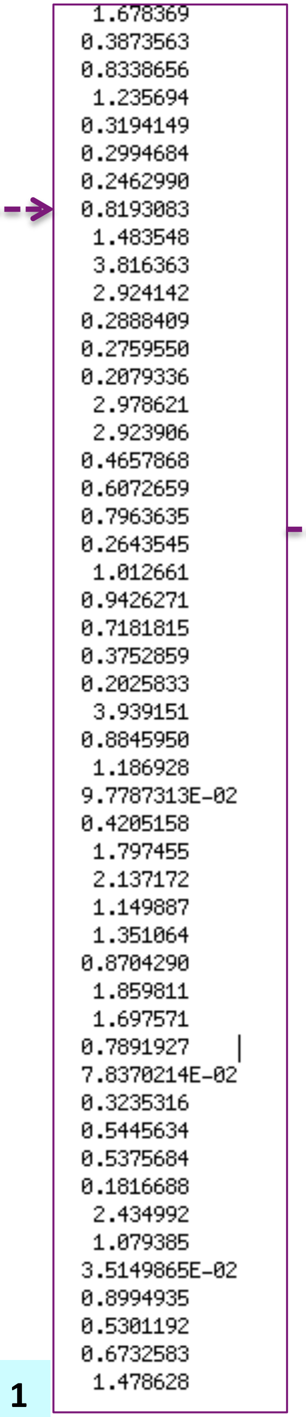
Maximum likelihood fit con modello esponenziale e stima del parametro vita media

Esercitazione del Corso di *Laboratorio Analisi Dati*
Secondo anno Magistrale / Primo Semestre
docente: Alexis Pompili

[credits to G.Cowan]

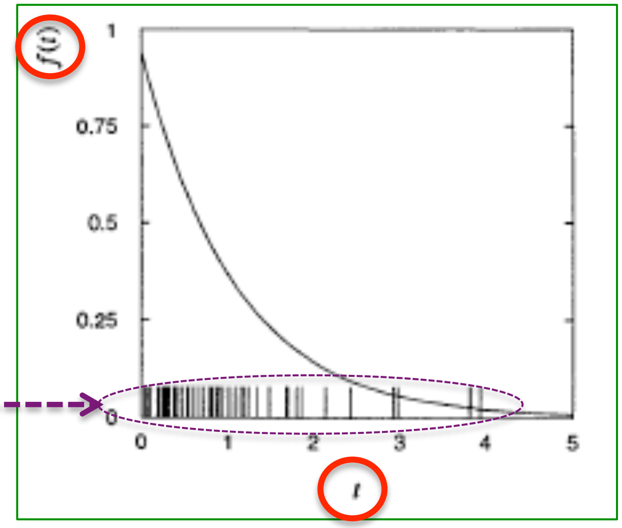
Data input file:

mltest.dat



```
 1.678369
0.3873563
0.8338656
 1.235694
0.3194149
0.2994684
0.2462990
0.8193083
 1.483548
 3.816363
 2.924142
0.2888409
0.2759550
0.2079336
 2.978621
 2.923906
0.4657868
0.6072659
0.7963635
0.2643545
 1.012661
0.9426271
0.7181815
0.3752859
0.2025833
 3.939151
0.8845950
 1.186928
9.7787313E-02
0.4205158
 1.797455
 2.137172
 1.149887
 1.351064
0.8704290
 1.859811
 1.697571
0.7891927
7.8370214E-02
0.3235316
0.5445634
0.5375684
0.1816688
 2.434992
 1.079385
3.5149865E-02
0.8994935
0.5301192
0.6732583
 1.478628
```

G.Cowan, p.73



Si tratta di 50 osservazioni generate (campione Monte Carlo) della variabile aleatoria esponenziale *t* :

