LP_Panel Build — End-to-End Report (Econ H191)

Purpose. This document is a self-contained playbook for reproducing and understanding the construction of the **mix-adjusted labor productivity panel** (`LP_Panel.tsv`). It records (i) inputs and their schemas/paths, (ii) the conceptual and mathematical definitions behind the LP proxy, and (iii) the code-level pipeline in **five gated stages (S1–S5)**, including QA and tunables. You can paste this report into any future chat to bootstrap context.

---

0) Executive Summary

We build six LP time series:

1. **Haifa — Port (Monthly)**: 2018-01 → 2021-08.
2. **Ashdod — Port (Monthly)**: 2018-01 → 2021-08.
3. **Haifa-Legacy — Terminal (Quarterly)**: 2021-Q3 → 2024-Q4.
4. **Haifa-Bayport (SIPG) — Terminal (Quarterly)**: 2021-Q3 → 2024-Q4.
5. **Ashdod-Legacy — Terminal (Quarterly)**: 2021-Q3 → 2024-Q4.
6. **Ashdod-HCT — Terminal (Quarterly)**: 2021-Q3 → 2024-Q4.

Core idea: LP is a **mix-adjusted proxy** defined as

$$\mathbf{LP} = \boldsymbol{w} \times \boldsymbol{\Pi}$$

- **w** (cargo-mix factor) = **tons/TEU**, **winsorized** within **(port, year)** and **rebased to mean 1** within the same group.
- **Π (Pi)** = a (port×month or terminal×year) **mix baseline** combining *quarter-constant terminal shares* with *terminal-year intrinsic productivity* $\Pi_{i,y}$.

We run a **deterministic five-stage pipeline** from raw files → S1/S2/S3 artifacts → S4 LP series → S5 stacked `LP_Panel.tsv`, with QA gates at each step and CLI tunables (e.g., **date windows**, **winsor bounds**).

---

1) Inputs, Locations, and Schemas

A) Monthly tons: ports & terminals

- **Path:** `Data/Output/monthly_output_by_1000_tons_ports_and_terminals.tsv`
- **Columns:** `PortOrTerminal`, `Month-Year` (MM-YYYY), `tons_k` (thousands).
We multiply `tons_k × 1000` *to obtain tons.*
- **Scope:** Ports (`Ashdod`, `Haifa`) and terminals that appear explicitly (`Ashdod HCT`, `Haifa SIPG`). Also includes `All Ports` and `Eilat` rows we **drop**.

Terminal canonicalization for tons (S1): - `Ashdod HCT` → `Ashdod-HCT` (port=`Ashdod`) - `Haifa SIPG` → `Haifa-SIPG` (port=`Haifa`)

Precedence rule for port-month tons (S1): - If terminal rows exist for a (port, month), **sum terminals** → `tons_port_m`. - Else, use the single port row → `tons_port_m`. - We track source as `tons_source ∈ {sum_terminals, port_row}`.

B) TEU: mixed frequency

- **Path:** `Data/Output/teu_monthly_plus_quarterly_by_port.tsv`
- **Columns:** `Port`, `Period`, `Freq ∈ {Monthly, Quarterly}`, `Year`, `MonthIndex (YYYYMM)`, `TEU_thousands`, `TEU`.
- **Rule:** Use `TEU` if present; else `TEU_thousands × 1000`.
- **Granularity split:**
- **Monthly (ports):** rows with `Freq=Monthly` and `Port ∈ {Ashdod, Haifa}` → **port-month TEU**.
- **Quarterly (terms encoded in `Port`):** rows with `Freq=Quarterly` and `Port ∈ {Haifa, Ashdod, Haifa SIPG, Ashdod HCT}` → **terminal-quarter TEU**.

Terminal canonicalization for TEU (S2): - `Haifa SIPG` → `Haifa-Bayport` (port=`Haifa`) - `Ashdod HCT` → `Ashdod-HCT` (port=`Ashdod`) - `Haifa` → `Haifa-Legacy` (port=`Haifa`) - `Ashdod` → `Ashdod-Legacy` (port=`Ashdod`)

We then compute **port-quarter TEU** as the **sum of terminal-quarter TEU**.

