R Codes (Scoring Logic)

Below is the **current R-score catalog** the war-room has approved so far. The fields ARE NOT EXACT, JUST IDEAS. We'll have to match it to the correct NON-NULL fields.

Use it as the single source of truth for your Signal-Engine v1.1.

-- "Pts" are the *default weights*; tweak later with outcome data.

#	Rule (trigger)	Points	Data source / field
R1	Probate filing ≤ 90 days old	**+15**	`filing_date`
R1.5	Probate mailing **matches** property/site address	**+10**	`probate_addr` vs `site_address`
R2	Probate **status = Open**	**+25**	`status`
R3	Probate subtype contains **Affidavit** or **Administration**	**+15**	`subtype`
R4	Decedent **last-name present**	**+10**	`decedent_last_name`
R5	Decedent full-name has **≥ 2 tokens**	**+10**	`decedent_full_name`
R5.5 notes on this	*Common surname* **and** base score < 60	**−5**	`decedent_last` ∈ `COMMON_SURNAMES`
R6	**≥ 1 deed match** to decedent/estate	**+15**	`rp_match_count ≥ 1`
R7	Deed still in **decedent's name** (no release)	**+20**	`deed_released = False`
R8	**≥ 2 probate surnames** hit property deeds	**+5**	`matched_names_count ≥ 2`
R9	**Current-year taxes delinquent**	**+15**	`total_current_due > 0`
R10	**Prior-year taxes owed**	**+25**	`prior_years_due > 0`

R11	**Homestead exemption present**	**−10**	`homestead_exemption = True`
R12	HCAD **owner last = grantee last**	**+10**	`owner_last == grantee_last`
R12.5	**Out-of-state mailing address**	**+15**	`owner_state ≠ property_state`
R13	HCAD **owner last = decedent last** (estate not retitled)	**+20**	`owner_last == decedent_last`
R14	**Year built < 1960**	**+10**	`year_built`
R15	**Poor condition** or **Grade C/C-**	**+15**	`physical_condition` , `grade`
R16	**Land-to-total value ≥ 0.40**	**+10**	`land_val / total_val`
R16.5	**Garage present** (`garage_area > 0`)	**+2**	`garage_area_sqft`
R17	**Large lot ≥ 7 500 sq ft**	**+5**	`lot_sqft_total`
R17.5	**Multi-segment lot** (`land_line_count > 1`)	**+5**	parsed land rows
R18	Probate filed **6–12 mo ago** → +10**> 12 mo** → +20	**+10/+20**	`months_since_filing`
R19 *(negative bucket)*	Active MLS listing or sold in ≤ 12 mo	**−20/−30**	future MLS feed
R20 (Neighborhood Demand & Liquidity)	Add points based on the **average time on market for comparable properties** in the immediate vicinity or zip code. Properties in high-demand areas, even if distressed, might be more attractive. A property's value is tied to its location. Understanding that a distressed property is	+5 to +15	Data Source: MLS data, historical sales data

	in a high-demand area where homes sell quickly adds a layer of confidence for an investor. This tells them their exit will be easier.		
R21 (Title/Lien Clarity)	**simplicity of the title chain** or absence of other major, known liens (e.g., city liens, HOA liens beyond taxes) could be a strong positive, as complex title issues are major deal killers.	+10 to +20	Data Source: Public lien records, title search data
R22 (Equity & Loan-to-Value Signals) notes on this	Incorporate a rule that considers the **estimated equity** in the property based on current market value vs. outstanding mortgage balances (if available from public records). Properties with high equity are often easier to acquire for less than market value.	+15 to +25	Data Source: Property valuation models, public mortgage records
R23 - Appraisal Velocity	For each property, parse the historical appraisal data. Calculate the year-over-year percentage increase for the last two years. A property that has been appreciating at 10-15% per year is in a fundamentally different class than one that is stagnant,	Tiers: >10% Avg YoY Growth +15 pts 3-10% Growth +5 pts	hcad_appraised_history_json

