***Estimación de Mínimos Cuadrados (OLS) Utilizando Maximum Likelihood: Unconstrained Optimization, Comando fminunc, estimando OLS a traves de MLE***

Recordemos que uno de los supuestos de OLS es que los errores se distribuyen como una normal con media cero y volatilidad sigma.

$$\varepsilon \sim N(0,\sigma^2) \tag{280}$$

Recordemos que $\varepsilon_i = y_i - \beta X_i$, entonces la probabilidad de que el error $\varepsilon_i$ ocurra es:

$$f(\varepsilon_i,\mu,\sigma) = \frac{e^{-[(\varepsilon_i)^2]/(2\sigma^2)}}{\sigma\sqrt{2\pi}} = \frac{e^{-[(y_i-\beta X_i)^2]/(2\sigma^2)}}{\sigma\sqrt{2\pi}} \tag{281}$$

De esta forma podemos armar la likelihood y log-likelihood function:

$$\text{Likelihood} = \prod_{i=1}^{n} \frac{e^{-[(y_i-\beta X_i)^2]/(2\sigma^2)}}{\sigma\sqrt{2\pi}} \quad \text{Log - Likelihood} = \sum_{i=1}^{n} \ln\left(\frac{e^{-[(y_i-\beta X_i)^2]/(2\sigma^2)}}{\sigma\sqrt{2\pi}}\right) \tag{282}$$

La log-likelihood se puede reexpresar así suponiendo que sólo hay un regresor para simplificar el álgebra:

$$\text{Log - Lik} = \sum_{i=1}^{n} \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \sum_{i=1}^{n} \frac{(y_i-\beta x_i)^2}{2\sigma^2} \tag{283}$$

$$\text{Log - Lik} = n\ln(1) - n\ln(\sigma) - n\ln(\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(y_i-\beta x_i)^2}{2\sigma^2}) = -n\ln(\sigma) - n\ln(\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(y_i-\beta x_i)^2}{2\sigma^2}$$

La FOC (first order condition) para Beta es:

$$\frac{\partial \ln L}{\partial \beta} = \sum_{i=1}^{n} \frac{(y_i-\beta x_i)x_i}{\sigma^2} = 0 \Rightarrow \sum_{i=1}^{n} x_i(y_i-\beta x_i) = 0 \tag{284}$$

Comparemos esta condición con la condición que Gauss le exigió a OLS. Recordemos que Gauss minimiza la suma de los errores al cuadrado:

$$Min_\beta \sum_{i=1}^{n} (y_i-\beta x_i)^2 \Rightarrow \text{FOC} = \sum_{i=1}^{n} -2x_i(y_i-\beta x_i) = 0 \Rightarrow \sum_{i=1}^{n} x_i(y_i-\beta x_i) = 0 \tag{285}$$

Como vemos, la condición de Gauss (OLS) es la misma que la que resulta de aplicar MLE. Es por esto que podemos sacar una conclusión fundamental: ***el valor estimado de Beta es idéntico tanto en OLS como en MLE.*** Vamos ahora a ver cuál es el valor estimado de sigma que resulta de aplicar MLE. La FOC (first order condition) para Sigma es:

$$\frac{\partial \ln L}{\partial \sigma} = \sum_{i=1}^{n} -\frac{1}{\sigma} + \frac{(y_i-\beta x_i)^2}{\sigma^3} = 0 \Rightarrow \sum_{i=1}^{n} \frac{(y_i-\beta x_i)^2}{\sigma^3} = \frac{n}{\sigma} \Rightarrow \sigma^2 = \sum_{i=1}^{n} \frac{(y_i-\beta x_i)^2}{n} = \sum_{i=1}^{n} \frac{(e_i)^2}{n} \tag{286}$$

Recordemos que en OLS el estimador de la varianza es:

$$\sigma^2 = \sum_{i=1}^{n} \frac{(e_i)^2}{n-k} \tag{287}$$

Es por esta razón que concluimos que OLS y MLE difieren en cuanto a la estimación de la varianza la cual es sesgada para el caso de MLE si bien en muestras grandes se hace insignificante.

***Estimando Unconstrained OLS a través de MLE***

***Definiendo Matrix de Regresores***

```
xx=normrnd(0,1,Size,2);
```

***Definiendo los Starting Values de la Optimización***

```
x0=Starting;
```

***Regression Shock***

```
Random=normrnd(0,sqrt(Beta(4,1)),Size,1);
```

***Definiendo OLS Relation***

```
y=Beta(1,1)*ones(size(xx,1),1)+Beta(2,1)*xx(:,1)+Beta(3,1)*xx(:,2)+Random;
```

***Definiendo Optimization Options***

```
options = optimset('LargeScale','off','Display','iter', 'TolX',PREC);
```

***Llamando al Optimizador***

```
[x] = fminsearch('MyOLSMLE',x0,options,y,xx);
```

***Importante: fminsearch MINIMIZA la función objetivo***

***Maximum Likelihood Estimation***

```
xmle=x;
```

***OLS Estimation***

```
xx=[ones(size(xx,1),1) xx];
xols=inv((xx')*((xx)))*((xx')*y);
Sigmaols=((sum((y-xols(1,1)*xx(:,1)-xols(2,1)*xx(:,2)-
xols(3,1)*xx(:,3)).^2))/(size(xx,1)-3));
xols=[xols;Sigmaols];
```

***Comparison Between OLS and MLE***

```
Comp=[xols xmle];
```

***Construcción de la Log-Likelihood Function Para Estimar OLS por MLE***

```
function [gos,x]=MyOLSMLE(x,y,xx);

Error=y-[x(1,1)*ones(size(xx,1),1)+x(2,1)*xx(:,1)+x(3,1)*xx(:,2)];

aa=((1/sqrt(2*pi*x(4,1)))*[exp(-0.5.*((Error).*(Error)/(x(4,1))))]);

Optimizador minimiza, entonces debemos definir negativa de la log-likelihood

gos=-sum(log(aa)/size(xx,1));
```

***Mixed Distributions***

Hasta ahora sólo analizamos la aplicación de MLE para casos relativamente triviales, es decir, casos en donde la estimación pudo haberse hecho perfectamente mediante métodos lineales. Por ejemplo, para el caso de la normal, tanto su media y varianza puede estimarse utilizando estimadores standards. Lo mismo ocurre para la media y varianza de las distribuciones Poisson y Bernoulli.

Sin embargo, la verdadera utilidad de MLE es para casos de funciones no lineales, para las que los métodos tradicionales de estimación no ofrecen solución. Vamos a ver varios de estos casos, para los que en realidad se creó el método de Maximum Likelihood.

