Eryk Wdowiak
original: 08 October 2012
updated: 14 October 2012

Econometrics
MLE coefficients and standard error

preliminaries:

```
(%i2) load(distrib)$
ratprint:false$
```

linear regression model:
y = alpha + beta*x + u

error term is normally distributed with mean zero and constant variance
u ~ N(0,gamma)
where: gamma == sigma^2

the log-likelihood function:

```
(%i3) 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);
```

(%o3)
$$\mathrm{loglik}(\alpha,\beta,\gamma):=\left(-\frac{N}{2}\right)\log(\gamma)-\frac{N}{2}\log(2\pi)+\left(-\frac{1}{2\gamma}\right)\sum_{i=1}^{N}(y_i-\alpha-\beta x_i)^2$$

We want to maximize the log-likelihood function, so let's first give ourselves
a visual image of what a likelihood function is and what it looks like.

Here, we'll randomly generate some data with alpha set to zero and look at the
shapes in "beta-gamma" space.

(%i14)
```
NN:100$
xx:random_normal(0,1,NN)$
uu:random_normal(0,1,NN)$
yy:0+2·xx+uu$

LnL_visual(beta,gamma):= −(NN/2)·log(gamma) − (NN/2)·log(2·%pi)
                    −(1/(2·(gamma)))·sum((yy[ii]−(beta·xx[ii]))^2,ii,1,NN)$
PDF_visual(beta,gamma):= ''(exp(LnL_visual(beta,gamma)))$

parms:lbfgs(−1·LnL_visual(beta,gamma), [beta,gamma], [1.0,1.0], 1e−4, [−1,0])$

print("")$
print("Maximizing the log of the likelihood function yields the following parameter estimate
print(beta,"=",subst(parms[1],beta))$
print(gamma,"=",subst(parms[2],gamma))$
```

Maximizing the log of the likelihood function yields the following parameter estimates:

β = 2.0699529966427

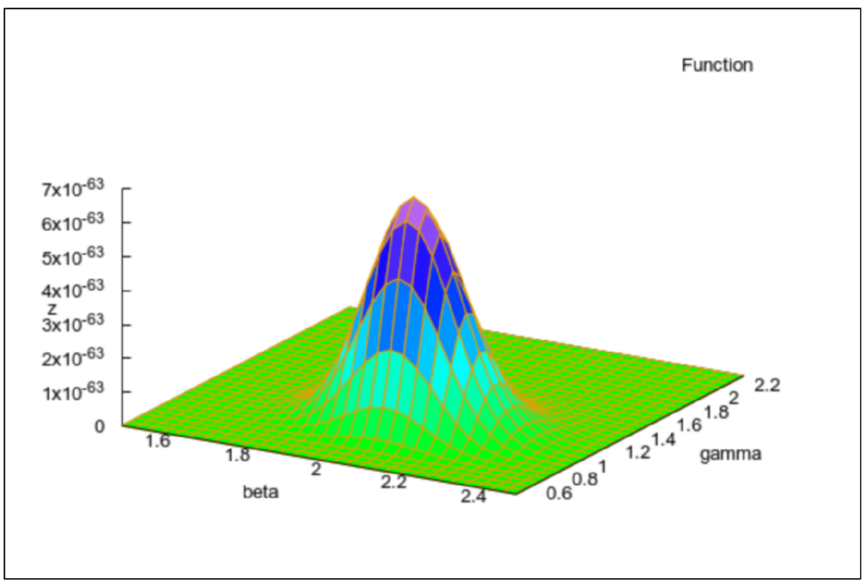
γ = 1.025179015053648

First, let's look at the likelihood function. Note that:

- the estimates of beta follow a t-distribution
- the estimates of gamma follow a chi-squared distribution

(%i15)
```
wxplot3d(PDF_visual(beta,gamma),[beta,1.5,2.5],[gamma,0.5,2.25])$
```

(%t15)

