```
R version 2.15.1 (2012-06-22) -- "Roasted Marshmallows"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ##  Eryk Wdowiak
> ##  29 april 2014
>
> ##  example of Maximum Likelihood Estimation (MLE)
> ##
> ##  We will assume that the S&P 500 daily returns are distributed Cauchy
> ##  and estimate the parameters of that distribution.
>
>
> ##  First, we have to import the data (courtesy of Yahoo! Finance).
>
> dta <- read.csv( file = "qn-06_sp500.csv" )
>
>
> ##  For percentage return, we'll take the log difference. We'll take advantage
> ##  of the fact that the data file is sorted in reverse chronological order.
> ##  So all we have to do is take the negative of log of difference.
> ##  Also tack on 100 so that we can work with "percentages"
>
> dsp <- -100*diff( log( dta$Adj.Close ))
>
>
> ##  Next, we need to define the Cauchy log-likelihood function. In practice
> ##  however, we'll work with the negative of that function because "optim"
> ##  looks for a minimum (as opposed to a maximum).
> ##  Note:  par[1] is "mu" (actually "x-null") and par[2] is "gamma."
>
> LLcauchy <- function( data , par ){
+   length(data)*log(pi) + length(data)*log(par[2]) +
+     sum( log( 1+ ((data-par[1])/par[2])^2 ) )
+ }
>
> ##  Let's also define the Normal log-likelihood function.
>
> LLnormal <- function( data , par ){
+   (length(data)/2)*log(2*pi) + (length(data)/2)*log(par[2]) +
+     (1/(2*par[2]))*sum( (data-par[1])^2 )
+ }
>
>
> ##  Run "optim" and print the results. I tried to run it with Cauchy and Normal,
> ##  but was only successful with Cauchy.  (That should tell you something).
>
> ( MLEcauchy <- optim( par = c(0,1), fn = LLcauchy, data = dsp, hessian = TRUE ))
$par
[1] 0.0549530 0.4276132
```

```
$value
[1] 21759.71

$counts
function gradient
      55       NA

$convergence
[1] 0

$message
NULL

$hessian
           [,1]       [,2]
[1,] 37700.6358   481.0126
[2,]   481.0126 50797.4725

>
>
> ##  Here's the Normal MLE that I was unable to run.
> ##( MLEnormal <- optim( par = c(0,1), fn = LLnormal, data = dsp, hessian = TRUE ))
>
>
> ##  Now, let's compare the predictions of the two distributions. If S&P daily
> ##  returns were distributed normally, then about 95 percent of the returns
> ##  should fall within two standard deviations of the mean.
>
> ##  We'll focus on the 95 percent interval.
>
> loCI <- 0.025
> hiCI <- 1 - loCI
>
> LOnorm <- mean(dsp) + qnorm(loCI)*sd(dsp)
> HInorm <- mean(dsp) + qnorm(hiCI)*sd(dsp)
>
>
> ##  To get the Cauchy 95 percent interval, we plug our estimated parameters
> ##  into the Cauchy cumulative density function (CDF).
>
> CDFcauchy <- function( par, mu, gamma ){ (1/2) + (1/pi)*atan( (par-mu)/gamma ) }
>
> loCDFcauchy <- function( par, mu, gamma){ ( CDFcauchy(par,mu,gamma) - loCI )^2 }
> hiCDFcauchy <- function( par, mu, gamma){ ( CDFcauchy(par,mu,gamma) - hiCI )^2 }
>
> LOcchy <- optim( par=c(0), fn = loCDFcauchy,
+                  method = "Brent", lower = -10, upper = 10,
+                  mu = MLEcauchy$par[1] , gamma = MLEcauchy$par[2]  )$par
>
> HIcchy <- optim( par=c(0), fn = hiCDFcauchy,
+                  method = "Brent", lower = -10, upper = 10,
+                  mu = MLEcauchy$par[1] , gamma = MLEcauchy$par[2]  )$par
>
>
> ##  Finally, for comparison with empirical reality, let's compute the median and
> ##  let's find the values at the 0.25 and 97.5 percent thresholds.