Supongamos que tenemos una distribución mixta:

$$\text{Funcion de Densidad de } y_i = \lambda[\frac{e^{-[(y_i-\mu_1)^2]/(2\sigma_1{}^2)}}{\sigma_1\sqrt{2\pi}}] + (1-\lambda)[\frac{e^{-[(y_i-\mu_2)^2]/(2\sigma_2{}^2)}}{\sigma_2\sqrt{2\pi}}] \quad (257)$$

Es decir, la variable aleatoria $y_i$, proviene de una distribución normal con media: $\mu_1$ y varianza $\sigma_1$ con probabilidad $\lambda$ y de una distribución normal con media: $\mu_2$ y varianza $\sigma_2$ con probabilidad $(1-\lambda)$. Es como si fuera una lotería, yo no sé exactamente de cuál de las dos distribuciones proviene la observación $y_i$. Como vemos, si suponemos que sólo sabemos que la variable aleatoria $y_i$ proviene de una de dos distribuciones normales, podemos estimar los 5 parámetros de la función $(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)$ utilizando MLE. Vemos que en este caso sería imposible aplicar algún método lineal de estimación.

$$Max_{\mu_1,\sigma_1,\mu_2,\sigma_2,\lambda} \sum_{i=1}^{n} \ln [\lambda(\frac{e^{-[(y_i-\mu_1)^2]/(2\sigma_1{}^2)}}{\sigma_1\sqrt{2\pi}}) + (1-\lambda)(\frac{e^{-[(y_i-\mu_2)^2]/(2\sigma_2{}^2)}}{\sigma_2\sqrt{2\pi}})] \quad (258)$$

Vemos entonces que podemos maximizar la log-likelihood con respecto a los 5 parámetros. Sin embrago, esta no es una tarea obvia dado que de acuerdo a la relación que exista entre $(\mu_1, \sigma_1, \mu_2, \sigma_2)$ mejor será la información que le proveeremos a MLE.

Por ejemplo, si $(\mu_1, \mu_2)$ son muy parecidas, al método le costará mucho distinguir de cuál de las dos distribuciones proviene una observación particular. En estos casos los resultados de la estimación van a ser muy malos. Por el contrario, si $(\mu_1, \mu_2)$ son muy distintas, al método le será fácil calibrar una log-likelihood consistente con este hecho.

***Varianza Asintotica: Teorema de Cramer Rao***

Como vimos, hasta ahora solamente nos concentramos en maximizar la Log-Likelihood Function pero no nos preguntamos si los estimadores MLE eran "buenos" o "malos". Es decir, no analizamos las propiedades de estos estimadores. Antes de analizar las propiedades, analicemos el siguiente teorema:

***La varianza de un estimador consistente del parámetro poblacional $\theta$ nunca es menor que:***

$$[-E(\frac{\partial \ln^2(L(\theta))}{\partial\theta^2})]^{-1} \quad (227)$$

Es decir, la varianza de un estimador consistente jamás es menor que la negativa de la esperanza de la inversa de la derivada segunda de la Log-Likelihood Function. Obvio, aplicar este teorema no es sencillo porque muchas veces la derivada segunda de la Log-Likelihood es una función no lineal cuya esperanza matemática puede ser muy difícil de calcular. Es por esta razón, que generalmente a este lower bound para la varianza se lo calcula así:

$$-[(\frac{\partial \ln^2(L(\hat{\theta}))}{\partial \hat{\theta}^2})^{-1}] \quad (228)$$

Es decir, al lower bound teórico se lo aproxima evaluando la negativa de la matriz hessiana en el valor del parámetro estimado. Cuál es la intuición de esta última expresión. En realidad es muy sencillo, recordemos que en primer lugar como estamos maximizando, las derivadas segundas en el máximo se tornan negativas. Es por eso que la expresión tiene el sigo menos delante, dado que la varianza nunca puede ser negativa. En segundo lugar, recordemos que la derivada segunda en el óptimo mide la curvatura que la función alcanza en eso óptimo. Cuanto mas curva es la función en el óptimo mas 'fuerte" es el máximo encontrado en relación a sus contrincantes próximos (recordemos que como la expresión es la inversa de la negativa de la hessiana, la hessiana en realidad esta dividiendo).
Si por ejemplo la matriz hessiana mostrase muy poca curvatura, el divisor seria relativamente pequeño y en consecuencia el lower bound para la varianza muy alto. Esto esencialmente nos dice que si el máximo encontrado no es mucho mejor que sus contrincantes próximos (o lo que es lo mismo, la hessiana muestra poca curvatura), el máximo encontrado exhibe una varianza alta ya que los contrincantes próximos se parecen mucho a él y en otra muestra podrían hasta ser mejores.

***<u>Propiedades de los Estimadores Maximum Likelihood</u>***

Las propiedades de los estimadores MLE para muestras pequeñas son en general, desconocidas. Es por eso que el método funciona mejor cuanto mas grande sea el tamaño de la muestra. Para muestras grandes, las propiedades asintoticas son las siguientes:

- Los estimadores MLE son ***CONSISTENTES,*** es decir, a medida que el tamaño de la muestra crece, los estimadores muestrales convergen al verdadero valor del parámetro poblacional;
- Son asintoticamente eficientes dado que alcanzan el límite mínimo de varianza definido por el teorema de Cramer-Rao:

$$\text{Varianza Asintotica}: \mathrm{V}(\hat{\theta}_{\mathrm{MLE}}) = -[(\frac{\partial \ln^2(L(\hat{\theta}))}{\partial \hat{\theta}^2})^{-1}] \quad (229)$$

La relevancia de esta propiedad es muy fuerte: en el límite, los estimadores MLE son los de menor varianza posible. No hay ningún otro que pueda tener menor varianza.

- Se distribuyen asintoticamente como una normal:

$$\hat{\theta} \to N(\theta, -[(\frac{\partial \ln^2(L(\hat{\theta}))}{\partial \hat{\theta}^2})^{-1}]) \quad (230)$$

Es muy importante entender por qué estamos poniendo tanto énfasis en la varianza de los estimadores MLE. La razón es muy sencilla: una vez estimados los parámetros, siempre es necesario testear hipótesis. Es imposible testear una hipótesis si no tenemos la matriz de varianza de los estimadores.

