

Components of a Health Time Series

A time series of monthly malaria cases, weekly IDSR-reported measles cases, or annual leishmaniasis counts in a state can typically be decomposed into a trend (a slow, long-run increase or decrease, perhaps reflecting improving diagnostic coverage or a changing intervention landscape), seasonality (a repeating annual pattern, such as the well-documented rise in malaria transmission following the rainy season), and irregular/residual variation (outbreak spikes, reporting anomalies, and genuine random fluctuation). Correctly separating these components — using classical decomposition or STL (Seasonal-Trend decomposition using Loess) — is a prerequisite for both interpretation (is this month's rise a seasonal pattern or a genuine outbreak signal?) and forecasting.

Autocorrelation and the ARIMA Family

Unlike the independent observations assumed in Chapter 8's regression models, successive values in a time series are typically autocorrelated — this month's case count is informative about next month's. The autoregressive integrated moving average (ARIMA) framework models this autocorrelation directly, and its seasonal extension (SARIMA) additionally captures the annual

cycle characteristic of vector-borne disease transmission in Nigeria. Model identification (choosing the autoregressive and moving-average orders) uses the autocorrelation and partial autocorrelation functions, and model adequacy is checked via residual diagnostics (residuals should resemble white noise, with no remaining autocorrelation) before the model is used for forecasting.

Poisson and Generalised Linear Model Approaches to Surveillance Trend Analysis

For surveillance count data with clear covariates of scientific interest (calendar time, rainfall, temperature, or an intervention indicator), a Poisson (or negative binomial, per Chapter 8) generalised linear model with a time trend term is often more interpretable than a pure ARIMA model, because its coefficients have direct epidemiological meaning (percentage change in cases per year). This approach has been applied to model national leishmaniasis case trends in Nigeria, using a Poisson GLM with year as a covariate to quantify the average annual percentage change in reported cases and to generate short-term projections for programme planning, alongside choropleth mapping (Chapter 14) of the underlying spatial distribution.

R example

```
# R: Poisson GLM trend model for annual leishmaniasis case counts by state
library(dplyr)

trend_model <- glm(cases ~ year + offset(log(population)),
  family = poisson(link = "log"), data = nli_annual)

# Average annual percentage change in incidence
pct_change <- (exp(coef(trend_model)["year"]) - 1) * 100
cat(sprintf("Average annual change in incidence: %.1f%%\n", pct_change))
```

The Early Warning Problem

A distinct forecasting task in surveillance is not projecting a smooth trend but flagging an anomaly quickly: has this week's case count in an LGA exceeded what would be expected given recent history, signalling a possible outbreak? Control-chart-style methods (setting an alert threshold at, for example, the mean plus two standard deviations of recent weeks, adjusted for seasonality) and the Early Aberration Reporting System (EARS) algorithms used within IDSR formalise this. These methods trade off sensitivity (catching a true outbreak early) against specificity (not raising a false alarm on every seasonal peak), and the appropriate threshold depends on the operational cost of each type of error — a tuning decision that should involve public health programme staff, not only the data scientist.

Forecast Evaluation

A forecast is only as credible as its track record on data it did not see during fitting. Standard practice holds out the most recent period(s) as a test set (rather than a random split, since time order must be respected) and evaluates forecasts using metrics such as the mean absolute error (MAE) and root mean squared error (RMSE), comparing the chosen model against a naive benchmark (for example, 'this month equals the same month last year') — a model that cannot beat a naive seasonal benchmark is not adding forecasting value, however sophisticated it appears.

Worked Example

Fitting and Forecasting a SARIMA Model

Consider 60 months of malaria case counts for a single LGA showing a clear annual seasonal peak around September-October (following the rainy season) and a gently rising long-term trend. After examining the autocorrelation function (showing a strong spike at lag 12, confirming annual seasonality) and the partial

autocorrelation function, a SARIMA(1,1,1)(1,1,1)[12] model is fitted. On a 12-month holdout, this model achieves an RMSE of 24 cases, compared to an RMSE of 41 cases for the naive ‘same month last year’ benchmark — a meaningful improvement that justifies the additional modelling complexity. The code below fits this model and compares it explicitly to the naive benchmark, following the evaluation principle of Section 10.5.