>
> loEMPmark <- round(length(dsp)*loCI)
> hiEMPmark <- length(dsp) - loEMPmark
>
> LOemp <- sort(dsp)[loEMPmark]
> HIemp <- sort(dsp)[hiEMPmark]
>
>
> ##  Now, let's summarize everything we just did.
```

```
>
> TXTnorm <- "\n"
> TXTnorm <- paste(TXTnorm, "NORMAL mean   :  ", round( mean(dsp), 4),"\n",sep="")
> TXTnorm <- paste(TXTnorm, "NORMAL std dev:  ", round(   sd(dsp), 4),"\n",sep="")
> TXTnorm <- paste(TXTnorm, "NORMAL predicts that 95 percent of the", sep="")
> TXTnorm <- paste(TXTnorm, " returns will lie within:\n", sep="")
> TXTnorm <- paste(TXTnorm, "lower 95 bound: ", round( LOnorm, 4), "\n",   sep="")
> TXTnorm <- paste(TXTnorm, "upper 95 bound:  ",round( HInorm, 4), "\n\n", sep="")
>
> TXTcchy <- "\n"
> TXTcchy <- paste(TXTcchy, "CAUCHY \"mean\" :  ", sep="")
> TXTcchy <- paste(TXTcchy, round( MLEcauchy$par[1] , 4), "\n", sep="")
> TXTcchy <- paste(TXTcchy, "CAUCHY predicts that 95 percent of the", sep="")
> TXTcchy <- paste(TXTcchy, " returns will lie within:\n", sep="")
> TXTcchy <- paste(TXTcchy, "lower 95 bound: ", round( LOcchy, 4), "\n"  , sep="")
> TXTcchy <- paste(TXTcchy, "upper 95 bound:  ",round( HIcchy, 4), "\n"  , sep="")
> TXTcchy <- paste(TXTcchy, "(As a technical matter, ", sep="")
> TXTcchy <- paste(TXTcchy, "Cauchy does not have a mean or variance)\n\n",sep="")
>
> TXTemp  <- "\n"
> TXTemp  <- paste(TXTemp,  "EMPIRIC median:  ", sep="")
> TXTemp  <- paste(TXTemp,  round( median(dsp) , 4), "\n", sep="")
> TXTemp  <- paste(TXTemp,  "EMPIRICALLY 95 percent of the", sep="")
> TXTemp  <- paste(TXTemp,  " returns lie within:\n", sep="")
> TXTemp  <- paste(TXTemp,  "lower 95 bound: ", round( LOemp, 4), "\n", sep="")
> TXTemp  <- paste(TXTemp,  "upper 95 bound:  ",round( HIemp, 4), "\n", sep="")
>
> outro <- paste("\n","But look at the extreme values:","\n",sep="")
> outroLO <- paste("",round(sort(dsp)[1:10],1), collapse="\t")
> outroHI <- paste(" ",round(sort(dsp,decreasing=TRUE)[1:10],1), collapse="\t")
> outro <- paste( outro, "lo:  ", outroLO, "\n", "hi:  ", outroHI, "\n\n", sep="")
>
> cat( TXTnorm , TXTcchy , TXTemp , outro )

NORMAL mean   :  0.0292
NORMAL std dev:  0.9749
NORMAL predicts that 95 percent of the returns will lie within:
lower 95 bound: -1.8816
upper 95 bound:  1.9399


CAUCHY "mean" :  0.055
CAUCHY predicts that 95 percent of the returns will lie within:
lower 95 bound: -5.3784
upper 95 bound:  5.4883
(As a technical matter, Cauchy does not have a mean or variance)


EMPIRIC median:  0.0468
EMPIRICALLY 95 percent of the returns lie within:
lower 95 bound: -1.8993
upper 95 bound:  1.9002

But look at the extreme values:
lo:   -22.9      -9.5    -9.4    -9.2    -8.6    -7.9    -7.1    -7      -7      -6.9
hi:    11         10.2    8.7     6.8     6.7     6.3     6.2     6.1     5.6     5.3

>
> ##  quit R without saving the workspace.
> q("no")
> proc.time()
   user  system elapsed
  0.640   0.028   0.655
```