$$f(t;\tau) = \frac{1}{\tau} e^{-\frac{t}{\tau}}$$

... dove, in generazione, la (vera) vita media e' assunta pari a τ=*1.0*

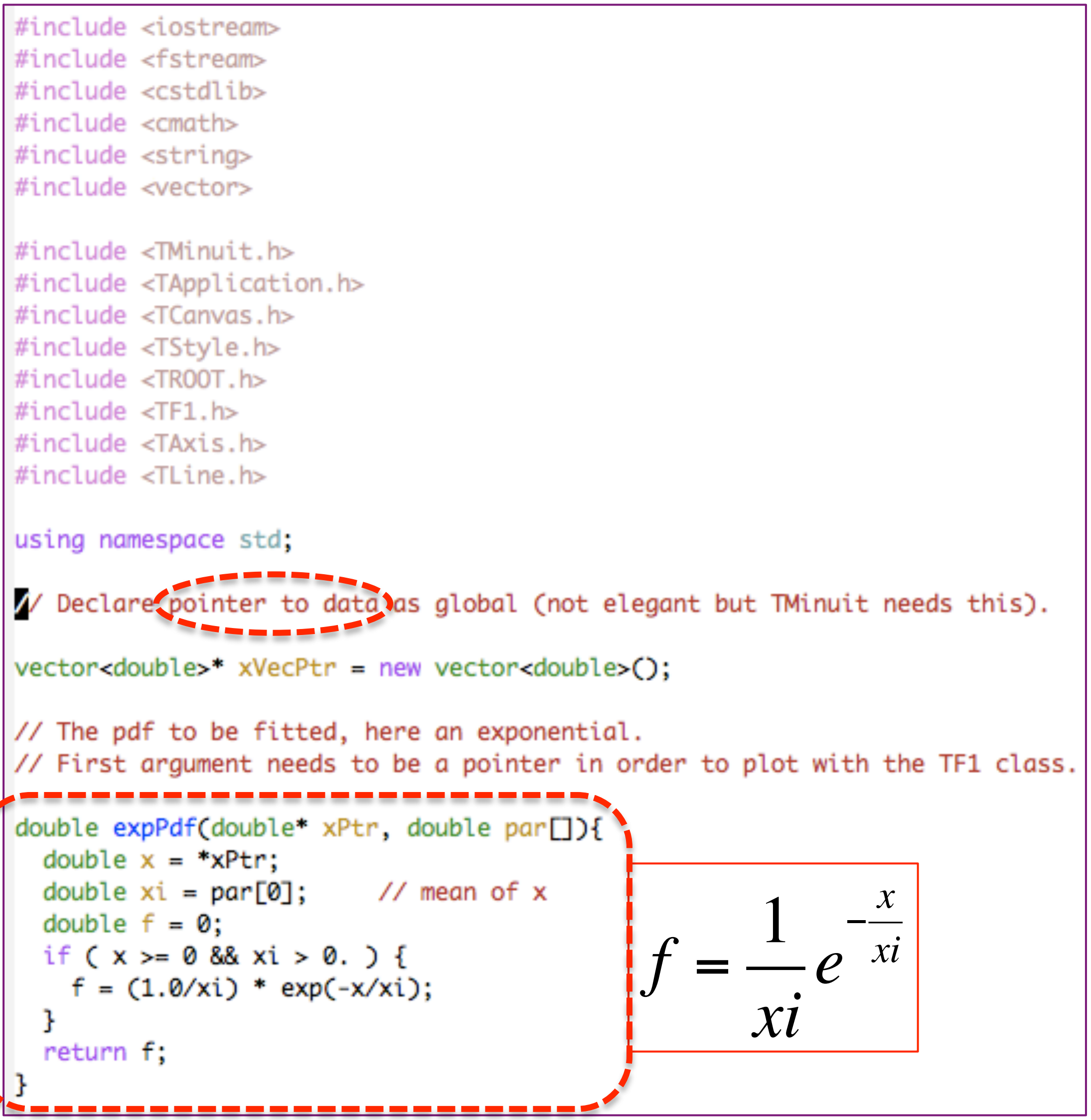
Sicuri che sono 50?
Verificate con:

$ *more mltets.dat* | *wc*

<u>Per ora non chiedetevi cosa e' la curva!</u>

Macro file (C^{++} program) [by G.Cowan]:

expFit.cc



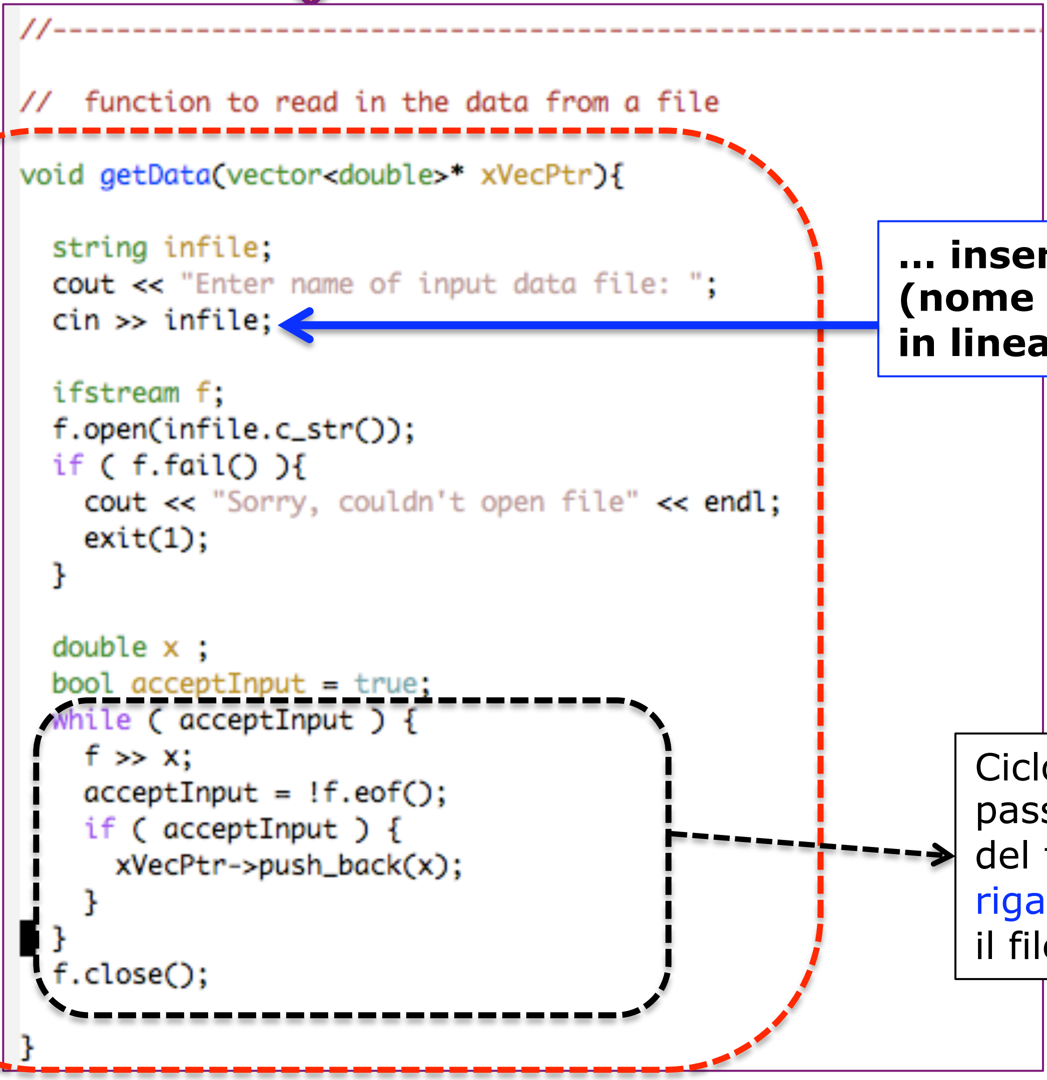
```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <cmath>
#include <string>
#include <vector>

#include <TMinuit.h>
#include <TApplication.h>
#include <TCanvas.h>
#include <TStyle.h>
#include <TROOT.h>
#include <TF1.h>
#include <TAxis.h>
#include <TLine.h>

using namespace std;

// Declare pointer to data as global (not elegant but TMinuit needs this).

vector<double>* xVecPtr = new vector<double>();

// The pdf to be fitted, here an exponential.
// First argument needs to be a pointer in order to plot with the TF1 class.

double expPdf(double* xPtr, double par[]){
  double x = *xPtr;
  double xi = par[0];      // mean of x
  double f = 0;
  if ( x >= 0 && xi > 0. ) {
    f = (1.0/xi) * exp(-x/xi);
  }
  return f;
}
```

