Time Series Analyses for Social Data

Yasin Shafi

18 February 2026

Start Off: Verifying Your Environment

1. **Environment check (required).** Submit proof that you successfully ran the full tutorial code on your machine. You may submit *one* of the following:
   - A screenshot or text file showing console output that includes the printed representation shapes (document–term matrix, Word2Vec document vectors, and transformer embeddings).

   - A screenshot of your `figures/` directory showing generated plots with timestamps.
   - A Git commit (hash or screenshot) that includes at least one generated figure or output file.
   - A short log file (e.g., `run_log.txt`) containing printed diagnostics and evaluation metrics.

Conceptual Questions

Please write a three to ten sentence explanation for **ONE** of the two questions below.

2. **Time leakage and honest evaluation.** In the coding tutorial, you computed RMSE under (i) a random train/test split and (ii) a time-based split (train on past, test on future).
   - Explain what *time leakage* is and why random splits usually inflate model performance for time-indexed data.
   - Explain why "train on past, test on future" is the default evaluation rule.
   - Describe what *rolling-origin* (backtesting) adds beyond a single time split.

Answer: Time leakage occurs when a model is trained on data that includes information from the future relative to the prediction point, which artificially inflates performance because the model "knows" outcomes it would never have access to in real deployment. Random train/test splits violate temporal ordering, so observations from later time periods end up in the training set and earlier ones in the test set, making the evaluation optimistic and misleading. "Train on past, test on future" is the default rule because it mirrors actual deployment conditions - in production, a forecasting model always predicts forward from the data it was trained on, so evaluation must respect that same direction. A single time split still has a limitation: it produces only one estimate of out-of-sample error from one particular cutoff point, which may be unrepresentative if that period happens to be unusually easy or hard to predict. Rolling-origin backtesting addresses this by repeatedly sliding the training window forward and generating many out-of-sample predictions across different time periods, giving a more reliable and stable estimate of true forecasting performance.

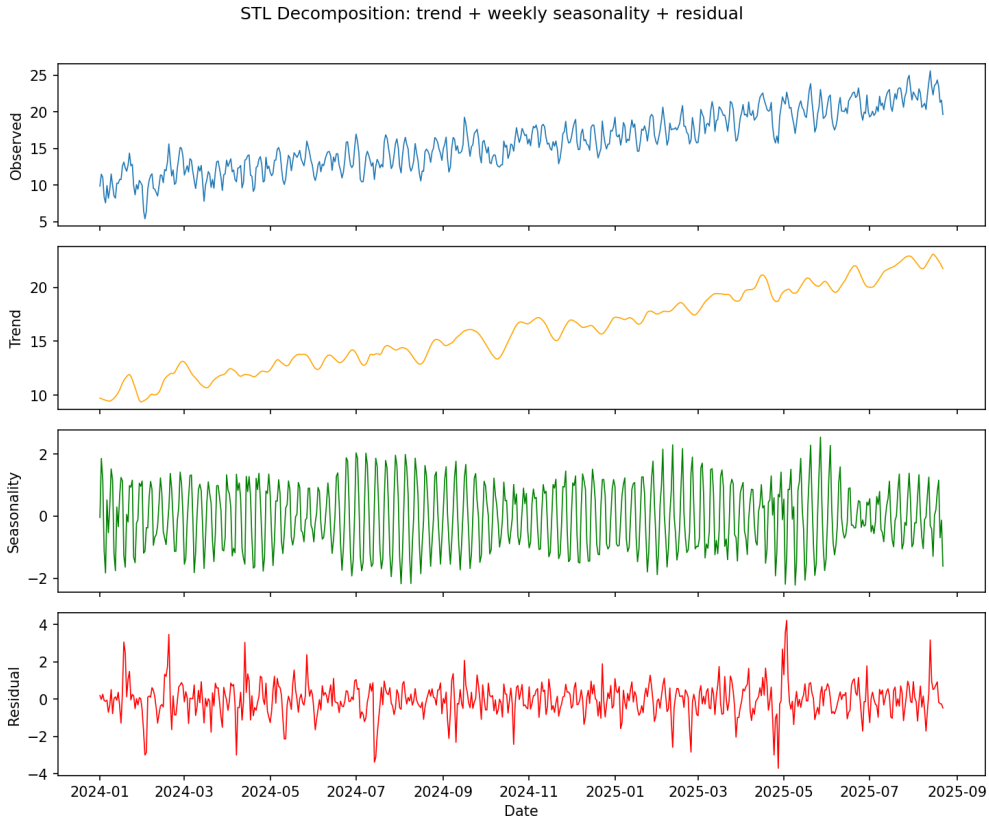
3. **Causal timing designs and placebos.** Interrupted Time Series (ITS) is often used when an intervention occurs at a known time t_0 but randomization is not available.
   - State one key identification assumption behind ITS.
   - Explain the difference between a "level change" and a "trend change" at t_0.
   - Propose one placebo diagnostic and explain what failure would look like (e.g., a fake intervention date, pre-trend check).

