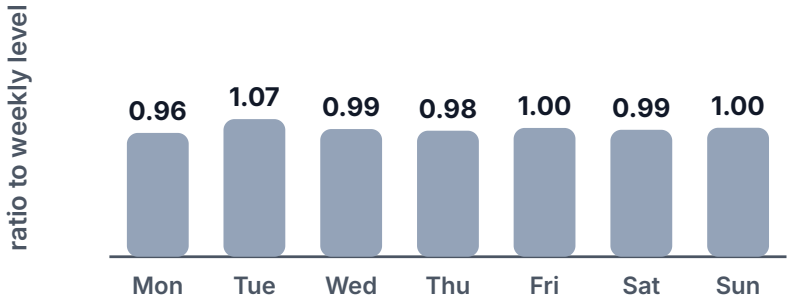
On 30 quiet days the expected count is 0.4 per day. About how many zero days does Poisson predict?

PART 1 · CALENDAR

The weekly reporting cycle



Reports by report date, divided by their centred seven day average (days 21–299).



The same cases counted by test date.

- Mondays carry 1.69 times the weekly level, Saturdays 0.74, Sundays 0.55. Monday alone brings about 24% of the week’s reports.
- A negative binomial regression with a smooth trend and one factor per weekday recovers the same effects from the counts alone: Mon 1.69, Tue 0.99, Wed 1.01, Thu 0.99, Fri 1.03, Sat 0.74, Sun 0.55.
- By test date every weekday lands within 7% of the weekly level: the pattern is made by reporting, not by transmission.

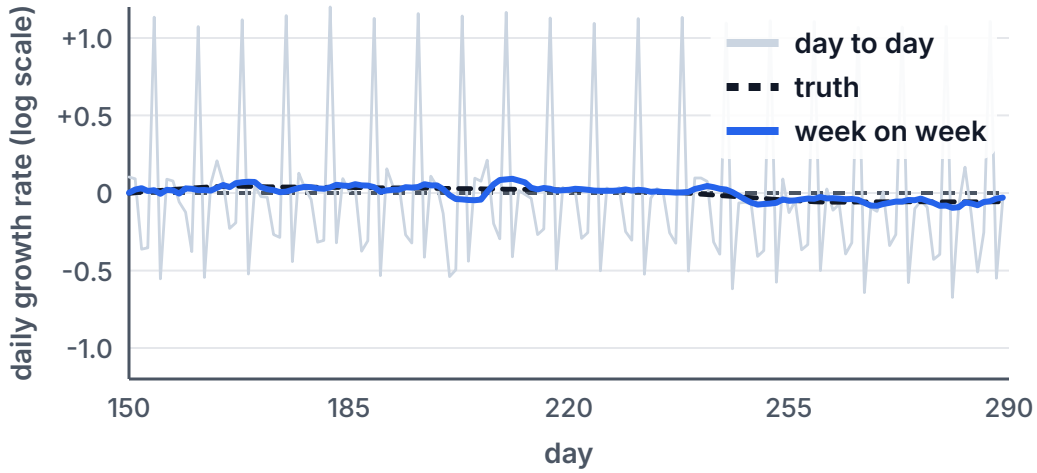
IN PLAIN WORDS

The virus does not keep a calendar; reporting offices do. When a count has a weekly rhythm, first ask which date it is counted by.

TRY IT

Mondays report 1.6 times the weekly level and Sundays 0.6 times. What growth from Sunday to Monday does the calendar alone create?

Daily growth: compare like with like



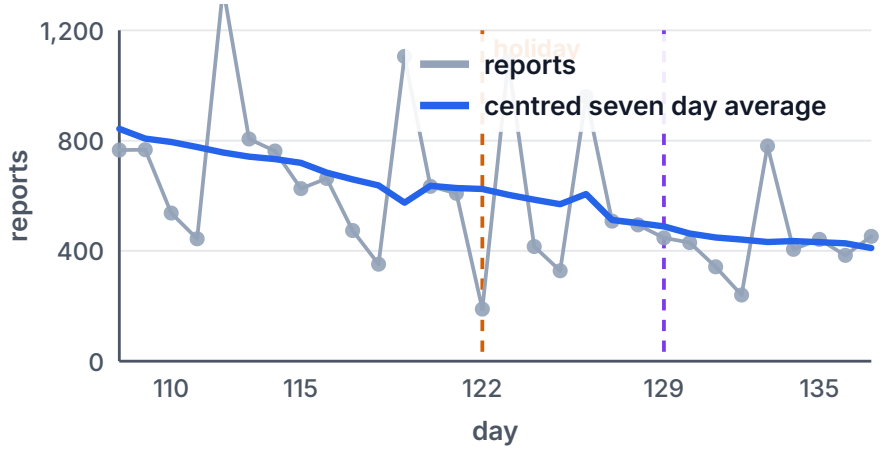
Our reports: growth from one day to the next (grey, clipped at ±1.2), from the same weekday a week earlier (blue), and the truth (dashed).