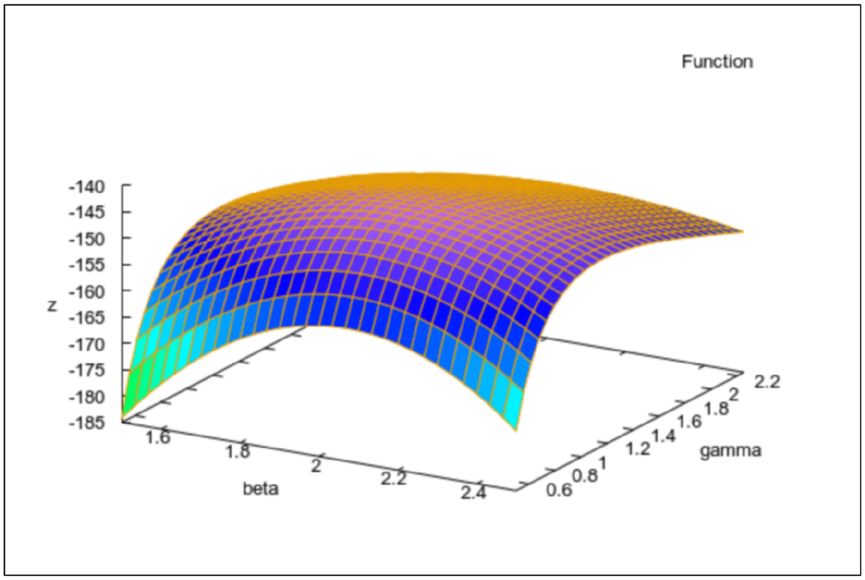


Now, let's look at the log likelihood function. You'll immediately notice the

more gradual shape, but what's important to remember is that the same parameter
values that maximize the likelihood function also maximize the log-likelihood
function. Why? Because the log-likelihood function is a monotone transformation
of the likelihood function.

```
(%i16) wxplot3d(LnL_visual(beta,gamma),[beta,1.5,2.5],[gamma,0.5,2.25])$
```

(%t16)



We want to find the parameter values that maximize the likelihood of

having generated this particular sample.

So maximize the log-likelihood function with respect to the parameters.

first-order conditions:

```
(%i19) 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))$
```

```
(%i24) print("")$
       print("The first order conditions are:")$

       print("d loglik"/"d alpha"," = ",dla(alpha,beta,gamma)," = 0")$
       print("which implies that the error terms are zero on average.")$
       print("(We made this assumption when we set up the likelihood function).")$
```

The first order conditions are:

$$\frac{d\ loglik}{d\ alpha} = \frac{\sum_{i=1}^{N} y_i - \beta\, x_i - \alpha}{\gamma} = 0$$

which implies that the error terms are zero on average.

(We made this assumption when we set up the likelihood function).

```
(%i28) print("")$
       print("d loglik"/"d beta"," = ",dlb(alpha,beta,gamma)," = 0")$
       print("which implies that the error terms are not correlated with the regressor.")$
       print("(This is one of the Gauss−Markov assumptions).")$
```

$$\frac{d\ loglik}{d\ beta} = \frac{\sum_{i=1}^{N} x_i\,(y_i - \beta\, x_i - \alpha)}{\gamma} = 0$$

which implies that the error terms are not correlated with the regressor.

(This is one of the Gauss−Markov assumptions).

```
(%i33) print("")$
       print("d loglik"/"d gamma"," = ",dls(alpha,beta,gamma)," = 0")$
       print("which implies that the error terms have constant variance: ")$
       print(solve(dls(alpha,beta,gamma),gamma)[1])$
       print("(We made this assumption when we set up the likelihood function).")$
```

$$\frac{d\ loglik}{d\ gamma} = \frac{\sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2}{2\,\gamma^2} - \frac{N}{2\,\gamma} = 0$$

which implies that the error terms have constant variance:

$$\gamma = \frac{\sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2}{N}$$

(We made this assumption when we set up the likelihood function).

second-order conditions:

```
(%i39) 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))$
```

```
(%i49) print("")$
       print("The second−order conditions are:")$
       print("")$
       print("own−partials must be negative:")$
       print("")$
       print("d^2 loglik"/"(d alpha)^2"," = ",daa(alpha,beta,gamma)," < 0")$
       print("")$
       print("d^2 loglik"/"(d beta)^2"," = ",dbb(alpha,beta,gamma)," < 0")$
       print("")$
       print("d^2 loglik"/"(d gamma)^2"," = ",dss(alpha,beta,gamma)," < 0")$
```

The second−order conditions are:

own−partials must be negative:

$$\frac{d\hat{}2\ loglik}{(d\ alpha)\hat{}2} = -\frac{N}{\gamma} < 0$$

$$\frac{d\hat{}2\ loglik}{(d\ beta)\hat{}2} = -\frac{\sum_{i=1}^{N} x_i^2}{\gamma} < 0$$

$$\frac{d\hat{}2\ loglik}{(d\ gamma)\hat{}2} = \frac{N}{2\,\gamma^2} - \frac{\sum_{i=1}^{N} (y_i - \beta\,x_i - \alpha)^2}{\gamma^3} < 0$$

```
(%i55) print("")$
print("when evaluated at the minimum:")$
solve(dls(alpha,beta,gamma),gamma)[1];

/* so we'll set */
gamma_hat:subst(solve(dls(alpha,beta,gamma),gamma)[1],gamma)$

print("therefore, at the minimum:")$
print("d^2 loglik"/"(d gamma_hat)^2"," = ",dss(alpha,beta,gamma_hat)," < 0")$
```

when evaluated at the minimum:

(%o52) $$\gamma = \frac{\sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2}{N}$$

therefore, at the minimum:

$$\frac{d\text{^}2\ loglik}{(d\ gamma\_hat)\text{^}2} = -\frac{N^3}{2\left(\sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2\right)^2} < 0$$

Set up the Hessian matrix.

```
(%i59) H:matrix(
           [daa(alpha,beta,gamma_hat),dab(alpha,beta,gamma_hat),das(alpha,beta,gamma_hat)
           [dab(alpha,beta,gamma_hat),dbb(alpha,beta,gamma_hat),dbs(alpha,beta,gamma_hat)
           [das(alpha,beta,gamma_hat),dbs(alpha,beta,gamma_hat),dss(alpha,beta,gamma_hat)]

       print("")$
       print("the Hessian matrix:")$
       print("H = ",1/gamma_hat,gamma_hat·H)$
```

the Hessian matrix:

$$-N \qquad -\sum_{i=1}^{N} x_i$$

$$H = \frac{N}{\sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2} \qquad -\sum_{i=1}^{N} x_i \qquad -\sum_{i=1}^{N} x_i^2 \qquad -$$

Note that the when the first order conditions are satisfied,
the numerators of the terms on the "borders" evaluate to zero.

```
(%i67)              H[1,3]:0$
                    H[2,3]:0$
H[3,1]:0$  H[3,2]:0$

print("")$
print("At the parameter values that maximize the likelihood function,")$
print("the Hessian matrix simplifies to:")$
print("H = ",1/gamma_hat,gamma_hat·H)$
```

At the parameter values that maximize the likelihood function,
the Hessian matrix simplifies to:

$$H = \frac{N}{\sum_{i=1}^{N} (y_i - \beta x_i - \alpha)^2} \begin{pmatrix} -N & -\sum_{i=1}^{N} x_i & 0 \\ -\sum_{i=1}^{N} x_i & -\sum_{i=1}^{N} x_i^2 & 0 \\ 0 & 0 & -\frac{N^2}{2 \sum_{i=1}^{N} (y_i - \beta x_i - \cdots} \end{pmatrix}$$

Let's furthur substitute in our estimate of gamma.

define "gamma hat": gh ==
sum((y[i]-beta*x[i]-alpha)^2,i,1,N)/N$

```
(%i72) ghH:''(gamma_hat·H)$
       ghH[3,3]:(−N/(2·gh))$
       H:''((1/gh)·ghH)$

       print("")$
       print("H = ",(1/gh),ghH)$
```

$$H = \frac{1}{gh}\begin{pmatrix} -N & -\sum_{i=1}^{N} x_i & 0 \\ -\sum_{i=1}^{N} x_i & -\sum_{i=1}^{N} x_i^2 & 0 \\ 0 & 0 & -\frac{N}{2\,gh} \end{pmatrix}$$

When the likelihood function (and, by extension, the log-likelihood function)
comes to a sharper peak, we have a more accurate estimate of alpha and beta.

The negative of the inverse of Hessian matrix is the "information matrix."

The square roots of the diagonal elements of the upper left block are the

standard errors of alpha and beta.

You can think of the lower right element as the "variance of the variance."
Our estimation strategy yields an estimate of the population variance, so
it too is measured with error.

```
(%i75) print("")$
detH:determinant(H)$
print("−1·(H^−1) = ",ratsimp(−1/detH),detH·invert(H))$
```

$$-1*(H\hat{}-1) = \frac{2\,gh^4}{N^2\left(\sum_{i=1}^{N} x_i^2\right) - N\left(\sum_{i=1}^{N} x_i\right)^2} \begin{pmatrix} \frac{N\sum_{i=1}^{N} x_i^2}{2\,gh^3} & -\frac{N\sum_{i=1}^{N}}{2\,gh^3} \\ -\frac{N\sum_{i=1}^{N} x_i}{2\,gh^3} & \frac{N^2}{2\,gh^3} \\ 0 & 0 \end{pmatrix}$$

which upon furthur simplification becomes:

```
(%i78) IM_left:''(ratsimp(((N^2)·gh)/detH))/(2·(gh^4))$
       IM_right:''(ratsimp((1/IM_left)·invert(H)))$

       print("−1·(H^−1) = ",IM_left,"",IM_right)$
```

$$-1*(H\hat{\ }-1) = -\frac{N\,gh}{\left(N\left(\sum_{i=1}^{N} x_i^2\right)-\left(\sum_{i=1}^{N} x_i\right)^2\right)} \begin{pmatrix} \frac{\sum_{i=1}^{N} x_i^2}{N} & -\frac{\sum_{i=1}^{N} x_i}{N} & \cdots \\ -\frac{\sum_{i=1}^{N} x_i}{N} & 1 & \cdots \\ 0 & 0 & \cdots \end{pmatrix}$$