	even if their current values are similar.		
R27 - Prime Redevelopment Candidate	This combines several of the existing data points into a more potent signal. A property might be a tear-down candidate if it has low improvement value relative to the land value, is in poor condition, and sits on a sizeable lot. Create a composite rule. This specifically targets properties where the value is in the dirt, not the structure—a prime target for builders and flippers."	+25	IF `hcad_land_market_value_total` is greater than `hcad_improvement_market_value` AND `hcad_physical_condition` is 'Poor' or 'C-' AND `hcad_lot_sqft_total` is large (e.g., > 7500 sq ft)
R28 - Entity Ownership Signal	Analyze the `hcad_owner_full_name` field for keywords that indicate corporate or trust ownership.	+10	hcad_owner_full_name `IF "LLC", "TRUST", "INC", or "CORP" is in the owner's name → +10 points` . This becomes even more powerful when you combine it with your existing out-of-state rule (`R12.5`). An out-of-state LLC is a very strong signal."
R24 Negative Permit Status	Building Permits: This is your most structured 'alternative' source. Create a dedicated Apify scraper for the city's building permit portal. You'd search for keywords like 'demolition,' 'stop work order,' or 'emergency repair.' This is a direct signal of physical distress.	+15	the city's building permit portal

R25: Fire and Public Safety Incidents: Verified Fire Incident	This is less structured. Use an Apify actor to scrape the public blotter or news feed of the local fire department. You can't just match addresses; you'll need a basic Natural Language Processing (NLP) layer—which can be a simple API call to a large language model—to interpret the text. For example: "Does the following text describe a significant fire at a residential structure? [Text from blotter]". A confirmed 'yes' could trigger	+30	Public Blotter and News Feeds
R26: Local News & Social Media	This is the most difficult but potentially rewarding. Set up automated searches on local news sites and hyper-local Facebook groups for the property address. A headline like "Car Crashes into Home on 123 Main St" is a high-value signal. This requires the most sophisticated filtering to avoid noise but can uncover distress that no public record will show for months.	+20	
R27: Basic Out-of-State Owner:	Needs to be dynamic. TX is the state for	+10	CASE WHEN mailing_state <>

	now but could change in the future. Account for that FIRST `hcad_mailing_address` or `hctax_mailing_address` state is not 'TX'. A simple but effective absentee signal.		property_state THEN 15 ELSE 0 END AS r12_5_score OR `get_state(hctax_mailing_address) != get_state(hcad_site_address)`
R999 (Owner Demographics/Behavioral Signals - if ethical & permissible)	While sensitive, exploring aggregated, anonymized demographic or behavioral data that might indicate **propensity to sell** (e.g., changes in household composition, public records of retirement, etc., always adhering to privacy regulations). This is a "stretch" but aligns with "Clearbit for Properties."	Future. High Risk	(Points and Data Source: Requires careful ethical and legal review, potentially inferred from public records or specialized third-party data).
R28: Confirmed Absentee Owner:	Both HCAD & HCTAX mailing addresses differ from the site address. A higher confidence absentee signal.	+15	`hcad_site_address != hcad_mailing_address` AND `hctax_site_address != hctax_mailing_address`
R29: High-Confidence Unattended Property	`R28` is true AND external vacancy/utility data confirms it. The gold standard for	+25	Rule `R28` is true AND a USPS vacancy flag or utility shutoff data is present. This is a powerful, high-impact signal.

	identifying vacant properties.		
R30: Administrative Drift:	`hcad_mailing_address` does not match `hctax_mailing_address`. A subtle signal of a less-attentive owner.	+5	◦ `hcad_mailing_address != hctax_mailing_address`
R31: Outdated Materials Signal:	`Roof Type`, `Interior Wall`, etc., suggest key components are at their end-of-life and require capital expenditure.	+10	If `Roof Type` is "Composition Shingle" and the property's `Year Remodeled` is older than 15 years (or null). Or if `Interior Wall` is "Plaster," indicating older construction. This suggests major components are at or near their end-of-life.
R32: Lower-Quality Construction:	`Foundation Type` or `Quality` description points to potential underlying issues or deferred maintenance.	+10	If the `Foundation Type` is "Pier & Beam" on a property built before 1970, it signals a higher potential for costly foundation issues. You can also assign points directly based on the `Quality` field (e.g., 'Average' = +5, 'Fair' = +10, 'Low' = +15).
R33: Value-Add Potential	Ideal For Flippers. `Building Style` is desirable but `Quality` is low, indicating a prime renovation candidate.	+5	If the property has a desirable `Building Style` (e.g., "Bungalow," "Ranch") but a low `Quality` score. This points to a classic "good bones, needs work" scenario that is ideal for flippers.

