Statistical Data Analysis
Problem sheet #7

Exercise 1: An experiment yields n time values $t_1, \ldots, t_n$, and a calibration value y, all of which are independent. The time measurements are all exponentially distributed with a mean of $\tau + \lambda$ and the calibration measurement, y, follows a Gaussian distributikon with a mean λ and a standard deviation σ. Suppose that σ is known and we want to estimate τ and λ.

1(a) Write down the likelihood function for τ and λ, and show that the Maximum Likelihood (ML) estimators for these parameters are

$$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n} t_i - y \,,$$

$$\hat{\lambda} = y \,.$$

1(b) Find the variances of $\hat{\tau}$ and $\hat{\lambda}$, and the covariance $\mathrm{cov}[\hat{\tau}, \hat{\lambda}]$. Use the fact that the variance of an exponentially distributed variable is equal to the square of its mean.

1(c) Show using a sketch how a contour of constant log-likelihood can be used to determine the standard deviations of $\hat{\tau}$ and $\hat{\lambda}$. Explain qualitatively how you would expect the variance of $\hat{\tau}$ to be different if the parameter λ were to be known exactly.

1(d) Show that the (co)variances of $\hat{\tau}$ and $\hat{\lambda}$ obtained from the matrix of second derivatives of the log-likelihood are the same as those found in (c). Use the fact that the inverse of a 2×2 matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \,.$$

Exercise 2: This exercise provides an introduction to the class `TMinuit`, used in `ROOT` for function minimization. You should turn in any code where you have made modifications and program output in the form of numerical values and plots as appropriate. You do not need to turn in code supplied to you that you did not modify.

The exercises uses `TMinuit` to carry out a Maximum Likelihood fit where we minimize the quantity $-2 \ln L$. For more information on TMminuit see

`root.cern.ch/root/html/TMinuit.html`

First we will generate some data using a simple Monte Carlo program. Download, build and test the program `makeData` from the course website. `makeData` generates values according to an exponential distribution

$$f(x;\xi) = \frac{1}{\xi} e^{-x/\xi} \quad (x \geq 0)$$

and writes the values to a file.

In a separate directory, download and build the program `expFit` from the course website. This program reads in the file of individual values provided by `makeData` and does a maximum likelihood fit of the parameter ξ nof the exponential pdf. Run `makeData` and generate a file with 200 data values. Use this as the input for `expFit` and find the estimate $\hat{\xi}$ and its standard deviation $\sigma_{\hat{\xi}}$.

Now modify `makeData` so that it generates values according to the pdf

$$f(x;\alpha,\xi_1,\xi_2) = \alpha \frac{1}{\xi_1} e^{-x/\xi_1} + (1-\alpha)\frac{1}{\xi_2} e^{-x/\xi_2} \ , \tag{1}$$

with $\alpha = 0.2$, $\xi_1 = 1.0$ and $\xi_2 = 5$. To do this, first generate a random number r uniform in $[0,1]$. If $r < \alpha$, then generate x according to an exponential with mean ξ_1, otherwise use ξ_2. Run the program and save 200 individual values to a text file.

Now modify the program `expFit` so that it reads in the values and carries out an ML fit of the parameters α, ξ_1 and ξ_2. You will have to supply start values and "step sizes" for the parameters. Choose start values not too far (say, within a factor of two) to the true values used in `makeData`. For the step sizes you can take, e.g., 0.1.

Try running the program with the minimum and maximum values (in the arrays `minVal` and `maxVal`) set equal to zero; this is equivalent to having no bounds on the parameters. If the program runs into a region of parameter space that it shouldn't, e.g., $\xi_1 < 0$, then you can place appropriate bounds on the parameter values. In the end it is best to see if you can rerun the fit with improved guesses for start values but without any bounds on the parameters.

Modify the program so it makes a reasonable plot of the fit (extend the limit of the horizontal axis as appropriate). Find the ML estimators and their covariance matrix using the routines `mnpout` and `mnemat`. This requires that you add lines of the form

```
double covmat[npar][npar];
minuit.mnemat(&covmat[0][0], npar);
```

where `npar` (here 3) is the number of fitted parameters. Determine as well the matrix of correlation coefficients.