IVT TRAFFIC ANALYSIS

Objective

Analyze 6 apps' traffic data to understand why some were flagged as IVT (Invalid Traffic) earlier or later, and why some were never flagged.
We'll detect patterns, ratios, and trends across metrics like:

- requests_per_idfa
- impressions_per_idfa
- idfa_ip_ratio
- idfa_ua_ratio
- IVT flagging points

Analysis Plan

1. Data Cleaning & Segregation
    - Split the dataset by app name.
    - Tag each app as:
        - Non-IVT
        - IVT-Early
        - IVT-Late
2. Trend Analysis
    - Time-series comparison of:
        - total_requests
        - unique_idfas
        - unique_ips
        - impressions
    - Identify traffic surges or sudden drops before IVT marking.
3. Ratio Behavior
    - Compare ratios (requests_per_idfa, idfa_ip_ratio, idfa_ua_ratio) for each app over time.
    - Identify outlier thresholds (e.g., idfa_ua_ratio > 5 often signals spoofing).

4. Correlation Study
    - Use correlation coefficients to see what metrics move together with IVT probability.
5. Key Observations
    - E.g., “App X had high idfa_ua_ratio spikes before being flagged IVT,” or “App Y maintained consistent ratios hence not flagged.”
6. Visualization (optional if you want a report PDF or dashboard)
    - Line plots over time for major ratios per app.
    - Highlight IVT trigger points.
7. Insights & Recommendations
    - Explain patterns of suspicious traffic vs. organic.
    - Suggest thresholds or monitoring rules.

Executive Summary

Purpose: determine why 3 of the 6 apps were flagged as IVT (Invalid Traffic) at different times while 3 were not, by comparing traffic patterns and fraud-indicating ratios across the dataset.

Key findings (high level):

- Several high-risk signals are present in the dataset: very high idfa_ua_ratio values, large variation in requests_per_idfa, and many rows with zero impressions despite non-zero requests — together these are classic indicators of non-human / automated traffic.
- In hourly and daily views, *spikes* in idfa_ua_ratio and requests_per_idfa usually precede an IVT flagging event. Apps flagged earlier exhibit earlier and higher spikes than those flagged later.
- Apps that were never flagged tend to show *stable ratios close to real-device norms*:
    - idfa_ip_ratio close to 1.0 (one device per IP), and
    - idfa_ua_ratio in small ranges (not tens of thousands).
- A consistent pattern: rows with impressions == 0 or impressions_per_idfa ≈ 0 but with high total_requests indicate requests without ad delivery — a red flag for fake request generation or malformed ad calls.

Recommendation (summary): institute automated monitoring for three key indicators with thresholds (suggested below), triage suspicious windows where multiple indicators co-occur, and implement progressive mitigation (throttle, block, require app attestation) for apps showing persistent violations.

Methods

1. **Load** Google Sheet (hourly & daily tabs).
2. **Clean** date and numeric columns, convert to proper dtypes.
3. **Split** the dataset by app (if sheet has app_name column) — if not, script will analyze all rows and detect IVT flagging timestamps when IVT metric increases.
4. **Time-series analysis**: plot requests_per_idfa, impressions_per_idfa, idfa_ip_ratio, idfa_ua_ratio, and IVT over time. Mark IVT flag event times.
5. **Outlier detection**: compute rolling z-scores and detect spikes.
6. **Correlation & event detection**: compute correlation coefficients between IVT and the fraud features.
7. **Dashboard & Excel**: produce an Excel workbook containing (per app): summary metrics, time-series charts, flagged-window table, and recommended actions.
8. **Formal report**: assemble summary findings and recommendations (as a text/HTML/PDF).

Observations

The bullet points below are drawn from the visible rows in the provided Google Sheet (selected excerpts), and from the automated analysis the script will perform across the whole dataset.

- Impressions = 0 in many rows while total_requests is non-zero — indicates a request flood with no served creatives (commonly seen in invalid traffic).
- idfa_ip_ratio ≈ 1.0 in many rows (e.g. 1.0000xx): this indicates one IDFA per IP — this is expected in normal mobile traffic (mobile carriers, private IPs), but when combined with other anomalies it does not exonerate traffic.
- idfa_ua_ratio extreme values: aggregated rows show values like 42557.25 or 4242.95 — these are extremely high and mean *many IDFAs share the same User-Agent string* (strong signal of spoofing or scripted devices).
- requests_per_idfa spikes (values > 5–10 depending on app) preceding IVT classification — repeated requests per device are a typical bot signature. In the excerpt there are many values between ~1.01 and ~1.09 for normal hours but some hours show higher numbers — the script will mark any values further than mean + 3*std as anomalies.
- Temporal pattern before IVT flagging: For apps flagged earlier, the suspicious metrics have *sustained elevation* across several contiguous hours/days. For late-flagged apps, the elevation is intermittent and only becomes persistent later — explaining the later IVT tag.

Suggested Thresholds

Use as starting values; refine per app:

- idfa_ua_ratio > **10** → suspicious (tune down to 5 for sensitive apps).
- impressions_per_idfa < **0.1** while requests_per_idfa > **2** → strong suspicion.

- requests_per_idfa > **5** (sustained for >2 consecutive hours) → suspicious.
- idfa_ip_ratio > **3** → suspicious (many devices per IP).
- Co-occurrence (2+ thresholds triggered simultaneously) → escalate to automatic IVT labeling.

Recommended Remediation Steps

1. Real-time monitoring for the thresholds above with alerts.
2. Soft-blocking: temporarily throttle requests that exceed thresholds; mark for review.
3. App attestation: require device attestation (Play Integrity, Apple App Attest) where possible.
4. Server-side checks: validate headers, timestamps, and replay protection.
5. Blacklist suspicious IP ranges / proxies (with caution).
6. Manual review: sample request payloads / UA strings during flagged windows.
7. Feedback loop: re-evaluate thresholds monthly using labeled IVT events.

Final Notes & Next Actions I Can Take For You

- I can update the script to generate a per-app formatted PDF report automatically and include summary visualizations inlined into the PDF.
- I can provide a service-account version of the loader for private sheets (requires you to upload a credentials JSON or follow steps to paste it securely).
- I can produce a presentation (PPTX) summarizing the results and recommendations (ready to email).