C) L_Proxy: terminal×month labor & Π

- **Path:** `Data/L_proxy/L_Proxy.tsv`
- **Minimum required columns:** `port`, `terminal`, `year`, `month`, `L_hours_i_m`, `Pi_teu_per_hour_i_y`.
- **Canonical terminals (S3):** `{Haifa-Bayport, Haifa-Legacy, Ashdod-HCT, Ashdod-Legacy}` with mapping for common variants (e.g., `Haifa SIPG` → `Haifa-Bayport`).
- **S3 constructs:**
- `S3_lproxy_clean.tsv` (terminal×month; canonical names; Int64 for dates; `month_index` = `year*12+month`; `quarter` from `month` if missing).
- `S3_terminal_year_pi.tsv` (unique **terminal×year** Π; **median** used if multiple values exist; variance recorded in QA).
- `S3_port_month_labor.tsv` (sum of terminal labor → **port×month** labor for the identity diagnostic).

---

2) Concept & Math (LP = w × Π)

2.1 Monthly (pre-reform, port×month)

1. **r (raw ratio)**: $r_{p,m} = \frac{\text{tons}_{p,m}}{\text{TEU}_{p,m}}$ using **port-month** tons and TEU.
2. **Winsorize** r within **(port, year)** to [low, high] quantiles (default 1–99%).
3. **Rebase to mean 1** within **(port, year)**: $w_{p,m} = r^{\text{clip}}_{p,m} / \overline{r^{\text{clip}}}_{(p,y)}$.
   A final guard ensures exact **mean(w)=1** per **(port, year)**.
4. **Π (port-month)** via **quarter-constant terminal shares** × **terminal-year Π**:

5. For each (port, year, quarter), pick shares in priority order: 1) `share_i_p_q` from L_Proxy, else 2) terminal **TEU_i_m** aggregated to quarter, else 3) terminal **TEU_i_q** from TEU file.
6. Then $\Pi_{p,q} = \sum_{i \in p,q} \mathrm{share}_{i,p,q} \times \Pi_{i,y}$. Broadcast to each month m in q.
7. **LP (monthly port)**: $\mathrm{LP}_{p,m} = w_{p,m} \times \Pi_{p,q(m)}$.
8. **Identity diagnostic** (optional): $\mathrm{LP}_{\mathrm{id},p,m} = \mathrm{TEU}_{p,m} / \mathrm{Labor}_{p,m}$.

2.2 Quarterly (post-reform, terminal×quarter)

1. **r (raw ratio)**: $r_{p,q} = \frac{\sum_{m \in q} \mathrm{tons}_{p,m}}{\mathrm{TEU}_{p,q}}$, where **TEU_{p,q} = \sum_i TEU_{i,q}**.
2. **Winsorize + rebase to mean 1** within **(port, year)** (exact guard applied).
3. **LP (terminal quarter)**: $\mathrm{LP}_{i,q} = w_{p,q} \times \Pi_{i,y}$.

Important: We **never infer** missing monthly TEU from quarterlies; we respect the file's frequency flags.

---

3) The Build Pipeline (S1–S5)

Each stage has explicit inputs → outputs, strict schemas, QA gates, and CLI usage. **Run in order.**

S1 — Tons from raw monthly file

Script: `Data/LP/Build_LP_Panel_S1_Tons.py`

Inputs - `--tons` → `Data/Output/monthly_output_by_1000_tons_ports_and_terminals.tsv`

Outputs - `S1_terminal_month_tons.tsv`
Columns: `port, terminal, year, month, month_index, tons_i_m` - `S1_port_month_tons.tsv`
Columns: `port, year, month, month_index, quarter, tons_port_m, tons_source` - `S1_port_quarter_tons.tsv`
Columns: `port, year, quarter, tons_port_q` - `S1_examples_port_precedence.tsv` (where both terminal sum and port row exist) - `S1_qa.tsv`, `_meta_s1.json`

QA gates - Keys unique by grain (terminal×month; port×month; port×quarter). - Source distribution log (`sum_terminals` vs `port_row`).

