

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

## Econometrics shape of the likelihood function

First, load the "distrib" package.

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

Now, randomly draw 10 values from the standard normal.

$$y = \alpha + \beta x + u$$

```
(%i5) N:10$  

x:random_normal(0,1,N)$  

u:random_normal(0,1,N)$  

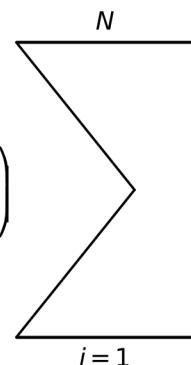
y:2+3*x+u$
```

Set up the log-likelihood function.

```
(%i6) loglik(alpha,beta,sigma):= -N·log(sigma) - (N/2)·log(2·%pi) -  

(1/2)·sum(((y[i]-alpha-(beta·x[i]))/sigma)^2,i,1,N);
```

(%o6)  $\text{loglik}(\alpha, \beta, \sigma) := (-N) \log(\sigma) - \frac{N}{2} \log(2\pi) + \left(-\frac{1}{2}\right) \sum_{i=1}^N \left(\frac{y_i - \alpha - \beta x_i}{\sigma}\right)^2$



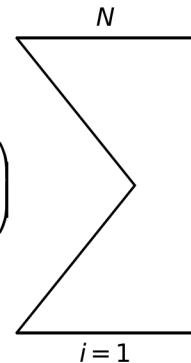
Derivatives must be taken with respect to  $\sigma^2$ , so define:

$\text{gamma} == \sigma^2$

and take derivatives with respect to gamma.

```
(%i7) loglik(alpha,beta,gamma):= -(N/2)·log(gamma) - (N/2)·log(2·%pi) -
(1/(2·gamma))·sum((y[i]-alpha-(beta·x[i]))^2,i,1,N);
```

$$(\%o7) \text{ loglik}(\alpha, \beta, \gamma) := \left( -\frac{N}{2} \right) \log(\gamma) - \frac{N}{2} \log(2 \pi) + \left( -\frac{1}{2 \gamma} \right) \sum_{i=1}^N (y_i - \alpha - \beta x_i)^2$$



Maximize it with respect to alpha, beta and gamma.

```
(%i11) sol:lbfgs(-loglik(alpha,beta,gamma),[alpha,beta,gamma],[2.01,2.99,1.01],0.0001,[-1,0]
alfa_hat:subst(sol[1],alpha)$
beta_hat:subst(sol[2],beta)$
gmma_hat:subst(sol[3],gamma)$
```

```
(%i15) print("")$
print(alpha," = ",alfa_hat)$
print(beta," = ",beta_hat)$
print(sigma^2," = ",gmma_hat)$
```

$$\begin{aligned}\alpha &= 2.084553181758032 \\ \beta &= 2.815648702811899 \\ \sigma^2 &= 1.397010319508444\end{aligned}$$

First-order conditions imply that:

```
(%i22) dla(alpha,beta,gamma):="(diff(loglik(alpha,beta,gamma),alpha))$
dlb(alpha,beta,gamma):="(diff(loglik(alpha,beta,gamma),beta))$
dls(alpha,beta,gamma):="(diff(loglik(alpha,beta,gamma),gamma))$

foc:float(solve([
    dla(alpha,beta,gamma)=0,
    dlb(alpha,beta,gamma)=0,
    dls(alpha,beta,gamma)=0],[alpha,beta,gamma]))$

alfa_foc:subst(foc[1][1],alpha)$
beta_foc:subst(foc[1][2],beta)$
gmma_foc:subst(foc[1][3],gamma)$
```

```
(%i26) print("")$  
    print(alpha," = ",alfa_foc)$  
    print(beta," = ",beta_foc)$  
    print(sigma^2," = ",gmma_foc)$
```

$\alpha = 2.08455755554346$   
 $\beta = 2.815645391810164$   
 $\sigma^2 = 1.397000323442532$