## ***Algoritmo de Matlab: Estimación Maximum Likelihood: Mixed Distributions***

```
Creando la Matriz de Datos Simulados
y1=normrnd(Mean1,Volatility1,Simulations,1);
y2=normrnd(Mean2,Volatility2,Simulations,1);
Defininendo la Mixed Distribution
S=binornd(1,lambda,Simulations,1)
y=y1.*S+y2.*(1-S);
Helping the Optimization
u=1.25;
v=0.75;
Res=zeros(1,6);
aq=min([Mean1*0.85;Mean1*1.25]);
while aq<=max([Mean1*0.85;Mean1*1.25]);
    bq=min([Volatility1*0.85;Volatility1*1.25]);
    while bq<=max([Volatility1*0.85;Volatility1*1.25]);
        cq=min([Mean2*0.85;Mean2*1.25]);
        while cq<=max([Mean2*0.85;Mean2*1.25]);
            dq=min([Volatility2*0.85;Volatility2*1.25]);
            while dq<=max([Volatility2*0.85;Volatility2*1.25]);
                eq=min([lambda*0.85;lambda*1.25]);
                while eq<=max([lambda*0.85;lambda*1.25]);
                    Definiendo la Funcion de Densidad para la Distribucion 1
                    aa=(eq.*((1./sqrt(2.*pi.*bq)).*exp(-0.5.*(((y-aq).^2)/(bq)))));
                    Definiendo la Funcion de Densidad para la Distribucion 2
                    bb=((1-eq).*((1./sqrt(2.*pi.*dq)).*exp(-0.5.*(((y-cq).^2)/(dq)))));
                    Definiendo la Log-Likelihood para la Mized Distribution
                    gos=-sum(log(aa+bb));
                    Res=[Res;[aq bq cq dq eq gos]];
                    eq=max([eq*u;eq*v]);
                end;
                dq=max([dq*u;dq*v]);
            end;
            cq=max([cq*u;cq*v]);
        end;
        bq=max([bq*u;bq*v]);
    end;
    aq=max([aq*u;aq*v]);
end;
Res=sortrows(Res(2:end,:),6);
Res=Res(2,1:5)'; Starting Values
Definiendo Valores Iniciales de Optimizacion
x0=Res;
Definiendo Restricciones Lineales
Aeq1=[];
Beq1=[];
Definiendo Restricciones No Lineales
Aeq2=[];
Beq2=[];
Definiendo Rangos Minimos y Maximos de Variacion para cada Parametro
lb=[-Inf;0.001;-Inf;0.001;0.001];
ub=[Inf;Inf;Inf;Inf;Inf];
Definiendo Opciones del Optimizador
options = optimset('LargeScale','off','TolX',PREC);
Realizando la Optimizacion
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyMixed',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y);
tval=zeros(size(x,1),1);
Matriz de Varianza/Covarianza
M=inv(HESSIAN);
Calculando t values
i=1;
while i<=size(x,1);
   tval(i,1)=x(i,1)/sqrt(M(i,i));
   i=i+1;
end;
hist([y1 y2],50)
Graficando el Histograma para Ambas Dsitirbuciones
axis([min(min(y1),min(y2)) max(max(y1),max(y2)) -Inf Inf])
title('MIXED DISTRIBUTION FROM A NORMAL');
```

***Definiendo MLE Mixed Distribution Function***

```
function [gos,x]=MyMixed(x,y);
```

***Definiendo la Funcion de Densidad para la Distribucion 1***

```
a=sqrt(2.*pi.*x(2,1));
aa=(x(5,1).*((1./a).*exp(-0.5.*(((y-x(1,1)).^2)/(x(2,1))))));
```

***Definiendo la Funcion de Densidad para la Distribucion 2***

```
b=sqrt(2.*pi.*x(4,1));
bb=((1-x(5,1)).*((1./b).*exp(-0.5.*(((y-x(3,1)).^2)/(x(4,1))))));
```

***Definiendo la Log-Likelihood para la Mized Distribution***

```
gos=-sum(log(aa+bb));
```

***Constrained Optimization***

***Testeo de Hipótesis en Maximum Likelihood, Likelihood Ratio Test***

La intuición detrás de un likelihood ratio test es muy sencilla. La idea es imponer una restricción en un (o unos) parámetro determinado. Una vez impuesta la restricción, maximizamos la "log-likelihood restringida". Obviamente, el valor de una función maximizada con restricciones es ***siempre*** menor al de la misma función maximizada sin restricciones. Cuanto mas fuerte sea la restricción, mayor será la diferencia entre la función maximizada restringida y la función maximizada sin restringir.

Para el caso específico de la estimación MLE, si la restricción es valida, imponerla no debería generar una reducción significativa en la log-likelihood function. Por lo tanto, el likelihood ratio test se basa en la diferencia entre la "unrestricted log-likelihood" y la "restricted log-likelihood". Definamos los siguientes conceptos:

$$L_U : \text{Unrestricted Likelihood} \Rightarrow \ln(L_U) : \text{Unrestricted Log - Likelihood} \tag{246}$$

$$L_R : \text{Restricted Likelihood} \Rightarrow \ln(L_R) : \text{Restricted Log - Likelihood}$$

$$L_R <= L_U \text{, porque un optimo restringido nunca es superior a uno sin restringir}$$

Definamos al likelihood ratio test de la siguiente forma:

$$\lambda = L_R / L_U \tag{247}$$

Esta función debe estar entre cero y uno. Si $\lambda$ es muy pequeño, esto implica que la restricción no es verdadera, es decir, no se condice con los datos de la muestra. La siguiente variable:

$$-2\ln(\lambda) = -2\ln(L_R / L_U) = -2[\ln(L_R) - \ln(L_U)] = 2[\ln(L_U) - \ln(L_R)] \tag{248}$$

Se distribuye como una chi-cuadrado, con grados de libertad igual a la cantidad de restricciones impuestas. Vamos entonces ahora, a testear la restricción de que:

$$\mu = 0.05 \tag{249}$$

Recordemos, que siempre simulamos los datos con un valor:

$$\mu = 0 \tag{250}$$

Entonces, debemos realizar las siguientes dos optimizaciones:

$$\text{Restringida} : Max_{\sigma} \sum_{i=1}^{n} \ln[e^{-[(y_i - 0.05)^2]/(2\sigma^2)} / \sigma\sqrt{2\pi}] = \ln(L_R) \tag{251}$$

Vemos aquí, que sólo maximizo con respecto a sigma, dado que la media se supone igual a 0.05. La segunda maximización, es la misma que la analizada anteriormente:

$$Max_{\mu,\sigma} \sum_{i=1}^{n} \ln[e^{-[(y_i - \mu)^2]/(2\sigma^2)} / \sigma\sqrt{2\pi}] = \ln(L_U) \tag{252}$$

Vemos aquí que maximizo con respecto a los dos parámetros, media y sigma. Entonces en este caso $2[\ln(L_U) - \ln(L_R)]$ se distribuye como una chi cuadrado con un grado de libertad.