PDF esponenziale

$$f = \frac{1}{xi} e^{-\frac{x}{xi}}$$

...continua...

Funzione di lettura dei dati da file esterno

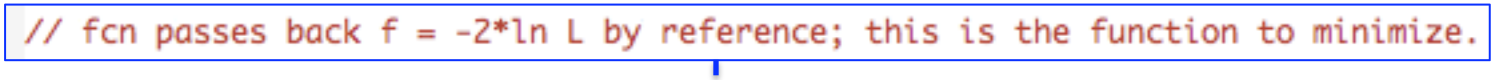


```
//-----------------------------------------------------------------

//  function to read in the data from a file

void getData(vector<double>* xVecPtr){

  string infile;
  cout << "Enter name of input data file: ";
  cin >> infile;

  ifstream f;
  f.open(infile.c_str());
  if ( f.fail() ){
    cout << "Sorry, couldn't open file" << endl;
    exit(1);
  }

  double x ;
  bool acceptInput = true;
  while ( acceptInput ) {
    f >> x;
    acceptInput = !f.eof();
    if ( acceptInput ) {
      xVecPtr->push_back(x);
    }
  }
  f.close();

}
```

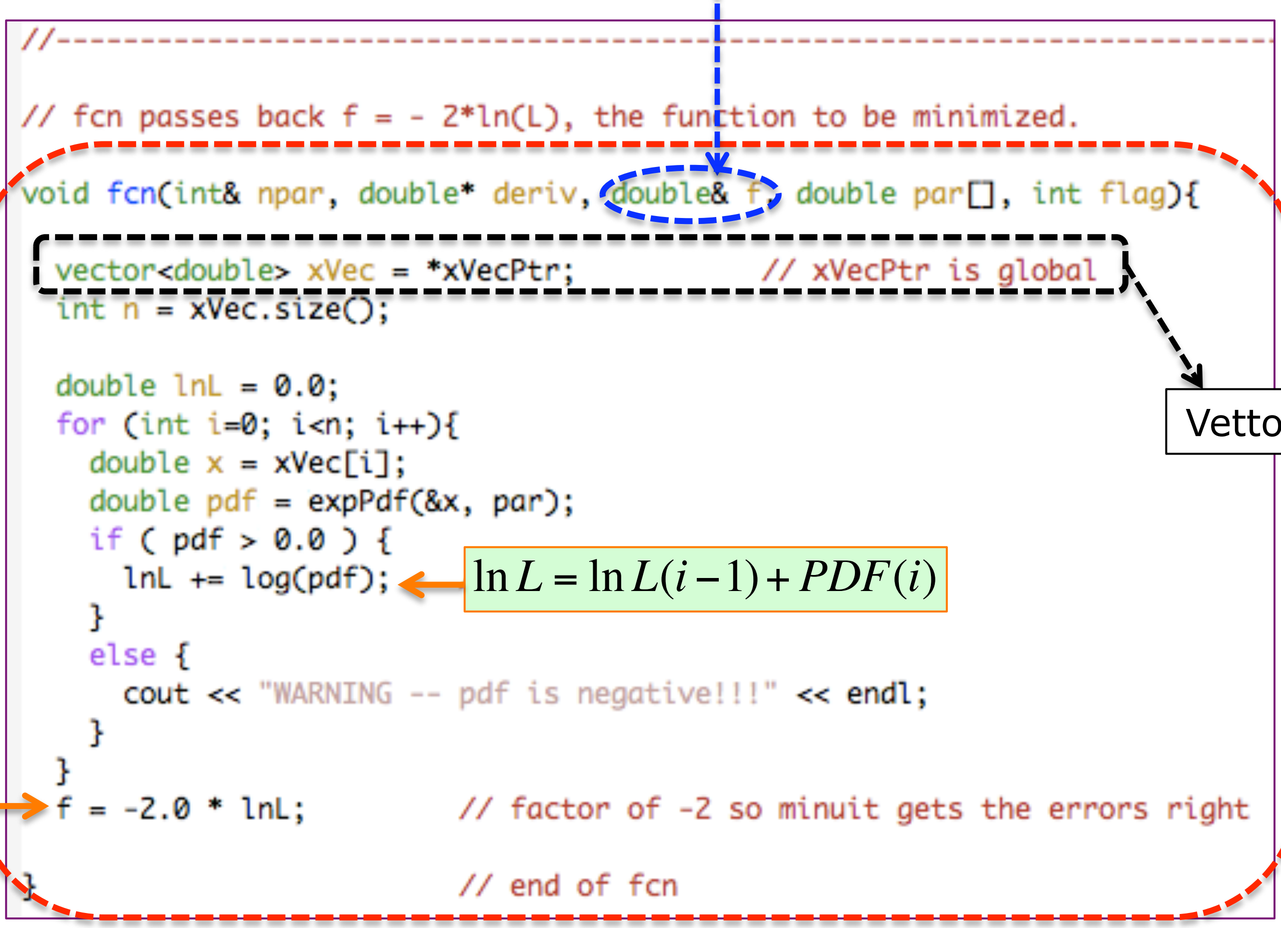
… inserire stringa (nome file estreno) in linea !