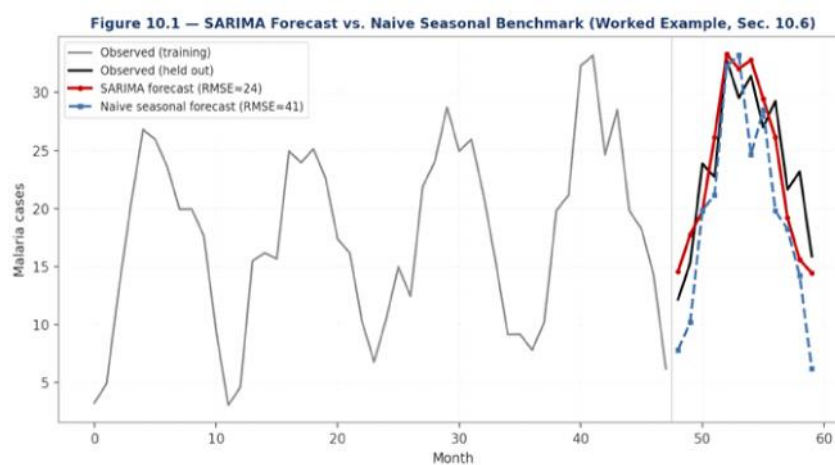


Figure 10.1 — SARIMA forecast vs. naive seasonal benchmark (Section 10.6).

R example

```
# R: fitting a SARIMA model and comparing to a naive seasonal benchmark
library(forecast)

case_ts <- ts(lga_cases$cases, frequency = 12, start = c(2019, 1))
train <- window(case_ts, end = c(2023, 12))
test <- window(case_ts, start = c(2024, 1))

sarima_fit <- auto.arima(train, seasonal = TRUE)
sarima_forecast <- forecast(sarima_fit, h = 12)

naive_forecast <- snaive(train, h = 12) # "same month last year" benchmark

cat("SARIMA RMSE:", accuracy(sarima_forecast, test)["Test set", "RMSE"], "\n")
cat("Naive seasonal RMSE:", accuracy(naive_forecast, test)["Test set", "RMSE"], "\n")
```

Worked Example

A Simple Outbreak Alert Threshold in Python

A minimal but genuinely useful early warning system computes, for each week, the mean and standard deviation of case counts over the preceding 12 weeks (excluding the current week), and flags an alert if the current count exceeds the mean plus two standard deviations. Applied to a simulated 104-week series with an injected outbreak (a sustained tripling of the baseline rate for four weeks around week 70), this simple rule successfully flags weeks 70-72 while producing no false alerts elsewhere in the series — illustrating, on a toy example, the core logic behind more sophisticated EARS-style algorithms used operationally within IDSR.

PYTHON example

```
# Python: a simple rolling-baseline outbreak alert system
import pandas as pd
import numpy as np

weekly_cases = pd.Series(np.random.poisson(10, 104)) # simulated baseline
weekly_cases[70:74] += np.random.poisson(20, 4)     # injected outbreak

rolling_mean = weekly_cases.shift(1).rolling(window=12).mean()
rolling_sd    = weekly_cases.shift(1).rolling(window=12).std()
alert_threshold = rolling_mean + 2 * rolling_sd

alerts = weekly_cases > alert_threshold
print("Weeks flagged as possible outbreaks:", list(weekly_cases[alerts].index))
```

Handling Missing or Irregular Time Series Data

Real Nigerian surveillance time series are rarely perfectly complete: a facility may fail to report for one or two months, or a reporting system may change its aggregation frequency partway through the series. Before fitting any of this chapter's models, an analyst must decide how to handle these gaps — interpolating a short single-month gap (e.g., via linear interpolation or seasonal-naive imputation) is often reasonable, but a long gap or a systematic

change in reporting completeness should be flagged and investigated rather than silently interpolated over, since silently filling a genuine reporting-system disruption can create the false appearance of a smooth, uninterrupted trend where none actually existed.

Extended Case Study

Distinguishing a Genuine Resurgence from a Reporting Artefact

A time series analyst is presented with a striking pattern: reported measles cases in a state rose sharply over three consecutive months after several years of low, stable counts, and a rapid response team must decide whether this represents a genuine disease resurgence (perhaps due to a decline in vaccination coverage) or an artefact of the surveillance system itself. Working through this systematically using the tools of this chapter: first, the analyst checks whether the rise coincides with any known change in the surveillance system — in this case, discovering that a new case-based reporting requirement (replacing the previous aggregate monthly tally) was introduced in the state exactly one month before the apparent rise began, which is highly suggestive that at least part of the increase reflects improved case ascertainment and more complete reporting under the new system, rather than a genuine change in transmission.