***Algoritmo de Matlab: Constrained Optimization, Likelihood Ratio Test***

***Valores Iniciales***

```
V0=[SVM; SVS];
```

***Cuando estas matrices se definen como vacias, la restriccion no aplica***

```
A1=[];
B1=[];
A2=[];
B2=[];
```

***Restringiendo Simultáneamente Ambos Parámetros***

```
lb=[LBR; LBRS];
ub=[UBR; UBRS];

options = optimset('LargeScale','off','TolX',PREC);
[V,FVAL,OUTPUT,EXITFLAG,LAMBDA,GRAD,HESSIAN] =
fmincon('LF',V0,A1,B1,[],[],lb,ub,[],options,y);
RestrictedLogLikelihood=-FVAL;
RPar=V;
```

***Unrestricted Log-Likelihood***

```
PREC=0.0001;
```

***Valores Iniciales***

```
V0=[SVM; SVS];

A1=[];
B1=[];
A2=[];
B2=[];
```

***Parámetros No Restringidos***

```
lb=[-Inf; 0.001];
ub=[Inf; Inf];

options = optimset('LargeScale','off','TolX',PREC);
[V,FVAL,OUTPUT,EXITFLAG,LAMBDA,GRAD,HESSIAN] =
fmincon('LF',V0,A1,B1,[],[],lb,ub,[],options,y);

UnrestrictedLogLikelihood=-FVAL;
UPar=V;
```

***Likelihood Ratio Test***

***Definiendo Valor del Estadístico***

```
Lambda=2*(UnrestrictedLogLikelihood-RestrictedLogLikelihood);
```

***Buscando el Critical Value***

```
ChiSquared=chi2inv(CL,DF);
```

***Algoritmo de Matlab: Log-Likelihood Function***

```
function [gos,V]=LF(V,y);
cc=(-((y-V(1,1)).*(y-V(1,1))./(2*V(2,1))-log(sqrt(2*pi*V(2,1))));
gos=-sum(cc);
```

***Primer Violación del Supuesto de Random Sampling: Truncation***

Existen ocasiones en las que la muestra que se utiliza para estimar el modelo está truncada. Por ejemplo, supongamos que se decide analizar el comportamiento de aquéllas personas que tengan ingresos superiores a un determinada nivel (llamemos a este nivel $a$). Cómo puede ocurrir esto?. Supongamos por ejemplo que se realiza un censo para recoger la información y el censo sólo se realiza en vecindarios de altos niveles de ingresos, obviamente, por definición de la metodología de recopilación de la información se viola el supuesto de muestreo aleatorio ya que solamente se consideran ingresos superiores a un determinado nivel. Obviamente, la violación de este supuesto genera sesgo en la estimación ya que la estimación se basará solamente en una proporción de la población. Recordemos que en términos generales, los coeficientes que se estiman reflejan el promedio de la muestra. Si la muestra está truncada, el promedio sólo se basará en una proporción de la muestra y en consecuencia ocurrirá lo que en la práctica se denomina como ***sample selection bias***, el cual es mucho mas común de lo que sería deseable en econometría.

Supongamos que la variable aleatoria $x$ está truncada en el nivel $a$, es decir, solamente se escogen aquellos valores de $x$ tal que: $x \geq a$. Se puede demostrar que la esperanza matemática de la variable aleatoria $x$ es:

$$E(x / x \geq a) = E(x) + \sigma_x \lambda; \quad \lambda = \frac{f(x)}{1 - F(a)} \tag{329}$$

En donde, $E(x)$ es la media ***INCONDICIONAL*** de la variable $x$, $\sigma_x$ es el desvío standard ***INCONDICIONAL*** y $\lambda$ es lo que normalmente se denomina como ***Inverse Mills Ratio***. Vamos a analizar profundamente las implicancias de la expresión precedente. En primer lugar, el término $\sigma_x \lambda$ es el ***SESGO*** en la media de $x$ si se intentase estimar la media incondicional de $x$ a través de la muestra truncada de $x$. Vemos que el sesgo tiene dos componentes.

Primero, el sesgo es función creciente de $\sigma_x$ porque cuanto mayor volatilidad exista en la población subyacente, mayor será la heterogeneidad de la misma. En estas circunstancias, el sesgo de truncar una muestra será mayor cuanto mas heterogénea sea la población subyacente. La razón es sencilla: si la estimación se basa en una proporción de una población que es muy heterogénea, la información que se utiliza en la estimación es muy poco representativa de la muestra total y de esta forma el sesgo será mas acentuado.

Segundo, recordemos que $\lambda = f(x)/(1 - F(a))$. Es importante entender bien qué significa esta expresión. Lo mas importante de la expresión es el denominador: $1 - F(a)$. Recordemos que $F(a)$ mide la probabilidad de que $x$ sea menor que $a$. Con lo cual, $1 - F(a)$ es una medida de la proporción de la población representada por la muestra truncada. Obviamente, cuanto mayor sea $a$ menor será la proporción de la población total representada en la muestra y en consecuencia mayor será el sesgo en la estimación de la media incondicional a partir de la muestra sesgada.

***Conclusión:*** ***Cuanto mayor sea el grado de heterogeneidad de la población subyacente (representada por $\sigma_x$) y mayor sea el nivel de truncation de la muestra (representado por $a$) mayor será el sesgo en la estimación de $E(x)$ si la misma se basa en la muestra truncada.***

Finalmente, es muy importante tener en cuenta cómo se define una función de densidad ***CONDICIONAL***. Supongamos que queremos determinar la probabilidad de ocurrencia de $x$ dado que $x \geq a$. En este caso, la función de densidad condicional es:

$$f(x / x \geq a) = \frac{f(x)}{1 - F(a)} \tag{330}$$

Esta expresión tiene mucho sentido. Primero, es importante destacar que $f(x)$ en este caso está definida ***SOLAMENTE*** para aquéllos valores de $x$ mayores o iguales que $a$. Dado que solamente observaremos $x's$ tales que $x \geq a$, para que esta nueva función de densidad sume uno, es necesario cambiar la escala de la función mediante el factor $1-F(a)$ que casualmente mide la probabilidad de que $x \geq a$. ***La expresión precedente es la función de densidad de una variable truncada***. Hecha esta aclaración, y si suponemos que la variable $x$ es continua resulta entonces:

$$\int_{x=a}^{\infty} \frac{f(x)}{1-F(a)} = 1 \tag{331}$$

El análisis anterior se puede obviamente extender al modelo regresional $\mathbf{Y} = \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\varepsilon}$. En este caso, el sesgo está dado porque sólo se seleccionan valores de la variable dependiente tales que $y_i \geq a$. Dado que ya hicimos un análisis intuitivo del serio problema de sesgo originado por truncar una muestra y tratar de estimar los parámetros de ***TODA*** la población a partir de la misma, ahora vamos a analizar cómo se puede corregir un modelo econométrico para que a pesar de utilizar una muestra truncada (o sea non-random) se puedan estimar sin sesgo los parámetros poblacionales $\boldsymbol{\beta}$. Como siempre, el error poblacional se puede expresar de la siguiente forma:

$$\varepsilon_i = y_i - \mathbf{x}_i\boldsymbol{\beta} \tag{332}$$

Si suponemos que $\varepsilon_i \sim N(0, \sigma^2)$, la función de densidad de $\varepsilon_i$ dado que $y_i \geq a$ es:

$$f(\varepsilon_i / y_i \geq a) = \frac{f(y_i - \mathbf{x}_i\boldsymbol{\beta})}{1-F(a-\mathbf{x}_i\boldsymbol{\beta})} = \frac{(1/\sigma\sqrt{2\pi})\exp[(y_i-\mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}{[1-\int_{-\infty}^{a-\mathbf{x}_i\boldsymbol{\beta}}(1/\sigma\sqrt{2\pi})\exp[(y_i-\mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]} \tag{333}$$

