



FREE WORKFLOW PACK / VERSION 1.2.0

Demand Sensing Router

Turn demand history and fresh operating signals into a planner-reviewable cadence decision for every SKU-location.

START IN 5 MINUTES

Run one local command with synthetic data. It profiles four SKUs, validates the known-good board, and shows the exact next files to replace.

THE DECISION

Route each SKU-location to one of three lanes:

- SENSE FREQUENTLY when a fresh signal can still change an operating decision.
- PLAN MONTHLY when the baseline cadence still fits the decision horizon.
- SPECIAL METHOD / REVIEW for intermittent, lumpy, lifecycle, or weak-evidence exceptions.

1 SAFE START	2 INPUT FILES	5 REVIEW OUTPUTS
--------------	---------------	------------------

HUMAN BOUNDARY

The pack prepares evidence and a recommendation. The planner approves policy, cadence, master data, and every system-of-record change.



01 / FIRST RUN

Prove the pack before using your data

Open Terminal in the extracted pack folder and run one command:

```
python3 START.py
```

**EXPECTED
RESULT**

4 profiles written / known-good routing board passed / next files printed. No external service or production data is used.

What the command proves

Python can read the included history and calculate ADI, CV2, and demand-pattern evidence.

The routing-board validator rejects unsupported frequent-sensing recommendations.

The package structure, examples, and expected output contract are intact.

Completed sample: what good looks like

SKU / location	Pattern evidence	Route	Why it belongs there
A100 / TOR	Smooth + fresh POS	Sense frequently	A 3-day response window can still change replenishment.
C310 / VAN	Stable + long horizon	Plan monthly	New information does not change the near-term decision.
D440 / MTL	Intermittent	Special review	Long zero gaps need a method decision, not a faster cadence.
F660 / TOR	Phase-in	Special review	Lifecycle evidence overrides a simplistic history route.

Choose the lightest tool path

Path	Best first use	Start with
Chat first	Learn the method with minimal setup	Run START.py, attach four files, paste the starter prompt.
Agent Skill	Reuse the method in a compatible tool	Install the complete skill folder; keep scripts and references together.
Claude Project / Cowork	Give a team one controlled workspace	Attach the skill, workers, and input files; run manually first.

02 / YOUR DATA

Replace two files, preserve the evidence contract

FILE 1 Demand history Copy sample-demand-history.csv to my-demand-history.csv. - sku / location: stable planning keys - period: one daily or weekly format - demand: numeric actual demand - one row per SKU-location-period	FILE 2 Routing context Copy sample-routing-context.csv to my-routing-context.csv. - dated signal + freshness - open response window - lifecycle + portfolio priority - decision supported + planner comment
--	---

Routing context: the fields that stop generic AI output

Context field	What it must answer	Example
signal_type + signal_date	What changed, and when was it observed?	POS / 2026-07-15
signal_freshness_days	Is the evidence recent enough for this review?	2
response_window_days	Can the supported decision still change?	3
lifecycle	Does phase-in/out require special treatment?	ACTIVE
decision_supported	Which operating decision can move?	REPLENISHMENT
previous_cadence	Did the route change since last review?	PLAN MONTHLY

NON-NEGOTIABLE A frequent-sensing route needs a dated trigger signal, an open response window, and a named operating decision. Missing evidence stays in planner review.

Run your history

```
python3 skills/demand-sensing-router/scripts/profile_demand.py my-demand-history.csv --output my-demand-profile.csv
```

Then attach the profile, context, routing policy, and data contract with the supplied chat-first or Cowork prompt.

03 / PLANNER REVIEW

Start with the exceptions, not the stable rows

Review order	Artifact	The question it should answer
1	Route-change log	Which SKU-location moved, and what new evidence caused it?
2	Data-quality exceptions	Which recommendation failed because evidence is missing, stale, or conflicting?
3	Demand-review brief	Which open response window or operating decision needs attention now?
4	Routing board	Does every lane match the policy and the visible evidence?
5	Run manifest	Which sources, dates, policy version, and checks produced this package?

Four planner questions

1. What evidence caused the route to move?	2. Is the signal fresh enough to matter?
3. Which decision can still change?	4. What evidence still needs to be challenged?

Fast troubleshooting

Problem	Fix
python3 not found	Use a managed Python 3 environment; the pack needs only the standard library.
Missing-column error	Copy the supplied CSV header exactly before replacing rows.
Duplicate-row error	Keep one row per SKU-location-period.
Frequent-sensing route rejected	Add a dated signal, open window, and supported decision or keep it in review.
Sensitive data concern	Anonymize identifiers or use an approved enterprise AI environment.

Ready-to-run checklist

☐ START.py passes with the synthetic sample.

☐ My two input files preserve the supplied fields and grain.

☐ Every frequent-sensing route has dated evidence and an open decision.

☐ Changed routes explain why they moved; failed checks stay visible.

☐ A planner approves the final cadence and any system update.

NEXT MOVE

Run the sample, inspect the completed outputs, copy the two input files, and build the first board with governed data. Keep the originals as your known-good reference.