Thesis VSCode Hierarchy Report — Paths, Roles, and Workflow (v2025-11-06)

This report summarizes the folder/file layout shown in your VS Code workspace and explains *where the key assets live*, *what each does*, and *how they connect to the econometric workflow*. Paths are written **relative to the repository root** `THESIS/`.

---

1) Top-level structure (root)

```
THESIS/
├─ .venv/                        # local Python environment (packages, interpreter)
├─ Data/                         # raw and constructed data (by domain)
└─ Design/                       # code, model configs, run outputs, diagnostics, visuals
```

What matters most: code lives under `Design/Code/…`, canonical analysis-ready outputs land in `Design/Output Data/`, while the *sources* and *constructed panels/proxies* sit under `Data/`.

---

2) Data/ — by domain

```
THESIS/Data/
├─ K/                            # capital (K) tracks, QA, and K/L mediator exports
├─ K_over_L instruments (V1)/    # dated engineering milestones (candidate instruments)
├─ L_proxy/                      # labor-hours proxy builds (terminal×month), QA, metadata
├─ LP/                           # LP pipeline products (S1–S5 outputs + stacked panel)
├─ LP (old)/                     # prior versions of LP artifacts (archive)
├─ Output/                       # tidy output backbones: TEU and tons series
├─ Raw data/                     # raw source dumps (CSVs/TSVs from PDFs, Excels, etc.)
└─ Time_controls/                # calendar controls: COVID, security-shock dummies, etc.
```

2.1 Data/K/ (capital tracks & mediators)

Purpose: Track-B (PPE/PIM) series, QA, and a prebuilt K/L mediator.

Key files: - `Data/K/_meta_K_B_Haifa_Legacy.json` — build meta (periods, coverage, splices). - `Data/K/haifa_financials_raw.tsv` — raw financials (annual/quarterly) used for Track-B. - `Data/K/K_B_monthly_Haifa_Legacy.tsv` — **monthly K series (Track-B) for Haifa-Legacy**. - `Data/K/qa_K_B_Haifa_Legacy.tsv` — QA checks for Track-B (continuity, splices, levels). - `Data/K/TrackB_Haifa_HPC_exc.2022.tsv` — Track-B variant excluding a revaluation window. - `Data/K/Mediator_K_over_L.tsv` — **tidy mediator**: `ln(K) − ln(L)` keyed by port×time. - `Data/K/Build_K/` *(folder)* — helper scripts/notes for capital construction.

Use in Model-1: The mediator file exists but is **not used** in the non-mediation pass.

2.2 Data/L_proxy/ (labor-hours proxy)

Purpose: Credible labor-hours at terminal×month, with identity-preserving annual totals.

Key files: - `Data/L_proxy/_meta_l_proxy.json` — build provenance & parameters. - `Data/L_proxy/containers_kpis_annual_wide_filled.tsv` — TEU/hour (ops productivity) wide table. - `Data/L_proxy/L_Proxy.tsv` — **canonical labor proxy** (terminal×month) consumed upstream. - `Data/L_proxy/labor_hours_monthly_terminal.tsv` — terminal×month L hours (denormalized export). - `Data/L_proxy/labor_hours_monthly_port.tsv` — port-level monthly L hours (aggregated). - `Data/L_proxy/labor_hours_QA.tsv` — coverage, pre-opening zeros, annual add-up checks. - `Data/L_proxy/labor_hours_missing_delta.tsv` (+ `.meta.json`) — delta audits vs baselines. - Subfolders: `construct_L/`, `build_labor_proxy_backfill/`, `Join/`, `verify_delta/` — scripts & logs.

2.3 Data/LP/ (LP pipeline S1–S5)

Purpose: Mixed-frequency LP pipeline outputs and the stacked panel consumed by design code.

Key files: - **Terminal-quarter LP:** - `Data/LP/LP_Haifa_SIPG_quarter.tsv` - `Data/LP/LP_Haifa_Legacy_quarter.tsv` - `Data/LP/LP_Ashdod_Legacy_quarter.tsv` - **Port-month LP:** - `Data/LP/LP_Haifa_port_month.tsv` - `Data/LP/LP_Ashdod_port_month.tsv` - **Stacked mixed-freq panel:** - `Data/LP/LP_Panel.tsv` ← **canonical stacked LP (levels)** used downstream. - **S-stage artifacts & QA:** - `Data/LP/S1_port_month_tons.tsv`, `Data/LP/S1_port_quarter_tons.tsv`, `Data/LP/S1_qa.tsv` - `Data/LP/S2_port_month_teu.tsv`, `Data/LP/S2_port_quarter_teu.tsv`, `Data/LP/S2_terminal_quarter_teu.tsv`, `Data/LP/S2_qa.tsv` - `Data/LP/S3_coverage_gaps.tsv`, `Data/LP/S3_coverage_vs_s2.tsv` - `Data/LP/S1_examples_port_precedence.tsv` — precedence examples (sum terminals > port row) - `Data/LP/qa_lp_report.tsv`, `Data/LP/Recheck_S4A_QA_Relaxed.py`

2.4 Data/Output/ (tidy backbones)

Purpose: Clean TEU and tons series used by the LP and L_proxy builds.

