Ejercicios Inferencia II

Jose M. Camacho[1]

[1]Universidad de Granada

April 19, 2020

Ejercicio 1:

Generemos una muestra aletoria simple ($m.a.s.$) de tamaño $n = 50$ de una población $X \sim N\left(\mu\ ,\ \sigma^2\right)$, con $\mu = 2$ y $\sigma^2 = 20$.

A partir de la m.a.s. así generada podemos estimar la media de la población, $\hat{\mu}_n \equiv \bar{X}_n$:

$$\bar{X}_n : \mathbb{R}^n \to \Theta = \mathbb{R} \tag{1}$$

Si generamos más muestras podemos ver $\overline{X}_n$ como una varuable aleatoria tal que $\overline{X}_n \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$. Ahora bien, a partir de $\overline{X}_n$ podemos generar otro estadístico T definido como:

$$T : \bar{X}_n \in \mathbb{R} \to \Theta \tag{2}$$

$$T(X_1, \cdots, X_n) = \begin{cases} 1 & si \quad \bar{X}_n \leq 0 \\ 0 & si \quad \bar{X}_n > 0 \end{cases}$$

A partir de la definición podemos ver que T es una variable binaria, por lo que seguirá una distribución de Bernoulli: $T \sim B\left(p\right)$, donde p es la probabilidad de que ocurra el suceso "éxito", $T = 1$ en este caso. Puesto que la distribución $B\left(p\right)$ es bien conocida conocida sabemos que $E\left[T\right] = p$.

Ahora bien, para el cáculo de p transformamos $\overline{X}_n \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right) \to Z \sim N\left(0, 1\right)$, mediante la transformación lineal:

$$Z = \frac{\bar{X}_n - \mu}{\sigma} \cdot \sqrt{n}$$

Con esta transformación la condición del suceso "éxito" de la variable aleatoria T, $\overline{X} \leq 0$ es equivalente a que:

$$Z = \frac{0 - \mu \cdot \sqrt{n}}{\sigma} = -\frac{\sqrt{2}}{2}$$

Calculemos ahora cuál es la probabilidad de que $Z < -\frac{\sqrt{2}}{2}$, para ello usamos la distribución de probilidad acumulada de la gaussiana $N(0,1)$:

$$\mathbb{F}(x) = \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$$

evaluamos en $\mu = 0$, $\sigma = 1,\ x = \frac{-\sqrt{2}}{2}$:

$$\mathbb{F}\left(x = \frac{-\sqrt{2}}{2}\right) = \frac{1}{2}\left[1 + erf(-0.5)\right] = \frac{1}{2}\left[1 - 0.5205\right] = 0.23975$$

a) Comprobar que $E[T]$ es un estimador de $F_Z\ \left(\frac{-\mu\cdot\sqrt{n}}{\sigma}\right)$. Siendo F_Z una función paramétrica de parámetro $\theta = \frac{-\mu\cdot\sqrt{n}}{\sigma}$, que nos proporciona la probabilidad de que $Z \leq \frac{-\mu\cdot\sqrt{n}}{\sigma}$.

Para que $E[T]$, con $T\ \sim B(p)$, esté bien definido como estimador de F_Z debe ocurrir que el espacio paramétrico, Θ, sobre el que aplica la función $E[T]$ coincida con el domino de F_Z:

$$F_Z\left(\frac{-\mu\cdot\sqrt{n}}{\sigma}\right) : \mathbb{R} \to \mathbb{R}$$

$$E[T] : T \in \{0,1\} \to \Theta = \mathbb{R}$$

b) Comprueba que T es estimador insesgado de F_Z usando la ***Ley Fuerte de los Grande Números.***

Condición de insesgadez: $\hat{E}_\theta\ \left[\ T\ \left(\overline{X}_n\right)\right]\ =\ F_Z\ (\theta)$, con $\theta = \frac{-\mu\sqrt{n}}{\sigma}$

Ley Fuerte de los Grandes Números:

Dada un sucesión infinita, $X_1,\ X_2,\cdots$, de variables aleatorias independientes e indenticamente distribuidas, con $E\ [\ |\ X_i\ |\ ]\ <\ \infty$, y valor esperado $p \Rightarrow P\ \left(lim_{n\to\infty}\bar{X}_n = p\right) = 1$.

Código:

```
clear all

n=50
m=100

mu=2
sigma=20

flag=-mu*sqrt(n)/sigma

for i=1 : m

    Z=normrnd(0,1)
    if(Z<flag)
        T(i)=1
    else
        T(i)=0
```

```
    end

end

mean(T)
```

Resultado:

$$\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\ = 0.29 \equiv \hat{p}$$

El valor real de la función parámetrica se ha calcuado antes

$$F_Z = \mathbb{F}_Z\left(x = -\frac{\mu\sqrt{n}}{\sigma}\right) = 0.23975$$

De donde podemos calcular el sesgo

$$\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right] = F_Z + b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = 0.05025$$

donde $b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = 0$ es el sesgo del estimador $\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]$

Según la ley de los grandes números si aumentamos el número de muestras de T , $m \to \infty$, el sesgo deberá denteder a cero, $b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) \to 0$:

$$m = 100 \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = -0.01975$$

$$m = 200 \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = 0.05525$$

$$m = 300 \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = -0.04892$$

$$m = 400 \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = 0.00775$$

$$m = 400 \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = -0.00575$$

$$m = 10000 \qquad b\left(\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]\right) = -0.00385$$