El paso que falta es calcular el logaritmo de la expresión precedente:

$$\ln(f(\varepsilon_i / y_i \geq a)) = \ln\left(\frac{(1/\sigma\sqrt{2\pi})\exp[(y_i-\mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}{[1-\int_{-\infty}^{a-\mathbf{x}_i\boldsymbol{\beta}}(1/\sigma\sqrt{2\pi})\exp[(y_i-\mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}\right) \tag{334}$$

Obviamente, la log-likelihood resulta en la sumatoria de la expresión precedente:

$$\log-\text{likelihood} : \sum_{i=1}^{n} \ln(f(\varepsilon_i / y_i \geq a)) \tag{335}$$

De esta forma, la estimación de la truncated regression se realiza mediante la optimización de la expresión precedente respecto a $\boldsymbol{\beta}$ y a $\sigma$ de la misma forma que para cualquiera de los otros modelos de MLE que analizamos previamente:

$$Max_{\boldsymbol{\beta},\sigma} : \sum_{i=1}^{n} \ln(f(\varepsilon_i / y_i \geq a)) \tag{336}$$

Para finalizar, es muy importante entender perfectamente qué significa $\boldsymbol{\beta}$ en este contexto de truncated regressions. El vector $\boldsymbol{\beta}$ es el mismo de siempre, es decir, corresponde a toda la población, este es el vector que debemos estimar en forma insesgada. El hecho de estimar el modelo a traves de una función de densidad condicional ***corrige el sesgo inherente a utilizar una muestra que no es aleatoria***.

Con lo cual, si estimamos el modelo de acuerdo a la expresión precedente, el vector que obtengamos de la optimización, $\mathbf{b}^{MLE}$, es un estimador insesgado del verdadero vector poblacional $\boldsymbol{\beta}$. Con lo cual, si la muestra es lo suficientemente grande, $\mathbf{b}^{MLE}$ convergerá a $\boldsymbol{\beta}$. Si por el contrario, hubiéramos ignorado el hecho de que la muestra está truncada aplicando por ejemplo standard OLS, el vector que hubiéramos obtenido de la estimación, $\mathbf{b}^{OLS}$, hubiera estimado sesgadamente al verdadero vector $\boldsymbol{\beta}$. Con lo cual, aun con una muestra grande, $\mathbf{b}^{OLS}$ nunca aproximaría al verdadero vector $\boldsymbol{\beta}$. Lo mismo se aplica para el otro parámetro poblacional estimado: $\sigma$.

***Conclusión:*** ***Como vemos, nuestro objetivo fue estimar los parámetros de una población que NUNCA estuvo representada en la muestra (porque la muestra está truncada) a partir precisamente de datos truncados. Lo interesante de este modelo es que aun VIOLANDO un supuesto básico (random sampling), la corrección realizada al modelo standard de OLS permite CONOCER la verdadera población a la cual nunca tuvimos acceso muestral.***

Finalmente, es muy importante resaltar que a la log-likelihood de este ***modelo le cuesta mucho converger***. La razón radica en la definición misma de la función de densidad truncada la cual es el ratio entre la función de densidad no truncada y la correspondiente función de distribución. Recordemos que la función de distribución ***es una integral*** que es función a su vez de una de las variables de optimización, $\sigma$, de ahí que el modelo resulte en una especificación muy difícil de optimizar incluso para una computadora.

De esta forma, tal como lo analizamos en casos anteriores, es en general necesario utilizar el procedimiento de grilla con respecto a $\sigma$. Lo que hacemos entonces es construir una grilla que para cada valor de $\sigma$ optimiza respecto a $\boldsymbol{\beta}$. Una vez terminado con la grilla, se escoge aquella optimización que resulta en la mayor log-likelihood la cual tendrá su correspondiente optimización para $\hat{\sigma}$ y $\mathbf{b}^{MLE}$. Estos valores a su vez pueden ahora ser utilizados como los starting values de una nueva y definitiva optimización (recordemos que el procedimiento de grilla es una gran ayuda para MLE cuando la log-likelihood es complicada o no tiene una forma bien definida). Esta optimización final nos generará la estimación correspondiente junto con la information matrix.

