MAP-Style Homicide Cluster Analyzer

A concise, field-ready guide for law-enforcement analysts

This README explains what the tool does, how to run it, and how to interpret the results when screening homicide data for low-clearance clusters. It assumes a single script named `map_cluster.py` (the one we've been iterating on).

---

1) What this tool does (in plain terms)

Goal: Rapidly surface *where and how* homicides go unsolved in your jurisdiction(s), so command staff can direct investigative reviews, quality-assurance (QA) checks, and targeted interventions.

Method (MAP-style, adapted): 1. **Recodes "solved"** from your data (default: use the dataset's own `Solved` field; the old Offender-Sex proxy is still available but discouraged). 2. **Derives victim-sex code** (1=Male, 2=Female, 9=Unknown) and creates **cluster IDs**: - **County view** (MURDGRP1) = County + Victim-Sex + Weapon - **MSA view** (MURDGRP2) = MSA + Victim-Sex + Weapon - If county/MSA is text, a **stable hash fallback** is used so groups don't collapse. 3. **Aggregates** cases per cluster: TOTAL, SOLVED, PERCENT cleared, and UNSOLVED. 4. **Optional filters**: female/male/all; clearance threshold; minimum support; decade slicing; metadata completeness stats for **Relationship** and **Circumstance**. 5. **Outputs** ranked CSVs and a console preview.

Key idea: Two recurring "low-clearance" archetypes usually appear: - **Method-driven** (e.g., strangulation/hanging). - **Data-gap–driven** (e.g., "Firearm, type not stated", "Other/unknown" weapon) where metadata holes travel with low clearance.

---

2) Data requirements (minimum viable columns)

Your CSV should include headers (case-insensitive accepted if you normalized them). Critical fields: - `CNTYFIPS` (text or numeric OK), `MSA` (text or numeric OK) - `VicSex`, `OffSex`, `Weapon` - `Solved` (values like `Yes/No` or `Y/N`) ← **recommended truth source** - Useful context (if present): `Relationship`, `Circumstance`, `Situation`, `Year`, `Month`, `Ori`, `Agency`, `VicAge`, `OffAge`

Notes: - Numeric sentinels like `OffAge=999` are treated as unknowns. - If a county/MSA is a name (e.g., "Anchorage, AK"), the tool uses the label and a stable hash so clusters remain distinct.

---

3) Installation (once per workstation)

- Python 3.9+ and `pip install pandas`.
- Save the script as `map_cluster.py` in a working folder.

- Your homicide CSV goes in the same (or supply full path).

Windows tip: Use `cmd.exe` or PowerShell. Examples below use `python` on PATH.

---

4) Quick start (copy-paste ready)

Baseline female/MSA scan (modern era, ≥2010):

```
python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female ^
  --relcirc --min-decade 2010 --min-total 15 --threshold 0.33 --top 20 --outdir out
```

County view:

```
python map_cluster.py SHR.csv --group county --solved-source field --focus-sex female ^
  --relcirc --min-decade 2010 --min-total 15 --threshold 0.33 --top 20 --outdir out
```

Tighten the bar (modern, ≥20 cases, ≤30% cleared):

```
python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female ^
  --relcirc --min-decade 2010 --min-total 20 --threshold 0.30 --top 20 --outdir out
```

Case-level dump for a flagged cluster:

```
python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female ^
  --dump-msa "St. Louis, MO-IL" --dump-weapon "Firearm, type not stated" ^
  --dump-out out/stl_firearm_not_stated_cases.csv --outdir out
```

Batch sweep (Windows CMD) over thresholds):

```
for %t in (0.25 0.28 0.30 0.33) do python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female --relcirc --min-decade 2010 --min-total 20 --threshold %t --top 10 --outdir out
```

---

5) Output files & how to read them

- `AGGREGATE_COUNTY.csv` or `AGGREGATE_MSA.csv`: all clusters (no filter), sorted by UNSOLVED.
- `FILTERED_*.csv`: the filtered/thresholded list shown in the console preview.
- `WEAPON_CODEBOOK.csv` (optional): mapping of weapon string→code for the run.
- Optional **case dumps**: per your `--dump-*` flags.

Important columns: - `PERCENT` = clearance rate in that cluster; `UNSOLVED` = TOTAL − SOLVED. - `REL_UNK_RATE` and `CIRC_UNK_RATE` = share of cases where Relationship/Circumstance are unknown/undetermined/unspecified/blank. - `REL_TOP1` / `CIRC_TOP1` = most frequent known category (e.g., `Acquaintance`, `Rape`, `Other arguments`). - `REPORT_GAP_IDX` (if present) ≈ average of the two unknown rates.