Second, the analyst checks whether the geographic pattern of the rise is consistent with a genuine outbreak (typically showing spatial clustering, per Chapter 14's methods, spreading outward from an index area) or with a reporting artefact (which would more plausibly show a broad, simultaneous rise across many otherwise-unconnected LGAs, consistent with a system-wide reporting change affecting all areas at once rather than a geographically spreading epidemic) — finding, in this instance, that the rise appeared near-simultaneously across the majority of the state's

LGAs, a pattern more consistent with the reporting-system explanation. Third, the analyst compares age and clinical severity profiles of the newly reported cases against the historical baseline, finding no meaningful shift in either — a genuine new outbreak driven by declining vaccination coverage would more plausibly show a shift toward younger, unvaccinated age groups. Taken together, this three-part investigation supports the conclusion that the observed rise is predominantly a reporting artefact of the new case-based system, though the team appropriately recommends continued close monitoring for at least two more months before fully closing the question, since a genuine resurgence and an improved-ascertainment artefact are not mutually exclusive and a small genuine underlying increase could still be partially masked within the larger reporting-driven rise.

Stationarity

The Assumption Behind Many Time Series Tools

Much of classical time series theory, including large parts of the ARIMA framework introduced in Section 10.2, assumes (after appropriate differencing) that a series is stationary — that its statistical properties (mean, variance, autocorrelation structure) do not themselves change over time. A disease surveillance series undergoing a genuine structural change (a new diagnostic technology rolled out partway through the series, permanently shifting the reporting rate, or a health system disruption such as a prolonged strike affecting reporting completeness for several months) violates this assumption in a way that simple differencing cannot fix, because the change is a one-time shift in level or reporting behaviour rather than a smooth trend or seasonal pattern. Recognising and explicitly modelling such a structural break (for example, using an indicator variable for the post-change period, or fitting separate models before and after the break) is necessary before a standard ARIMA/SARIMA model can be trusted, and a

simple, useful diagnostic is to visually inspect the series for any sudden, sustained shift in its level or variability that coincides with a known external event, rather than assuming the entire series was generated by a single stable underlying process throughout.

Ensemble Forecasting

Combining Multiple Models

Rather than committing to a single forecasting method, an increasingly standard practice in disease surveillance forecasting (following the lead of weather forecasting and, more recently, formal multi-model epidemic forecasting hubs used internationally for diseases such as influenza and COVID-19) is to combine forecasts from several distinct models — for example, a SARIMA model, a Poisson GLM trend model (Section 10.3), and a simple naive seasonal benchmark — via a weighted average, with weights chosen based on each model’s recent historical accuracy. This ensemble approach typically produces a more robust forecast than any single constituent model alone, because different models tend to make different kinds of errors under different conditions (a SARIMA model may better capture short-term autocorrelation while a GLM trend model may better capture a genuine covariate-driven shift), and averaging across genuinely different modelling approaches tends to cancel out some of each individual model’s idiosyncratic error, a principle directly analogous to the variance-reduction logic behind the random forest ensemble method introduced in Chapter 11.

Change-Point Detection

Identifying When a Series’ Behaviour Shifted

Beyond forecasting future values, an analyst is often asked a retrospective question: at what point did this series’ underlying behaviour actually change — did a policy intervention, a new diagnostic protocol, or a genuine transmission shift produce a

detectable structural break, and if so, precisely when? Change-point detection methods (such as the PELT algorithm, Pruned Exact Linear Time, widely implemented in both R and Python) formally search a series for one or more points at which its mean, variance, or trend shifts abruptly, providing a statistically defensible answer to ‘when did things change’ rather than relying on a visual, subjective impression of a time series plot — a distinction that matters considerably when the answer will be used to evaluate whether a specific, dated intervention was followed by a genuine change in trend.