El sesgo es una variable aleatoria (pues $b = \hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right] - F_Z, \text{con} \hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right] \sim N\left(p, \frac{p(1-p)}{n}\right)$) por lo que, al aumentar el valor de m , no siempre se aprecia una reducción del sesgo respecto al caso anterior. Sin embargo, en los casos anteriores sí se que se puede observar una tendencia: el sesgo tiende a cero conforme aumentamos el tamaño de las muestras. Por tanto, según la ley fuerte de los grandes números $\hat{E}_\theta\ \left[\,T\ \left(\overline{X}_n\right)\right]$ es un estimador asintoticamente insesgado (o consistente) de $F_Z = \mathbb{F}_Z\left(x = -\frac{\mu\sqrt{n}}{\sigma}\right)$. (Aquí no entiendo muy bien la diferencia entre estimador consistente e insesgado, ya que el primero se hace insesgado en el límite. ¿Acaso la media un estimador insesgado arroja siempre un valor correcto del parámetro sin importar el tamaño de la muestra?)

Si tomamos también $n \ \to \infty$ la variable aleatoria $\overline{X}_n$ se acercará cada vez más a $N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$,sin embargo, también ocurriría lo siguiente:

$$F_Z = lim_{n\to\infty}\mathbb{F}_Z\left(x = -\frac{(\mu=2)\sqrt{n}}{(\sigma=20)}\right) = \mathbb{F}_Z\left(\to -\infty\right) = 0$$

Es decir, que la probabilidad de que $\bar{X}_n \leq 0$ tiende a cero conforme aumenta n, pues el valor se acerca cada vez más al valor real μ. Por lo que T sería un vector de componentes nulas.

c) Generar una estimación de la varianza de T, $S^2_{T,m}$, a partir de una m.a.s. de tamaño $m = 100$ del estadístico T, utilizando el método de los momentos

El método de los momentos es un método que consiste en la derivación de ecuaciones que relaccionan los momentos poblacionales con los momentos muestrales y despejar parámetro de interes. En este caso la ecuación es simplemente una igualación entre la varianza poblacional y la muestral:

$$S_\theta\ \left[\, T\ \left(\overline{X}_n\right)\right] = Var\left(F_Z\left(\frac{-\mu\cdot\sqrt{n}}{\sigma}\right)\right)$$

En el apartado anterior se hizo lo mismo con la media poblacional y la muestral.

Código:

```
clear all

n=50
m=100

mu=2
sigma=20

flag=-mu*sqrt(n)/sigma

for i=1 : m

    Z=normrnd(0,1)
    if(Z<flag)
        T(i)=1
    else
        T(i)=0
    end

end

var(T)
```

Resultado:

$$S_\theta\ \left[\, T\ \left(\overline{X}_n\right)\right] = 0.19434$$

El valor poblacional puede sacarse asumiendo que $F_Z = p$ (valor poblacional de la media de $T \sim B(p)$, y usando el hecho de que $Var\ [B(p)] = p \cdot (1-p)$:

$$Var\ [B(p)] = 0.23975 \cdot (1 - 0.23975) = 0.18227$$

$$S_\theta\ \left[T\ \left(\overline{X}_n\right)\right] = Var\ [B(p)] + b\left(S_\theta\ \left[T\ \left(\overline{X}_n\right)\right]\right) \qquad b\left(S_\theta\ \left[T\ \left(\overline{X}_n\right)\right]\right) = 0.01157$$

d) Generar $\omega = 30$ m.a.s. de tamaño $m = 100$ de T y obtener una aproximación del Error Cuadrático Medio asociado a la estimación de momentos del parámetro varianza de T.

$$E.\hat{C}.M. \equiv \hat{\epsilon} = \frac{1}{30} \sum_{\omega=1}^{30} \left(S^2_{T,100} - \sigma^2_T\right)^2$$

donde σ^2_T es la varianza poblacional y :

$$S^2_{T,100} = \frac{1}{100} \sum_{\omega=1}^{100} \left(T_i\left(\overline{X}_n\right) - \hat{E}_\theta\ \left[T\ \left(\overline{X}_n\right)\right]\right)^2$$

a su vez cada $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. En total tenemos $n = 30$ m.a.s. $X_i \sim N\left(\mu, \sigma^2\right)$ para cada la generación de cada valor de $T \sim B(p)$, del cual, a su vez se generarán $m = 100$ muestras. Calculamos la varianza de los $m = 100$ valores de T , $S^2_{T,100}$. Finalmente, generamos $\omega = 30$ valores de $S^2_{T,100}$ y le calculamos su error cuadrático medio $\hat{\epsilon}$.

```
clear all

n=50
m=100
omega=30

mu=2
sigma=20
var_poblacional=0.18227
flag=-mu*sqrt(n)/sigma



for j=1 : omega
  for i=1 : m

      Z=normrnd(0,1)
      if(Z<flag)
          T(i)=1
      else
          T(i)=0
```

```
      end

    end
    S2(j)=var(T)
end

ECM=sum((S2.-var_poblacional).*(S2.-var_poblacional))
ECM=ECM./omega
```

$$\boxed{\hat{\epsilon} = 5.0302e - 4} \tag{3}$$

Ejercicio 2:

La distribución de Erlang , $Erlang_{n,\lambda}(x)$, es una distribución biparametrica que opera sobre $X \in [0, \infty)$, y que puede ser definida como suma de n variables aleatorias independientes siguiendo una distribución exponencial de parámetro λ:

$$X \sim Erlang_{n,\lambda}(x) : X = \sum_{i=1}^{n} Y_i \quad \text{donde} \quad Y_i \sim e^{-\lambda y_i}$$

A su vez podemos general las muestras de variables aleatorias exponenciales usando el método de la transformada inversa:

Sea X una variable aleatoria con función de probabilidad acumulada $F_X(x)$, que admite inversa: $u = F_X(x) \in [0,1]$ entonces la variable transformada u sigue una uniforme $u \sim [0,1]$.

Este resultado puede utilizarse para generar una muestra aleatoria siguiendo cualquier distribución de probabilidad: $u = F_X(x) \Leftrightarrow x = F_X^{-1}(u)$. Para el caso de la exponencial:

$$Y \sim e^{-\lambda y} \Leftrightarrow F_Y(y) = 1 - e^{-\lambda y} \quad \text{con} \quad 0 \leq x, \lambda > 0$$

$$u \sim [0,1] \quad , \quad u = 1 - e^{-\lambda y} \Leftrightarrow y = \frac{-1}{\lambda} Ln(1-u)$$

Generar una m.a.s. de tamaño $m = 100$ de una distribución Erlang de parámetros $n = 30$ y $\lambda = 1.5$ a partir de un array de variables exponenciales independientes, a su vez generadas con el método de la transformada inversa:

a) A partir de la muestra generada, estimar el parámetro media y varianza poblacionales mediante el método de los momentos.

La media y la varianza de la población es conocida:

$$X \sim Erlang_{n,\lambda}(x) \qquad E[X] = \frac{n}{\lambda} \qquad Var[X] = \frac{n}{\lambda^2}$$

Definimos los estimadores de $E[X]$ y $Var[X]$. así como sus sesgos como sigue:

$$\bar{X} = E[X] + b(\bar{X}) \qquad S^2 = Var[X] + b(S^2)$$

Código Matlab:

```
clear all

lambda=1.5
n=30
m=100

for i=1:m
  X(i)=0
  for j=1:n
    Y=(-1.0/lambda)*log(1.0-rand)
    X(i)=X(i)+Y
   end
end

mean(X)

var(X)
```

Resultados:

$$\boxed{\bar{X} = 19.990 \qquad b(\bar{X}) = 0.01}$$

$$\boxed{S^2 = 14.003 \qquad b(S^2) = -0.67}$$

b) Generar una m.a.s. de tamaño $n = 3000$ de variables aleatorias exponenciales: $X_1, \cdots, X_N$, con $X_i \sim e^{-\lambda x_i}$, $i = 1, \cdots, N$, $\lambda = 1.5$. Determinar una estimación de tiempo medio de supervivencia y de la función de supervivencia en el intervalo $[0, 50]$ con paso de discretización temporal $\Delta t = 0.5$.

La función de supervivencia es una función que mide la probabilidad de que un sujeto sobreviva más allá de un tiempo dado. Si asumimos que el tiempo desde el origen al evento, t, es una variable aleatoria, $\mathbb{T}$, podemos obtener la función asociada que representa la probabilidad de que el evento ocurra en un tiempo mayo que t: $P(T > t) \equiv \mathbb{S}(t)$, función que vale 0 en $t = 0$ y que se aproxima a 1 conforme $t \to \infty$.

$$\mathbb{S}_{\mathbb{T}}(t) = 1 - F_{\mathbb{T}}(t) = 1 - \int_{-\infty}^{t} F_Y(y)dy = e^{-\lambda t}$$

Puesto que ahora la variable aleatoria es el tiempo de supervivencia t a su valor esperado lo llamaremos tiempo medio de supervivencia: $E[t]$. La media de una densidad de probabilidad exponencial exponencial es bien conocida: $E[t] = \frac{1}{\lambda}$.

Código Matlab para representar la estimación de la densidad de probabilidad exponencial:

```
clear all
lambda=1.5
n=3000
for j=1:n
    X(j)=(-1.0/lambda)*log(1.0-rand)
end
At=0.5
tmax=50
t=0:At:tmax
N=round(tmax/At)+1
```

```
z1=histc(X,t)
z1=z1./n

plot(t,z1)
```

Resultados:

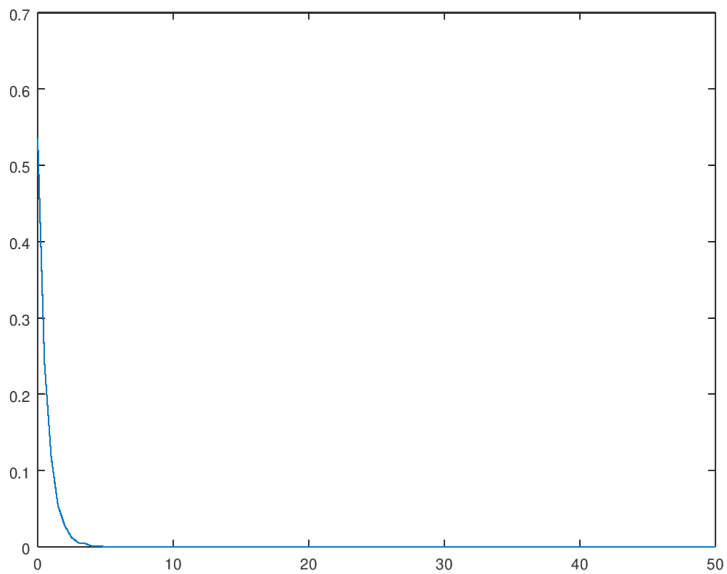

Figure 1: Estimación de la función de densidad de probabilidad obtenida a partir de la m.a.s. exponencial anterior. $f_{\mathbb{T}}(t) = \lambda e^{-\lambda t}$

