Time Series Analyses for Social Data

Jared Edgerton

Why Time Series Matters

Big social data are often *time-indexed*:

- Platform logs (daily/weekly metrics)
- Online experiments (A/B outcomes over time)
- Economic indicators (APIs, nowcasts)
- Event data (counts by day)
- Text streams (topic prevalence over time)

Time structure is not noise.

Time Series vs Panel vs Event Streams

Common structures:

- **Time series:** one unit tracked over time (national approval, platform KPI)
- **Panel / TSCS:** many units tracked over time (state-by-week, user-by-day)
- **Event stream:** irregular events (timestamps; bursts)

The data structure determines the model and validation.

Repeated Sampling Over Time

Common in big social data:

- repeated cross-sections (new sample each wave; e.g., weekly surveys)
- rotating panels (some respondents persist; some refresh)
- rolling cohorts (e.g., new users each week)

Core validity issue:

- changes in the outcome can reflect **composition change** (who is sampled), not behavior change.

Practical checks:

- compare marginal distributions of key covariates across waves
- weight or post-stratify when composition drifts
- cohort-specific plots (e.g., signup-week cohorts)

Panel Data at Scale (TSCS)

When you have many units tracked over time (state-by-week, user-by-day):

- two-way structure: unit i and time t
- key advantage: you can control for **unit fixed differences** and **common time shocks**

Typical baseline workflow:

- unit FE + time FE, clustered SE, and explicit time structure

Pitfalls in big data panels:

- serial correlation in errors
- staggered adoption / dynamic effects
- leakage from using future information in features

Survival / Time-to-Event Outcomes

Big-data experiments often generate survival outcomes:

- time until churn
- time until first purchase
- time until next login
- time until conversion / event

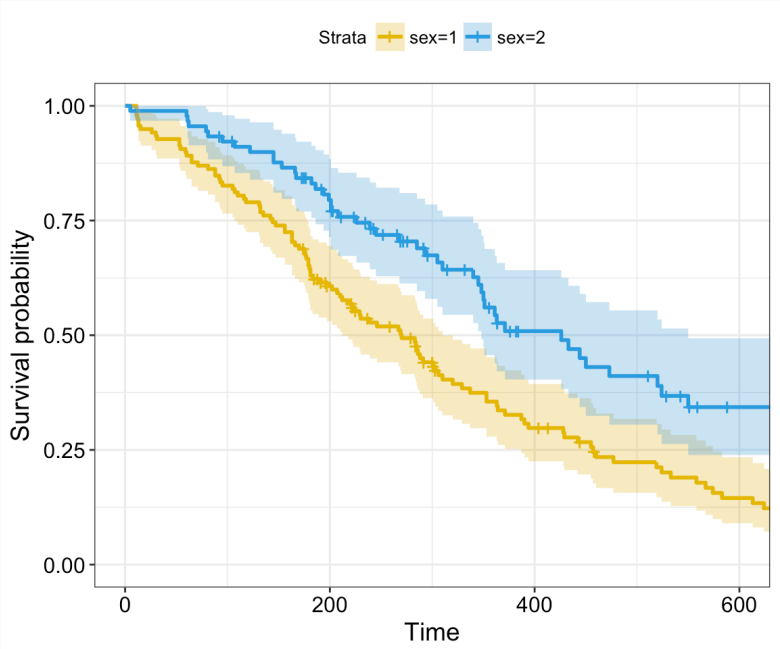
Key concepts:

- right censoring (user has not churned yet by end of observation)
- hazard vs survival curve
- cohort survival curves (Kaplan–Meier style)

RCT connection:

- compare survival curves across treatment/control
- effects can be time-varying (short-run bump vs long-run fade)

Survival trend example



Forecasting vs Causal Timing

Two common goals:

- **Forecasting:** predict future Y_{t+h} given history
- **Causal timing:** estimate effect of an intervention at time t_0

Big-data experiments often combine both:

- forecast baseline trend
- detect treatment effect on top of trend

The Biggest Mistake: Time Leakage

Random train/test splits break time:

- The model sees the future
- Evaluation looks unrealistically good
- Deployment fails

Correct default:

- train on past, test on future
- rolling windows / backtesting

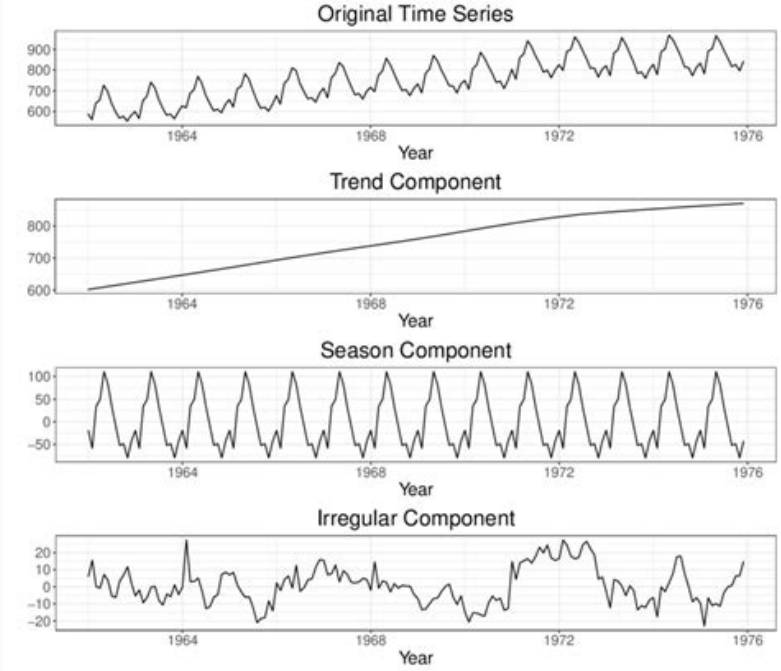
A Minimal Time Series DGP

A simple starting point: $Y_t = \mu_t + \varepsilon_t$

Trend + autocorrelation: $\mu_t = \alpha + \delta t + \phi(Y_{t-1} - \mu_{t-1})$

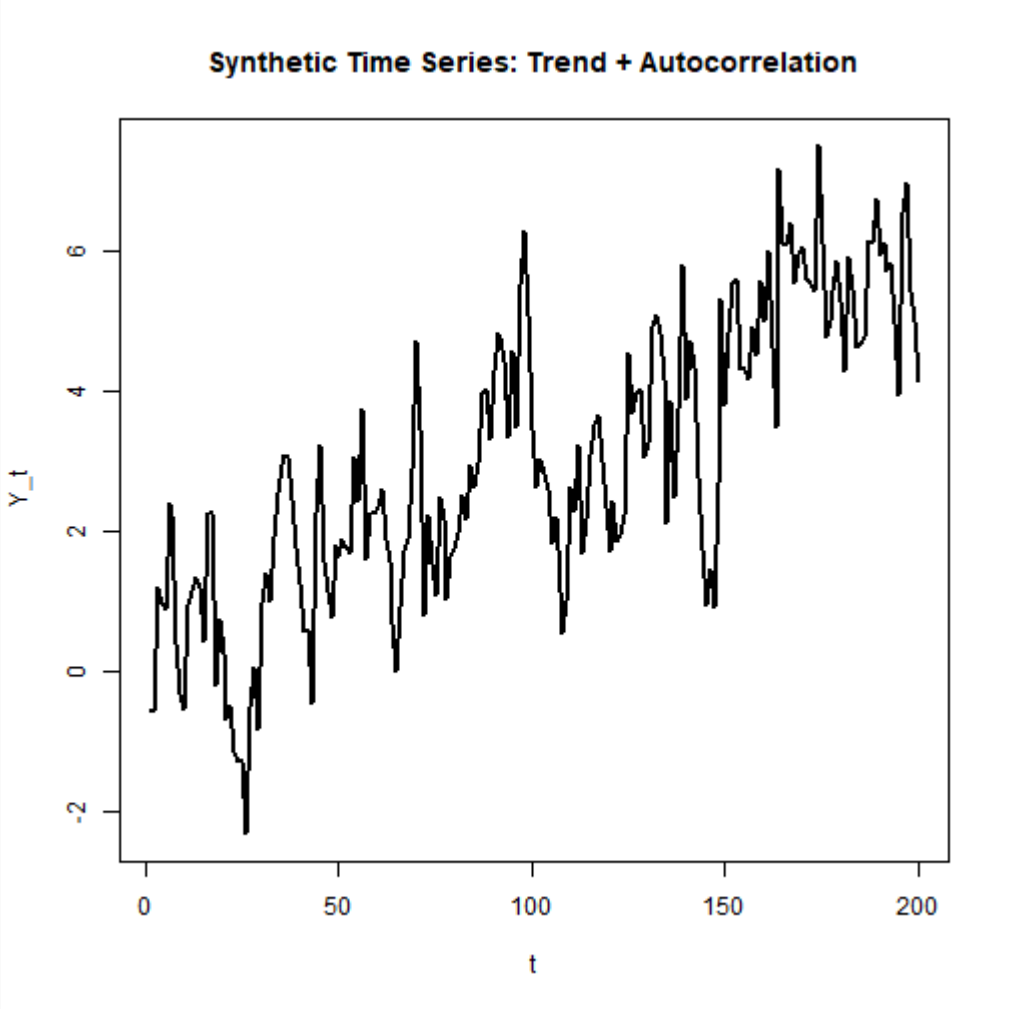
Key idea: observations are dependent over time.