DAILY GROWTH RATE, DAYS 21–299	NOISE
day to day: log(today / yesterday)	±0.52
week on week: log(today / same day last week) / 7	±0.027 (error)

- A day-to-day wobble of 0.5 on the log scale is a factor of 1.65: enough to make a slow decline look like a 65% jump.
- The same trap waits for anything built on daily counts. A reproduction number estimated from raw reports rises every Monday and falls every weekend unless the weekday effect is removed first.
- Compare the same weekday, or use a seven day average, before reading growth.

PART 1 · CALENDAR

Holidays, and two tools



Our reports around the holiday on day 122, with the centred seven day average.

HOLIDAY	DAY BEFORE	HOLIDAY	DAY AFTER
122	609	189	1,089
303	164	44	250

- The seven day average barely notices: the dip and the catch-up cancel inside its window. Week-on-week comparisons do not: day 122 looks like a 70% drop against day 115, and day 129 like 2.4 times growth against the holiday.
- **Tool 1, decompose**: split the series into trend, weekly pattern and remainder (for example STL: seasonal-trend decomposition with local smoothing), then model the trend.
- **Tool 2, calendar terms in the model**: weekday and holiday indicators sit inside the regression or forecasting model, which learns them with the trend and carries them into its forecasts.

$$\log \mu_t = s(t) + w_{\text{weekday}(t)} + h\,1[t \text{ is a holiday}]$$

IN PLAIN WORDS

Either way, model the reporting process, so you do not mistake it for the epidemic.

Check yourself: Part 1

CHECK YOURSELF

1. A negative binomial count has mean 900 and k = 25. What are its variance and standard deviation? And for a Poisson count with the same mean?

2. Why does counting cases by test date instead of report date remove most of the weekly pattern in our system?

3. A reproduction number estimated from daily reports jumps every Monday. Give two fixes.

TIP

Answers are at the end of the deck. Try them before you look.

Applied Computational Mathematics Lab

PART 2

Forecasts and nowcasts

Learning from the past, and seeing the present.



ARIMA in three ingredients

- **Differencing (I)**: work with changes instead of levels; this removes a trend.
- **Auto-regression (AR)**: today's change depends on the last few changes.
- **Moving average (MA)**: today's surprise echoes for a few days.
- **Seasonal ARIMA** adds the same pieces at a lag of 7 days, for the weekly pattern.

$$\phi(B)\,(1-B)(1-B^7)\,z_t = \theta(B)\,(1+\Theta B^7)\,\varepsilon_t, \qquad z_t = \log(y_t+1)$$

B shifts back one day (B $z_t = z_{t-1}$); φ and θ are short polynomials in B with p and q terms.

ORDER (P, D, Q) × (0, 1, 1) WEEKLY	CHOSEN AT ORIGINS
(0, 1, 1)	2
(1, 1, 1)	6
(2, 1, 1)	4
(1, 1, 2)	5

Our fits: log counts over the last 120 days before each origin; order chosen by the AIC at each of 17 origins.

ORIGIN OF THE METHOD

Box and Jenkins (1970) set out the ARIMA approach: make the series stationary, identify the orders, fit, check the residuals, forecast.

TRY IT

Why fit on log counts rather than raw counts? Think about the noise you saw in Part 1.

PART 2 · BACKTEST

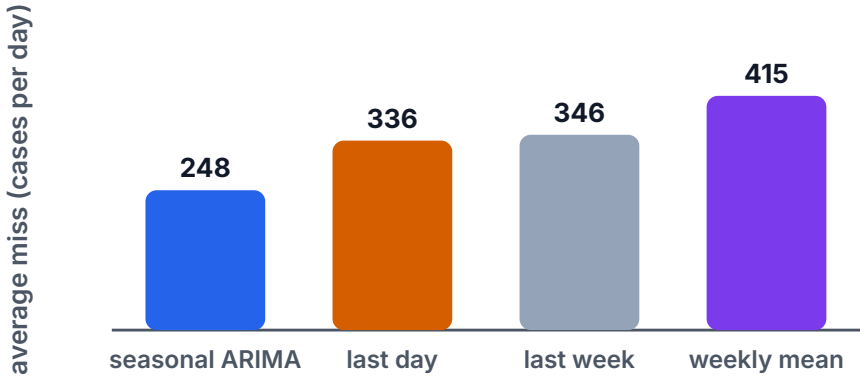
The honest test: rolling origins

A ROLLING-ORIGIN BACKTEST

1. pretend it is day 100: fit only on data before it
2. forecast the next 14 days, with intervals
3. compare with what happened; record the errors
4. move the origin forward 14 days and repeat (17 origins, days 100–324)
5. average the errors; check how often the intervals held the truth

IN PLAIN WORDS

A model must beat simple baselines before it earns any trust. And a forecast without honest intervals is half a forecast.



Our backtest: average count in the forecast windows 1,088; seasonal ARIMA missed by 23% of it.

