Eryk Wdowiak
original: 22 April 2012
updated: 30 April 2014

Quantitative Analysis
shape of the likelihood function

First, load the "distrib" package.

```
(%i1) load(distrib)$
```

Now, randomly draw 100 values from the standard normal.

```
(%i2) N:100$
      x:random_normal(0,1,N)$

      mnx:(1/N)*sum(x[i],i,1,N)$
      sdx:sqrt((1/N)*sum((x[i]-mnx)^2,i,1,N))$

      print("")$
      print("mean of x: ",mnx)$
      print("std. dev.: ",sdx)$
```

mean of x: -0.035296155553136
std. dev.: 1.015933462196797

Set up the log-likelihood function.

Deriviatives must be taken with respect to sigma^2, so define:

gamma == sigma^2

and take derivatives with respect to gamma.

```
(%i9) loglik(mu,gamma):= -(N/2)*log(gamma) - (N/2)*log(2*%pi)
          - (1/(2*gamma))*sum((x[i]-mu)^2,i,1,N);
```

(%o9) $\mathrm{loglik}(\mu,\Gamma):=\left(-\frac{N}{2}\right)\log(\Gamma)-\frac{N}{2}\log(2\pi)+\left(-\frac{1}{2\Gamma}\right)\sum_{i=1}^{N}(x_i-\mu)^2$

Maximize it with respect to mu and gamma.

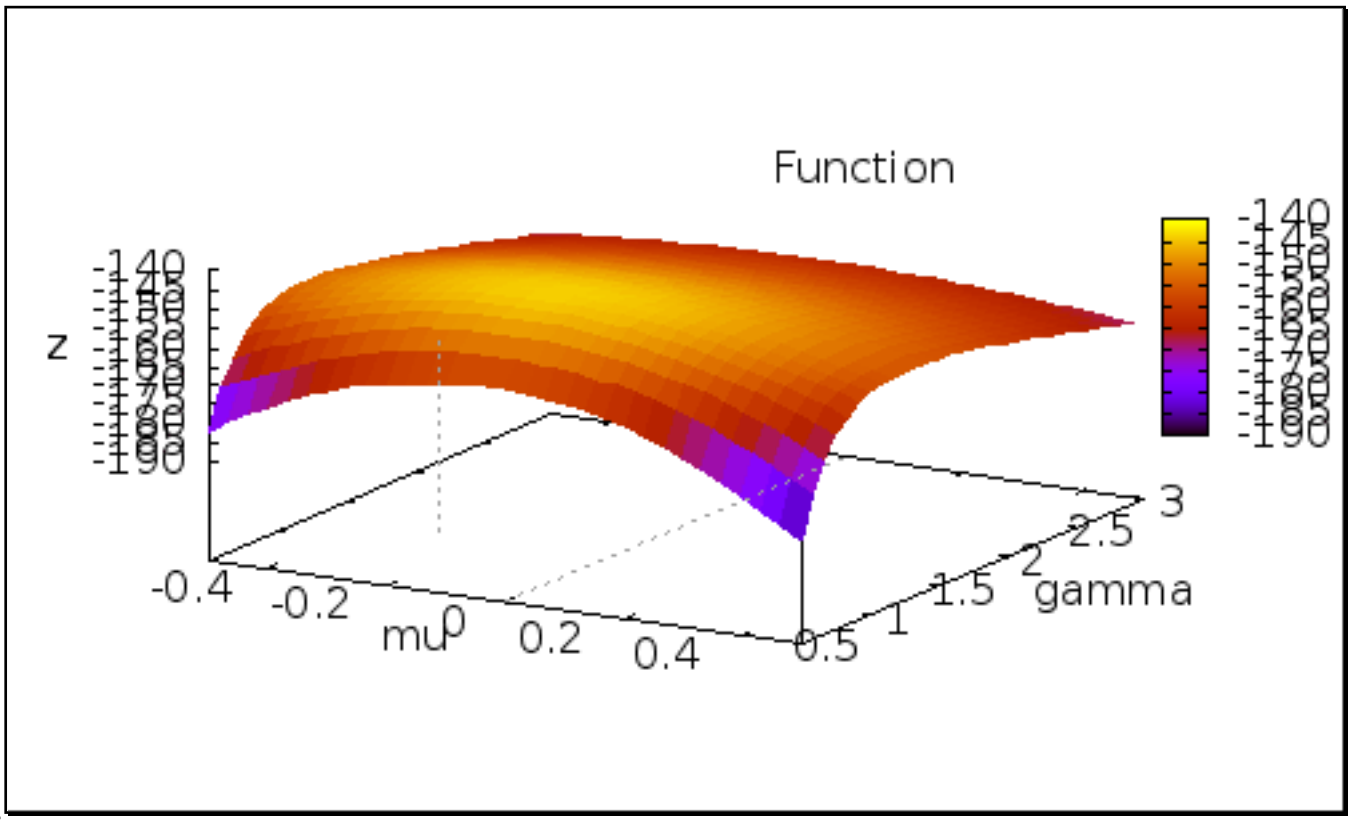
```
(%i10) sol:lbfgs(-loglik(mu,gamma),'[mu,gamma],[0.01,0.99],0.0001,[-1,0])$
       mu_max:subst(sol[1],mu)$
       gamma_max:subst(sol[2],gamma)$

       print("")$
       print(mu," = ",mu_max)$
       print(sigma^2," = ",gamma_max)$
```