Run

```
python "Data/LP/Build_LP_Panel_S1_Tons.py"
  --tons "Data/Output/monthly_output_by_1000_tons_ports_and_terminals.tsv"
  --out  "Data/LP"
```

S2 — TEU from mixed-frequency file

Script: `Data/LP/Build_LP_Panel_S2_TEU.py`

Inputs - `--teu` → `Data/Output/teu_monthly_plus_quarterly_by_port.tsv`

Outputs - `S2_port_month_teu.tsv`
Columns: `port, year, month, month_index, TEU_port_m, is_pre_reform` - `S2_terminal_quarter_teu.tsv`
Columns: `port, terminal, year, quarter, TEU_i_q` - `S2_port_quarter_teu.tsv`
Columns: `port, year, quarter, TEU_port_q` - `S2_qa.tsv`, `_meta_s2.json`

QA gates - Keys unique by grain (port×month; terminal×quarter; port×quarter). - **Additivity**: port-quarter TEU equals sum of terminal-quarter TEU. - Counts of zero/neg TEU (not fatal; ratios masked later).

Run

```
python "Data/LP/Build_LP_Panel_S2_TEU.py"
  --teu "Data/Output/teu_monthly_plus_quarterly_by_port.tsv"
  --out "Data/LP"
```

S3 — L_Proxy harmonization

Script: `Data/LP/Build_LP_Panel_S3_LProxy.py`

Inputs - `--lproxy` → `Data/L_proxy/L_Proxy.tsv` - `--s2_term_quarter` → `Data/LP/S2_terminal_quarter_teu.tsv`

Outputs - `S3_lproxy_clean.tsv` (terminal×month; canonical names & strict dtypes). - `S3_port_month_labor.tsv` (port×month sums of labor hours). - `S3_terminal_year_pi.tsv` (unique terminal×year Π; **median** chosen if multiple values exist). - `S3_coverage_vs_s2.tsv`, `S3_coverage_gaps.tsv` (terminal-years used in S2 but missing Π and/or labor months). - `S3_qa.tsv`, `_meta_s3.json`.

QA gates - Uniqueness (terminal×month; terminal×year Π).
- Π variance counts (how often terminal-year Π had >1 distinct values). - Coverage checks vs S2.

Run

```
python "Data/LP/Build_LP_Panel_S3_LProxy.py"
  --lproxy "Data/L_proxy/L_Proxy.tsv"
  --s2_term_quarter "Data/LP/S2_terminal_quarter_teu.tsv"
  --out "Data/LP"
```

S4 — Compute LP (monthly ports; quarterly terminals)

Script: `Data/LP/Build_LP_Panel_S4.py`

Inputs (defaults) - `--s1_port_month_tons Data/LP/S1_port_month_tons.tsv` - `--s1_port_quarter_tons Data/LP/S1_port_quarter_tons.tsv` - `--s2_port_month_teu Data/LP/S2_port_month_teu.tsv` - `--s2_term_quarter_teu Data/LP/S2_terminal_quarter_teu.tsv` - `--s2_port_quarter_teu Data/LP/S2_port_quarter_teu.tsv` - `--s3_lproxy_clean Data/LP/S3_lproxy_clean.tsv` - `--s3_port_month_labor Data/LP/S3_port_month_labor.tsv` - `--s3_term_year_pi Data/LP/S3_terminal_year_pi.tsv`

Tunables - `--winsor_low 0.01`, `--winsor_high 0.99` (winsor bounds). - **Range switches:** - `--monthly_start YYYYMM`, `--monthly_end YYYYMM` (default: `201801`, `202108`). - `--quarterly_start YYYYQ`, `--quarterly_end YYYYQ` (default: `2021Q3`, `2024Q4`).

Outputs - **Monthly ports**: `LP_Haifa_port_month.tsv`, `LP_Ashdod_port_month.tsv` Columns: `port, year, month, month_index, quarter, TEU_port_m, tons_port_m, tons_source, w, w_source, Pi_port_q, LP_mix, LP_id`. - **Quarterly terminals**: `LP_Haifa_Legacy_quarter.tsv`, `LP_Haifa_SIPG_quarter.tsv`, `LP_Ashdod_Legacy_quarter.tsv`, `LP_Ashdod_HCT_quarter.tsv` Columns: `port, terminal, year, quarter, TEU_i_q, w, Pi_teu_per_hour_i_y, LP_mix`. - QA: `S4A_qa.tsv` (monthly ports), `S4B_qa.tsv` (quarterly terminals), combined `qa_lp_report.tsv`, `_meta_s4.json`.

