# Master Dashboard Audit — Safe‑Route Plan & Weekly Spot‑Check (v0.1)

You chose the **safe route** (good call). Below is a compact plan you can keep in the repo for weekly spot‑checks and a roadmap to **hone the audit** specifically for the **Master Dashboard**.

## 1) Where this sits in the pipeline

A1 Build → **A2 Enrich** (lines + Off/Def DVOA) → A3 Kickoff helpers → **Audit (safe baseline)** → Export Member HTML → (later) Master Dashboard.

## 2) Weekly spot‑check (5‑minute pass)

Paste from repo root; ranges quoted to avoid PowerShell parsing issues.

# Safe baseline audit (includes lines consistency when --roadmap is provided)  
python -m scripts.millions\_audit\_week --season 2025 --week 1 `  
 --planner "picks/millions/millions\_planner.csv" `  
 --roadmap "picks/millions/millions\_roadmap\_game.csv" `  
 --out\_dir "picks/millions/diagnostics" `  
 --fail\_on\_missing 0.01 `  
 --spread\_range="-20,20" `  
 --dvoa\_range="-0.6,0.6" `  
 --line\_tolerance 0.5

Open: picks/millions/diagnostics/millions\_week\_audit.md. If **no red** and only acceptable yellows, export member HTML.

Optional quick sanity:

python - <<'PY'  
import pandas as pd  
p = pd.read\_csv(r"picks/millions/millions\_planner.csv")  
cols = [c for c in (  
 'team\_off\_dvoa\_proj','team\_def\_dvoa\_proj','opp\_off\_dvoa\_proj','opp\_def\_dvoa\_proj'  
) if c in p.columns]  
for c in cols:  
 s = pd.to\_numeric(p[c], errors='coerce')  
 if s.notna().any():  
 print(f"{c:22s} min={s.min(): .3f} max={s.max(): .3f} mean={s.mean(): .3f}")  
print('\nKickoff key present:', '\_kickoff\_sort\_key' in p.columns)  
PY

## 3) Safe‑baseline checks (what the audit enforces today)

* **Schema**: season, week, team, opponent present.
* **Missingness** (≤1% on key fields): team, opponent, circa\_line, team\_total\_dvoa\_proj, opp\_total\_dvoa\_proj.
* **Ranges**: spreads in **[−20, +20]**; total DVOA in **[−0.6, +0.6]** (fractional; exporter formats %).
* **HOME/AWAY** validity; **duplicate matchups**;
* **Alias** optional (when you pass an alias file);
* **Lines consistency** vs roadmap with tolerance (when --roadmap provided).

This baseline is intentionally conservative to prevent noisy failures during weekly runs.

## 4) Master Dashboard — honing plan (next iterations)

Add these checks (toggleable with CLI flags) to tighten master data without breaking weekly flow:

1. **Totals coverage & range**
   * Require open\_total OR current\_total per game.
   * Range sanity (e.g., 30–70 by default), CLI‑tunable.
2. **Line deltas**
   * current\_spread\_\* − open\_spread\_\* magnitude flag (e.g., > 3.0 pts).
   * Trend direction notes (info only).
3. **DVOA scale guardrail**
   * Auto‑detect percent‑points (abs max > 1.5) and warn/fix suggestion.
4. **Kickoff ordering invariant**
   * Assert \_kickoff\_sort\_key exists and is strictly increasing within weekday buckets; Monday last.
5. **Join coverage**
   * Planner↔Roadmap key coverage: 100% match for the target week; list unmatched.
6. **Schema drift**
   * Fail fast on unexpected duplicate columns or \*\_x/\_y suffixes.

We’ll wire these behind flags like --require\_totals 1, --max\_line\_delta 3.0, etc., then set stricter defaults for the **Master Dashboard** job while leaving the **Member** job on the safe baseline.

## 5) Export member HTML (only after audit passes)

python scripts/millions\_export\_member\_html.py `  
 --season 2025 `  
 --week 1 `  
 --planner "picks/millions/millions\_planner.csv" `  
 --out "picks/millions/exports/member\_week\_1.html" `  
 --sort\_by kickoff `  
 --show\_rest 0

## 6) Config knobs you can tune per run

* **Missingness threshold**: --fail\_on\_missing 0.01 (1%).
* **Spread range**: --spread\_range="-20,20".
* **DVOA range**: --dvoa\_range="-0.6,0.6" (fractions).
* **Lines tolerance**: --line\_tolerance 0.5 when comparing to roadmap.

Use equals form (--arg="value") in PowerShell to avoid the *expected one argument* error.

## 7) Backlog for the audit tool (when ready)

* Add **Totals** to roadmap → planner → exporter; audit both presence and range.
* Add **injury/weather badges** to exporter; audit presence of injuries\_key\_\* when Master Dashboard uses them.
* Emit a tiny **HTML summary** alongside the Markdown for quick browser viewing.
* Make a single make/batch target to run: build → enrich → audit → export.

## 8) What to commit now

* This doc (e.g., docs/master\_audit\_plan\_v0.1.md).
* The audit run output in picks/millions/diagnostics/ (kept in .gitignore if you prefer local only).
* Any exporter patches you’ve already applied (kickoff sorting, badge logic).

**TL;DR:** Keep the safe baseline for weekly member exports, do a 5‑minute spot‑check using the command above, and we’ll add stricter checks (totals, deltas, invariants) as we bring the **Master Dashboard** online.