μ = -0.035295710574827

σ^2 = 1.032121024914478

To see the maximum, plot the log likelihood function in "mu-gamma" space.

```
(%i16) wxplot3d(loglik(mu,gamma),[mu,-0.5,0.5],[gamma,0.5,3.0])$
(%t16)
```



Check to see if second-order conditions are satisfied.

```
(%i17) /* set up the Hessian matrix */
       dxx(mu,gamma):=''(diff(diff(loglik(mu,gamma),mu),mu))$
       dyy(mu,gamma):=''(diff(diff(loglik(mu,gamma),gamma),gamma))$
       dxy(mu,gamma):=''(diff(diff(loglik(mu,gamma),mu),gamma))$
       H:matrix(
           [dxx(mu_max,gamma_max),dxy(mu_max,gamma_max)],
           [dxy(mu_max,gamma_max),dyy(mu_max,gamma_max)])$
```

```
(%i21) print("")$
       print("own-partials must be negative:")$
       print("")$
       print("d^2 loglik(mu,gamma)"/"(d mu)^2"," = ",dxx(mu_max,gamma_max))$
       print("")$
       print("d^2 loglik(mu,gamma)"/"(d gamma)^2"," = ",dyy(mu_max,gamma_max)
       print("")$
       print("")$
       print("the cross-partial:")$
       print("")$
       print("d^2 loglik(mu,gamma)"/"d mu d gamma"," = ",dxy(mu_max,gamma_max
       print("")$
```

own-partials must be negative:

$$\frac{d^2\ loglik(mu,gamma)}{(d\ mu)^2} = -96.88786255302382$$

$$\frac{d^2\ loglik(mu,gamma)}{(d\ gamma)^2} = -46.93626905889457$$

the cross-partial:

$$\frac{d^2\ loglik(mu,gamma)}{d\ mu\ d\ gamma} = 4.1771261415998478\ 10^{-5}$$

```
(%i33) print("")$
       print("the Hessian matrix:")$
       print("H = ",H)$
       print("")$
       print("determinant of Hessian must be positive")$
       print("det(H) = ",determinant(H))$
       print("")$
```

the Hessian matrix:

$$H = \begin{bmatrix} -96.88786255302382 & 4.1771261415998478\ 10^{-5} \\ 4.1771261415998478\ 10^{-5} & -46.93626905889457 \end{bmatrix}$$

determinant of Hessian must be positive

det(H) = 4547.554785328178

```
(%i40) info:-1*invert(H)$
       print("")$
       print("the information matrix:")$
       print("-1*(H^-1) = ",info)$
       print("")$
       print(mu,": ",mu_max,"   se:",sqrt(info[1,1]))$
       print(sigma^2,": ",gamma_max,"   se:",sqrt(info[2,2]))$
       print("")$
```

the information matrix:

-1(H^-1) =* $\begin{bmatrix} 0.010321210249149 & 9.18543335657341\ 10^{-9} \\ 9.18543335657341\ 10^{-9} & 0.021305485503026 \end{bmatrix}$

μ *:* -0.035295710574827 *se:* 0.10159335730819

σ^2 *:* 1.032121024914478 *se:* 0.14596398700716