> Current max theoretical score ≈ +230 (before negatives).
>
> Feel free to rescale or cap later.

Implementation pointers

- **Python-only path**: extend `probate_rules()`, `property_rules()`, `tax_rules()`, and add a new `building_land_rules()` for R14-R17.5.
- **Hybrid Airtable formulas**: create `R14 ... R18` columns exactly as you did for earlier rules (`IF(year_built < 1960, 10, 0)`, etc.) and add them to the master `Signal_Score` roll-up.

All newly scraped fields (`year_built`, `lot_sqft_total`, `land_val`, `physical_condition`, `owner_state`, etc.) just feed into these rules.

You can start with these weights today; after a few closed deals, run a quick regression or weighting tweak (Deming loop) to refine.

A simple summation of points is a great start, but it assumes all these signals are independent. They're not. A `Probate filing ≤ 90 days old` (`R1`) combined with `Prior-year taxes owed` (`R10`) is likely more than the sum of their parts (25 + 25 = 50 pts). This combination tells a story of sudden transition coupled with existing financial strain.

- **Recommendation:** As you gather outcome data, move beyond a simple sum. The mention of a 'Deming loop' to tweak weights is key. I urge you to take this a step further. Your 'R' codes are the features for a machine learning model. Instead of manually tweaking points, you can use logistic regression to determine the optimal weights based on which leads actually convert.
- **A Point of Caution:** Be mindful of rules like `R5.5` **(Common surname and base score < 60, -5 pts)**. This is a heuristic patch for weak entity resolution. While practical, it can also penalize legitimate leads. The better long-term solution is to invest heavily in improving the accuracy of your name-matching algorithms so this rule becomes unnecessary."

"You have a process with defined rules. This is the foundation of quality control. The goal is to continuously reduce variation and improve the predictability of your output—a high-quality lead.

My attention is drawn to the note: 'tweak later with outcome data'. This is the philosophy of the Deming Cycle (Plan-Do-Check-Act). It is good that you plan to do this, but you must be rigorous.

- **Recommendation:** Don't just tweak individual point values. Measure the performance of the *entire system*. For every 100 leads you score above a certain threshold (say, 100 points), how many result in a closed deal? How many were false positives? That percentage is your key quality metric.

- The negative rules, like `R11` **(Homestead exemption present, -10 pts)** and `R19` **(Active MLS listing, -20/-30 pts)**, are critical for reducing noise. Are these penalties aggressive enough? A property on the MLS is fundamentally not 'off-market.' You might consider making `R19` an exclusionary rule, not a point deduction. If a property is on the MLS, it should perhaps be disqualified from this specific 'off-market' engine entirely to ensure the purity of your product. This improves the quality for the customer who is paying you specifically for leads they can't find on Zillow."

You should build a quality check into the system itself. Poor data quality is a risk to your entire scoring model.

- **Process Improvement Idea: Internal Data Quality Score**
  - **Logic:** You have a `needs_review_flag` and an `error` column, which is excellent. Formalize this. Create an internal 'Data Health' score for each record. For every key field that is present, valid, and in the correct format (e.g., `hcad_physical_condition`, `hcad_year_built`, etc.), add a point to this health score.
  - **Implementation:** If a record's 'Data Health' score is below a certain threshold, automatically flag it for review. This does two things: First, it prevents 'garbage in, garbage out' by stopping low-quality data from corrupting your `match_score_total`. Second, it gives you a metric to track the quality of your data pipeline over time. Your goal should be to continuously improve the average 'Data Health' score of your records."

dbt Integrations for ML and Python to keep the system healthy and self learning

some notes on this