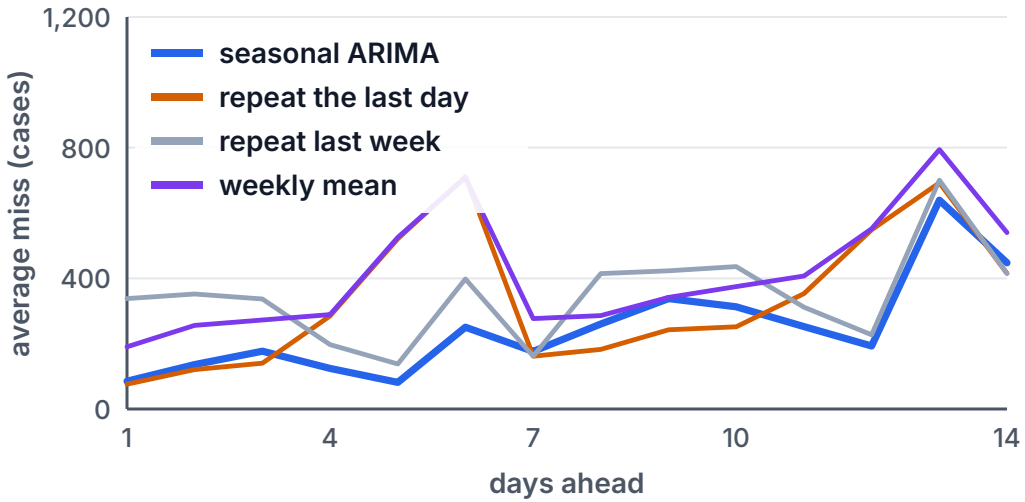
INTERVAL	HELD THE TRUTH
80%	84.9%
95%	94.5%

TRY IT

17 origins with 14 days each: how many forecast–outcome pairs feed the average?

PART 2 · BACKTEST

Errors grow with the horizon



Our backtest: average miss by days ahead, over the same 17 origins.

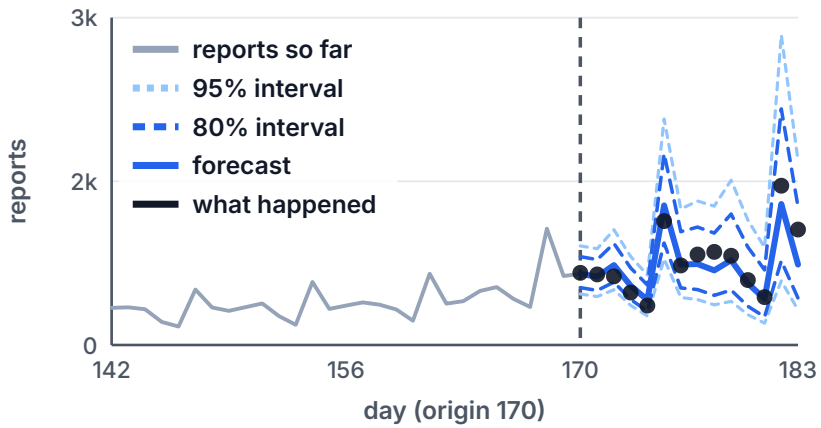
- One day ahead seasonal ARIMA missed by 85 cases on average; two weeks ahead by 448.
- Repeating the last day wins the first day ahead (76), then collapses whenever the last day was a Monday or a weekend day.
- Every origin falls on the same weekday, so the zigzag follows the weekly reporting cycle.
- Report errors by horizon: a single average hides that a model good at one day ahead may be poor at two weeks.

IN PLAIN WORDS

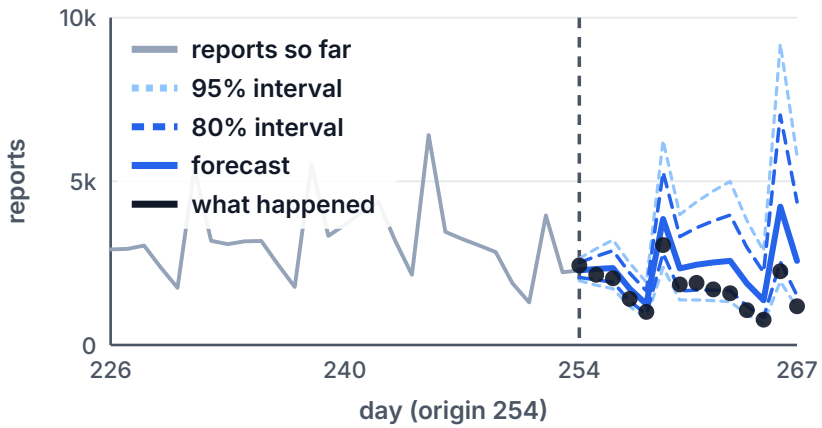
Intervals must widen with the horizon too. Our 95% intervals held the truth 95% of the time overall; check that by horizon as well.

PART 2 · BACKTEST

Two forecasts, drawn



Origin 170, early in the second wave: forecast 737 two weeks ahead; truth 1,058.