Ciclo while con cui passo ad ***x*** il valore del file esterno riga x riga … finche' il file non finisce !

…continua…



```
// fcn passes back f = -2*ln L by reference; this is the function to minimize.
```



```
//----------------------------------------------------------------------

// fcn passes back f = - 2*ln(L), the function to be minimized.

void fcn(int& npar, double* deriv, double& f, double par[], int flag){

  vector<double> xVec = *xVecPtr;           // xVecPtr is global
  int n = xVec.size();

  double lnL = 0.0;
  for (int i=0; i<n; i++){
    double x = xVec[i];
    double pdf = expPdf(&x, par);
    if ( pdf > 0.0 ) {
      lnL += log(pdf);
    }
    else {
      cout << "WARNING -- pdf is negative!!!" << endl;
    }
  }
  f = -2.0 * lnL;          // factor of -2 so minuit gets the errors right

}                          // end of fcn
```

Vettore di dati

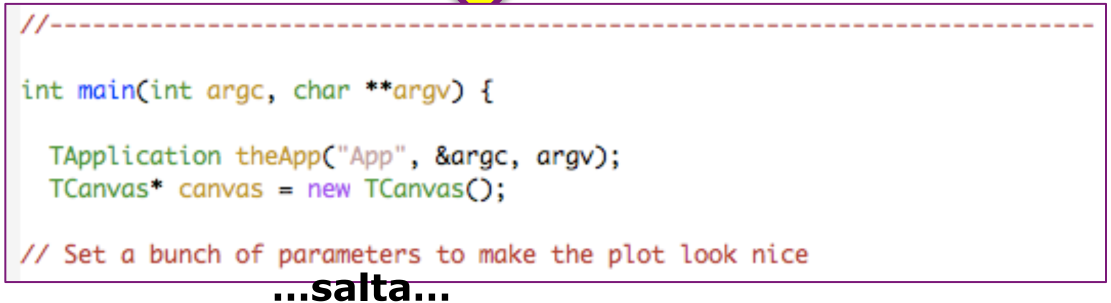
$\ln L = \ln L(i-1) + PDF(i)$

Fornisce la funzione (NLL) [*] da minimizzare file esterno

$\mathscr{L}' = 2*(-\ln L)$

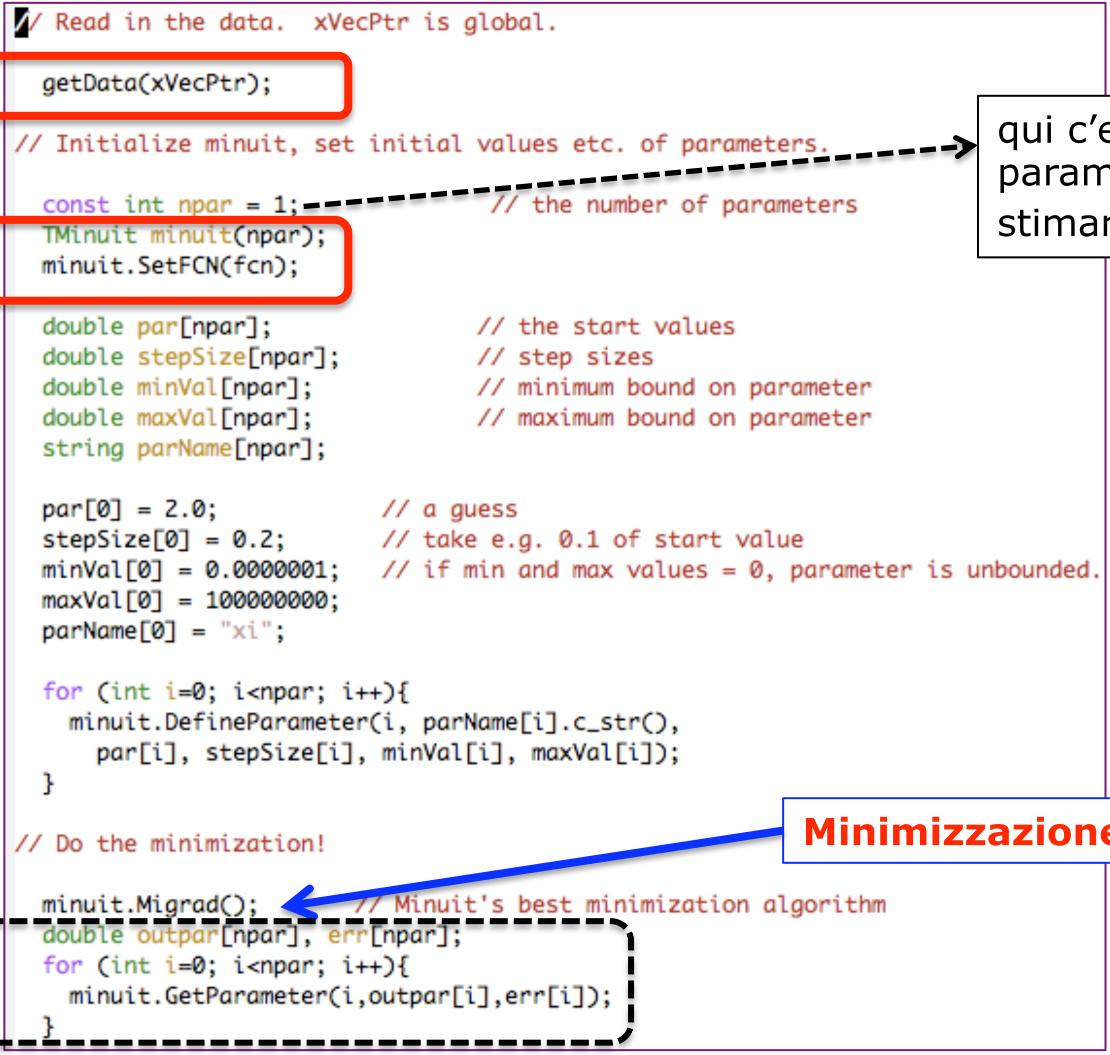
…continua…

[*] NLL=Negative Log Likelihood $\mathscr{L} = -\ln L$



```
//--------------------------------------------------------------------------

int main(int argc, char **argv) {

