Data Analysis Coding Assignment

A file all_transactions.csv has been provided that contains two fields:
Customer ID, Date of Transaction (in format YYYYMMDD)
The data contains the transactions history of customers from Jan 1997 – Jun 1998. A customer could have done multiple transactions in this dataset.

The following tasks have to be performed on this dataset:
(Note: Try to finish as many tasks as possible in this assignment and send back to us what you have completed. Our evaluation criteria consists of many different aspects such as how a difficult problem is tackled, accuracy of the results for different steps, how the code is organized and so on)

1. Extract the transactions dataset from Jan 1997– Sep 1997 for calibration purposes. The rest of the dataset from Oct 1997 – Jun 1998 can be used for validation (however, validation would not be a part of this assignment). In case a customer has done two or more transactions on the same day, count it as only one transaction i.e. remove duplicate rows from the data.
   E.g. For customer with ID 1, we would consider only the following data for calibration period:

   1 19970101
   1 19970118
   1 19970802

   Save the extracted transactions in a new csv file called cal_period_transactions.csv. This file is the input for the next step.

2. Calculate for each customer three parameters:
   - *x*: Number of transactions done by customer minus 1, it is 0 in case the customer made only one transaction
   - *tx*: Duration in weeks between customer's last transaction and first transaction, it is 0 in case the customer made only one transaction
   - *T*: Duration in weeks between end of calibration period (01.10.1997) and customer's first transaction

   E.g.: Customer with ID 1 would have the following parameters estimated:

x	2
tx	30.43
T	38.86

   Save these parameters *Customer ID*, *x*, *tx*, *T* for each customer in a new csv file called summary_customers.csv. This file is the input for the next step.

3. The aim of this assignment is to fit a model to the data, so that predictions about the customers' purchase behavior in future can be made. The model is defined by four parameters namely *r*, α, *a*, *b* and they are always greater than 0. For fitting the model (i.e. finding the parameters *r*, α, *a*, *b*), Maximum Likelihood Estimation (MLE) can be used. The log likelihood (*LL*) function which is to be maximized is given as follows:

$$LL(r,\alpha,a,b)=\sum_{i=1}^{N}\ln(L(r,\alpha,a,b\mid X_i=x_i,tx_i,T_i)) \quad \text{(Eq. 1)}$$

Here *N* is the number of customers, *ln* is natural logarithm and the individual likelihood (*L*) for each customer *i* is:

$$L(r,\alpha,a,b\mid X_i=x_i,tx_i,T_i)=A1_i\cdot A2_i\cdot\left(A3_i+\delta_{x_i>0}A4_i\right)$$

Where $\delta_{x>0}$ is 1 if $x > 0$, 0 otherwise and:

$$A1_i=\frac{\Gamma(r+x_i)\alpha^r}{\Gamma(r)},\ A2_i=\frac{\Gamma(a+b)\Gamma(b+x_i)}{\Gamma(b)\Gamma(a+b+x_i)},$$

$$A3_i=\left(\frac{1}{\alpha+T_i}\right)^{r+x_i},\ A4_i=\left(\frac{a}{b+x_i-1}\right)\left(\frac{1}{\alpha+tx_i}\right)^{r+x_i}$$

Γ symbol represents the Gamma function.

Eq. 1 can be converted to negative log likelihood (*NLL*) that is to be minimized as follows (note that the division by *N* is just done for normalization):

$$NLL(r,\alpha,a,b)=\sum_{i=1}^{N}(-1)\left[\ln(A1_i)+\ln(A1_i)+\ln\left(\exp^{\ln(A3_i)}+\delta_{x_i>0}\exp^{\ln(A4_i)}\right)\right]/N \quad \text{(Eq. 2)}$$

Where:

$$\ln(A1_i)=\ln(\Gamma(r+x_i))+r\ln(\alpha)-\ln(\Gamma(r)),$$
$$\ln(A2_i)=\ln(\Gamma(a+b))+\ln(\Gamma(b+x_i))-\ln(\Gamma(b))-\ln(\Gamma(a+b+x_i)),$$
$$\ln(A3_i)=-(r+x_i)\ln(\alpha+T_i)$$
$$\ln(A4_i)=\ln(a)-\ln(b+x_i-1)-(r+x_i)\ln(\alpha+tx_i)$$

Implement the objective function given by Eq. 2 as an optimizable (minimizable) function. Note that the natural logarithm of the Gamma function is already available in different programming languages, so this can be used directly. Details about the function are:
Input parameters: *x*, *tx*, *T* for all customers
Model parameters: *r*, α, *a*, *b*
Output parameter: The value of the function (NLL i.e. Eq. 2) evaluated at some value of *r*, α, *a*, *b* given the customers' data *x*, *tx* and *T*.

4. Minimize this function using a numerical method for non-linear optimization problems e.g. the Nelder-Mead algorithm (which does not require calculation of derivatives). Alternately, other algorithms e.g. L-BFGS can be used, gradient calculations can be performed using numerical approximations.

   The starting point for parameters *r*, α, *a*, *b* can be taken as the result of evaluating Gaussian functions with mean 1.0 and standard deviation 0.05. Also, these four parameters are always greater than 0, so in case during optimization, if any of these parameters tend to go towards negative, this can be handled for example by returning infinity as the output parameter of the objective function.

The parameters *tx* and *T* have to be normalized before passing them to the function. For this, do the following:

- Get the maximum value of *T* from the available values
- Calculate the normalization factor by dividing 10.0 by the maximum value of *T*
- Multiply every element in *tx* and *T* vector by this factor

Also, the model parameter α has the same units like *tx* and *T*, so it also has to be multiplied by the normalization factor inside the objective function before any evaluations. To summarize, while calculating Eq. 2, use scaled α, *tx* and *T* values.

Find the optimal values of *r*, α, *a*, *b* that minimize the objective function and define a strategy to check convergence from different starting points. Save these parameter estimates in a csv file called estimated_parameters.csv.

The programming can be done in Scala (preferred)/Python (preferred)/Java/R. External libraries can be used. Provide the code and the generated csv files, along with a documentation describing the code.