Econ H191 — Code-Oriented Event-Study Strategy (v2025-11-11)

Goal: Fill Tables 1–5 (main) and A–C (core appendices) using a reproducible Python-first pipeline (with Stata mirrors when useful), fed by the canonical quarterized LP panel and optional mediators. This is a **do-from-scratch playbook**: conceptual steps → concrete scripts → file contracts → QA and rerun protocol when direct **L** arrives.

---

0) Executive Summary

1. **Assemble inputs** → `LP_Panel_quarterized.tsv` (port×terminal×quarter, LP in levels), clocks from `model1_params.yaml`, and shock toggles.
2. **Prepare event-study design matrices** with not-yet-treated (NYT) comparisons, terminal & quarter FE, optional port trends and shock windows.
3. **Estimate**: (i) dynamic β_k paths, (ii) average-post windows, (iii) small-N inference (CR2, wild-cluster, Fisher RI/IM optional), (iv) TEU-weighted variants.
4. **Export analysis-ready CSVs** with strict schemas consumed by `08_make_tables.py` and `09_make_inline_results_tex.py`.
5. **Render LaTeX tables** (main + trimmed appendices) and figures (paths, forest plots), log a run manifest.
6. **Rerun switch** when direct monthly **L** arrives: rebuild LP → re-quarterize → re-estimate; tables auto-refresh.

---

1) Inputs & Contracts

1.1 Canonical inputs

- `Design/Output Data/LP_Panel_quarterized.tsv`
- **Columns:** `port, terminal, year, quarter, lp` (levels; take logs in estimation)
- **Terminal conventions:** pre-entry port-level rows use `terminal=""`; post-entry split into `Legacy` and entrant (Haifa: `SIPG`; Ashdod: `HCT`).
- `Design/Code/Econometrics/model1_params.yaml` (clocks & toggles)
- **Keys (expected):** `event_clocks` (by `port×terminal`), `shock_windows` (COVID 2020–21; security 2023/24), `specs` (`baseline`, `+port_tr`, `+tr_shocks`), `windows` (e.g., `[1,4]`).
- (Optional) `Data/K/Mediator_K_over_L.tsv` (for placeholders / later mediation pass)
- (Optional) `Design/Output Data/panel_port_quarter_full.csv` (provenance metadata)

1.2 Output file contracts (consumed by table/figure makers)

- **Dynamic paths (by port×terminal×spec):**
- `Design/Output Data/es_dynamic_<port>_<terminal>_<spec>.csv`

- **Schema:** `port, terminal, spec, k, beta, se, p_cr2, p_wild, n_bins_support`
- **Average-post windows:**
- `Design/Output Data/es_avgpost_<port>_<terminal>.csv`
- **Schema:** `port, terminal, spec, window_lo, window_hi, beta, se, p_cr2, p_wild, beta_teu, se_teu, p_wild_teu`
- **Privatization (Haifa-Legacy clock):** `Design/Output Data/es_haifa_priv.csv`
- **Schema:** rows for avg-post & k∈{0..4} mini-grid; same fields as above
- **Diagnostics:** `Design/Output Data/es_support_coverage.csv`
- **Schema:** `spec, pct_nyt, pct_already_treated, cohorts_used, mean_comp_per_k, min_comp_any_k`
- **Run manifest:** `Design/Output Data/es_run_manifest.json`
- **Keys:** timestamps, git hash (if present), input file hashes, yaml snapshot, specs run, shock toggles

---

2) End-to-End Task Graph (Scripts & Responsibilities)

(01) Build Panel — `01_panel1_build.py` - Reads `Data/LP/LP_Panel.tsv`, prioritizes sources, outputs `panel_port_quarter_full.csv` + meta.

(02) Quarterize Canonical LP — `02_model1_combine_lp_quarter.py` - Drops monthly rows, anti-joins port-quarter from (01), writes `LP_Panel_quarterized.tsv`.

(03) Enrich — `03_lp_enrich_stepwise.py` - Attaches clocks, shock flags, spec IDs from YAML; creates `lp_es_ready.parquet` with `ln_lp`.