  TApplication theApp("App", &argc, argv);
  TCanvas* canvas = new TCanvas();

// Set a bunch of parameters to make the plot look nice
```

...salta...



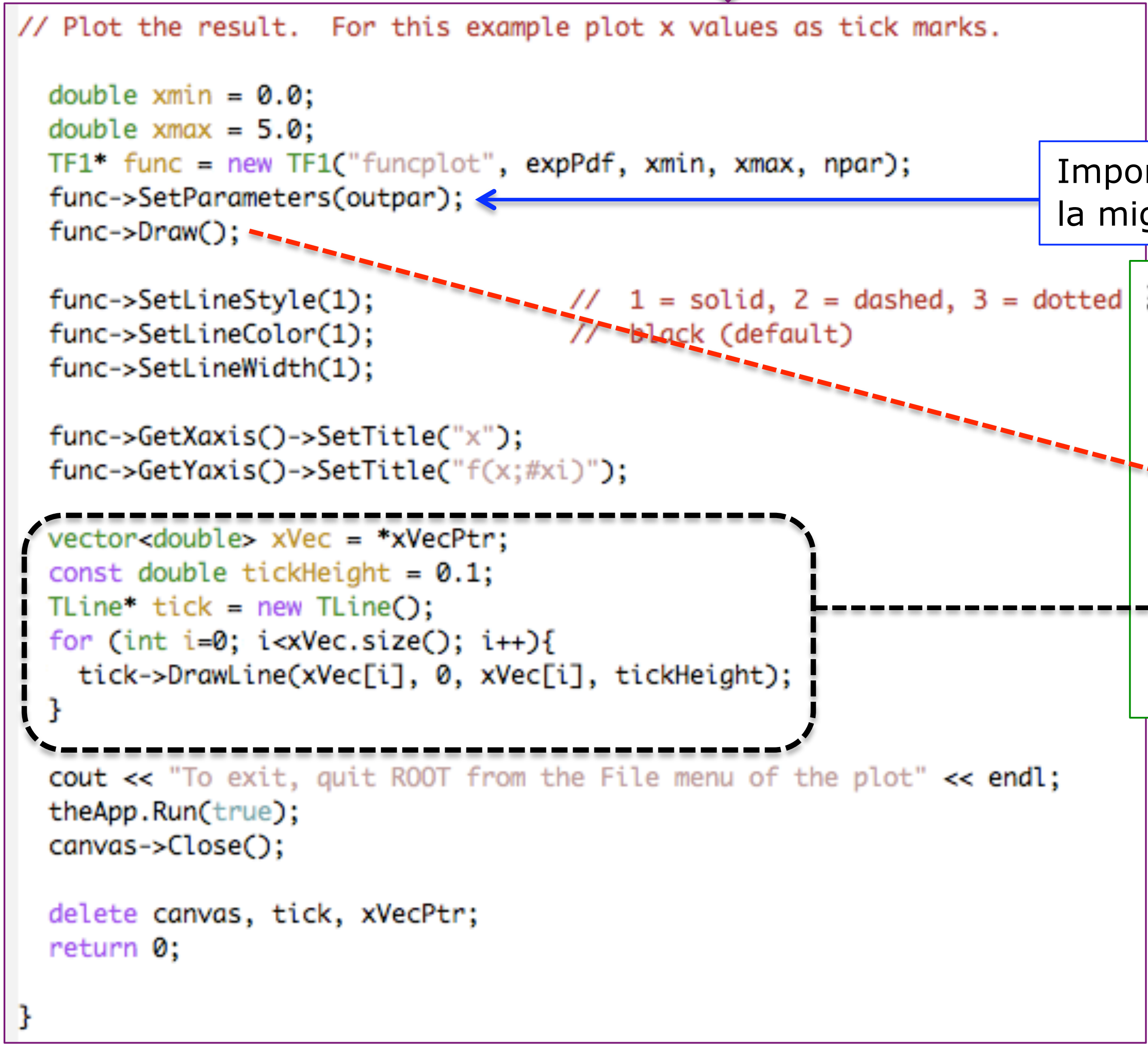
```
// Read in the data.  xVecPtr is global.

  getData(xVecPtr);

// Initialize minuit, set initial values etc. of parameters.

  const int npar = 1;              // the number of parameters
  TMinuit minuit(npar);
  minuit.SetFCN(fcn);

  double par[npar];               // the start values
  double stepSize[npar];          // step sizes
  double minVal[npar];            // minimum bound on parameter
  double maxVal[npar];            // maximum bound on parameter
  string parName[npar];

  par[0] = 2.0;            // a guess
  stepSize[0] = 0.2;       // take e.g. 0.1 of start value
  minVal[0] = 0.0000001;   // if min and max values = 0, parameter is unbounded.
  maxVal[0] = 100000000;
  parName[0] = "xi";

  for (int i=0; i<npar; i++){
    minuit.DefineParameter(i, parName[i].c_str(),
      par[i], stepSize[i], minVal[i], maxVal[i]);
  }

// Do the minimization!

  minuit.Migrad();       // Minuit's best minimization algorithm
  double outpar[npar], err[npar];
  for (int i=0; i<npar; i++){
    minuit.GetParameter(i,outpar[i],err[i]);
  }
```

qui c'e' 1 solo parametro da stimare: τ

Minimizzazione !!