Decomposition: Trend + Seasonality +



Takeaway: many "time series" patterns are structured (trend + seasonality), not i.i.d. noise.

Visualization: Autocorrelated Trend



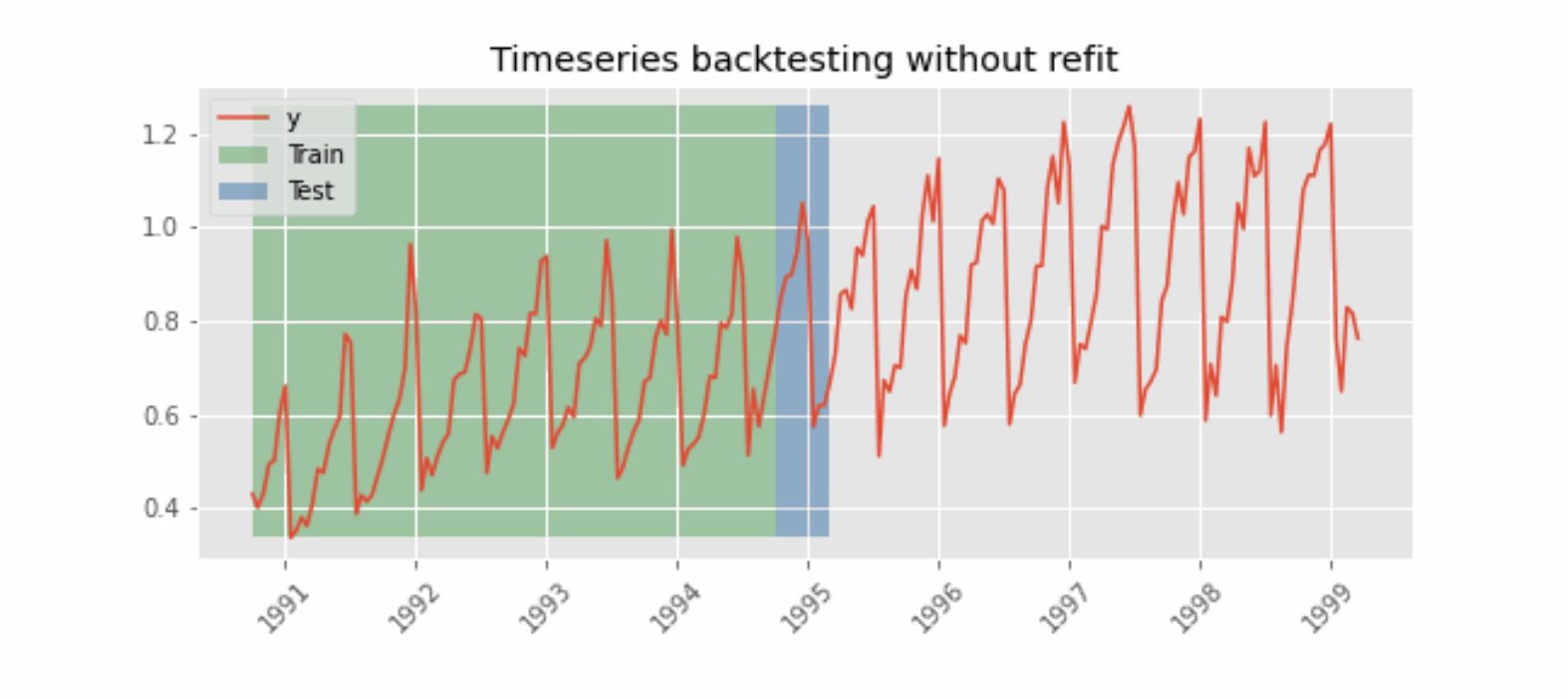
Rolling-Origin Evaluation (Backtesting)

