1 Description of Module 3

You are tasked with building an algorithm to detect fraudulent credit card transactions for a bank. The bank has data on each individual's credit card transactions, with Y_t representing the credit card charge at time t; see Table 1 for an example. Assume that each charge was either by the

Time	Name of Transaction (POT)	Amount
Mar 2, 2018	University of Wisconsin, Rec Services	\$ 70.30
Mar 2, 2018	Aldo Cafe	\$ 8.57
Mar 2, 2018	University of Wisconsin, Parking Service	\$ 10.00
Mar 7, 2018	United Airlines (ORD to SFO)	\$ 179.34
Mar 7, 2018	University of Firenzia, Italia	EU 400.00
Mar 7, 2018	Bank Fee (Foreign Conversion Fee)	\$ 4.00
Mar 8, 2018	Trader Joes, Madison	\$ 20.39
Mar 10, 2018	Best Buy	\$ 15,032.39
Mar 12, 2018	Transfer from Checking Account (Paid Credit Card bill)	\$ −837.32
⋮	⋮	⋮

Table 1: A sample credit card transaction data for one individual. Each row presents one credit card transaction at a particular time and location costing a particular amount.

individual or some bad actor/fraudster. As a bank, you want to tell your customers immediately of any fraudulent charges right away. It is unknown how many (or which) of these transactions were actually done by the individual or the fraudster. Assume that there are more 'honest' transactions by the individual than fraudulent ones.

You are tasked with **developing an algorithm to detect fraudulent credit card transactions** so that banks can notify their customers of any fraudulent behavior immediately. A couple of things you must take into consideration are

1. The data comes in streams (not in batches!), with Y_t representing the credit card charge at time t and X_t representing the point of transaction (POT).
2. Your algorithm needs to produce real-time predictions about any fraudulent behavior.
3. Your algorithm should be reasonably secure/robust from fraudsters who are constantly trying to avoid detection.
4. Your algorithm needs to scale, especially to make sure future updates that may be necessary for your algorithm can be deployed quickly and efficiently.
5. Your algorithm must be robust to things like currency conversions, credit card fees, misspelling of POTs, refunds, etc.

2 Deliverables

You'll submit a 2-page pdf summary that includes (1) an R code (embedded) that runs your algorithm, (2) a working demo of your code (with realistic simulated data), (3) some benchmarks for your code, and (4) a concise, clear explanation of your algorithm. Your summary will be graded

initially by your peers where your peers will judge your proposal. I'll compile their feedback and give a Fail, Pass, or High Pass.

Your 2-page pdf MUST NOT CONTAIN any identifiable information (e.g. your name, your lecture group, etc.). This is to ensure fair grading.