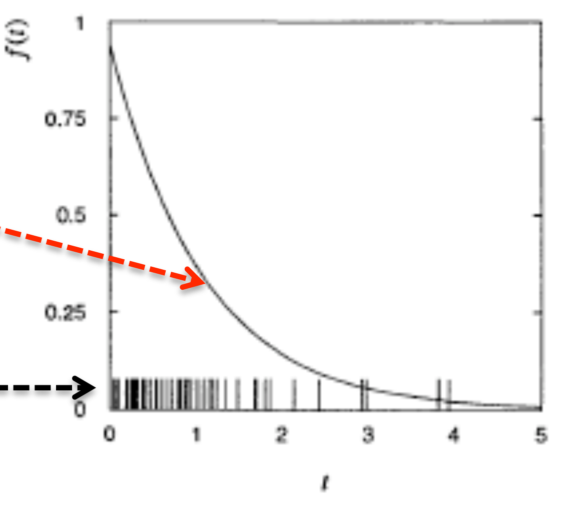
Recupera le migliori stime dei parametri!

...continua...



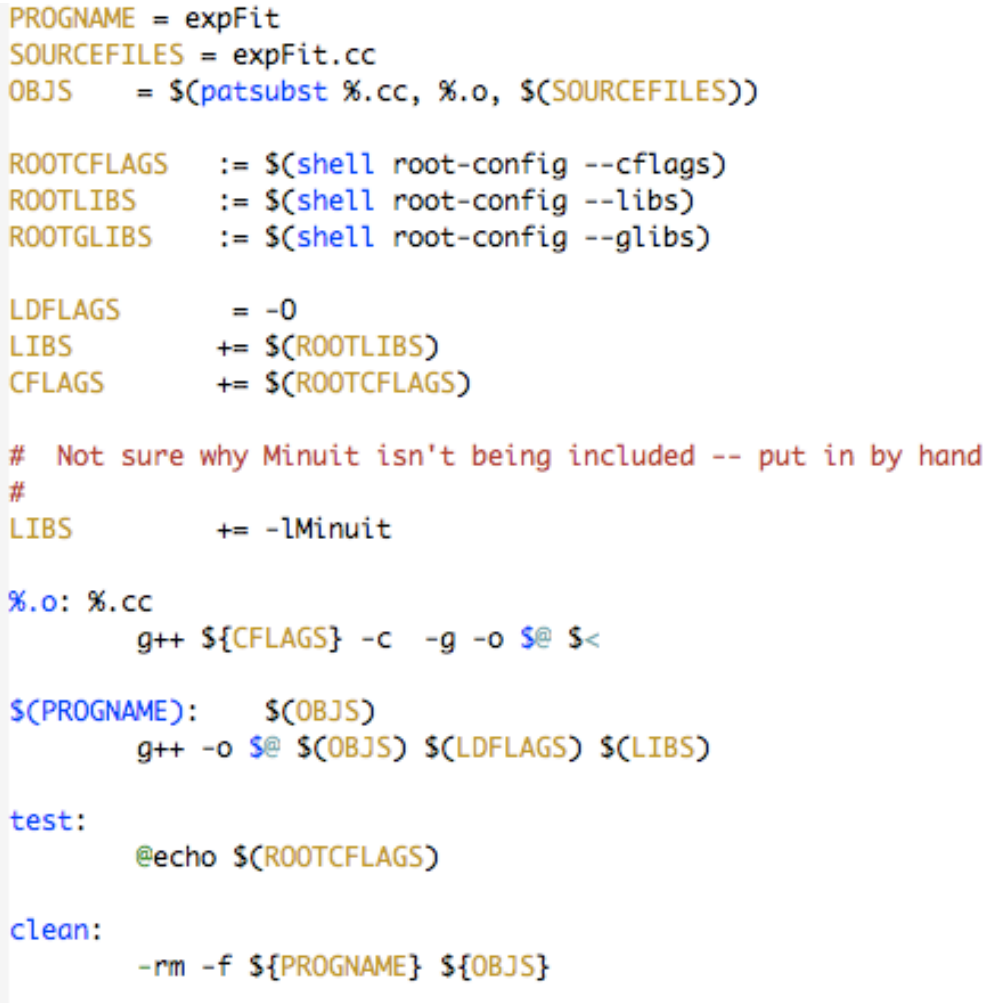
```
// Plot the result.  For this example plot x values as tick marks.

  double xmin = 0.0;
  double xmax = 5.0;
  TF1* func = new TF1("funcplot", expPdf, xmin, xmax, npar);
  func->SetParameters(outpar);
  func->Draw();

  func->SetLineStyle(1);           //  1 = solid, 2 = dashed, 3 = dotted
  func->SetLineColor(1);           //  black (default)
  func->SetLineWidth(1);

  func->GetXaxis()->SetTitle("x");
  func->GetYaxis()->SetTitle("f(x;#xi)");

  vector<double> xVec = *xVecPtr;
  const double tickHeight = 0.1;
  TLine* tick = new TLine();
  for (int i=0; i<xVec.size(); i++){
    tick->DrawLine(xVec[i], 0, xVec[i], tickHeight);
  }

  cout << "To exit, quit ROOT from the File menu of the plot" << endl;
  theApp.Run(true);
  canvas->Close();

  delete canvas, tick, xVecPtr;
  return 0;

}
```

Imponi alla PDF esponenziale
la migliora stima del parametro τ !