## ***Código de Estimacion en Presencia de Truncation***

```
xx=[ones(Size,1) lognrnd(RegressorsMean,RegressorsSigma,Size,3)
normrnd(MeanError,SigmaError,Size,1)];
y=Beta1*xx(:,1)+Beta2*xx(:,2)+Beta3*xx(:,3)+Beta4*xx(:,4)+xx(:,5);
yReference=y;
xReference=xx(:,1:4);

Helping Estimate Sigma with a Grid
Res=zeros(1,6);
Sigma=SigmaError-0.02;
t=1;
while Sigma<=SigmaError+0.02;
    x0=[0.10;0.20;0.30;0.40];
    lb=[-Inf;-Inf;-Inf;-Inf];
    ub=[Inf;Inf;Inf;Inf];
    Aeq1=[];
    Beq1=[];
    Aeq2=[];
    Beq2=[];
    PREC=0.0001;
    options =
    optimset('LargeScale','off','Display','iter', 'MaxFunEvals',300, 'TolX',PREC);
    [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
    fmincon('MyOLSMLEBis',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma);
    Res=[Res;[x(1,1) x(2,1) x(3,1) x(4,1) Sigma FVAL]];
    t=t+1;
    Sigma=Sigma+0.001;
end;
Res=Res(2:end,:);
FinalRes=sortrows(Res,[size(Res,2)]);
Final=FinalRes(1,:);

Final "Helped" Optimization
x0=[Final(:,1:size(FinalRes,2)-1)'];
lb=[-Inf;-Inf;-Inf;-Inf;0.0001];
ub=[Inf;Inf;Inf;Inf;Inf];
Aeq1=[];
Beq1=[];
Aeq2=[];
Beq2=[];
PREC=0.0001;
options = optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisHelped',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx);
Estimate=x;
VarCovar=inv(size(y,1)*HESSIAN);

Initiating Truncation Process
A=[y xx];
B=sortrows(A,[1]);
C=B(TruncationLevel:end,:);
y=C(:,1); Truncated Dependent Variable
xx=C(:,2:size(C,2));
Trunc=y(1,1);

Helping Estimate Sigma IGNORING Truncation
ResTruncated=zeros(1,6);
Sigma=SigmaError-0.02;
t=1;
while Sigma<=SigmaError+0.02;
    x0=[0.10;0.10;0.10;0.10];
    lb=[-Inf;-Inf;-Inf;-Inf];
    ub=[Inf;Inf;Inf;Inf];
    Aeq1=[];
    Beq1=[];
    Aeq2=[];
    Beq2=[];
    PREC=0.0001;
    options =
    optimset('LargeScale','off','Display','iter', 'MaxFunEvals',300, 'TolX',PREC);
    [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
```

```
    fmincon('MyOLSMLEBis',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma);
    ResTruncated=[ResTruncated;[x(1,1) x(2,1) x(3,1) x(4,1) Sigma FVAL]];
    t=t+1;
    Sigma=Sigma+0.001;
end;
ResTruncated=ResTruncated(2:end,:);
FinalResTruncated=sortrows(ResTruncated,[size(ResTruncated,2)]);
FinalTruncated=FinalResTruncated(1,:);

Final "Helped" Optimization IGNORING Truncation
x0=[FinalTruncated(:,1:size(FinalResTruncated,2)-1)'];
lb=[-Inf;-Inf;-Inf;-Inf;-Inf];
ub=[Inf;Inf;Inf;Inf;Inf];
Aeq1=[];
Beq1=[];
Aeq2=[];
Beq2=[];
PREC=0.0001;
options = optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisHelped',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx);
EstimateTruncated=x;
VarCovarTruncated=inv(size(y,1)*HESSIAN);

Correcting the Truncation Anomaly
Helping Estimate Sigma with a Grid-Truncation Correction
ResTruncationCorrected=zeros(1,6);
Sigma=0.05; This model is EXTREMELY sensitive to the starting levels of
SIGMA!!!!!!!!!!!!!!!!!!
while Sigma<=0.15;
    x0=[0.10;0.10;0.10;0.10];
    lb=[0.001;0.001;0.001;0.001];
    ub=[Inf;Inf;Inf;Inf];
    Aeq1=[];
    Beq1=[];
    Aeq2=[];
    Beq2=[];
    PREC=0.0001;
    options =
    optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
    [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisTruncated',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma,Trunc);
    EstimationCorrectedTruncation=x;
    VarCovarCorrectedTruncation=inv(size(y,1)*HESSIAN);
    if FVAL==-Inf;
        FVAL=1000000;
    end;
    ResTruncationCorrected=[ResTruncationCorrected;[x' Sigma FVAL]];
    Sigma=Sigma+0.001;
end;
ResTruncationCorrected=ResTruncationCorrected(2:end,:);
ResTruncationCorrected=sortrows(ResTruncationCorrected,[size(ResTruncationCorrected,2)]);
ResTruncationCorrectedFin=ResTruncationCorrected(1,1:size(ResTruncationCorrected,2)-1)';
Sigma=ResTruncationCorrectedFin(end,1);
x0=[ResTruncationCorrectedFin(1:end,:)];
lb=[0.001;0.001;0.001;0.001;0.001];
ub=[Inf;Inf;Inf;Inf;Inf];
Aeq1=[];
Beq1=[];
Aeq2=[];
Beq2=[];
PREC=0.0001;
options = optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisTruncatedBis',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma,Trun
c);
EstimationCorrectedTruncation=x;
VarCovarCorrectedTruncation=inv(size(y,1)*HESSIAN);
EstimateTruncatedCorrected=[x];
FinalResults=[Estimate EstimateTruncated EstimateTruncatedCorrected]
```

***Definicion de la Funcion Objetivo***

```
function [gos,x]=MyOLSMLEBisTruncated(x,y,xx,Sigma,Trunc);

ErrorT=y-(x(1,1)*xx(:,1)+x(2,1)*xx(:,2)+x(3,1)*xx(:,3)+x(4,1)*xx(:,4));
ErrorTrunc=Trunc-(x(1,1)*xx(:,1)+x(2,1)*xx(:,2)+x(3,1)*xx(:,3)+x(4,1)*xx(:,4));
Density=((1/sqrt(2*pi*(Sigma^2)))*[exp(-0.5.*((ErrorT).*(ErrorT)/((Sigma^2))))]);

Density=NORMPDF(ErrorT,0,Sigma);
Cumulative=1-NORMCDF(ErrorTrunc,0,Sigma); SIGMA is EXOGENOUS
f=Density./Cumulative;
gos=-sum(log(f))/size(y,1);


function [gos,x]=MyOLSMLEBisTruncatedBis(x,y,xx,Sigma,Trunc);

ErrorT=y-(x(1,1)*xx(:,1)+x(2,1)*xx(:,2)+x(3,1)*xx(:,3)+x(4,1)*xx(:,4));
ErrorTrunc=Trunc-(x(1,1)*xx(:,1)+x(2,1)*xx(:,2)+x(3,1)*xx(:,3)+x(4,1)*xx(:,4));
Density=((1/sqrt(2*pi*(x(5,1)^2)))*[exp(-0.5.*((ErrorT).*(ErrorT)/((x(5,1)^2))))]);
Cumulative=1-NORMCDF(ErrorTrunc,0,x(5,1)); SIGMA is ENDOGENOUS
f=Density./Cumulative;
gos=-sum(log(f))/size(y,1);
```

***Segunda Violación del Supuesto de Random Sampling: Censoring***

Una muestra tiene el problema de censoring cuando por alguna razón el comportamiento implícito en el data no puede reflejarse como en realidad hubiera querido ser. Qué quiere decir esto?. Supongamos que estamos analizando la cantidad de vacaciones anuales que le gustaría tomarse a empleados en relación de dependencia como función de su ingreso por ejemplo. En general, es imposible poder tomarse mas de un mes por año, con lo cual, para todos aquellos empleados que se tomaron un mes entero de vacaciones el econometrista no sabe en realidad si esa fue la cantidad que hubieran querido tomarse si el límite de 30 días no hubiera existido. Es aquí en donde la información está ***CENSURADA*** dado que el comportamiento del individuo no puede revelarse como hubiera querido si no hubiera existido el límite de 30 días máximo de vacaciones anuales. Como vemos, este problema es una forma de violar el supuesto de random sampling en el sentido que la variable dependiente no puede tomar su verdadero valor y por el contrario está acotada por un límite máximo.