QA gates - Keys unique by grain. - **Mean(w)=1 by (port, year)** in both monthly and quarterly outputs (exact enforcement in S4; guard pass applied at the end of the quarterly build too). - **Annual preservation (monthly only)**: within (port, year), **abs(mean(LP) − mean(Π)) ≤ 0.05**. *(Relaxed from extremely tight tolerance to account for winsorization noise.)*

Run

```
python "Data/LP/Build_LP_Panel_S4.py"
  --winsor_low 0.01 --winsor_high 0.99
  --monthly_start 201801 --monthly_end 202108
  --quarterly_start 2021Q3 --quarterly_end 2024Q4
  --out "Data/LP"
```

S5 — Stack six series into LP_Panel

Script: `Data/LP/Build_LP_Panel_S5_Stack.py`

Inputs (defaults) - `--haifa_m Data/LP/LP_Haifa_port_month.tsv` - `--ashdod_m Data/LP/LP_Ashdod_port_month.tsv` - `--haifa_legacy_q Data/LP/LP_Haifa_Legacy_quarter.tsv` - `--haifa_sipg_q Data/LP/LP_Haifa_SIPG_quarter.tsv` - `--ashdod_legacy_q Data/LP/LP_Ashdod_Legacy_quarter.tsv` - `--ashdod_hct_q Data/LP/LP_Ashdod_HCT_quarter.tsv`

Outputs - `LP_Panel.tsv` (unified schema) - `S5_qa.tsv` (uniqueness by series×grain; span report; mean(w) by (freq, port, year); NA tallies) - `S5_spans.tsv`, `_meta_s5.json`

Unified schema (LP_Panel.tsv) `series_id, level, freq, port, terminal, year, month, quarter, month_index, quarter_index, TEU, tons, L_hours, w, Pi, LP, LP_id, tons_source`

Run

```
python "Data/LP/Build_LP_Panel_S5_Stack.py" --out_dir "Data/LP"
```

4) “Switchboard” — Change Date Windows (without editing code)

- **Monthly ports (Haifa/Ashdod)**: in S4, set `--monthly_start YYYYMM` and `--monthly_end YYYYMM`.
  Example: include 2017 data if available → `--monthly_start 201701`.
- **Quarterly terminals (four series)**: in S4, set `--quarterly_start YYYYQ` and `--quarterly_end YYYYQ`.
  Example: extend to 2025Q2 → `--quarterly_end 2025Q2`.
- **Winsor bounds**: `--winsor_low/--winsor_high` in S4 (applies to both monthly and quarterly `w`).

Workflow when changing windows: re-run **S4** with new flags → re-run **S5** (stacker). S1–S3 usually do not need to change unless you updated the raw input files.

5) Acceptance Criteria (before using `LP_Panel.tsv`)

1. **Keys unique by grain** in each stage’s outputs (S1–S5).
2. **Additivity (S2)**: `TEU_port_q` equals sum of terminal-quarter `TEU_i_q`.
3. **Mean(w)=1 by (port, year)** in monthly and quarterly outputs (S4 QA and S5 QA summary).
4. **Annual preservation (monthly)**: within (port, year), **abs(mean(LP) − mean(Π)) ≤ 0.05**.
5. **No forbidden rows**: no `Eilat`, no `All Ports`.
6. **LP arithmetic**: `LP` equals `w × Π` (exact up to floating noise).

6) Common Pitfalls & Remedies

- **Mixed frequency confusion (TEU)**: Never infer monthly TEU from quarterly (or vice versa). Always branch by `Freq`.
- **Terminal naming drift**: Use the canonical mapping in S2/S3 (e.g., `Haifa SIPG` → `Haifa-Bayport`).

- **Tons precedence**: If terminal rows exist for a (port, month), prefer their sum over the port row.
- **Zero/negative TEU/tons**: Ratios for `w` and `LP_id` are masked to NA. Investigate large blocks of zeros.
- **Outlier tails**: If `w` looks volatile, adjust `--winsor_low/high` slightly (e.g., 0.02/0.98).

