R Analytics Module 3: Anomaly Control Center

1. Technical Significance

The **Anomaly Control Center** identifies detecting fiscal leakage and fraud in logistics data. Traditional "rules-based" engines (like "if cost > 5000") produce too many false positives. Our approach uses **Unsupervised Machine Learning** (specifically **Isolation Forest**) to automatically "learn" what a normal shipment looks like and flag anything that deviates statistically, without human rules.

2. The R Technique: Isolation Forest

We rely on the `isotree` package in R for high-performance outlier detection.

A. Isolation Forest (iForest)

- **Purpose:** To detect anomalies in high-dimensional data (Cost, Weight, Distance, Carrier, Route).
- **Algorithm:** `isotree` package.
- **Mechanism:**
  - It builds an ensemble of random decision trees.
  - *Key Insight:* Anomalies are "easy to isolate." If you randomly split data, outlines will be separated in very few splits (short path length). Normal points are clumped together and take many splits to isolate (long path length).
  - The "Anomaly Score" is inversely proportional to the average path length.
- **Why it's superior:** It requires **no labeled training data**. It finds "unknown unknowns"—fraud patterns you didn't know existed.

B. Gaussian Mixture Models (GMM) - *Secondary Validation*

- We use GMM to cluster "normal" behavior. If a point falls outside the 99% probability contour of any normal cluster, it reinforces the Isolation Forest verdict.

3. Workflow & Architecture

1. **Data Ingestion:**
   - The user opens the Anomaly Dashboard.
   - The system sends recent invoices/shipments to `backend/r_analytics_service.py`.
2. **R Execution (`anomaly.R`):**
   - Data is serialized and piped to R.
   - R runs `detect_anomalies()`.
3. **Statistical Processing:**
   - **Preprocessing:** Categorical variables (Carrier Name) are encoded.

- **Modeling:** The Isolation Forest constructs 100 random trees.
- **Scoring:** Every invoice gets a score from 0.0 (Global Normal) to 1.0 (Extreme Outlier).

4. **Output:**
   - R returns a list of indices that are outliers, along with their severity scores.

4. Sample Data & Results

Input Data (Sample)

```
[
  { "id": "INV_001", "cost": 5000, "weight": 1000, "lane": "MUM-DEL" }, // Standard
  { "id": "INV_002", "cost": 5200, "weight": 1050, "lane": "MUM-DEL" }, // Standard
  { "id": "INV_003", "cost": 18000, "weight": 200, "lane": "MUM-DEL" }  // ANOMALY! (High Co
]
```

R Processing Output

```
{
  "anomalies": [2],           // Index of the 3rd item
  "scores": [0.35, 0.38, 0.85], // 0.85 is very high
  "categories": ["normal", "normal", "critical_anomaly"]
}
```

User Facing Result

- **Alert:** "Critical Anomaly Detected (Score 0.85)"
- **Explanation:** "This invoice value (18k) is 300% higher than expected for weight 200kg on this lane."
- **Action:** "Blocked for Payment."