Origin 254, just after the second peak: forecast 2,571; truth 1,185.

- Linear models extend the recent trend. Their weak spot is turning points: the past cannot say that a wave is about to turn.
- A mechanistic model can know about depletion of susceptibles or a policy change; a purely statistical one cannot. Combining them is a theme of later modules.

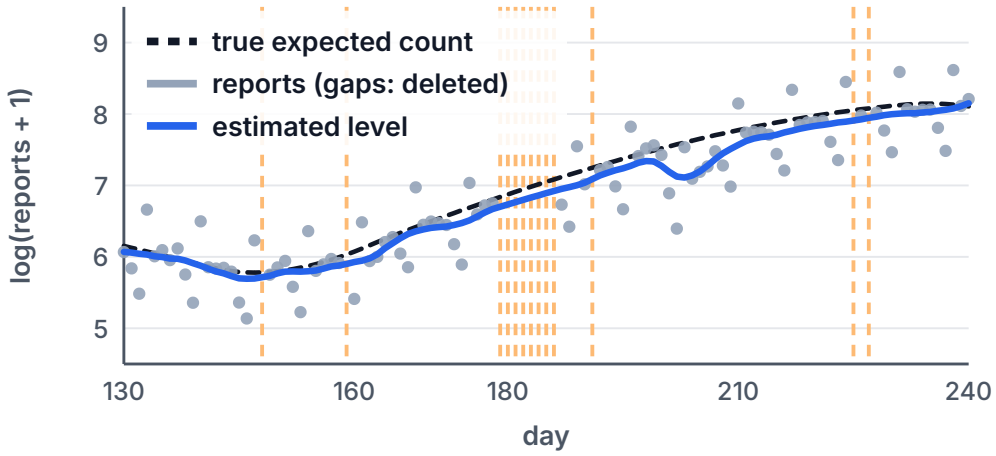
State space models and the Kalman filter

$$\ell_t = \ell_{t-1} + b_{t-1} + \eta_t, \quad b_t = b_{t-1} + \zeta_t, \quad z_t = \ell_t + w_{\text{weekday}(t)} + \varepsilon_t$$

- A hidden state (here the level ℓ and trend b of log counts, plus a weekday effect) evolves with noise; we see only noisy observations of it.
- The Kalman filter (Kalman, 1960) updates the state day by day: predict, then correct with the new observation, weighted by how noisy each part is.
- A missing day is simply a day with no correction: the filter predicts through it. The smoother then uses later days too.

IN PLAIN WORDS

ARIMA models can be written in state space form; the state space view makes missing data, changing noise and several data streams natural.

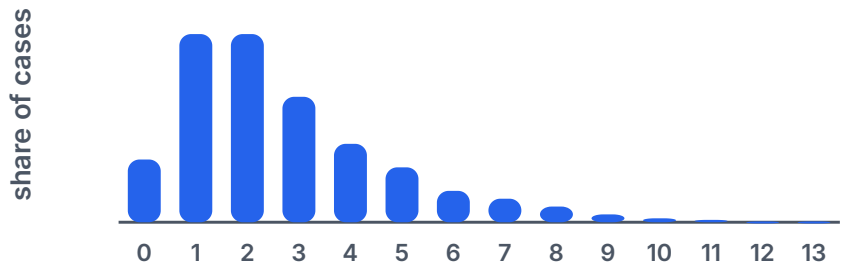


Our fit (local linear trend + weekday effect, days 20–299): 22 days deleted, 15 at random and days 180–186 (orange).

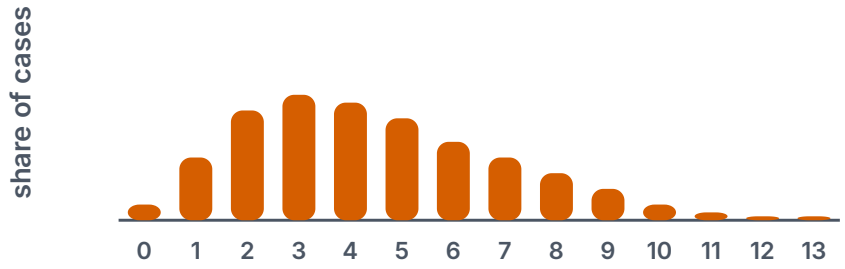
SMOOTHED LEVEL VS TRUTH (LOG SCALE)	ERROR
on the deleted days	0.137 (about 15%)
on all days	0.156

PART 2 · DELAYS

Reports arrive late



Usual delay from test to report: mean 2.8 days.



Slow labs (test dates 200–239): mean 4.5 days.

- Usually only 8% of a day's cases are reported on the test day itself, 56% within three days (delays 0–2). Seeing them all takes two weeks.
- Right truncation: the newest data are cut off before they are complete. The most recent days always look too low.
- Counting only what has arrived, the last seven test dates look 37% lower than they finally turn out, on average, every time.

IN PLAIN WORDS

A falling curve at the right edge of a dashboard is the most common false alarm, and the most common false comfort.