(04) Shared-Pre Sample Builder — `04_build_panel_terminal_sharedpre.py` - Ensures common pre-periods within port×terminal pairs for NYT design; writes `lp_es_sharedpre.parquet`.

(05) Prep Model 1 — `05_prep_model1_terminal.py` - Builds design matrices: event dummies (omit k=−1), FE (terminal, quarter), optional port trends & shocks.

(06) Estimate ES — `06_run_es.py` - Runs dynamic ES and avg-post windows per (port×terminal×spec); computes CR2 & wild-cluster p; writes dynamic & avg-post CSVs.

(07) Pooled & Figures — `07_pooled_main_and_figs.py` - Produces pooled entrant vs legacy, forest plots, path charts; writes `Design/visuals` PNG/PDFs.

(08) Tables — `08_make_tables.py` - Loads CSVs → renders main Tables 1–5.

(09) Inline — `09_make_inline_results_tex.py` - Generates small macros (e.g., `\newcommand{\HaifaSIPGAvgPost}{...}`) for paper text.

(Plot) — `Plot_LP_Series.py` - Level checks: LP over time (port & terminal), for narrative figures.

Note: Stata mirrors (optional): `run_es.do` for wildboottest/ri; keep identical schemas on export.

---

3) Estimation Logic (Concept → Code)

3.1 Core ES specification

- **Outcome:** `y_it = ln(lp_it)`
- **Unit:** terminal `i` in quarter `t` within port `p(i)`
- **Spec:**

$$y_{it} = \sum_{k/\ =-1} \beta_k \cdot 1\{K_{it} = k\} + \gamma_i + \delta_t + \tau_{p(i)} \cdot t\,[\,+\,\mathrm{shocks}_t\,] + \varepsilon_{it}$$

- `γ_i`: terminal FE, `δ_t`: calendar-quarter FE
- `τ_{p}·t`: optional **port linear trend** (+PortTr)
- `shocks_t`: optional COVID (2020–21) + late-2023/24 windows (+Tr&Shocks)
- **NYT**: composition excludes already-treated comparisons
- **Omitted bin:** k=−1 (donut)

Average-post: $\overline{\beta}_{[a,b]} = \frac{1}{b-a+1}\sum_{k=a}^{b} \beta_k$ with SE via delta or re-estimation over mean indicator.

TEU-weighted: weight obs by terminal TEU share for the window (export as separate columns).

3.2 Small-N inference

- **CR2**: small-sample correction for clustered variance (clusters=ports)
- **Wild-cluster bootstrap**: Rademacher weights at cluster level with 9,999 draws (export `p_wild`)
- **(Optional)** Fisher **randomization inference** on clock labels; IM t-tests over two clusters (report cautiously)

3.3 Code blueprint (Python)

Design matrix helper - Inputs: `lp_es_sharedpre.parquet`, YAML - Builds: - Full event-time dummies: `D_k` for k∈K, drop `k=-1` - FE encodings: use demean-within FE strategy or explicit dummies handled by absorbing in the estimator - Shock flags: `covid_flag`, `security2324_flag` - Port trends: `port_trend = (t_index) × port dummies`

Estimator - Regress with `linearmodels.PanelOLS` or `statsmodels` OLS on de-meaned data; cluster by port - Compute CR2 via `statsmodels.stats.sandwich_covariance.cov_cluster` with small-sample adjust - Wild-cluster: implement custom bootstrap (Rademacher ±1 at port level), re-estimating or using score bootstrap

Exports - Dynamic CSV per (port×terminal×spec) - Avg-post CSV (unweighted & TEU-weighted) - Support coverage CSV (ATT composition proxy) from design matrix counts

---

4) Fill Tables (order of operations)