El ***problema de censoring es totalmente diferente al de truncation***. Recordemos que truncation ocurre cuando la muestra se refiere solamente a una proporción de la población. Pero el data truncado se releva de la forma que desea el individuo, es decir no existe ningún límite máximo o mínimo que impida conocer la verdaderas decisiones contenidas en la muestra. Es decir, en presencia de truncation la información está limitada porque la población no está totalmente representada en la muestra, pero las respuestas contenidas en la muestra no están limitadas en absoluto.

Por el contrario, en presencia de censoring, la población está totalmente representada a través de la muestra, es decir, en este sentido existe pleno random sampling. Sin embargo, el comportamiento de algunos (o todos) los individuos contenidos en la muestra no puede reflejarse en la forma en la que hubieran deseado de no haber existido el límite máximo de vacaciones anuales. Es decir, en presencia de censoring, la información está limitada porque algunas o todas las respuestas no pueden reflejarse libremente como lo hubieran hecho si el límite no hubiera existido, pero la población está totalmente representada en la muestra.

La pregunta entonces es, cómo podemos hacer para modelar un proceso estocástico que exhibe censoring. Cuando el data exhibe censoring, se lo puede dividir en dos grandes grupos. El primer grupo está compuesto por todas aquellas respuestas que ***NO*** superan el límite (en nuestro ejemplo, los treinta días de vacaciones). Dado que las respuestas no superaron el límite, las decisiones implícitas en las mismas fueron hechas con absoluta libertad en el sentido que el límite no las censuró. Por ejemplo, si dado el límite de 30 días, un empleado decidió tomarse 27 días, su comportamiento no fue distorsionado por el límite y se revela con absoluta exactitud y libertad. El segundo grupo está compuesto por todas aquellas respuestas que ***SON IGUALES*** al límite. Dado que en este caso las respuestas son iguales al límite, no se sabe si esto fue así porque los individuos decidieron libremente tomarse sólo treinta días o porque no pudieron tomarse mas días dado el límite máximo que censuró su comportamiento. Dado estos dos grupos, el modelo de censor regression es:

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i \tag{337}$$

Para el primer grupo, dado que no ofrece censoring, su tratamiento debe ser idéntico que el correspondiente al caso de standard OLS el cual utiliza la función de densidad del error:

$$\text{Prob}(y_i = y_i / y_i < \bar{y}) = f(\varepsilon_i) = f(y_i - \mathbf{x}_i\boldsymbol{\beta}) = (1/\sigma\sqrt{2\pi})\exp[(y_i - \mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2) \tag{338}$$

Sin embargo, para el segundo grupo la expresión se complica un poco mas:

$$\text{Prob}(y_i = \bar{y}) = \text{Prob}(y_i \geq \bar{y}) = \text{Prob}(x_i\beta + \varepsilon_i \geq \bar{y}) = \text{Prob}(\varepsilon_i \geq \bar{y} - x_i\beta) = 1 - F(\varepsilon_i) \tag{339}$$

Recordemos que $F(\varepsilon_i)$ es la función de distribución de la normal:

$$F(\varepsilon_i) = \int_{-\infty}^{\bar{y}-\mathbf{x}_i\boldsymbol{\beta}} (1/\sigma\sqrt{2\pi}) \exp[(y_i - \mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2) \tag{340}$$

Vemos que para el segundo grupo, la probabilidad de ocurrencia se parece a un Probit. De esta forma vemos que para cada uno de estos dos grupos la probabilidad de ocurrencia está representada en forma diferente. Esta es la primera vez que armamos una log-likelihood con densidades diferentes para distintos integrantes de la muestra. De esta forma, la log-likelihood es:

$$\log - \text{likelihood} = \sum_{y_i \le \bar{y}} \ln[f(y_i - \mathbf{x}_i\boldsymbol{\beta})] + \sum_{y_i = \bar{y}} \ln[1 - F(y_i - \mathbf{x}_i\boldsymbol{\beta}] \tag{341}$$

Esto a su vez es:

$$\ln L = \sum_{y_i \le \bar{y}} \ln[\frac{\exp[(y_i - \mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}{\sigma\sqrt{2\pi}}] + \sum_{y_i = \bar{y}} \ln[1 - \int_{-\infty}^{\bar{y}-\mathbf{x}_i\boldsymbol{\beta}} \frac{\exp[(y_i - \mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}{\sigma\sqrt{2\pi}}] \tag{342}$$

Vemos precisamente que para aquellas repuestas que no ofrecen censoring ( $y_i \le \bar{y}$ ) quedan incluidas en el primer término de la sumatoria, mientras que las que ofrecen censoring ( $y_i = \bar{y}$ ) se incluyen en el segundo término. Lo que ahora hay que hacer es maximizar la log-likelihood con respecto a $\boldsymbol{\beta}$ y a $\sigma$ :

$$Max_{\boldsymbol{\beta},\sigma} : \sum_{y_i \le \bar{y}} \ln[\frac{\exp[(y_i - \mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}{\sigma\sqrt{2\pi}}] + \sum_{y_i = \bar{y}} \ln[1 - \int_{-\infty}^{\bar{y}-\mathbf{x}_i\boldsymbol{\beta}} \frac{\exp[(y_i - \mathbf{x}_i\boldsymbol{\beta})^2/(2\sigma^2)]}{\sigma\sqrt{2\pi}}] \tag{343}$$

Como ya nos podemos imaginar, a este modelo también le cuesta converger por lo que aplicamos el mismo procedimiento de grilla para el tratamiento de $\sigma$ como lo explicamos para el caso de truncation. Finalmente, a este modelo se lo conoce con el nombre de Tobit.

## ***Código de Estimacion en Presencia de Censoring***

```
xx=[ones(Size,1) lognrnd(RegressorsMean,RegressorsSigma,Size,3)
normrnd(MeanError,SigmaError,Size,1)];
y=Beta1*xx(:,1)+Beta2*xx(:,2)+Beta3*xx(:,3)+Beta4*xx(:,4)+xx(:,5);
yReference=y;
xReference=xx(:,1:4);
```

***Helping Estimate Sigma with a Grid***

```
Res=zeros(1,6);
Sigma=SigmaError-0.02;
t=1;
while Sigma<=SigmaError+0.02;
    x0=[0.10;0.20;0.30;0.40];
    lb=[-Inf;-Inf;-Inf;-Inf];
    ub=[Inf;Inf;Inf;Inf];
    Aeq1=[];
    Beq1=[];
    Aeq2=[];
    Beq2=[];
    PREC=0.0001;
    options =
    optimset('LargeScale','off','Display','iter', 'MaxFunEvals',300, 'TolX',PREC);
    [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
    fmincon('MyOLSMLEBis',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma);
    Res=[Res;[x(1,1) x(2,1) x(3,1) x(4,1) Sigma FVAL]];
    t=t+1;
    Sigma=Sigma+0.001;
end;
Res=Res(2:end,:);
FinalRes=sortrows(Res,[size(Res,2)]);
Final=FinalRes(1,:);
```