TRY IT

With the usual delays, what share of a test date's cases has arrived after one full day (delays 0 and 1)?

PART 2 · DELAYS

The reporting triangle

delay in days (reported on test date + delay)

test date	0	1	2	3	4	5	6	7	8	9	10	11	12	13
136	30	112	98	52	38	28	20	12	3	4	3	1	1	0
137	24	112	100	76	48	24	21	15	8	2	1	1	1	
138	21	84	68	41	32	24	9	13	12	0	3	0		
139	34	119	91	63	49	25	17	15	9	4	2			
140	16	64	74	53	30	25	11	6	6	2				
141	28	86	84	46	28	28	20	13	6					
142	16	66	75	42	30	11	14	7						
143	19	90	72	40	42	23	15							
144	24	43	43	25	24	13								
145	28	95	75	53	29									
146	28	91	84	41										
147	17	82	78											
148	36	93												
149	26													

Our reporting triangle as seen at the start of day 150: grey cells have not happened yet.

- One row per test date, one column per delay. Cell (t, d) counts cases tested on day t and reported d days later.
- On day T only the cells with t + d < T exist. The newest rows are incomplete: test date 149 shows 26 cases; two weeks later it had 352.
- Complete rows (older than the longest delay) tell us the delay distribution; incomplete rows tell us what has arrived so far.

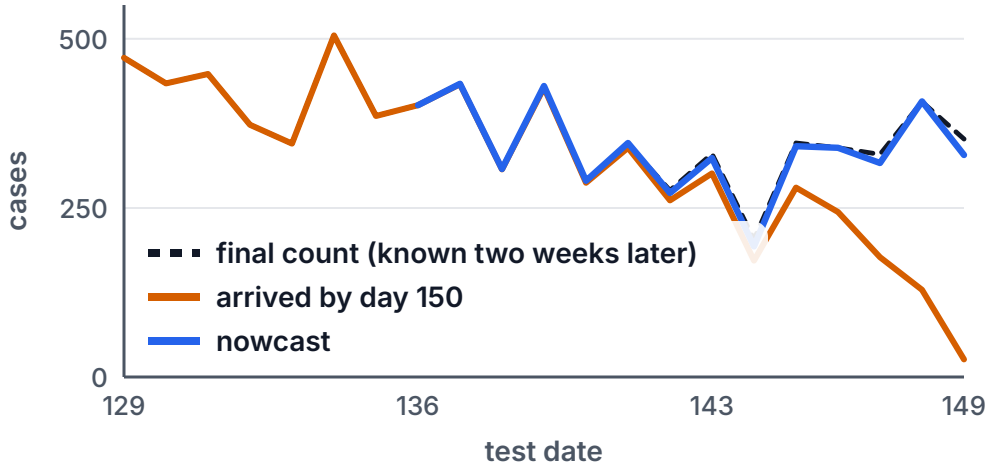
TRY IT

On day T, which cells of the row for test date T − 5 are visible? What share of its cases do you expect to have seen, with the usual delays?

Nowcasting fills in the triangle

$$\widehat{N}_t = \frac{\text{arrived}_t}{F(T-t)}, \qquad F(n) = \sum_{d<n} \hat{p}_d$$

- Estimate the delay distribution $\hat{p}$ from complete rows in a recent window (our version: the last 60 complete test dates).
- Scale each recent day up by the share of its cases you expect to have seen so far, F(T − t).
- Our simple version cut the error for the last seven days from 37% to 12% (average size, outside the weeks when delays changed).
- The remaining cases arrive by chance: the newest day, scaled up most, is the least certain. A good nowcast reports that uncertainty with the number.



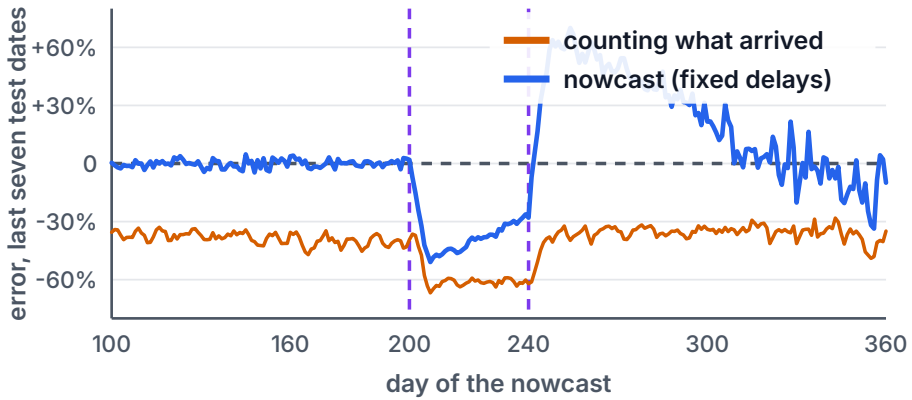
Our example at the start of day 150: what had arrived (orange), the nowcast (blue), and the final count (dashed).