Usando el comando **mean(X)** después del codigo anterior obtenemos la estimación del tiempo medio de supervivencia:

$$\hat{\mu}_{\mathbb{T}} = E[\mathbb{T}] + b(\hat{\mu}_{\mathbb{T}})$$

$$\boxed{\hat{\mu}_{\mathbb{T}} = 0.66484 \qquad b(\hat{\mu}_{\mathbb{T}}) = -0.00182}$$

Código Matlab para la representación de $\mathbb{S}_{\mathbb{T}}(t)$ y $F_{\mathbb{T}}(t)$

```
clear all
lambda=1.5
n=3000
for j=1:n
    X(j)=(-1.0/lambda)*log(1.0-rand)
end
At=0.5
tmax=50
t=0:At:tmax
```

```
N=round(tmax/At)+1
z1=histc(X,t)
z1=z1./n
%función de supervivencia
for i= 1 : N
    F(i)=0
    for j=1 : i
      F(i)=F(i)+z1(j)
    end
    S(i)=1.0-F(i)
end

plot(t,F,t,S)
```

Resultados:

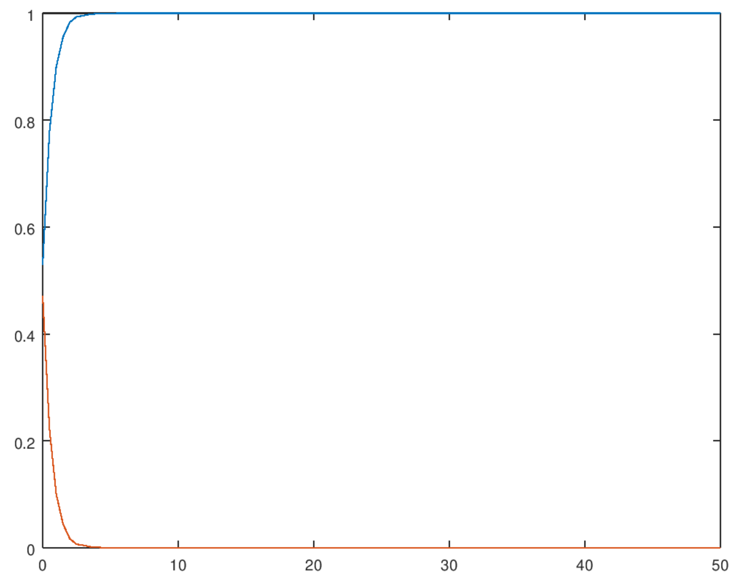

Figure 2: Representación de las estimaciones de las funciones $\hat{F}_{\mathbb{T}}(t)$, linea azul y de $\hat{\mathbb{S}}_{\mathbb{T}}(t)$, linea roja.

c) Usando la m.a.s. del apartado anterior estimar, mediante el método de los momentos, la función de riesgo y la función de riesgo acumulada en el intervalo $[0\ ,\ 50]$, con $\Delta t = 0.5$

La función de riesgo proporciona la probabilidad instantanea de que un evento ocuura en un instante t sin haber ocurrido hasta ese momento: $h_{\mathbb{T}}(t) = \frac{f_{\mathbb{T}}(t)}{S_{\mathbb{T}}(t)}$. Analogamente se define la función de riesgo acumulada como: $H_{\mathbb{T}}(t) = \int_0^t h_{\mathbb{T}}(s)ds$, y nos porporciona la probabilidad de que el evento ocurra entre 0 y t.

Código Matlab:

```
clear all
lambda=1.5
```

```
n=3000
for j=1:n
    X(j)=(-1.0/lambda)*log(1.0-rand)
end
At=0.5
tmax=50
t=0:At:tmax
N=round(tmax/At)+1
f=histc(X,t)
f=f./n

%función de supervivencia
for i= 1 : N
    F(i)=0
    for j=1 : i
      F(i)=F(i)+f(j)
    end
    S(i)=1.0-F(i)
end

%función de riesgo
for i= 1 : N
  if(abs(S(i)) < 0.0001)
      h(i)=0
  else
      h(i)=f(i) / S(i)
  end
end

for i= 1 : N
  H(i)=0
  for j=1 : i
  H(i)=H(i)+h(j)
  end
end

plot(t,H)
```

Resultados:

Función de Riesgo

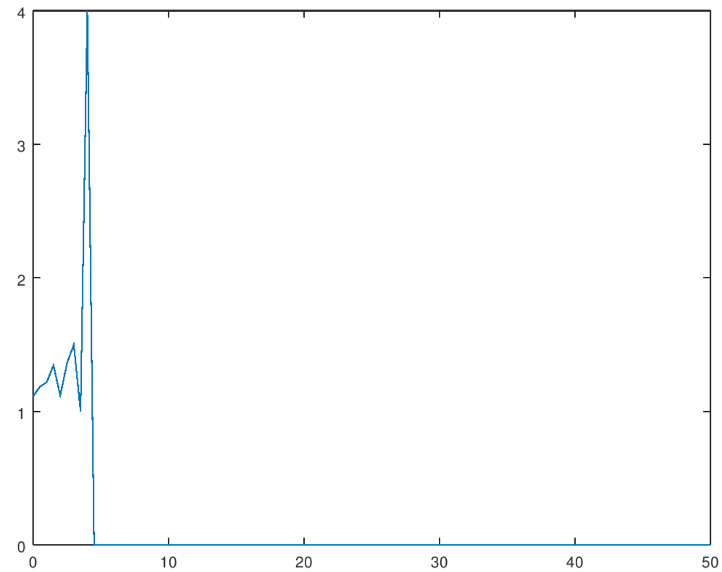

Figure 3: -

Función de Riesgo Acumulado

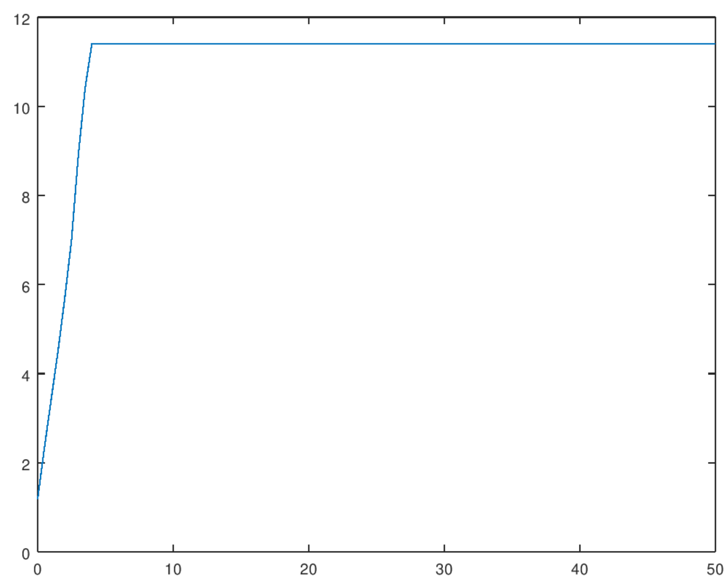

Figure 4: -

d) Generar 200 muestras de tamaño $m = 100$ de $X \sim Erlang_{n,\lambda}(x)$, $n = 30$ y $\lambda = 1.5$. Calcular los errores cuadráticos medios de los estimadores calculados en apartados anteriores.

El programa debe recorrer buclues que contienen del orden de 10^5 iteraciones ($200 \times 100 \times 30$), el lenguaje de programación Matlab resulta ser inpracticablemente lento. Paso entonces a usar un lenguaje de programación de alto nivel como C++.

Error cuadrático medio: estimación de la media de $Erlang_{n,\lambda}(x)$

```
n=30;
m=100;
q=200;
lambda=1.5;


for(l=1;l<=q;l++)
{
    for(i=1;i<=m;i++)
    {
        X[i][l]=0.0;
        for(j=1;j<=n;j++)
        {
            u=gsl_rng_uniform (r);
            Y=(-1.0/lambda)*log(1.0 - u);
            X[i][l]=X[i][l]+Y;
        }

    }
}


for(l=1;l<=q;l++)
{
    MEAN1[l]=0.0;
    MEAN2[l]=0.0;
    for(i=1;i<=m;i++)
    {
        MEAN1[l]=MEAN1[l]+X[i][l];
        MEAN2[l]=MEAN2[l]+X[i][l]*X[i][l];
    }
    MEAN1[l]=MEAN1[l]/m;
    MEAN2[l]=MEAN2[l]/m;
    VAR[l]=MEAN2[l]-MEAN1[l]*MEAN1[l];
}


MEAN_MEAN=0.0;
MEAN_VAR=0.0;
VAR_MEAN=0.0;
VAR_VAR=0.0;

for(l=1;l<=q;l++)
{
```

```
        MEAN_MEAN=MEAN_MEAN+MEAN1[l]/q;
        MEAN_VAR=MEAN_VAR+VAR[l]/q;
    }

    for(l=1;l<=q;l++)
    {
        VAR_MEAN=VAR_MEAN+(MEAN1[l]-MEAN_MEAN)*(MEAN1[l]-MEAN_MEAN)/q;
        VAR_VAR=MEAN_VAR+(VAR[l]-MEAN_VAR)*(MEAN1[l]-MEAN_VAR)/q;
    }

    ECM_MEAN=sqrt(VAR_MEAN/q);
    ECM_VAR=sqrt(VAR_VAR/q);
    cout<<MEAN_MEAN<<"   "<<ECM_MEAN<<"     "<<MEAN_VAR<<"     "<<ECM_VAR<<endl;
```

