Model 1 — Python-first Event-Study & RDiT Plan (Callaway–Sant'Anna / Sun–Abraham)

This report is a **self-contained, code-oriented roadmap** for running the non-mediation econometrics for the Econ H191 thesis. Save this alongside your repo. If a new chat forgets context, re-upload this file and continue.

---

0) Purpose & Outputs

Goal: Estimate the **total effect** of reforms on productivity — **without** mediation — using:

1. **Dynamic event-study DiD** with **not-yet-treated** controls (Callaway–Sant'Anna / Sun–Abraham).
2. **Cross-port RDiT** windows as a validation.

Outcome: $Y_{p,t} = \ln(\mathrm{LP}_{p,t})$, using **mix-adjusted LP** already built in your pipeline. We will **not** re-engineer LP (no re-computing w or Π). We only quarterize the pre-period.

Artifacts produced (paths relative to repo root): - `Thesis/Design/Code/Econometrics/output/meta/panel_port_quarter.csv` - `Thesis/Design/Code/Econometrics/output/tables/table_es_lp.csv` - `Thesis/Design/Code/Econometrics/output/figures/fig_es_lp_{reform}_{port}.png`

---

1) Repository Layout (as in VS Code screenshots)

```
Thesis/
  Data/
    LP/
      LP_Panel.tsv
      Output/
        teu_monthly_plus_quarterly_by_port.tsv
        monthly_output_by_1000_tons_ports_and_terminals.tsv
    L_proxy/
      L_Proxy.tsv                # Not used for main estimates (diagnostics
only)
    K/                           # Not used for Model 1
  Design/
    Code/
      Econometrics/
        model1_params.yaml       # You created this file
        model1_build_panel.py    # Build port×quarter panel (quarterize
pre-period only)
```

```
        model1_es_did.py          # Event-study in Python via pyfixest (Sun-
Abraham)
        model1_es_did.do          # Stata fallback (csdid)
        output/
          meta/
          tables/
          figures/
```

---

2) Parameter File (YAML)

Create/confirm `Thesis/Design/Code/Econometrics/model1_params.yaml`:

```yaml
events:
  competition:
    Haifa:  "2021Q3"
    Ashdod: "2022Q4"
  privatization:
    Haifa:  "2023Q1"
    Ashdod: null

bins:
  leads_ref: -1            # omit h = -1 as reference (or change to -2)
  post_1: [0, 4]           # 0–4 quarters
  post_2: [5, 12]          # 5–12 quarters

robustness:
  drop_2020: true
  truncate_at_2023Q3: true
  include_spillover_flag: true

paths:
  lp_panel: "Thesis/Data/LP/LP_Panel.tsv"
  teu:      "Thesis/Data/LP/Output/teu_monthly_plus_quarterly_by_port.tsv"
  tons:     "Thesis/Data/LP/Output/
monthly_output_by_1000_tons_ports_and_terminals.tsv"

output:
  panel_q:   "Thesis/Design/Code/Econometrics/output/meta/
panel_port_quarter.csv"
  es_tables: "Thesis/Design/Code/Econometrics/output/tables/table_es_lp.csv"
  figs:      "Thesis/Design/Code/Econometrics/output/figures"
```

---

3) Build Panel Script — `model1_build_panel.py`

Purpose

Assemble a **port×quarter** panel (2018Q1–2024Q4) with **no LP recomputation**. Only quarterize the pre-period and attach reform clocks and not-yet-treated masks.

Environment

```
pip install pandas pyyaml numpy
```

Required columns (expected in LP_Panel.tsv)

- Keys/structure: `port`, `freq` (Monthly/Quarterly), `level` (port/terminal), time variables (`month` or `year` + `quarter`).
- Outcome: `LP` (mix-adjusted). If `ln_LP` is present, we may use it directly; else compute `Y = ln(LP)`.