ONE CELL, BY HAND

Test date T − 1 shows 26 cases; with 8% expected by then, the nowcast is 26 / 0.08, about 325. The final count was 352.

PART 2 · NOWCASTING

When the delays change



Our nowcast with delays estimated from the last 60 complete test dates, day by day. Between the purple lines (days 200–240) the labs were slow.

LAST SEVEN TEST DATES	USUAL WEEKS	SLOW LABS (DAYS 205–244)
counting what arrived	-37%	-61%
nowcast, average error	±12%	-34%

- When labs slow down, fewer cases have arrived than the old delays assume: the nowcast falls short just as the second wave grows.
- When the labs speed up again, the nowcast still uses the slow delays from its window and overshoots, by up to 70%.
- Watch the delays themselves, and estimate them from recent data.

GOING FURTHER

Bayesian nowcasting, such as NobBS (McGough et al., 2020), treats the whole triangle as data, lets the delays change smoothly over time, and returns an interval for each day.

Check yourself: Part 2

CHECK YOURSELF

4. In a backtest, a model's 80% intervals held the truth on 60% of forecast days. What does that tell you, and what would you change?

5. Delays put 8% of cases on the test day, 24% one day later and 24% two days later. At the start of day T, test date T − 2 shows 300 cases. Nowcast it.

6. Why must a backtest fit only on data before each origin, and why use many origins?

TIP

Answers are at the end of the deck.

Applied Computational Mathematics Lab

PART 3

Learning machines

Machine learning, deep learning and language models.



From a series to a table

EACH ROW (MADE AT AN ORIGIN T, FOR H DAYS AHEAD)	
14 daily lags	log cases on days T − 1 … T − 14, minus their weekly mean
level	the mean of log cases over the last 7 days
horizon	h, from 1 to 14
calendar	weekday of the target day (7 indicators)
target	log cases on day T − 1 + h, minus the weekly mean

- Every past origin gives 14 rows, one per horizon. Training uses only rows whose targets were already observed at the forecast origin.
- Subtracting the weekly mean lets one model learn shapes that hold at 100 or at 5,000 cases.

DIRECT

One model for all horizons

The horizon is a feature (our choice), or one model per horizon. No errors fed back.

RECURSIVE

One day at a time

Forecast a day, feed it back as a lag, repeat. Simple, but errors pile up from day to day.

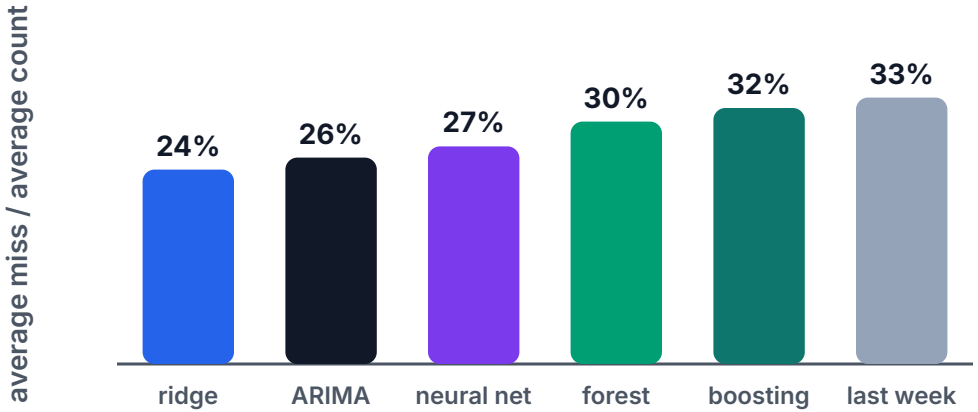
THE MAIN TRAP

Leakage: anything in the features that would not have been known at the forecast origin. Common forms: a random train/test split of days, smoothing with centred windows, and scaling with statistics of the whole series.

TRY IT

How many features does each row have?