Forecasting should mimic deployment:

- Fit on $1..t$
- Predict $t+1$
- Slide forward and repeat

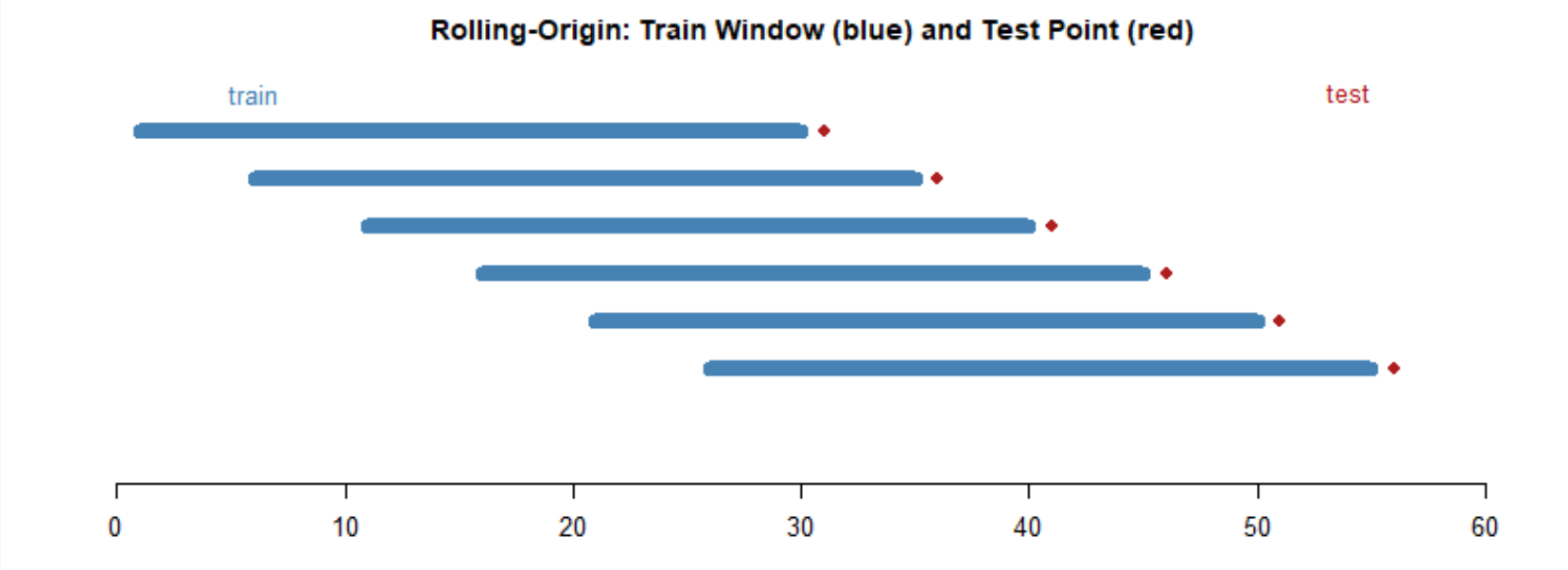
This produces *many* out-of-sample errors.