1. **Table 1 (Avg post by port×terminal)**
2. Run `06_run_es.py` with specs: `baseline`, `+port_tr`, `+tr_shocks` → produce avg-post CSV; compute implied %Δ via `100*(exp(beta)-1)`.
3. Add `Share post L=proxy` from metadata (0 when direct L is available).
4. **Table 2 (Haifa privatization)**
5. Switch clock to `privatization` for `Haifa-Legacy`; re-run avg-post + mini dynamics (k=0..4).
6. Placebo: apply same clock to `Haifa-SIPG` and export.
7. **Tables 3 & 4 (Dynamic ES)**
8. Export compact grid: lead mean `(-4..-2)`, then k=0..4; leave full grid to Appendix A.
9. Include `Leads F-test (p)` using a joint test over pre bins.
10. **Table 5 (Robustness)**
11. Balanced sample, Excl. 20–21, Excl. 23–24 toggles are governed by YAML filters; re-run avg-post; export TEU-weighted variants.
12. **Appendix A (full k) + B (pretrend/placebo) + C (window sensitivity)**
13. Auto-populate from the same CSVs; ensure titles and captions are **not empty** and exactly match LaTeX labels.

---

5) YAML Schema (minimal contract)

```
specs:
  baseline:
    port_trends: false
    use_shock_windows: false
  port_tr:
    port_trends: true
    use_shock_windows: false
  tr_shocks:
    port_trends: true
    use_shock_windows: true
windows:
  avgpost_default: [1,4]
  compact_leads: [-4,-2]
```

```
shocks:
  covid: [2020Q1, 2021Q4]
  security2324: [2023Q4, 2024Q4]
clocks:
  haifa_entry:
    SIPG: 2021Q4
    Legacy: 2021Q4
  ashdod_entry:
    HCT: 2021Q4
    Legacy: 2021Q4
  haifa_privatization:
    Legacy: 2022Q4
```

> **Note:** Actual dates come from the dossier; above is a schematic. The code should parse either `YYYYQ#` or `{year,quarter}`.

---

6) Pseudocode (core routines)

A) Load & enrich

```
lp = pl.read_parquet("Design/Output Data/lp_es_sharedpre.parquet")  # or
read_csv
params = yaml.safe_load(open("Design/Code/Econometrics/model1_params.yaml"))

# Construct k grid per clock
def add_event_time(df, clock_name):
    # df has columns: port, terminal, year, quarter
    df = df.with_columns(
        pl.when(pl.col("date") >= clock_date)
          .then(((pl.col("year")*4+pl.col("quarter")) - event_index)).alias("k")
          .otherwise(((pl.col("year")*4+pl.col("quarter")) -
event_index)).alias("k")
    )
    return df
```

B) Design matrix & FE

```
# Dummy build (drop k=-1)
K = sorted(df["k"].unique())
K = [kk for kk in K if kk != -1]
X = pd.get_dummies(df["k"], prefix="k")[ [f"k_{kk}" for kk in K] ]
```

```
# Absorb terminal & quarter FE by within transformation, or keep as dummies if
memory allows
```

C) Estimation

```
# Using statsmodels OLS on demeaned y and demeaned X (within by terminal and
quarter)
res = sm.OLS(y_tilde, X_tilde).fit(cov_type='cluster', cov_kwds={'groups':
port_ids})
# CR2 adjustment via sandwich; wild-cluster p via bootstrap draws at port level
```

D) Avg-post

```
post_idx = [k for k in K if 1 <= k <= 4]
beta_bar = betas.loc[post_idx].mean()
# SE via delta:
V = cov.loc[post_idx, post_idx]
se_bar = np.sqrt((1/len(post_idx))**2 * V.values.sum())
```

E) TEU weights

```
w = teu_share.groupby(["port","terminal"]).apply(lambda s:
s.loc[post_idx].mean())
# Re-estimate with weights or compute weighted average of β_k; export both
```

F) Wild-cluster

```
for b in range(B):
    s = draw_rademacher_by_port()
    y_b = y_tilde * s[port_id]
    beta_b = OLS(y_b, X_tilde).fit().params
    store(beta_b)
# p_wild = 2*min( mean(beta_b>=|beta_hat|), mean(beta_b<=-|beta_hat|) )
```

---

7) Figure Generation

- **Event-time paths:** one panel per port; SIPG/HCT vs Legacy; specs in columns; 95% CR2 bands; pre bins shaded; shock windows gray.