Four learners against seasonal ARIMA

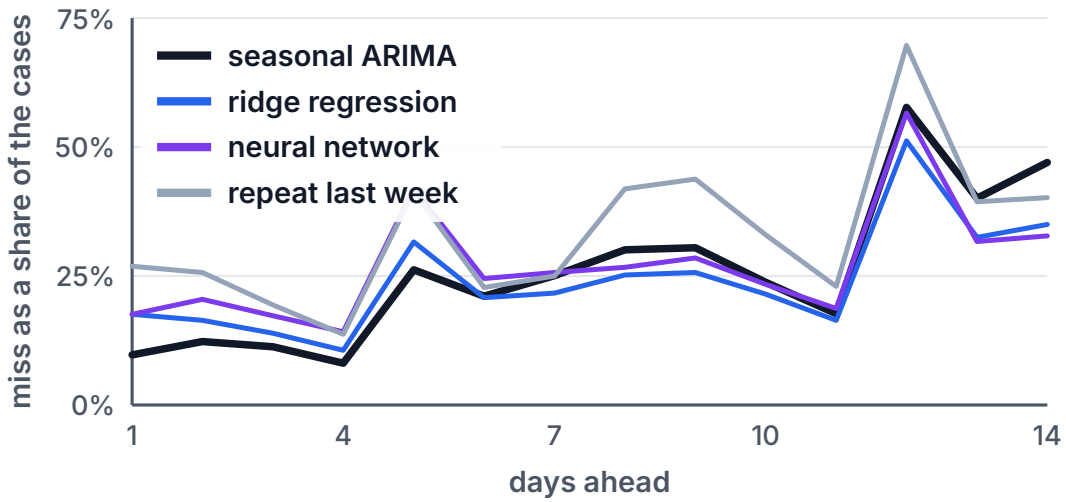


Our backtest: origins every 7 days, days 150–325 (26 two-week forecasts); average count 1,254.

LEARNER	SETTINGS (SCIKIT-LEARN)	MISS
ridge regression	penalty 1	24.3%
random forest	150 trees, 5 rows per leaf	30.3%
boosted trees	200 rounds, rate 0.05	32.0%
neural network	two layers of 32 and 16	27.2%
seasonal ARIMA	as in Part 2	25.8%
repeat last week	baseline	33.3%

- On one year of data the simplest learner did best, and the flexible ones barely beat repeating last week.
- Machine learning needs data: longer histories, more places, more signals. With one short series there is little to learn from.

By horizon and by phase



Our backtest: average miss by days ahead, as a share of the average count.

PHASE (FORECASTS)	SEASONAL ARIMA	RIDGE	NEURAL NETWORK	LAST WEEK
growing (10)	22%	21%	20%	29%
falling (12)	35%	47%	41%	76%
turning (4)	20%	18%	25%	16%

Each forecast's miss divided by its own window's average count, averaged by phase. Turning: the growth rate changes sign or by more than 0.25 per week.

- One day ahead seasonal ARIMA was best (10%); two weeks ahead the neural network (33%) and ridge (35%) beat it (47%).
- When cases fall toward small numbers, relative errors jump for every method.

Deep learning for time series

RECURRENT

RNN

A hidden state carried from day to day.

GATED

LSTM, GRU

Gates choose what to remember and what to forget (Hochreiter and Schmidhuber, 1997).

CONVOLUTIONAL

Temporal convolution

Filters look back through fixed windows of past days.

ATTENTION

Transformer

Attention weighs every past day for each prediction (Vaswani et al., 2017).

IN PLAIN WORDS

They shine with many series and many signals at once: regions, mobility, searches, hospital data. On one short series they overfit easily without careful regularisation and early stopping.

TEST ANY LEARNER LIKE A FORECASTER

1. only the past in training, at every origin
2. rolling origins that cover rises, peaks and declines
3. simple baselines to beat: last week, seasonal ARIMA
4. errors by horizon and by phase, not only one average
5. intervals checked against outcomes

WHY THE SPLIT MATTERS

A random split of days into training and test lets the model see the day before and the day after each test day. It looks excellent and forecasts badly.

TRY IT

Name one signal besides case counts that could help a deep model forecast, and one way it could mislead.

Language models as careful readers

- Much of what we learn about an outbreak first appears as text: bulletins, news, situation reports.
- A language model predicts the next word from the words before it. That lets it turn messy text into summaries, labels or structured records.
- The cautious use: not as the forecaster but as a reader that fills in a fixed form, and nothing else.

THE FORM (OUR EXAMPLE)

disease	…
place	…
report date	…
new confirmed cases	…
deaths	…
hospital pressure (low, medium, high, unknown)	…
source	…

WEAKNESS 1

Prompt-sensitive

Answers change with the wording of the request.

WEAKNESS 2

Unsupported facts

It can state things that are not in the text.

WEAKNESS 3

Drift

It changes when reporting language changes, or when the model is updated.

IN PLAIN WORDS

Never let its output flow straight into a model. Check it.

A toy check

- Our simulated reports for days 150–299 play the bulletins. On a share of days the reader slips: it drops a digit, adds one, or copies the cumulative total.
- The rule: flag a value above 4 times, or below a quarter of, the median of the last 14 accepted values, and send it to a person.
- No language model was run: the slips are simulated, to show what a simple check can catch.

READER SLIPS ON	2% OF DAYS	5% OF DAYS
seven day average off by more than 20%, unchecked	8.7%	21.9%
the same, with the rule and a person	0.4%	1.1%
slips caught	91.5%	94.3%
false alarms per 150 days	0	0

Our runs: 400 random trials per column.

SAFEGUARDS

1. a fixed form, and ask for nothing else
2. answer only from trusted sources given to it, not from memory
3. process the output with fixed code, not more prompting
4. simple plausibility checks, and a person for the hard cases
5. log every input and output so a mistake can be traced

IN PLAIN WORDS

They can also summarise papers or draft code. Asked for forecasts, they face the same rolling-origin test. They assist; people stay accountable.

TRY IT

Why compare with the median of recent values, and not the mean?