Applied Exercises

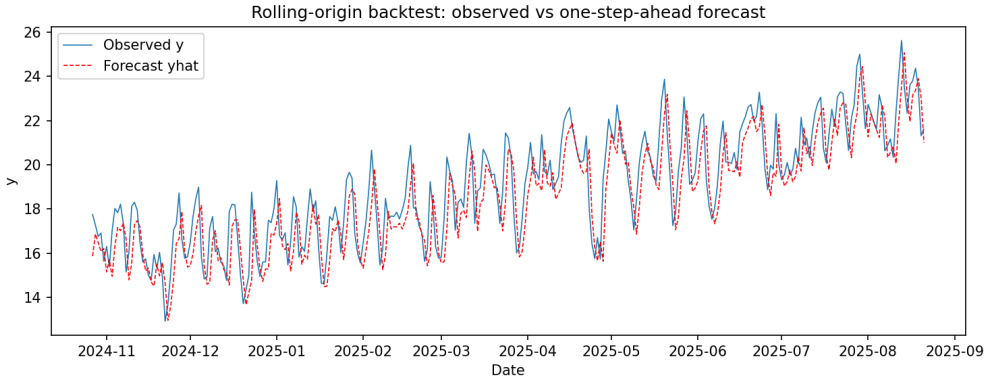
Use the code in `timeseries.R` and the lecture slides to answer the following questions.

Response: Find this week's work here: `https://github.com/yasinshafi/soda501/tree/main/06_timeseries/problem_set`

4. **Decomposition: trend + seasonality + residual (save a figure).** Extend the synthetic daily time series in `timeseries.R` by decomposing it into components.
   - Convert the daily series to a `ts` object with weekly frequency (`frequency = 7`).
   - Run a seasonal-trend decomposition (e.g., `stl()`).
   - Create a single figure showing the observed series and the decomposed components.
   - Save the figure as `outputs/figures/decomposition.png`.



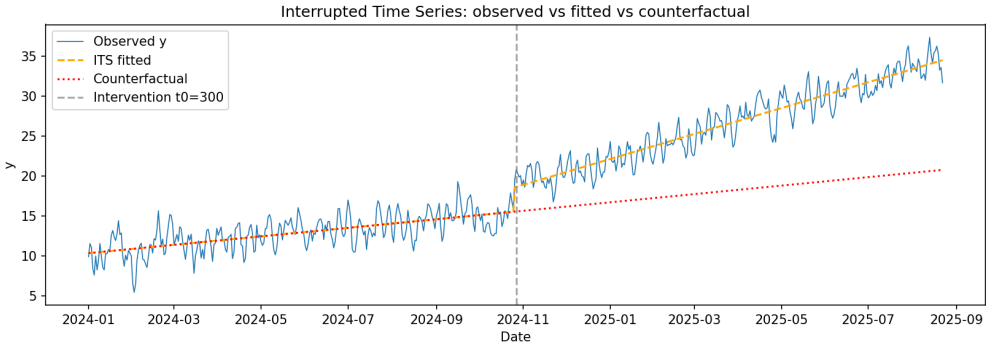
5. **Rolling-origin evaluation (backtesting) for forecasting RMSE.** The lecture emphasizes that evaluation should mimic deployment (fit on past, predict future). Implement a rolling-origin backtest that produces *many* out-of-sample errors instead of a single split.
   - Choose a forecast horizon $h = 1$ day ahead.
   - Choose an initial training window (e.g., first 300 days).
   - For each time t from the end of the initial window to the end of the series:
     (a) fit an AR(1) model to data up to t (use `arima(..., order=c(1,0,0))` or `forecast::auto.arima` if available),
     (b) forecast y_{t+1},
     (c) store the one-step-ahead error $e_{t+1} = y_{t+1} - \hat{y}_{t+1}$.
   - Compute RMSE across the backtest errors and compare it to the single time-split RMSE from the tutorial.
   - Save:
     – a CSV of the backtest errors (`date`, `y`, `yhat`, `error`) as `outputs/tables/backtest_errors.csv`, and
     – a line plot of y_t and $\hat{y}_t$ over the test region as `outputs/figures/backtest_forecast.png`.



6. **Interrupted Time Series (ITS): level and slope change + placebo date.** Create a synthetic intervention at time t_0 on top of a trend (use the lecture's ITS framing).
   - Pick an intervention date t_0 around the middle of the series (report your choice).
   - Create an intervention indicator $I[t \geq t_0]$ and a post-intervention time counter $(t - t_0)\, I[t \geq t_0]$.
   - Fit an ITS regression:

$$y_t = \alpha + \delta t + \tau_1 I[t \geq t_0] + \tau_2 (t - t_0) I[t \geq t_0] + \varepsilon_t$$

   - Plot three lines on the same figure:
     – observed y_t,
     – fitted values from the ITS model,
     – counterfactual values setting $\tau_1 = \tau_2 = 0$ (pre-period trend extended forward).
   - Run a placebo ITS with a *fake* intervention date in the pre-period and report whether it produces a large "effect." (Briefly interpret.)
   - Save:
     – coefficient table for the real ITS and placebo ITS as `outputs/tables/its_results.csv`, and
     – your main ITS figure as `outputs/figures/its_plot.png`.



Challenge Question (Optional — if you finish early)

Choose **ONE** option.

(a) **Switchback experiment (time-block randomization).** Simulate a switchback design where treatment alternates by time blocks (e.g., every 7 days) and estimate the effect.
   - Create a block assignment over time (alternating 0/1).

- Generate an outcome with trend + noise + a treatment effect that operates only during treated blocks.
- Estimate the difference in means across treated vs control blocks.
- Placebo: shift assignment by 1 block (lag the assignment) and show the estimated effect shrinks toward 0.
- Save a plot of the time series with shaded treated blocks as `outputs/figures/switchback.png`.

(b) **Sequential testing / peeking inflation (simulation).** Simulate repeated monitoring where you compute a p-value every day after a minimum start date.

- Under a true null (no treatment effect), compute daily "p-values" for a naive treated vs control comparison.
- Show that the probability of getting at least one p-value < 0.05 is much larger than 0.05.
- Briefly explain why this happens and name one control strategy (alpha spending, pre-specified stopping).