$$\boxed{\bar{X} = 20.0255 \pm 0.0248315}$$

$$\boxed{S^2 = 13.4946 \pm 0.25917}$$

Error cuadrático medio: estimación de la densidad de probabilidad exponencial

Código:

```
    for(i=1;i<=m;i++)
    {
        for(j=1;j<=n;j++)
        {
            u=gsl_rng_uniform (r);
            Y=(-1.0/lambda)*log(1.0 - u);
            X[i][j]=Y;
        }

    }



    for(l=1;l<=m;l++)
    {
        space=0.0;
        aa=1;

        while(space<tmax)
        {
            hist[aa][l]=0;
            for(i=1;i<=n;i++)
            {

                if(space<=X[i][l] && X[i][l]<(space+At))
                {
                    hist[aa][l]++;
                }
```

```
            }
            space=space+At;
            aa++;
        }
    }


    for(i=1;i<=int(tmax/At);i++)
    {
        pdf[i]=0.0;
        mean1_hist[i]=0.0;
        mean2_hist[i]=0.0;
        for(l=1;l<=m;l++)
        {
            mean1_hist[i]=mean1_hist[i]+hist[i][l]/(1.0*n);
            mean2_hist[i]=mean2_hist[i]+hist[i][l]*hist[i][l]/(1.0*n);
        }

        pdf[i]=mean1_hist[i]/m;
        pdf2[i]=mean2_hist[i]/m;

        var_hist[i]=mean2_hist[i]-pdf[i]*pdf[i];
        ECM_pdf[i]=sqrt(var_hist[i]/q);

        fprintf(f1,"%f %f %f \n",i*At,pdf[i],ECM_pdf[i]);
    }
```

Resultado:

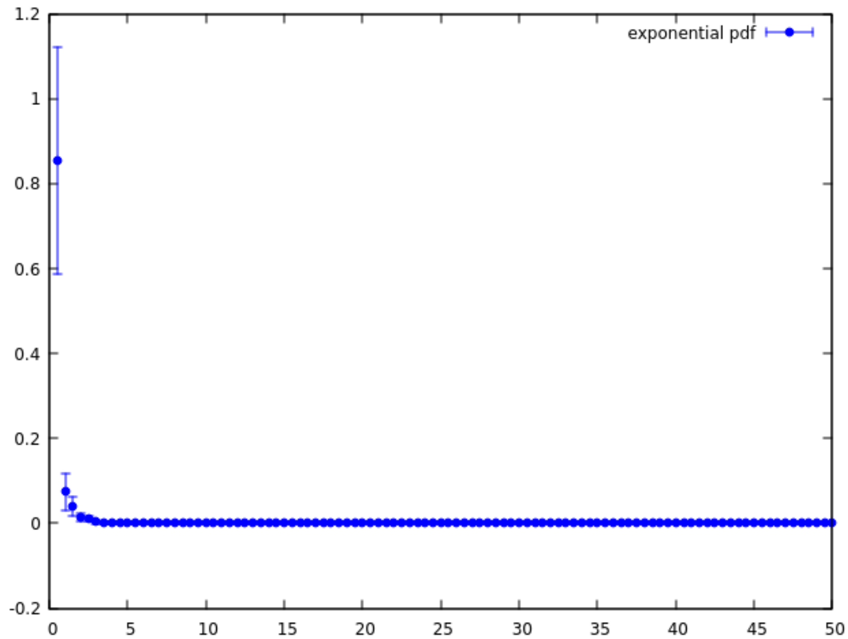

Figure 5: -

Error cuadrático medio: estimación de la distribución de probabilidad acumulada y la función de supervivencia

Código:

```
for(i=1;i<=m;i++)
    {
        for(j=1;j<=n;j++)
        {
            u=gsl_rng_uniform (r);
            Y=(-1.0/lambda)*log(1.0 - u);
            X[i][j]=Y;
        }

    }



    for(l=1;l<=m;l++)
    {
        space=0.0;
        aa=1;

        while(space<tmax)
        {
            hist[aa][l]=0;
            for(i=1;i<=n;i++)
```

```
        {

            if(space<=X[i][l] && X[i][l]<(space+At))
            {
                hist[aa][l]++;
            }

         }

         pdf[aa][l]=hist[aa][l]/(1.0*n);

         space=space+At;
         aa++;
    }
}




for(l=1;l<=m;l++)
{
    for(i=1;i<=int(tmax/At);i++)
    {
        F[i][l]=0.0;
        for(j=1;j<=i;j++)
        {
            F[i][l]=F[i][l]+pdf[j][l];
        }
    }
}

//Cumulative probability function
for(i=1;i<=int(tmax/At);i++)
{
    F_mean1[i]=0.0;
    F_mean1[i]=0.0;
    for(l=1;l<=m;l++)
    {
        F_mean1[i]=F_mean1[i]+F[i][l];
        F_mean2[i]=F_mean2[i]+F[i][l]*F[i][l];
    }
    F_mean1[i]=F_mean1[i]/(1.0*m);
    F_mean2[i]=F_mean2[i]/(1.0*m);
    var_F[i]=F_mean2[i]-F_mean1[i]*F_mean1[i];
    ECM_F[i]=sqrt(var_F[i]/m);
    fprintf(f1,"%f %f %f \n",i*At,F_mean1[i],ECM_F[i]);
}

//Survival function
for(i=1;i<=int(tmax/At);i++)
{
```

```
        S_mean1[i]=0.0;
        S_mean1[i]=0.0;
        for(l=1;l<=m;l++)
        {
            S_mean1[i]=S_mean1[i]+(1.0-F[i][l]);
            S_mean2[i]=S_mean2[i]+(1.0-F[i][l])*(1.0-F[i][l]);
        }
        S_mean1[i]=S_mean1[i]/(1.0*m);
        S_mean2[i]=S_mean2[i]/(1.0*m);
        var_S[i]=S_mean2[i]-S_mean1[i]*S_mean1[i];
        ECM_S[i]=sqrt(var_F[i]/m);
        fprintf(f2,"%f %f %f \n",i*At,S_mean1[i],ECM_S[i]);
    }
```