Key files: - `Data/Output/monthly_output_by_1000_tons.tsv` - `Data/Output/monthly_output_by_1000_tons_ports_and_terminals.tsv` - `Data/Output/teu_monthly_by_port.tsv` - `Data/Output/teu_quarterly_by_port.tsv` - `Data/Output/teu_monthly_plus_quarterly_by_port.tsv`

(Other Data folders such as `Raw data/`, `Time_controls/`, and `K_over_L instruments (V1)/` store the raw PDFs/Excels converted to TSV, event-instrument calendars, and calendar dummies; they follow the same tidy naming conventions.)

---

3) Design/ — code, configs, outputs

```
THESIS/Design/
├─ Code/
│  ├─ Econometrics/
│  │  ├─ model1_combine_lp_quarter.py   # merges quarterized port rows → canonical LP (quarter)
│  │  ├─ model1_params.yaml             # event dates, binning, shock toggles, paths
│  │  └─ panel1_build.py                # builds port×quarter panel_port_quarter_full.csv
│  ├─ Old code/                         # archived scripts
│  └─ Visualization/                    # plotting utilities
│
├─ Input Data/                          # (reserved) external inputs used by code
├─ Output Data/
│  ├─ _meta_panel_port_quarter_full.json
│  ├─ LP_Panel_quarterized.tsv          # **canonical quarterized LP (levels)**
│  └─ panel_port_quarter_full.csv       # port×quarter LP assembled from sources
│
├─ diagnostics/                         # logs, QA summaries from code runs
└─ visuals/                             # saved figures (ES paths, pretrends, etc.)
```

How the three econometrics scripts interact: 1) `Design/Code/Econometrics/panel1_build.py` → reads `Data/LP/LP_Panel.tsv` and prioritizes sources to create `Design/Output Data/panel_port_quarter_full.csv` (LP **levels** at port×quarter) with a small meta JSON. 2) `Design/Code/Econometrics/model1_combine_lp_quarter.py` → takes `Data/LP/LP_Panel.tsv`, drops monthly rows, **anti-joins** the quarterized port rows from Step 1, and writes the **5-column** canonical `Design/Output Data/LP_Panel_quarterized.tsv` (`port, terminal, year, quarter, lp`). 3) Estimation (not shown in screenshots) will consume `LP_Panel_quarterized.tsv`, aggregate terminals via **mean(log LP)** to **port×quarter**, attach **event clocks** from `model1_params.yaml`, add **time controls**, then run the **NYT event-study**.

---

4) Quick access — high-priority files & paths

- **Stacked LP (mixed-freq)** — `Data/LP/LP_Panel.tsv`
- **Quarterized LP (canonical)** — `Design/Output Data/LP_Panel_quarterized.tsv`
- **Port×quarter build (source mix)** — `Design/Output Data/panel_port_quarter_full.csv`
- **Build meta** — `Design/Output Data/_meta_panel_port_quarter_full.json`
- **Labor proxy (terminal×month)** — `Data/L_proxy/L_Proxy.tsv`
- **Mediator ln(K/L)** — `Data/K/Mediator_K_over_L.tsv`
- **Econometrics YAML** — `Design/Code/Econometrics/model1_params.yaml`
- **Build scripts** — `Design/Code/Econometrics/panel1_build.py`, `model1_combine_lp_quarter.py`

---

5) Minimal workflow pointers (non-mediation Model-1)

1. **Confirm inputs**: `Data/LP/LP_Panel.tsv` exists; time-control TSVs present; YAML set.
2. **Run Step A**: `panel1_build.py` → `Design/Output Data/panel_port_quarter_full.csv` (+ meta).
3. **Create canonical LP (quarter)**: `model1_combine_lp_quarter.py` → `Design/Output Data/LP_Panel_quarterized.tsv`.
4. **Estimation** (not shown here): aggregate to port×quarter using mean(log LP), merge event clocks from YAML, add shocks, produce ES figures and CSVs in `Design/Output Data/` and `Design/visuals/`.

Notes & conventions to remember

- Keep **LP in levels** in files; take logs **inside estimation**.
- **Terminal is blank** (`""`) for pre-reform port-level rows; post-entry splits into `Legacy` and entrant (`SIPG/Bayport`, `HCT`).
- Maintain **additivity** (sum terminal-quarter TEU = port-quarter TEU) and **mean(w)=1)** by (port,year) upstream.
- Small-N inference and **time controls** (COVID 2020–2021; security shock late-2023/2024) live in the estimation stage.

End of report.