Interpretation pattern: - **Method-driven pockets** (e.g., *Strangulation - hanging*): often operational challenge even with decent metadata. - **Data-gap pockets** (e.g., *Firearm, type not stated*): usually correctable via QA/training; clearance tends to improve when metadata improves.

---

6) Flags (cheat sheet)

Flag	What it does	Typical values
`csv` (positional)	Input CSV path	`SHR65_23.csv`
`--group`	Cluster by county or MSA	`county` | `msa`
`--solved-source`	How to mark SOLVED	`field` (recommended) | `offsex` (legacy proxy)
`--focus-sex`	Victim sex filter	`female` (default) | `male` | `all`
`--threshold`	Keep clusters with **PERCENT ≤ t**	`0.33` default; try `0.30`, `0.25`
`--min-total`	Require at least N cases in a cluster	e.g., `15`, `20`
`--by-decade`	Adds DECADE to grouping/outputs	toggle
`--min-decade`	Drop cases before given decade	`2000`, `2010`
`--relcirc`	Adds Relationship/Circumstance stats	toggle
`--min-known-rel`	Require share of known Relationship ≥ k	e.g., `0.30`

Flag	What it does	Typical values
`--top`	How many rows to print	e.g., `10`, `20`
`--outdir`	Output directory	`out`
`--no-filter`	Produce aggregates only	toggle
`--dump-msa`, `--dump-weapon`, `--dump-out`	Export case-level rows matching **MSA + weapon**	strings + path

---

7) Recommended analyst workflow (LEA context)

1. **Scan (broad):** `--group msa --solved-source field --focus-sex female --relcirc --min-total 15 --threshold 0.33` (optionally add `--min-decade 2000`).
2. **Validate:** sanity-check that `PERCENT` is consistent with the dataset's `Solved` values (you're already using `--solved-source field`).
3. **Triage:** separate **method-driven** vs **data-gap–driven** clusters. Modern, large-support pockets (e.g., ≥2010, ≥20 cases) get priority.
4. **Deep-dive:** use case dumps to sort by **ORI/Agency**, year, `Relationship`, `Circumstance`. Look for one or two submitters driving the pocket.
5. **Action:**
6. **Data gaps:** coding retraining, form fixes, NIBRS/SHR crosswalks.
7. **Method pockets:** specialized investigative playbooks (e.g., asphyxia: ligature trace, forensic timelines, victimology linkage).
8. **Re-run** with the same flags to measure lift post-intervention.

---

8) Hiring note: what skills this analyst needs

- **Pandas proficiency** (groupby, filtering, joins), **CSV hygiene**, and comfort with Windows/PowerShell.
- Ability to produce **actionable triage memos**: explain *why* a cluster popped (method vs data gap), *who* (agencies/ORIs), *when* (decade), and *what next* (QA or investigative).
- Basic **CJIS/PII** hygiene and chain-of-custody discipline for data extracts.

---

9) Troubleshooting (fast answers)

- **Empty table:** Your thresholds are too strict. Loosen `--threshold` (e.g., `0.33`) or lower `--min-total`, or widen the era via `--min-decade 2000`.
- **"Anchorage/Unknown" mega-clusters:** Use the latest script with **hash fallback** and label grouping; switch to `--group msa` if county codes are messy.
- **OffSex proxy inflated unsolved:** Always use `--solved-source field` when the dataset has a `Solved` column.

- **KeyError on label columns:** Don't aggregate a column that's also in your groupby keys (we patched `aggregate`).
- **Regex "unknown" didn't count 'undetermined':** The updated helper counts `unknown/not determined/undetermined/unspecified/blank`.

---

10) Privacy, security & ethics

- Keep extracts on secured drives; follow agency data-handling SOPs.
- Limit case-level dumps to personnel with a need-to-know; redact PII where policy requires.
- Treat "data-gap" findings as opportunities for **training and system fixes**, not blame. The target is **better clearance**, not scorekeeping.

---

11) Appendix: example recipes

Modern near-outliers (female/MSA):

```
python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female ^
  --relcirc --min-decade 2010 --min-total 20 --threshold 0.33 --top 20 --outdir out
```

Method pockets only (strangulation):

```
python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female ^
  --relcirc --min-decade 2000 --min-total 15 --threshold 0.33 --top 20 --outdir out
```

Data-gap pockets only (weapon under-specified):

```
python map_cluster.py SHR.csv --group msa --solved-source field --focus-sex female ^
  --relcirc --min-decade 2000 --min-total 15 --threshold 0.33 --top 20 --outdir out
```

Version notes - Hash fallback for text county/MSA is deterministic (MD5 modulo). Re-running with the same inputs yields the same clusters. - The tool never uploads data; all processing is local.

Prepared for agency leadership evaluating analyst workflows that use the MAP-style clustering approach.