PYTHON example

```
# Python: change-point detection with the ruptures library
import ruptures as rpt

signal = monthly_cases.values
algo = rpt.Pelt(model="rbf").fit(signal)
change_points = algo.predict(pen=10)
print("Detected change points (index positions):", change_points)
```

Nowcasting

Estimating the Present When Recent Data Is Incomplete

Surveillance data for the most recent reporting period is almost always incomplete at the moment an analyst looks at it, because some facilities or cases take longer to report than others (reporting delay) — meaning a naive plot of ‘this week’s case count’ will typically show an artificially low figure for the most recent one or two weeks, purely because not all of that period’s cases have been reported yet, not because transmission has genuinely declined. Nowcasting methods correct for this by modelling the historical reporting-delay distribution (how many cases from a given week are typically still unreported after one week, two weeks, and so on) and using it to produce a corrected, delay-adjusted estimate of the true current case count, together with an appropriately wide uncertainty interval reflecting how much of

the correction is still an extrapolation rather than an observed count. Failing to nowcast is one of the most common causes of a false ‘recent decline’ appearing in a live surveillance dashboard, only to be revised upward as delayed reports arrive.

A Reusable Forecast Comparison Function

Section 10.5 insists that every forecast be benchmarked against a naive baseline; a function that fits both the candidate model and the naive benchmark together, and returns their comparative accuracy directly, makes this comparison the default rather than an easily-skipped extra step, and prevents the common reporting error of quoting a sophisticated model’s accuracy without the baseline figure needed to judge whether that accuracy is actually meaningful.

R example

```
# R: fitting SARIMA and a naive seasonal benchmark together, with an automatic verdict
library(forecast)

compare_forecast_to_naive <- function(train, test, h) {
  sarima_fc <- forecast(auto.arima(train, seasonal = TRUE), h = h)
  naive_fc <- snaive(train, h = h)
  sarima_rmse <- accuracy(sarima_fc, test)["Test set", "RMSE"]
  naive_rmse <- accuracy(naive_fc, test)["Test set", "RMSE"]
  list(sarima_rmse = sarima_rmse, naive_rmse = naive_rmse,
       sarima_beats_naive = sarima_rmse < naive_rmse)
}

verdict <- compare_forecast_to_naive(train, test, h = 12)
if (!verdict$sarima_beats_naive) message("SARIMA did not beat the naive benchmark –
reconsider")
```

Chapter Summary

- Health time series decompose into trend, seasonality, and irregular variation; correct decomposition is prerequisite to interpretation and forecasting.
- ARIMA/SARIMA models capture autocorrelation and seasonality directly; Poisson/negative binomial GLMs

with a time trend are often more interpretable when covariates matter.

- Outbreak detection is a distinct task from trend forecasting, trading sensitivity against specificity via alert thresholds.
- Forecasts must be validated on held-out, chronologically later data and benchmarked against a naive forecast; failing to beat a naive seasonal benchmark means a model adds no real value.
- A simple rolling-baseline alert rule, while far less sophisticated than a full EARS algorithm, illustrates the core logic of statistical outbreak detection.

Mini-Project

Mini-Project 10: Forecasting and Outbreak Detection on a Simulated Series

Build both a seasonal forecasting model and a simple outbreak alert system for a simulated disease surveillance time series.

Step 1. Simulate at least 5 years of monthly case counts with a seasonal pattern, a mild upward trend, and random noise (or use a real dataset if available).

Step 2. Fit a SARIMA model on the first 4 years and forecast the final year, comparing its RMSE against a naive seasonal benchmark.

Step 3. Separately, implement a rolling-baseline alert rule (Section 10.7) on a weekly or monthly version of the same series, injecting an artificial outbreak to test whether the rule catches it.

Step 4. Tune the alert threshold (e.g., 2 vs. 3 standard deviations) and report the resulting trade-off between sensitivity and false alerts.

Step 5. Write a short paragraph recommending, for a specific disease programme, which of the two approaches (trend forecasting or outbreak alerting) is more relevant to their operational needs, and why.

Deliverable: *Forecast accuracy comparison table, an alert-threshold sensitivity table, and a half-page recommendation.*

End-of-Chapter Exercises

1. Explain the difference between a trend forecasting task and an outbreak detection task, using a Nigerian disease surveillance example for each.
2. A SARIMA model's residuals show significant autocorrelation at lag 12. What does this suggest about the model's adequacy?