Backtesting Example (Train/Test Split



This is a visual reminder: evaluation must mimic deployment (train on past, test on future).

Visualization: Rolling Windows



Experiments Measured Over Time

In platform experiments, outcomes are often time series:

- daily active users
- retention
- content consumption
- clicks / engagement

Big sample sizes do not remove:

- time trends
- seasonality
- instrumentation shifts
- interference and spillovers

Interrupted Time Series (ITS)

A simple causal timing design:

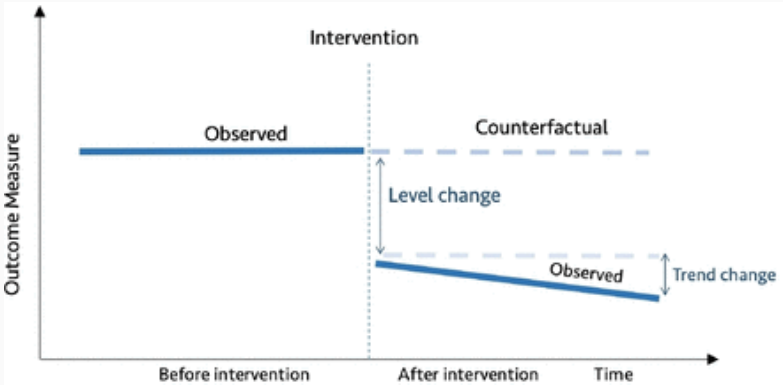
- define an intervention time t_0
- compare pre vs post while accounting for trend

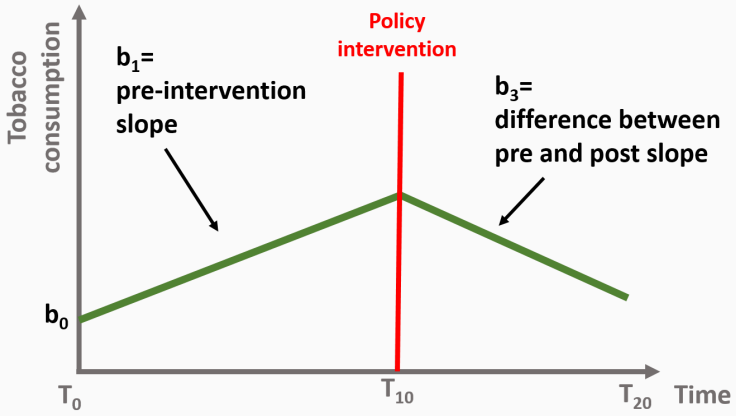
$$Y_t = \alpha + \delta t + \tau \mathbf{1}[t \geq t_0] + \varepsilon_t$$

Placebo checks:

- fake intervention dates
- pre-trend diagnostics

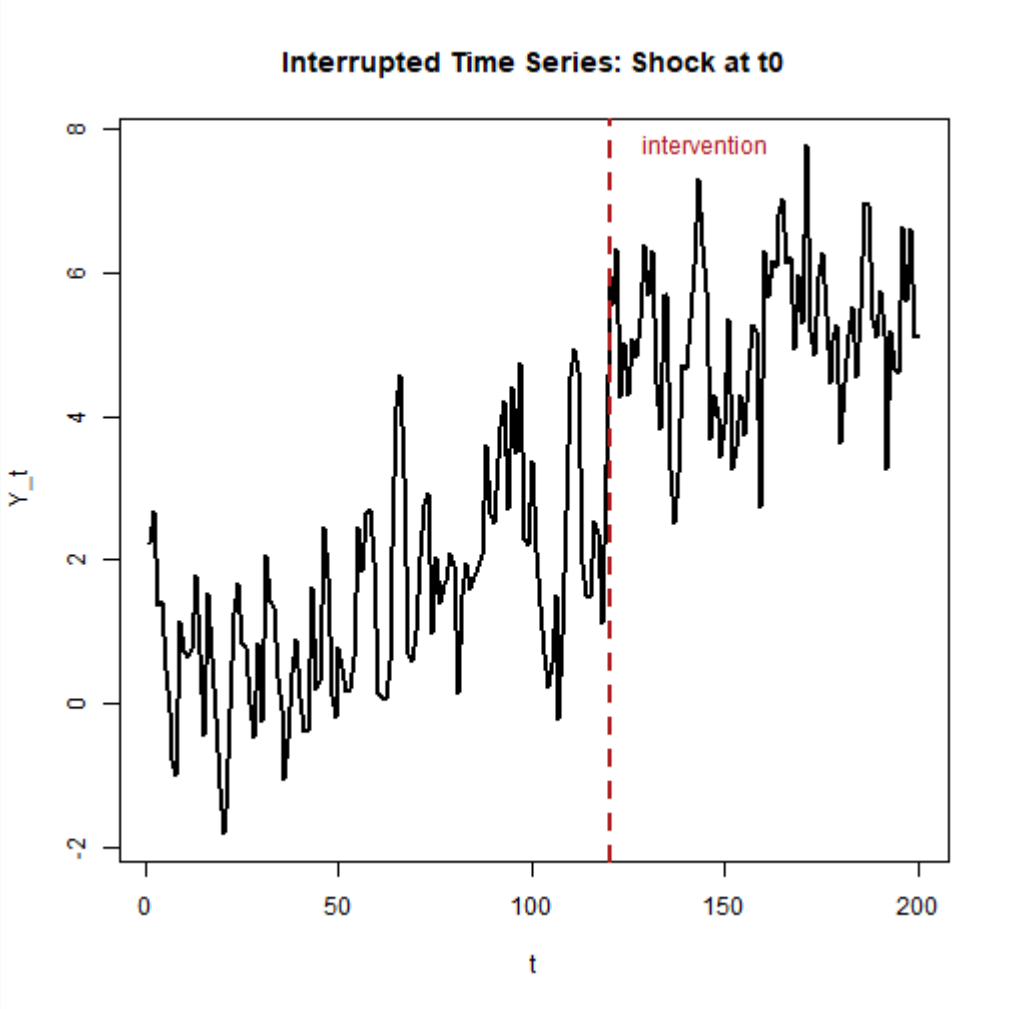
ITS Intuition: Level Change vs Trend



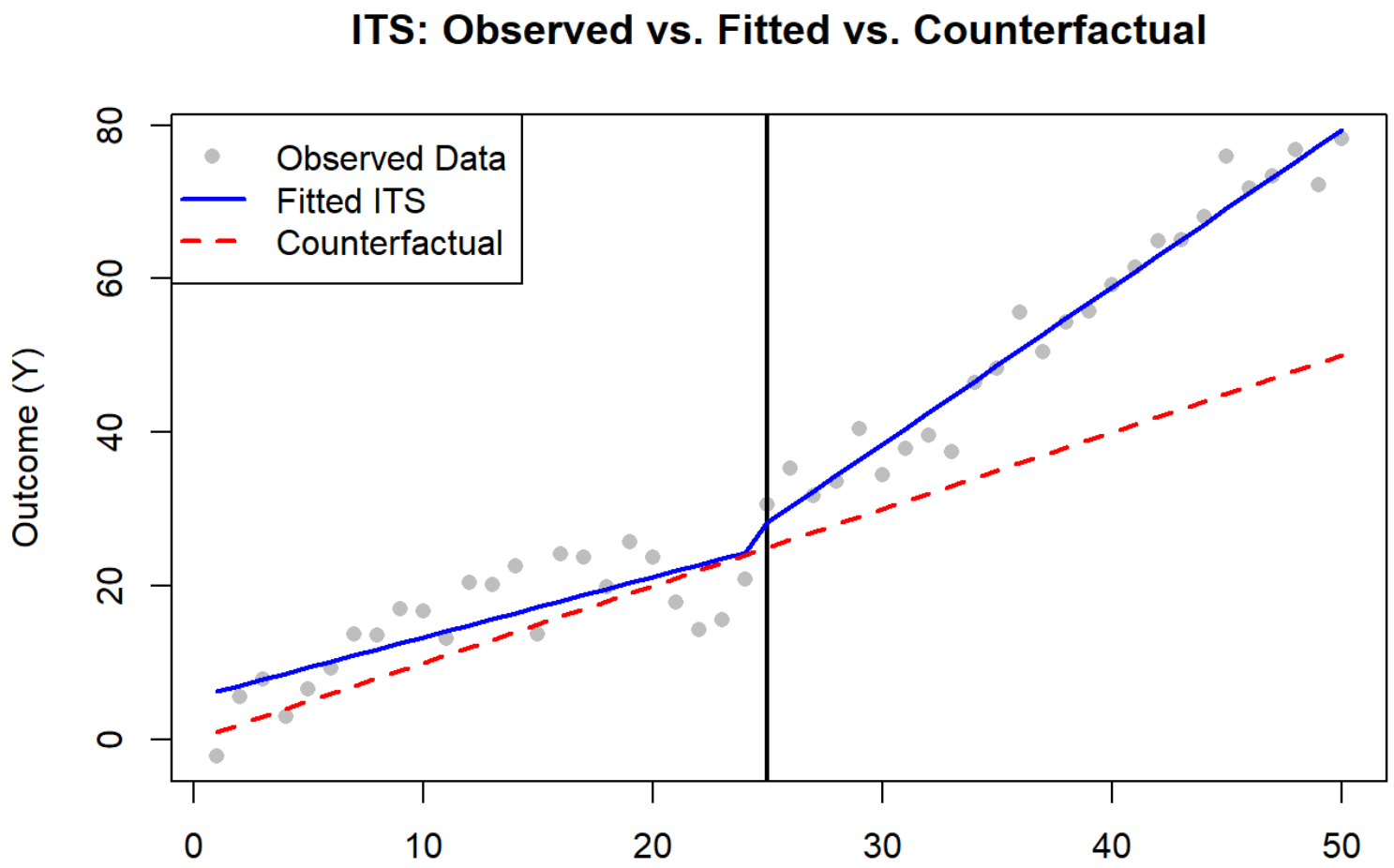


Use these schematics to clarify what "level change" and "trend change" mean before fitting any model.

Visualization: Intervention on Top of



ITS: Observed vs. Fitted vs.



Switchback Experiments (Big Data)