- **Forest plots (avg-post):** rows = groups; columns = specs; show unweighted & TEU-weighted with open/filled markers.
- **Series sanity plots:** LP levels by terminal with vertical lines at entries/privatization.

Exports under `Design/visuals/...` with deterministic filenames used in the paper.

---

8) QA & Diagnostics (checklist)

1. **Identity checks**: terminal-quarter counts match across enrich→prep→estimate.
2. **Design coverage**: `es_support_coverage.csv` shows no empty bins in reported windows; min comps $\geq$1.
3. **Pretrend tests**: Leads F-tests non-rejection pre; flag regressions with p<0.10.
4. **Cluster sanity**: CR2 and wild p in same ballpark; document any divergence.
5. **Shock toggles**: coefficients stable under Excl. 20–21/23–24.
6. **Table rendering**: no missing titles/captions; `\bse` macros resolved; no Unicode beta.

---

9) Rerun Protocol when Direct L Arrives

1. Replace `Data/L_proxy/L_Proxy.tsv` with direct L (monthly).
2. Rebuild LP pipeline (S1–S5) → regenerate `LP_Panel.tsv`.
3. Rerun `01` → `02` to refresh `LP_Panel_quarterized.tsv`.
4. Re-estimate via `03` – `07`; all tables/figures refresh.
5. Update Table 1–5 **"Share post L=proxy"** → 0%.

---

10) Stata Mirror (optional)

- Use `reghdfe` (absorb terminal, quarter); wild cluster via `boottest`, CR2 via `ivreg2` small-sample options; export CSV with same schemas using `esttab` / `postfile`.
- For Fisher RI / IM: custom ado or Mata script; keep export keys identical.

---

11) Run Order (one-button wrapper)

A wrapper (`make_es_main.py`) can orchestrate:

1. Snapshot YAML → manifest
2. Call 03→06 for `baseline`, `port_tr`, `tr_shocks`
3. Call privatization clock for Haifa-Legacy + SIPG placebo
4. Build pooled & figures (07)
5. Render LaTeX (08, 09)
6. Validate CSV schemas & LaTeX compile

---

12) Deliverable Map → Tables

- **Table 1**: `es_avgpost_*.csv` (+ implied %Δ column from script 08)
- **Table 2**: `es_haifa_priv.csv` (avg-post + k∈0..4), SIPG placebo subset
- **Table 3**: `es_dynamic_Haifa_*.csv` compact rows (lead avg, 0..4)
- **Table 4**: `es_dynamic_Ashdod_*.csv` compact rows (lead avg, 0..4)
- **Table 5**: `es_avgpost_*.csv` under balanced/exclusion toggles, plus TEU-weighted
- **Appendix A–C**: full dynamic grid; pretrend/placebo; window sensitivity

---

13) Risks & Mitigations

- **Two clusters**: emphasize wild-cluster and CR2; add IM t as sensitivity.
- **Mixed frequency leaks**: guard with asserts that only quarterly rows enter estimation.
- **Clock ambiguity**: keep alt clocks in YAML; label results by clock.
- **Proxy L**: Surface `Share post L=proxy` in main tables until direct L replaces it.

---

14) Naming & Style Conventions

- Filenames: lower-snake; include `port_terminal_spec` tokens; avoid spaces.
- CSV headers: all lowercase; no spaces; units implicit or described in README.
- Dates: `YYYYQ#` or numeric `year, quarter` — **not both** in the same file.

---

15) Minimal README (to ship with outputs)

- Purpose, input hashes, yaml snapshot, run date, code commit, table/figure index.
- Reproduce command: `python make_es_main.py --specs baseline,port_tr,tr_shocks --render`.

This document is the canonical blueprint for filling the main and appendix tables. Keep it alongside the LaTeX file and the YAML; update the version header on material changes.