Check to see if second-order conditions are satisfied.

```
(%i33) /* set up the Hessian matrix */
```

```
daa(alpha,beta,gamma):="(diff(diff(loglik(alpha,beta,gamma),alpha),alpha))$  
dab(alpha,beta,gamma):="(diff(diff(loglik(alpha,beta,gamma),alpha),beta))$  
das(alpha,beta,gamma):="(diff(diff(loglik(alpha,beta,gamma),alpha),gamma))$  
  
dbb(alpha,beta,gamma):="(diff(diff(loglik(alpha,beta,gamma),beta),beta))$  
dbs(alpha,beta,gamma):="(diff(diff(loglik(alpha,beta,gamma),beta),gamma))$  
  
dss(alpha,beta,gamma):="(diff(diff(loglik(alpha,beta,gamma),gamma),gamma))$  
  
H:matrix(  
    [daa(alfa_foc,beta_foc,gmma_foc),dab(alfa_foc,beta_foc,gmma_foc),das(alfa_foc,beta_foc,gmma_foc)],  
    [dab(alfa_foc,beta_foc,gmma_foc),dbb(alfa_foc,beta_foc,gmma_foc),dbs(alfa_foc,beta_foc,gmma_foc)],  
    [das(alfa_foc,beta_foc,gmma_foc),dbs(alfa_foc,beta_foc,gmma_foc),dss(alfa_foc,beta_foc,gmma_foc)])
```

```
(%i48) print("")$  
    print("own-partials must be negative:")$  
    print("")$  
    print("d^2 loglik"/"(d alpha)^2," = ",daa(alfa_foc,beta_foc,gmma_foc))$  
    print("")$  
    print("d^2 loglik"/"(d beta)^2," = ",dbb(alfa_foc,beta_foc,gmma_foc))$  
    print("")$  
    print("d^2 loglik"/"(d gamma)^2," = ",dss(alfa_foc,beta_foc,gmma_foc))$  
    print("")$  
    print("")$  
    print("the Hessian matrix:")$  
    print("H = ",H)$  
    print("")$  
    print("determinant of Hessian must be negative")$  
    print("det(H) = ",determinant(H))$
```

*own-partials must be negative:*

$$\frac{d^2 \loglik}{(d \alpha)^2} = -7.15819447726231$$

$$\frac{d^2 \loglik}{(d \beta)^2} = -5.948390501417307$$

$$\frac{d^2 \loglik}{(d \gamma)^2} = -2.561987408715431$$

*the Hessian matrix:*

$$H = \begin{pmatrix} -7.15819447726231 & 0.1862052458522994 & -3.185702699150288 \cdot 10^{-15} \\ 0.1862052458522994 & -5.948390501417307 & -6.143855205504127 \cdot 10^{-15} \\ -3.185702699150288 \cdot 10^{-15} & -6.143855205504127 \cdot 10^{-15} & -2.561987408715431 \end{pmatrix}$$

*determinant of Hessian must be negative*

$$\det(H) = -108.999917354472$$

Calculate the standard error of alpha, beta and sigma^2.

```
(%i56) info:-1·invert(H)$

print("")$  

print("the information matrix:")$  

print("-1·(H^-1) = ",info)$  

print("")$  

print(alpha," = ",alfa_foc," , se: ",sqrt(info[1,1]))$  

print(beta," = ",beta_foc," , se: ",sqrt(info[2,2]))$  

print(sigma^2," = ",gmma_foc," , se: ",sqrt(info[3,3]))$
```

*the information matrix:*

$$-1*(H^{-1}) = \begin{pmatrix} 0.1398138818508779 & 0.004376659238730881 & -1.843471282610332 \cdot 10^{-1} \\ 0.004376659238730881 & 0.1682497066510845 & -4.089187041353097 \cdot 10^{-1} \\ -1.843471282610332 \cdot 10^{-16} & -4.089187041353097 \cdot 10^{-16} & 0.3903219807397085 \end{pmatrix}$$

$\alpha = 2.08455755554346$  , se: 0.3739169451240181

$\beta = 2.815645391810164$  , se: 0.4101825284566427

$\sigma^2 = 1.397000323442532$  , se: 0.6247575375613395