Resultado

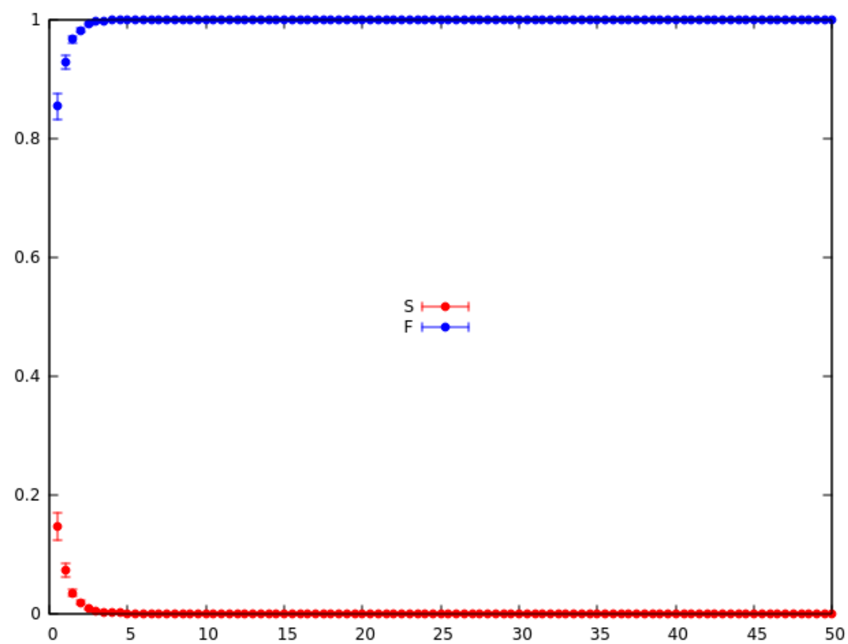

Figure 6: -

Error cuadrático medio: función de riesgo y función de riesgo acumulado

Código:

```
    for(j=1;j<=m;j++)
    {
        for(i=1;i<=n;i++)
        {
            u=gsl_rng_uniform (r);
            Y=(-1.0/lambda)*log(1.0 - u);
            X[i][j]=Y;
        }
```

```
}



for(l=1;l<=m;l++)
{
    space=0.0;
    aa=1;

    while(space<tmax)
    {
        hist[aa][l]=0;
        for(i=1;i<=n;i++)
        {

            if(space<=X[i][l] && X[i][l]<(space+At))
            {
                hist[aa][l]++;
            }

         }

         pdf[aa][l]=hist[aa][l]/(1.0*n);
         space=space+At;
         aa++;
    }

}


for(l=1;l<=m;l++)
{
    for(i=1;i<=int(tmax/At);i++)
    {
        F[i][l]=0.0;
        for(j=1;j<=i;j++)
        {
            F[i][l]=F[i][l]+pdf[j][l];
        }
    }
}


//Risk Function
for(l=1;l<=m;l++)
{
    for(i=1;i<=int(tmax/At);i++)
    {
        if(abs(1.0-F[i][l])>0.01)
        {
```

```
                h[i][l]=pdf[i][l]/(1.0-F[i][l]);
            }
            else
            {
                h[i][l]=0.0;
            }
        }
    }

    //Survival function
    for(i=1;i<=int(tmax/At);i++)
    {
        h_mean1[i]=0.0;
        h_mean1[i]=0.0;
        for(l=1;l<=m;l++)
        {
            h_mean1[i]=h_mean1[i]+(h[i][l]);
            h_mean2[i]=h_mean2[i]+(h[i][l])*(h[i][l]);
        }
        h_mean1[i]=h_mean1[i]/(1.0*m);
        h_mean2[i]=h_mean2[i]/(1.0*m);
        var_h[i]=h_mean2[i]-h_mean1[i]*h_mean1[i];
        ECM_h[i]=sqrt(var_h[i]/m);
        fprintf(f1,"%f %f %f \n",i*At,h_mean1[i],ECM_h[i]);
    }
```

Resultado:

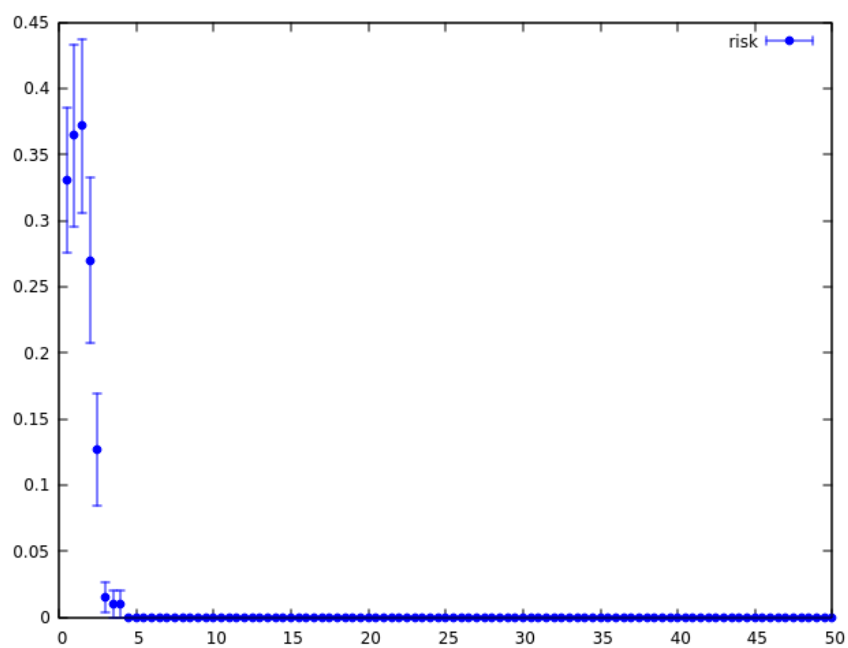

Figure 7: -

Código:

```
for(j=1;j<=m;j++)
{
    for(i=1;i<=n;i++)
    {
        u=gsl_rng_uniform (r);
        Y=(-1.0/lambda)*log(1.0 - u);
        X[i][j]=Y;
    }

}



for(l=1;l<=m;l++)
{
    space=0.0;
    aa=1;

    while(space<tmax)
    {
        hist[aa][l]=0;
        for(i=1;i<=n;i++)
        {

            if(space<=X[i][l] && X[i][l]<(space+At))
            {
                hist[aa][l]++;
            }

         }

         pdf[aa][l]=hist[aa][l]/(1.0*n);
         space=space+At;
         aa++;
    }

}


for(l=1;l<=m;l++)
{
    for(i=1;i<=int(tmax/At);i++)
    {
        F[i][l]=0.0;
        for(j=1;j<=i;j++)
        {
            F[i][l]=F[i][l]+pdf[j][l];
```

```
        }
    }
}


//Risk Function
for(l=1;l<=m;l++)
{
    for(i=1;i<=int(tmax/At);i++)
    {
        if(abs(1.0-F[i][l])>0.0001)
        {
            h[i][l]=pdf[i][l]/(1.0-F[i][l]);
        }
        else
        {
            h[i][l]=0.0;
        }
    }
}

//Cumulated Risk Function
for(l=1;l<=m;l++)
{
    for(i=1;i<=int(tmax/At);i++)
    {
        H[i][l]=0.0;
        for(j=1;j<=i;j++)
        {
            H[i][l]=H[i][l]+h[j][l];
        }
    }
}

//Survival function
for(i=1;i<=int(tmax/At);i++)
{
    H_mean1[i]=0.0;
    H_mean1[i]=0.0;
    for(l=1;l<=m;l++)
    {
        H_mean1[i]=H_mean1[i]+(H[i][l]);
        H_mean2[i]=H_mean2[i]+(H[i][l])*(H[i][l]);
    }
    H_mean1[i]=H_mean1[i]/(1.0*m);
    H_mean2[i]=H_mean2[i]/(1.0*m);
    var_H[i]=H_mean2[i]-H_mean1[i]*H_mean1[i];
    ECM_H[i]=sqrt(var_H[i]/m);
    fprintf(f1,"%f %f %f \n",i*At,H_mean1[i],ECM_H[i]);
}
```

Resultado:

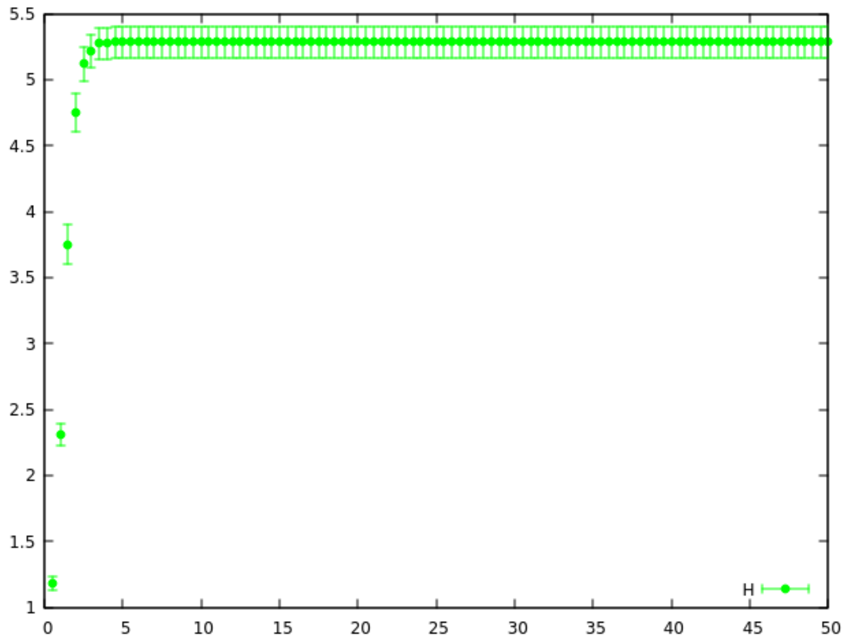

Figure 8: -