Credit Card Transactions

Goal

One of the greatest challenges in fraud, and in general in that area of data science related to catching illegal activities, is that you often find yourself one step behind.

Your model is trained on past data. If users come up with a totally new way to commit a fraud, it often takes you some time to be able to react. By the time you get data about that new fraud strategy and retrain the model, many frauds have been already committed.

A way to overcome this is to use unsupervised machine learning, instead of supervised. With this approach, you don't need to have examples of certain fraud patterns in order to make a prediction. Often, this works by looking at the data and identify sudden clusters of unusual activities.

This is the goal of this challenge. You have a dataset of credit card transactions and you have to identify unusual/weird events that have a high chance of being a fraud.

Challenge Description

Company XYZ is a major credit card company. It has information about all the transactions that users make with their credit card.

Your boss asks you to do the following:

- Your boss wants to identify those users that in your dataset never went above the monthly credit card limit (calendar month). The goal of this is to automatically increase their limit. Can you send him the list of Ids?
- On the other hand, she wants you to implement an algorithm that as soon as a user goes above her monthly limit, it triggers an alert so that the user can be notified about that.We assume here that at the beginning of the new month, user total money spent gets reset to zero (i.e. she pays the card fully at the end of each month). Build a function that for each day, returns a list of users who went above their credit card monthly limit on that day.
- Finally, your boss is very concerned about frauds cause they are a huge cost for credit card companies. She wants you to implement an unsupervised algorithm that returns all transactions that seem unusual and are worth being investigated further.

Data

We have 2 table downloadable by clicking **here**.

The 2 tables are:

```
"cc_info" - general information about the credit card and its holder
```

Columns:

- **credit_card** : credit card number. Can be joined to credit_card in the table below
- **city** : where the credit card holder lives
- **state** : in which state the credit card holder lives
- **zipcode** : credit card holder zip code
- **credit_card_limit** : this is the credit card monthly limit. Credit card holders should be careful in not going above this limit in total money spent per month. The limit is by calendar month

---

```
    transactions - information about each transaction that happens between Aug,
1 and Oct, 30 for the credit cards in cc_info.
```

Columns:

- **credit_card** : credit card number. Can be joined to credit_card in the other table
- **date** : when the transaction happened (GMT time)
- **transaction_dollar_amount** : transaction amount in dollars
- **Long** : longitude of where the transaction happened
- **Lat** : latitude of where the transaction happened

Example

```
    Let's check one transaction
```

head(transactions,1)

Column Name	Value	Description
credit_card	1003715054175576	this is the credit card number
date	2015-09-11 00:32:40	the transaction happened on Sept, 11 around midnight GMT time
transaction_dollar_amount	43.78	the credit card was charged 43.78$
Long	-80.17413	the transaction happened at this longitude
Lat	40.26737	and this latitude. It means it happened here, in Pennsylvania.

```
    Let's check info about that credit card
```

subset (cc_info, credit_card==1003715054175576)

Column Name	Value	Description
credit_card	1003715054175576	same as in the example above
city	Houston	city where the credit card holder lives
state	PA	it makes sense. Before we saw the cc holder bought something in Pennsylvania and now we found out that she actually also lives in Pennsylvania
zipcode	15342	her zip code
credit_card_limit	20000	her monthly credit card limit is 20K USD