To start Splus, enter `Splus -e` (-e is for editor mode)

```
%  Splus -e

S-PLUS : Copyright (c) 1988, 1995 MathSoft, Inc.
S : Copyright AT&T.
Version 3.3 Release 1 for DEC alpha, OSF1 V3.2 : 1995
Working data will be in .Data

> # anything after "#" is a comment
> # Enter the Survival data:  Y is survival time in weeks
> # wbc is the White Blood cell count

> Y <- c(65, 156,100,134,16,108,121,4,39,143,56,26,22,1,1,5,65)

> wbc <- c(2300,750,4300,2600,6000,10500,10000,17000,5400,7000,9400,
           32000,35000,100000,100000,52000,100000)
> x <- log(wbc/10000)

> # Define the liklihood as a function of Y and theta

> loglik <- function(Y, theta0, theta1) {
  mu <- theta0*exp(- theta1*x)
  ll <- sum(log(dexp(Y, 1/mu)) )
return(ll) }

> #  Find Starting values

> summary(lm(log(Y) ~ x))

Call: lm(formula = log(Y) ~ x)
Residuals:
    Min      1Q Median    3Q   Max
 -1.722 -0.6108 0.2539 0.667 2.515

Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept)   3.5423   0.3027    11.7025   0.0000
          x  -0.8178   0.2140    -3.8207   0.0017

Residual standard error: 1.233 on 15 degrees of freedom
Multiple R-Squared: 0.4932
F-statistic: 14.6 on 1 and 15 degrees of freedom, the p-value is 0.001671

Correlation of Coefficients:
  (Intercept)
x -0.1562

> # find the posterior mode with a flat prior (mle)

> post.obj <- ms(~ -loglik(Y, theta0, theta1), start=list(theta0=exp(3.5),theta1=.8))
```

```
# find coefficients
> summary(post.obj)
Final value: 83.87739

Solution:
             Par.
theta0 56.8487796
theta1  0.4818293

Convergence: RELATIVE FUNCTION CONVERGENCE.

Computations done:
 Iterations Function Gradient
          7        9       20
> # define the hessian matrix  2nd derivatives of logpost wrt theta

>hessian <- function(t0, t1, y, x) {

  h <- matrix(0,2,2)
  h[1,1] <- sum(1/t0^2 - 2*exp(t1 * x) *y/ (t0^3))
  h[2,2] <- sum(-exp(t1*x)* x^2* y/t0)
  h[1,2] <- sum(exp(t1*x)*x*y/ (t0^2))
  h[2,1] <- h[1,2]
return(h)
      }

> # Observed Fisher information

> solve(-hessian(56.85, .482, Y, x))
            [,1]       [,2]
[1,] 195.2119084 0.40602991
[2,]   0.4060299 0.03228799

#  Approximate standard errors

> sqrt(diag(solve(-hessian(56.8, .48, Y, x))))
[1] 13.9474208  0.1796889
>
```