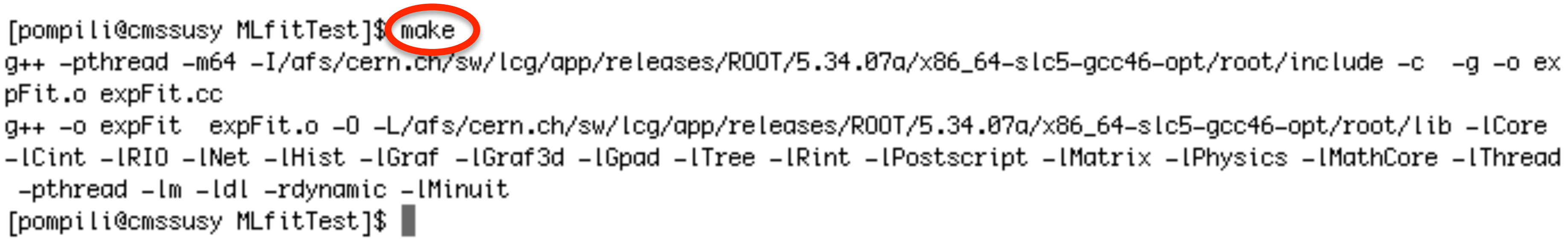
Make **e' un'*utility* Unix/Linux che costruisce automaticamente i programmi eseguibili e le librerie da codice sorgente leggendo ed eseguendo file chiamati *makefiles*.**

Nel nostro caso il file e': *GNUmakefile*



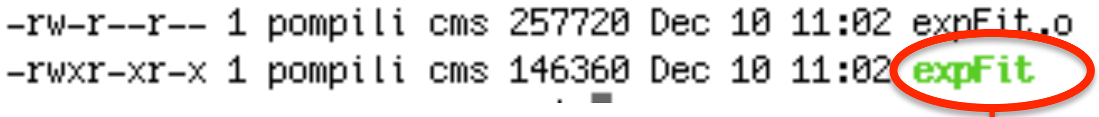
```
PROGNAME = expFit
SOURCEFILES = expFit.cc
OBJS     = $(patsubst %.cc, %.o, $(SOURCEFILES))

ROOTCFLAGS    := $(shell root-config --cflags)
ROOTLIBS      := $(shell root-config --libs)
ROOTGLIBS     := $(shell root-config --glibs)

LDFLAGS        = -O
LIBS          += $(ROOTLIBS)
CFLAGS        += $(ROOTCFLAGS)

#  Not sure why Minuit isn't being included -- put in by hand
#
LIBS          += -lMinuit

%.o: %.cc
	g++ ${CFLAGS} -c  -g -o $@ $<

$(PROGNAME):    $(OBJS)
	g++ -o $@ $(OBJS) $(LDFLAGS) $(LIBS)

test:
	@echo $(ROOTCFLAGS)

clean:
	-rm -f ${PROGNAME} ${OBJS}
```

Il comando da dare e' semplicemente (nella dir dove c'e' il *GNUmakefile*):



```
[pompili@cmssusy MLfitTest]$ make
g++ -pthread -m64 -I/afs/cern.ch/sw/lcg/app/releases/ROOT/5.34.07a/x86_64-slc5-gcc46-opt/root/include -c  -g -o ex
pFit.o expFit.cc
g++ -o expFit  expFit.o -O -L/afs/cern.ch/sw/lcg/app/releases/ROOT/5.34.07a/x86_64-slc5-gcc46-opt/root/lib -lCore
-lCint -lRIO -lNet -lHist -lGraf -lGraf3d -lGpad -lTree -lRint -lPostscript -lMatrix -lPhysics -lMathCore -lThread
 -pthread -lm -ldl -rdynamic -lMinuit
[pompili@cmssusy MLfitTest]$
```

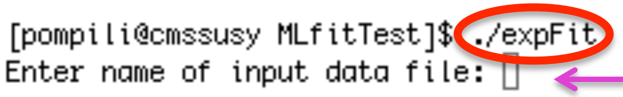
...creando i file:



```
-rw-r--r-- 1 pompili cms 257720 Dec 10 11:02 expFit.o
-rwxr-xr-x 1 pompili cms 146360 Dec 10 11:02 expFit
```

eseguibile

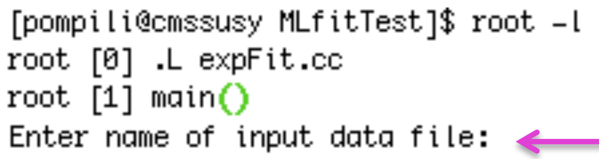
Dopodiche' basta eseguirlo:



```
[pompili@cmssusy MLfitTest]$ ./expFit
Enter name of input data file: 
```