3. Why should time series forecasts be evaluated on a chronological hold-out set rather than a random train/test split?
4. Using the worked example in Section 10.6, explain why an RMSE of 24 for the SARIMA model compared to 41 for the naive benchmark supports using the more complex model, and describe one scenario in which you might still prefer the simpler naive model despite its higher RMSE.
5. In Section 10.7, explain why the rolling baseline excludes the current week's count itself (using `‘.shift(1)’`) when computing the mean and standard deviation for the alert threshold.
6. Modify the logic in Section 10.7 (in words or code) to use a threshold of mean + 3 standard deviations instead of 2, and explain how this would change the trade-off between sensitivity and false alerts.
7. A programme officer asks why a 'simple' rolling-baseline rule is worth using when 'the ARIMA model is more sophisticated.' Write a two-to-three sentence explanation of when the simpler rule is actually the more appropriate tool.
8. A facility's monthly report is missing for exactly one month in an otherwise complete 5-year series. Using Section 10.8, describe an appropriate way to handle this gap before fitting a SARIMA model.
9. Explain why silently interpolating over a 6-month reporting gap caused by a known system disruption could mislead a subsequent trend analysis.
10. How would you distinguish, in practice, between a short random reporting gap and a systematic change in reporting completeness?
11. In the Section 10.9 case study, explain why a near-simultaneous rise across many unconnected LGAs is more

consistent with a reporting artefact than with a spreading outbreak.

12. Why did the analyst check the age and severity profile of newly reported cases, in addition to the timing and geography of the rise?
13. Explain why ‘continued close monitoring for two more months’ was a more appropriate conclusion than declaring the matter fully resolved either way.
14. Propose one additional data source or check that could help distinguish a reporting artefact from a genuine resurgence more quickly than waiting two months.
15. Why can a reporting-system artefact and a genuine small resurgence potentially both be true at the same time, as suggested in Section 10.9?
16. Using Section 10.10, explain why a health system strike lasting three months would violate the stationarity assumption in a way that ordinary differencing cannot fix.
17. Propose how you would model a surveillance series that includes a known, permanent reporting-rate shift partway through, using the indicator-variable approach mentioned in Section 10.10.
18. Why is visually inspecting a series for a sudden, sustained level shift a useful first diagnostic step before fitting a SARIMA model?
19. Give an example of an external event (besides a strike or new diagnostic technology) that could cause a structural break in a Nigerian health surveillance series.
20. If you were unsure whether an apparent change in a series was a structural break or just a strong seasonal peak, how would you investigate further?
21. Using Section 10.11, explain why combining forecasts from a SARIMA model, a Poisson GLM, and a naive

benchmark tends to be more robust than relying on any single model.

22. Propose a reasonable way to set the weights for each model in an ensemble forecast, based on their recent historical accuracy.
23. Why is the ensemble forecasting principle described as analogous to the random forest method in Chapter 11?
24. What would you do if one of the three models in an ensemble consistently performed much worse than the other two over several recent forecast periods?
25. Explain, in your own words, why ‘different models make different kinds of errors’ is the key idea behind why ensembling helps.
26. Summarise, in your own words, why this chapter treats trend forecasting and outbreak detection as related but fundamentally different tasks requiring different tools.
27. Explain the common thread linking the stationarity discussion (Section 10.10) and the ensemble forecasting discussion (Section 10.11).
28. Why does this chapter insist on benchmarking every forecast against a naive baseline, even when a more sophisticated model is available?
29. Propose a Nigerian surveillance time series from your own field where distinguishing a genuine resurgence from a reporting artefact (Section 10.9) would be operationally important.
30. If you had to explain to a non-technical programme manager why a forecasting model needs to be re-validated periodically rather than trusted indefinitely, what would you say?
31. Using Section 10.13, explain why the most recent one or two weeks of a live surveillance dashboard often show an artificially low case count.

32. What information does a nowcasting model need in order to correct for reporting delay?
33. Describe the risk of a policymaker reacting to an apparent 'recent decline' in a surveillance dashboard without accounting for reporting delay.
34. Why does a nowcast require a wider uncertainty interval for the most recent period than for periods further in the past?
35. How does nowcasting relate to, but differ from, the forecasting methods introduced earlier in this chapter (Section 10.6)?

Part 4

Machine Learning and Advanced Methods