Preprocess event data for analysis

Welcome to the exciting world of process mining!

The following exercise will guide you through a typical transformation and data analysis to prepare data from an IT system for process mining.

In order to do process mining, we need to derive an event log in the following format:

Case ID	Action	Start	Complete	Attribute 1
1	A	01.01.2016 10:54:23	01.01.2016 10:55:02	X
1	B	01.01.2016 13:27:16	01.01.2016 14:27:16	Y
2	A	02.01.2016 08:12:45	02.01.2016 08:42:31	Z

This table shows a very simple event log. Each event has to have at least a case identifier (Case ID), a name of the activity that has been executed (Action), a start and a complete timestamp. Optionally, an event can have an arbitrary number of attributes. In the case above, we only have one attribute (Attribute 1).

Each line in the event log represents exactly one event. Thus, we have 3 events that represent the two actions A and B. Action A has been executed two times while action B only ran once. The three events belong to two different cases: 1 and 2. Case 1 has the events A and B from the 1st of January while case B has the activity A from 2nd of January.

Tasks

The following tasks can be completed with the language and frameworks of your choice. Using Spark and Scala would be awesome, but is definitely not a requirement. Please also document all your quality checking and analysis steps that you might additionally do during the assignment. Submit your solution by simply packing your scripts and giving us a quick intro on how to run it.

I. Understanding the data

The first thing to do, is to get an understanding of the data and format in the CSV that you received. The data is provided in a tab separated text file. The *eLetter_ID* is the case identifier and the *ActivityName* is the activity. So far so good. But what about the start and complete timestamps? It seems there are none. But if we look closer, we can see that there is a Time column, where the first row starts with value 0. Actually, this gives us the time passed from a specific point in time. Luckily, our client provided this fix point. It's the 01.01.2016 10:00:00. If we interpret the value from the Time column as seconds and add them to this date, we get the actual time of the event. Hence, the first row with value 0 happend at 01.01.2016 10:00:00 and the second row with value 600 happened at 01.01.2016 10:10:00, which is ten minutes later.

We've got a timestamp, great! But wait...didn't we need two timestamps: Start and Complete?

Yes, we do. If you look closely at the data, you will see that there are almost always two subsequent rows that share the same *ActivityName* and *eLetter_ID* . The first of two rows is always the start and the second is the complete time of an event.

So we've got some first insights about the structure of the data. Let's get some initial numbers to understand what we're looking at. Please answer the following questions for the provided data set.

1. How many events do we have?
2. How many events do we have for each activity name?
3. How many cases do we have?
4. There's an *eLetter_Type* attribute. What values does it have and how often do they occur?

II. Transforming the data into an event log

Now that we know a bit about the structure and have a few basic numbers, we can start to transform the data into the format explained in the beginning of this assignment.

Looking back at the numbers we just calculated, we can see that there is something strange. Our initial assumption was that there are always two lines per activity, one representing the start and the other on the complete time. So how can we have an odd number of events for certain activities? It seems the last event of such an activity in a case comes from something that has only been started but not yet completed. These events we need to remove.

Thus, the tasks for the transformation are:

1. Calculating the timestamps for each row.
2. Combining two subsequent rows for the same activity into one with start and complete timestamp.
3. Removing all rows for which we do not have a matching second row.

III. Checking the transformed data

Once we transformed the data, we always need to do some sanity checks to ensure good data quality. In fact, we need to make sure we didn't screw up the data while transforming it. In order to do so, please complete the following steps so we can discuss the results with our client:

1. Answer the questions from section I again for the transformed data
2. Calculate the difference between start and complete time for each event and show the distribution
3. (Optional): Surprise as with something interesting you found in the data :-)
4. (Super Bonus): Add something fun to your analysis that has something to do with penguins. For example, you may use the standard Twitter search API to get the number of tweets about penguins for each day. This you could add as an extra column to each event so we know how much people talked about penguins while our process was running :-)[1]

IV. Presenting the analysis results

Prepare a short 10 minute presentation in German of your analysis results. Focus on aspects that the client might be interested in.

Happy data crunching!

[1] Note: If you really look into this, you'll notice that you will only get the last seven days from twitter. You may just use the workday as aggregation and reference. So if the event was on a Tuesday, we know how many people talked about penguins on our reference Tuesday from last week.