Fraud Detection and Transaction Pattern Mining in R: A Case Study for a U.S. Fintech Startup

1. Background

A New York-based fintech startup offering virtual debit cards and digital wallets experienced increasing fraud complaints and suspicious activity reports. Manual fraud detection was time-consuming and prone to oversight. The client sought a pattern-based fraud detection system that could flag anomalous transactions for review without relying on predefined rules.

2. Objective

- To identify suspicious transaction patterns using unsupervised anomaly detection
- To segment transaction behavior by customer profile using clustering
- To reduce the burden on manual fraud teams by prioritizing high-risk flags

3. Data Used

Source: Encrypted transaction logs exported from the company's internal payment gateway

Structure:

- 1.4 million transaction records over 6 months
- Key
  fields: Transaction_ID, User_ID, Amount, Merchant_Category, Time_of_Day, Location, Device_ID, Failed_Attempts, Txn_Volume_Past_24hr, Account_Age_Days

4. Modeling Methodology

4.1 Data Cleaning and Feature Engineering

- Used dplyr, lubridate, and stringr for transformations
- Created composite features:
  - Avg_Txn_Amount_Per_Day
  - Merchant_Frequency_Score
  - Distance_Between_Txn_Locations using geolocation API
- Normalized all numeric features using scale()

4.2 Clustering for Behavior Profiling

- Applied **DBSCAN** to identify dense regions of normal activity
- Flagged points outside dense clusters as potential anomalies

```
library(dbscan)
db_result <- dbscan(scaled_data, eps = 0.5, minPts = 10)
```

4.3 Isolation Forest for Anomaly Detection

- Used isotree package to score anomalies
- Combined with clustering output for dual-layer detection

```
library(isotree)
iso_model <- isolation.forest(scaled_data)
scores <- predict(iso_model, scaled_data)
anomaly_flags <- ifelse(scores > 0.65, 1, 0)
```

4.4 Rule Layer (Hybrid)

- Layered post-processing rules:
    - Flag transactions with Amount > 90th percentile AND New_Device = TRUE
    - Flag repeat micro-transactions in <30 min with changing Device_IDs

5. Results

Metric	Before (Manual-Only)	After (R-based System)
Monthly flagged transactions	~1,000	~3,800
Manual false positives	77%	28%
Avg. fraud resolution time	3.2 days	0.9 days
Precision of anomaly model	–	0.82

6. Interpretation and Recommendations

- **Isolation Forest** outperformed DBSCAN alone in identifying irregular amounts from new devices

- Hybrid model reduced noise by excluding consistent high-spend users from unnecessary alerts
- Suggested updating clustering model every 15 days to accommodate evolving user behavior
- Recommended integrating alert system with internal fraud dashboard (e.g., Looker/Power BI)
- Added explainability layer showing top contributing features per flag (Shapley-style)

7. Reporting Output

- **Fraud Analytics R Markdown Report (30 pages)**
  - Heatmaps of flagged transactions by hour
  - Cluster profiles: normal vs. anomalous users
  - Visual flowcharts of anomaly scoring logic
- **Interactive R Shiny Prototype**
  - Upload new transaction files → Auto-flag anomalies
  - Tabs: Risk Scoring, Visualization, Export CSV
- **Code Modules Delivered**
  - clean_transaction_data.R, fraud_flag_model.R, clustering_logic.R, shiny_ui_server.R

8. Business Outcome

- Reduced fraud review team's workload by **60%**
- Flagged **\$187,000** worth of suspicious transactions in the first 45 days
- Detected and reported **8 fraudulent merchant rings** using pattern clusters
- Prototype extended to mobile wallet division in Q2 rollout