Steps (must-do)

1. **Load YAML** and **LP_Panel.tsv**.
2. **Construct time**: `qtr = YYYYQ#`; `t_index` = integer index over sorted `qtr`.
3. **Quarterize pre-period (port-month → port-quarter)**:
4. For `freq=="Monthly"` & `level=="port"`, compute `Y_q = mean(ln(LP_m))` within quarter (geometric mean). If fewer than 2 months available in a quarter, set NA.
5. For rows already at quarter frequency (post), keep as is.
6. **Merge** to a single **port×quarter** table: `port, qtr, t_index, Y (= ln LP)` plus provenance `source`.
7. **Attach reform clocks** using YAML: `g_comp[port]`, `g_priv[port]`. Build `tau_comp = t_index - g_comp[port]` and `tau_priv = t_index - g_priv[port]` (nullable if never treated).
8. **Not-yet-treated masks**:
9. Haifa – competition: keep `t_index < g_comp["Ashdod"]`.
10. Ashdod – competition: keep `t_index < g_comp["Haifa"]`.
11. Analog for privatization (when Ashdod's date exists).
12. **Shock trims** (optional per YAML): drop 2020; truncate at 2023Q3.
13. **Write** `panel_port_quarter.csv` to `output/meta`.

QA gates (print/fail if violated)

- Unique `(port, qtr)` after build.
- No missing `Y` in analysis rows.
- Presence of `g_comp`/`g_priv` for treated ports.
- Sample counts for each not-yet-treated subset.

Note: We deliberately **do not** recompute `w`, `Π`, or enforce `mean(w)=1`. This script is LP-light by design.

---

4) Estimation Script — `model1_es_did.py` (Python via pyfixest)

Environment

```
pip install pyfixest matplotlib pandas pyyaml numpy
```

Model

For each `reform ∈ {competition, privatization}` and `treated_port ∈ {Haifa, Ashdod (if treated)}`: - Subset to **not-yet-treated** comparison (e.g., for Haifa competition, drop quarters where Ashdod is already treated for competition). - Fit Sun–Abraham ES with unit & time FE and **time-clustered** SEs:

```
from pyfixest.estimation import feols
from pyfixest.utils import to_formula

# df_sub has columns: Y, port, t_index, G (first treated time for this reform)
formula = to_formula("Y ~ sunab(G, t_index) | port + t_index")
est = feols(formula, data=df_sub, vcov="cluster:t_index")
```

Deliverables

- **Dynamic path** (event time h, est, se, ci) saved to `table_es_lp.csv` with identifiers `{reform, treated_port, spec}`.
- **Pre-trend test**: joint Wald test that all **leads** (h < 0, excluding the omitted bin) equal zero.
- **TE windows**: averages over `[0..4]` and `[5..12]` quarters using linear combos of the ES coefficients and their VCOV.
- **Figures**: `fig_es_lp_{reform}_{treated_port}.png` — coefficient path with 95% CIs, h=0 marked, omitted pre-bin highlighted.

Optional robustness (run on toggles in YAML)

- Drop 2020; truncate at 2023Q3; include a spillover check in a wider sample (add a rival-post dummy in a FE regression as a sensitivity only).

---

5) Stata Fallback — `model1_es_did.do` (if you prefer Stata for estimation)

Requirements: `csdid` (Callaway–Sant'Anna), `reghdfe`.

Sketch:

```
clear all
import delimited using "Thesis/Design/Code/Econometrics/output/meta/
panel_port_quarter.csv", varnames(1)

* Example: Haifa – competition, not-yet-treated subset
keep if t_index < g_comp_ashdod   // compute/merge this constant from YAML or a
small mapping file

csdid Y, ivar(port) time(t_index) gvar(g_comp) method(dripw) notyet vce(cluster
t_index)
estat event, window(-8 12) ref(-1) graph name(haifa_comp, replace)
* Export coeff tables and CIs to CSV and mirror Python figure style
```

---

6) Run Order & Artifacts

1. `python Thesis/Design/Code/Econometrics/model1_build_panel.py`
2. `python Thesis/Design/Code/Econometrics/model1_es_did.py`

Outputs: - `panel_port_quarter.csv` (analysis panel) - `table_es_lp.csv` (dynamic ES, pre-trends, TE windows) - `fig_es_lp_*` (publication-ready figures)

---

7) Guardrails (what not to do)

- No TWFE on the full staggered sample.
- Do not allow already-treated units as controls. Always use **not-yet-treated** subsamples.
- Do not recompute LP components. Quarterize pre-period only.
- Keep **separate clocks** for competition vs privatization.
- Cluster by **time** (t_index) for conservative inference with 2 units.

---

8) Next Up (after Model 1)

- Implement **RDiT** checks around each event date (±4 to ±8 quarters, HAC errors, donut around 0).

- When ready to add mediation: build $M = \ln(K/L)$ for Haifa first (Track B), run ES on M (first stage), then 2SLS mediation on the same clocks.

End of report.