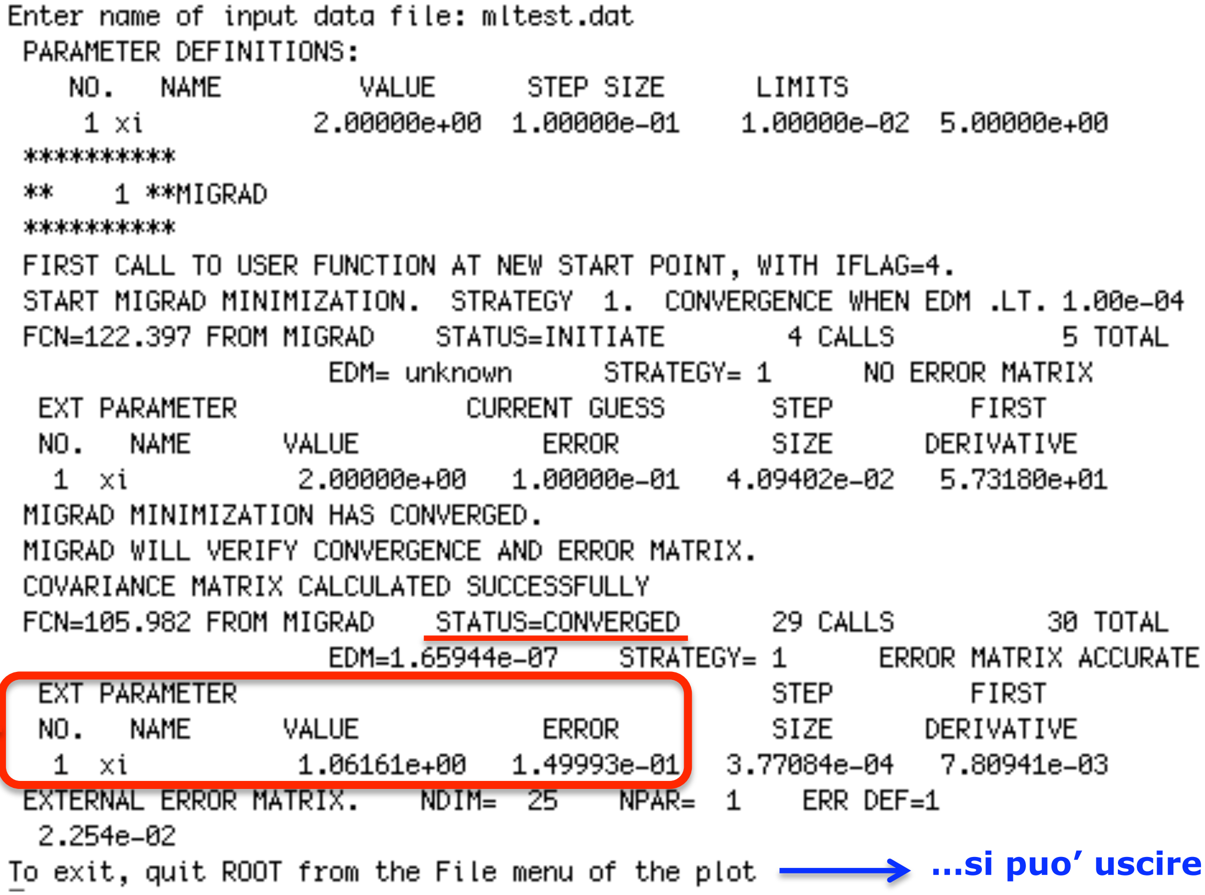
mltest.dat **(scrivere da tastiera)**

In alternativa si puo' procedere cosi':
(da dentro ROOT)
[in tal caso la compilazione e' affidata all'*interpreter* CINT]



```
[pompili@cmssusy MLfitTest]$ root -l
root [0] .L expFit.cc
root [1] main()
Enter name of input data file:
```

Il risultato a schermo e':

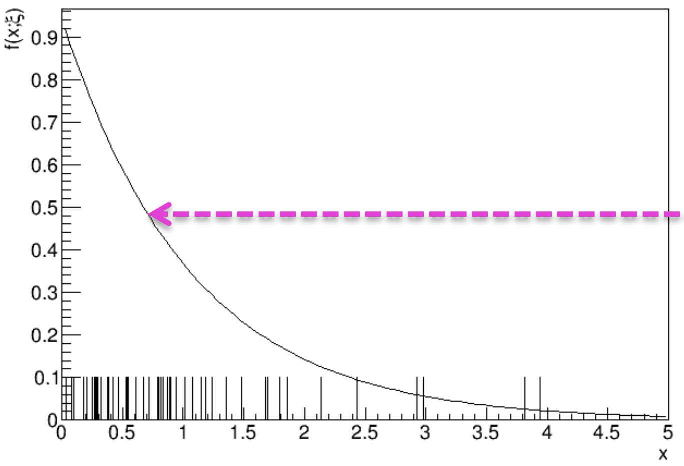


```
Enter name of input data file: mltest.dat
 PARAMETER DEFINITIONS:
    NO.   NAME         VALUE      STEP SIZE      LIMITS
     1 xi           2.00000e+00  1.00000e-01    1.00000e-02  5.00000e+00
 **********
 **    1 **MIGRAD
 **********
 FIRST CALL TO USER FUNCTION AT NEW START POINT, WITH IFLAG=4.
 START MIGRAD MINIMIZATION.  STRATEGY  1.  CONVERGENCE WHEN EDM .LT. 1.00e-04
 FCN=122.397 FROM MIGRAD    STATUS=INITIATE        4 CALLS          5 TOTAL
                     EDM= unknown      STRATEGY= 1      NO ERROR MATRIX
  EXT PARAMETER               CURRENT GUESS       STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  xi          2.00000e+00   1.00000e-01   4.09402e-02   5.73180e+01
 MIGRAD MINIMIZATION HAS CONVERGED.
 MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
 COVARIANCE MATRIX CALCULATED SUCCESSFULLY
 FCN=105.982 FROM MIGRAD    STATUS=CONVERGED      29 CALLS         30 TOTAL
                     EDM=1.65944e-07    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  xi          1.06161e+00   1.49993e-01   3.77084e-04   7.80941e-03
 EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  1    ERR DEF=1
  2.254e-02
To exit, quit ROOT from the File menu of the plot
```

$$\hat{\tau} \pm \hat{\sigma}_{\hat{\tau}} \cong 1.062 \pm 0.150$$

…si puo' uscire !

…mentre il plot nella canvas:



$$f(t;\hat{\tau}) = \frac{1}{\hat{\tau}} e^{-\frac{t}{\hat{\tau}}}$$