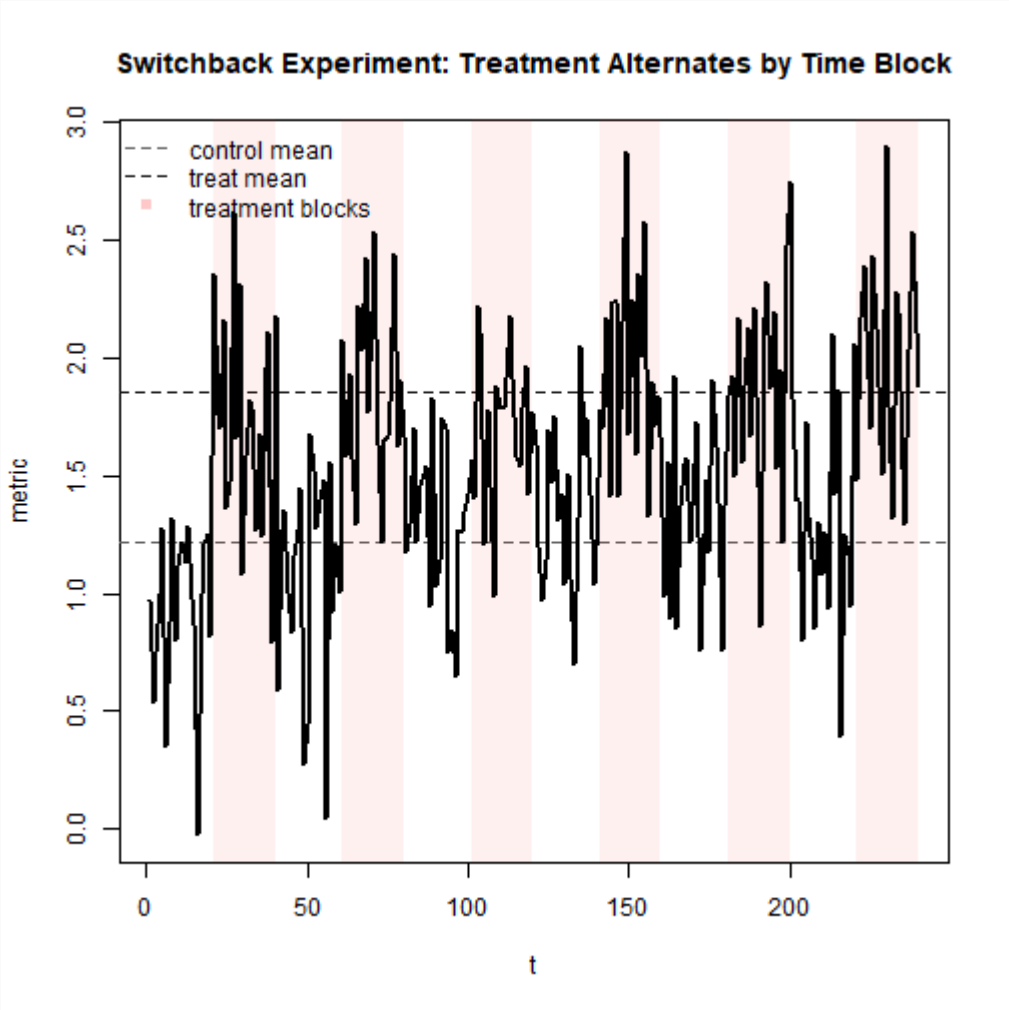
When interference is likely (e.g., congestion):

- randomize by *time blocks* (minutes/hours/days)
- alternate control/treatment across blocks
- compare outcomes across blocks

Common in:

- marketplaces
- rideshare
- recommender systems

Visualization: Switchback Assignment



Sequential Testing and Peeking

If you monitor outcomes every hour/day:

- repeatedly testing inflates false positives
- big n makes tiny fluctuations look 'real'

Controls:

- pre-specified stopping rules
- alpha spending / group sequential designs
- holdout windows

Minimal R Workflow (Forecasting)

```r
# 1) Sort by time
df ← df[order(df$date), ]

# 2) Split: past → future
train ← subset(df, date < as.Date('2026-01-01'))
test  ← subset(df, date ⩾ as.Date('2026-01-01'))

# 3) Fit baseline (example)
fit ← arima(train$y, order = c(1,0,0))

# 4) Forecast and score
pred ← predict(fit, n.ahead = nrow(test))$pred
rmse ← sqrt(mean((test$y - pred)^2))
```

Minimal R Workflow (Experiment Over

```r
# df has: date, metric, treated_block (0/1)

# Difference in means across time blocks
ate ← with(df, mean(metric[treated_block==1]) - mean(metric[treated_block==0]))

# Placebo: shift assignment by 1 block (should break alignment)
df$treated_placebo ← dplyr::lag(df$treated_block)
ate_placebo ← with(df, mean(metric[treated_placebo==1], na.rm=TRUE) -
                       mean(metric[treated_placebo==0], na.rm=TRUE))
```

Documentation and Reproducibility

Time series work is sensitive to:

- exact time zone
- aggregation rules (hourly vs daily)
- missingness imputation
- leakage (feature windows)

Always log:

- temporal split rule
- data snapshot date
- preprocessing choices

Discussion

- Where does time leakage happen in your own work?
- Which outcomes are most fragile to instrumentation shifts?
- When does an A/B test become a time-series problem?
- What makes a good" placebo for a temporal design?