---

7) Regenerate From Scratch — One-Liners

```
# S1 — Tons
python "Data/LP/Build_LP_Panel_S1_Tons.py"
  --tons "Data/Output/monthly_output_by_1000_tons_ports_and_terminals.tsv"
  --out  "Data/LP"

# S2 — TEU
python "Data/LP/Build_LP_Panel_S2_TEU.py"
  --teu "Data/Output/teu_monthly_plus_quarterly_by_port.tsv"
  --out "Data/LP"

# S3 — L_Proxy
python "Data/LP/Build_LP_Panel_S3_LProxy.py"
  --lproxy "Data/L_proxy/L_Proxy.tsv"
  --s2_term_quarter "Data/LP/S2_terminal_quarter_teu.tsv"
  --out "Data/LP"

# S4 — LP series (adjust ranges/winsor as needed)
python "Data/LP/Build_LP_Panel_S4.py"
  --winsor_low 0.01 --winsor_high 0.99
  --monthly_start 201801 --monthly_end 202108
  --quarterly_start 2021Q3 --quarterly_end 2024Q4
  --out "Data/LP"

# S5 — Stack into LP_Panel
python "Data/LP/Build_LP_Panel_S5_Stack.py" --out_dir "Data/LP"
```

---

8) Appendix

A) Canonical terminal mapping

- **TEU file (`Port` → terminal)**:
- `Haifa SIPG` → `Haifa-Bayport` (port: `Haifa`)
- `Ashdod HCT` → `Ashdod-HCT` (port: `Ashdod`)
- `Haifa` → `Haifa-Legacy` (port: `Haifa`)
- `Ashdod` → `Ashdod-Legacy` (port: `Ashdod`)
- **L_Proxy file (common variants)**:

- `Haifa SIPG`, `Haifa Bayport`, `Haifa-SIPG` → `Haifa-Bayport`
- `Ashdod HCT`, `Ashdod Hct` → `Ashdod-HCT`
- `Haifa Legacy`, `Haifa-Legacy` → `Haifa-Legacy`
- `Ashdod Legacy`, `Ashdod-Legacy` → `Ashdod-Legacy`

B) Schemas at a glance

- **S1_terminal_month_tons.tsv**: `port, terminal, year, month, month_index, tons_i_m`
- **S1_port_month_tons.tsv**: `port, year, month, month_index, quarter, tons_port_m, tons_source`
- **S1_port_quarter_tons.tsv**: `port, year, quarter, tons_port_q`
- **S2_port_month_teu.tsv**: `port, year, month, month_index, TEU_port_m, is_pre_reform`
- **S2_terminal_quarter_teu.tsv**: `port, terminal, year, quarter, TEU_i_q`
- **S2_port_quarter_teu.tsv**: `port, year, quarter, TEU_port_q`
- **S3_lproxy_clean.tsv**: terminal×month rows; key columns canonicalized; `month_index`, `quarter`; numeric types coalesced
- **S3_port_month_labor.tsv**: `port, year, month, month_index, L_hours_port_m`
- **S3_terminal_year_pi.tsv**: `terminal, year, Pi_teu_per_hour_i_y`
- **LP_Haifa/Ashdod_port_month.tsv**: `port, year, month, month_index, quarter, TEU_port_m, tons_port_m, tons_source, w, w_source, Pi_port_q, LP_mix, LP_id`
- **LP_*_quarter.tsv**: `port, terminal, year, quarter, TEU_i_q, w, Pi_teu_per_hour_i_y, LP_mix`
- **LP_Panel.tsv**: `series_id, level, freq, port, terminal, year, month, quarter, month_index, quarter_index, TEU, tons, L_hours, w, Pi, LP, LP_id, tons_source`

C) Glossary

- **Π (Pi)**: terminal-year intrinsic productivity (`Pi_teu_per_hour_i_y`), used in mix baselines.
- **w (mix factor)**: winsorized tons/TEU ratio, rebased to mean 1 by (port,year).
- **LP_mix**: mix-adjusted LP = `w × Π` (monthly port or terminal quarterly, depending on series).
- **LP_id**: identity diagnostic = TEU / labor hours (monthly port only).
- **month_index**: `year*12 + month` (Int64).
- **quarter_index**: `year*4 + qcode(Qk)` (Int64).

End of report.