***Final "Helped" Optimization***

```
x0=[Final(:,1:size(FinalRes,2)-1)'];
lb=[-Inf;-Inf;-Inf;-Inf;0.0001];
ub=[Inf;Inf;Inf;Inf;Inf];
Aeq1=[];
Beq1=[];
Aeq2=[];
Beq2=[];
PREC=0.0001;
options = optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisHelped',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx);
Estimate=x;
VarCovar=inv(size(y,1)*HESSIAN);
```

***Initiating Censoring***

```
A=[y xx];
B=sortrows(A,[1]);
B=[B (1:size(y,1))'];
What=find(B(:,size(B,2))==CensoringLevel);
Censoring=B(What,1);
Indicator=zeros(size(y,1),1);
i=1,
while i<=size(y,1);
    if B(i,1)>=Censoring;
        B(i,1)=Censoring;
        Indicator(i,1)=1;
    end;
    i=i+1;
end;
y=B(:,1); Final Censored Dependent Variable
xx=B(:,2:size(B,2)-1);
```

***Helping Estimate Sigma with a Grid IGNORING Censoring***

```
ResTruncated=zeros(1,6);
Sigma=SigmaError-0.02;
t=1;
while Sigma<=SigmaError+0.02;
    x0=[0.10;0.10;0.10;0.10];
```

```
    lb=[-Inf;-Inf;-Inf;-Inf];
    ub=[Inf;Inf;Inf;Inf];
    Aeq1=[];
    Beq1=[];
    Aeq2=[];
    Beq2=[];
    PREC=0.0001;
    options =
    optimset('LargeScale','off','Display','iter', 'MaxFunEvals',300, 'TolX',PREC);
    [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
    fmincon('MyOLSMLEBis',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma);
    ResTruncated=[ResTruncated;[x(1,1) x(2,1) x(3,1) x(4,1) Sigma FVAL]];
    t=t+1;
    Sigma=Sigma+0.001;
end;
ResTruncated=ResTruncated(2:end,:);
FinalResTruncated=sortrows(ResTruncated,[size(ResTruncated,2)]);
FinalTruncated=FinalResTruncated(1,:);

Final "Helped" Optimization IGNORING Censoring
x0=[FinalTruncated(:,1:size(FinalResTruncated,2)-1)'];
lb=[-Inf;-Inf;-Inf;-Inf;-Inf];
ub=[Inf;Inf;Inf;Inf;Inf];
Aeq1=[];
Beq1=[];
Aeq2=[];
Beq2=[];
PREC=0.0001;
options = optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisHelped',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx);
EstimateTruncated=x;
VarCovarTruncated=inv(size(y,1)*HESSIAN);

Correcting the Censoring Anomaly
Helping Estimate Sigma with a Grid-Truncation Correction
ResTruncationCorrected=zeros(1,6);
Sigma=0.05; This model is EXTREMELY sensitive to the starting levels of
SIGMA!!!!!!!!!!!!!!!!!!!
while Sigma<=0.15;
    x0=[0.10;0.10;0.10;0.10];
    lb=[0.001;0.001;0.001;0.001];
    ub=[Inf;Inf;Inf;Inf];
    Aeq1=[];
    Beq1=[];
    Aeq2=[];
    Beq2=[];
    PREC=0.0001;
    options =
    optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
    [x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisCensored',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma,Indicato
r);
    EstimationCorrectedTruncation=x;
    VarCovarCorrectedTruncation=inv(size(y,1)*HESSIAN);
    if FVAL==-Inf;
        FVAL=1000000;
    end;
    ResTruncationCorrected=[ResTruncationCorrected;[x' Sigma FVAL]];
    Sigma=Sigma+0.001;
end;
ResTruncationCorrected=ResTruncationCorrected(2:end,:);
ResTruncationCorrected=sortrows(ResTruncationCorrected,[size(ResTruncationCorrected,2)]);
ResTruncationCorrectedFin=ResTruncationCorrected(1,1:size(ResTruncationCorrected,2)-1)';

Sigma=ResTruncationCorrectedFin(end,1);
x0=[ResTruncationCorrectedFin(1:end,:)];
lb=[0.001;0.001;0.001;0.001;0.001];
ub=[Inf;Inf;Inf;Inf;Inf];
Aeq1=[];
Beq1=[];
```

```
Aeq2=[];
Beq2=[];
PREC=0.0001;
options = optimset('LargeScale','off','Display','iter', 'MaxFunEvals',100, 'TolX',PREC);
[x,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN] =
fmincon('MyOLSMLEBisCensoredBis',x0,Aeq1,Beq1,Aeq2,Beq2,lb,ub,[],options,y,xx,Sigma,Indic
ator);
EstimationCorrectedTruncation=x;
VarCovarCorrectedTruncation=inv(size(y,1)*HESSIAN);
EstimateTruncatedCorrected=[x];

FinalResults=[Estimate EstimateTruncated EstimateTruncatedCorrected]
```

***Definicion de la Funcion Objetivo***

```
function [gos,x]=MyOLSMLEBisCensored(x,y,xx,Sigma,Indicator);

Estim=zeros(size(y,1),1);

ErrorT=y-(x(1,1)*xx(:,1)+x(2,1)*xx(:,2)+x(3,1)*xx(:,3)+x(4,1)*xx(:,4));

t=1;
while t<=size(y,1);
    if Indicator(t,1)==0;
        Estim(t,1)=((1/sqrt(2*pi*(Sigma^2)))*[exp(0.5.*((ErrorT(t,1)).^2/((Sigma^2))))]);
    elseif Indicator(t,1)==1;
        Estim(t,1)=1-NORMCDF(ErrorT(t,1),0,Sigma);
    end;
    t=t+1;
end;

gos=-sum(log(Estim))/size(y,1);



function [gos,x]=MyOLSMLEBisCensoredBis(x,y,xx,Sigma,Indicator);

Estim=zeros(size(y,1),1);

ErrorT=y-(x(1,1)*xx(:,1)+x(2,1)*xx(:,2)+x(3,1)*xx(:,3)+x(4,1)*xx(:,4));

t=1;
while t<=size(y,1);
    if Indicator(t,1)==0;
      Estim(t,1)=((1/sqrt(2*pi*(x(5,1)^2)))*[exp(0.5.*((ErrorT(t,1)).^2/((x(5,1)^2))))]);
    elseif Indicator(t,1)==1;
      Estim(t,1)=1-NORMCDF(ErrorT(t,1),0,x(5,1));
    end;
    t=t+1;
end;

gos=-sum(log(Estim))/size(y,1);
```