Check yourself: Part 3

CHECK YOURSELF

7. A random train/test split of days gave a forecast error of 5%; a rolling-origin backtest gave 30%. Which do you trust, and why?

8. Why might ridge regression beat a neural network on one year of daily counts?

9. A language model extracts daily counts and, on 3% of days, returns ten times the true value. Name two safeguards that limit the damage.

TIP

Answers are on the next slides.

Answers: warm-up and Parts 1–2

- **(a)** False: cases are a share (here 25%) of infections, about 6 days later, with noise. **(b)** False: Poisson ranges held only 40% of our counts. **(c)** False: the Monday jump is reporting catch-up. **(d)** True: right truncation; 37% low on average here.
- **1.** Variance $900 + 900^2 / 25 = 33{,}300$, standard deviation about 182; Poisson: variance 900, standard deviation 30.
- **2.** The weekly pattern comes from when offices report, not from when people are tested; by test date the weekend hold-back does not exist.
- **3.** Remove the weekday effect first (regression with weekday terms, or by test date), or estimate from seven day averages or week-on-week ratios.
- **4.** The intervals are too narrow (overconfident). Use a noisier observation model (negative binomial, log scale), check coverage by horizon, and widen until coverage matches.
- **5.** Delays 0 and 1 have passed: 32% arrived, so about $300 / 0.32 = 940$. **6.** Otherwise the future leaks into the fit; many origins cover rises, peaks and declines and give a stable average.

TRY-IT ANSWERS

Try-it answers: with 10% found, ρ falls and counts are smaller and relatively noisier. Mean 400, k = 30: standard deviation 76 against 20 for Poisson. $30 \times e^{-0.4}$, about 20 zero days. Sunday to Monday: 1.6 / 0.6 = 2.7 times. Log counts make the noise about the same size at every level. 17 × 14 = 238 pairs. After one full day: 8% + 24% = 32%. Row T − 5: delays 0 to 4 are visible, about 82% of its cases.

Answers: Part 3

- **7.** The rolling-origin result: a random split puts neighbouring days in training, so the model is tested on information it would never have had at the forecast origin.
- **8.** One year gives few rows and a strong, simple structure. A flexible model has many parameters to fit from little data and overfits; regularised linear models do well with little data.
- **9.** For example: a plausibility rule against the median of recent values, with a person for flagged days; a fixed form; answers only from trusted sources; logging every input and output.

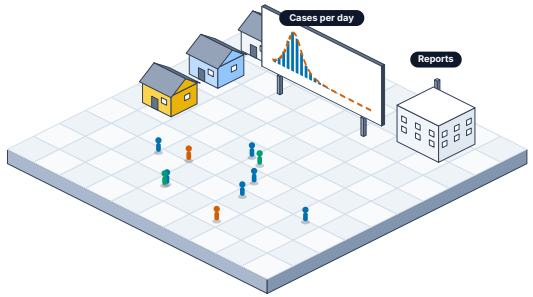
TRY-IT ANSWERS

Try-it answers: each row has 14 lags + 1 level + 1 horizon + 7 weekday indicators = 23 features. A helpful extra signal: hospital admissions or wastewater; it can mislead when testing or sampling policy changes. The median ignores a single wild value; the mean would be pulled by it and let the next slip pass.

CREDITS

Sources and credits

- This deck accompanies Chapter 7 of T. Lazebnik, Epidemic Modeling: Principles, Algorithms, and Applications (Academic Press, 2026), the course textbook. It follows the chapter's topics in our own words; no text, figures, tables, exercises or code are copied from the book.
- The text, figures, examples and simulations were written, drawn and run for ACML; nothing is copied from the book. Numbers labelled "our run" come from our own code; counts from agencies and papers are credited where they appear.
- All numbers and curves come from our own simulated surveillance system and our own fits (production/explains/sims/epi_m07_sims.py), with made-up values chosen to show mechanisms, not to describe any real pathogen, place or surveillance system. No language model was run for the toy check.
- Named works, described in our own words: ARIMA (Box and Jenkins, 1970); the Kalman filter (Kalman, 1960); STL decomposition (Cleveland et al., 1990); NobBS nowcasting (McGough et al., 2020); LSTM (Hochreiter and Schmidhuber, 1997); the transformer (Vaswani et al., 2017).
- Companion videos: **Epidemic Modeling**, module 7, episodes 26–31. More from the lab: acml.teddylazebnik.com



Applied Computational Mathematics Lab

END OF MODULE 7

Data-driven models

YOU COVERED

- Counts are not cases
- Seasonality and reporting artefacts
- ARIMA, SARIMA and state-space models
- Nowcasting under reporting delay
- Machine learning and deep learning forecasts
- Large language models in epidemiology

Module 7 complete



UP NEXT · MODULE 8

Parameter estimation and calibration

Fitting models to data, and knowing when you can't

Based on Chapter 7 of T. Lazebnik, Epidemic Modeling: Principles, Algorithms, and Applications (Academic Press, 2026)