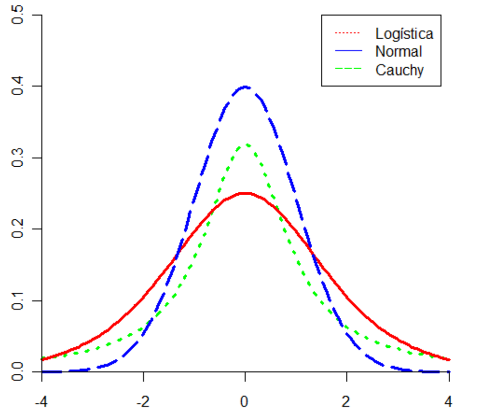
Boletín 2 – Ejercicio 10

Crear un generador para la distribución logística, cuya función de densidad es

$$f(x)=\frac{e^{\frac{-(x-\alpha)}{\beta}}}{\beta\left(1+e^{\frac{-(x-\alpha)}{\beta}}\right)^2},\quad x\in\mathbb{R},\alpha\in\mathbb{R},\beta>0$$

y estudiar su bondad de ajuste.

La distribución logística es una distribución de probabilidad continua que aparece, sobre todo, en la regresión logística y en un tipo de redes neuronales artificiales llamadas *feedforward*. Puesto que la función de distribución es sencilla de calcular e invertir, usaremos el método de inversión para crear el generador. Sin embargo, puesto que la forma de la función de densidad tiene cierto parecido con la de la distribución normal (y por tanto, con la de Laplace y la de Cauchy), utilizaremos también el método de aceptación-rechazo. El siguiente código nos permite visualizar la gráfica de la derecha:



Funciones de densidad de Cauchy, normal y logística de parámetros (0,1)

```
# Gráfica con las funciones de densidad de Cauchy,
# Normal y Logística de parámetros 0 y 1.
curve(dcauchy(x,0,1),axes=FALSE,-4,4,ylab="",xlab="",
    col="green",lty=3,lwd=3,ylim=c(0,0.5))
curve(dlogis(x,0,1),-4,4,add=TRUE,col="red",lty=1,
    lwd=3)
curve(dnorm(x,0,1),-4,4,add=TRUE,col="blue",lty=5,
    lwd=3)
axis(1,pos=0)
axis(2,pos=-4)
legend(1.5, 0.5, c("Logística", "Normal", "Cauchy"),
    col=c("red","blue","green"),lty=c(3,1,5))
title(sub="Funciones de densidad de Cauchy, normal y
    logística de parámetros (0,1)",col.sub="gray10",
    cex.sub=1,font.sub=1,line=2)
```

Primero vamos a usar el método de inversión. Lo primero que necesitamos es calcular la función de distribución a partir de la función de densidad dada:

$$F(x)=\int_{-\infty}^{x} f(y)dy=\int_{-\infty}^{x}\frac{e^{\frac{-(y-\alpha)}{\beta}}}{\beta\left(1+e^{\frac{-(y-\alpha)}{\beta}}\right)^2}dy=\int_{\infty}^{1+e^{\frac{-(x-\alpha)}{\beta}}}\frac{-1}{t^2}dt=\frac{1}{t}\Bigg]_{\infty}^{1+e^{\frac{-(x-\alpha)}{\beta}}}=\frac{1}{1+e^{\frac{-(x-\alpha)}{\beta}}}$$

A partir de ella, vamos a hallar su inversa, que será la que usemos para generar la variable aleatoria:

$$u=\frac{1}{1+e^{\frac{-(x-\alpha)}{\beta}}}\Rightarrow e^{\frac{-(x-\alpha)}{\beta}}=\frac{1}{u}-1\Rightarrow\frac{-(x-\alpha)}{\beta}=\ln\left(\frac{1}{u}-1\right)\Rightarrow x=\alpha-\beta\ln\left(\frac{1}{u}-1\right)$$

de donde $F^{-1}(u)=\alpha+\beta\ln\left(\frac{u}{1-u}\right)$. Con ella, la función para nuestro generador queda de la siguiente manera:

```
# Generador de valores para la distribución logística
# Método de inversión
log_inv<-function(n,alpha=0,beta=1){
    t<-proc.time()
    x<-runif(n,0,1)
    x<-alpha+beta*log(x/(1-x))
    t<-proc.time()-t
    return(list("datos"=x,"tiempo"=t))
}
```

siendo *n* el número de valores que se quieren generar y *alpha* y *beta* los parámetros de la distribución.

Como la distribución logística es una distribución de probabilidad continua, para comprobar la bondad de ajuste de las secuencias de valores creadas, usaremos el test de Kolmogorov-Smirnov. La siguiente función nos devolverá el p-valor y el estadístico del contraste para la secuencia de datos *x*:

```
# Contraste de Kolmogorov-Smirnov
TEST_KS<-function(x,...){
    result<-ks.test(x, ...)
    sal<-c(result$p.value,result$statistic)
    return(sal)
}
```

Una vez tenemos esas dos funciones, ya podemos realizar experimentos con nuestro generador para comprobar la bondad de ajuste a la distribución logística. En este caso, repetiremos K=100 veces la generación de N=1000 valores y la realización del contraste K-S. El código siguiente nos permitirá hacer eso:

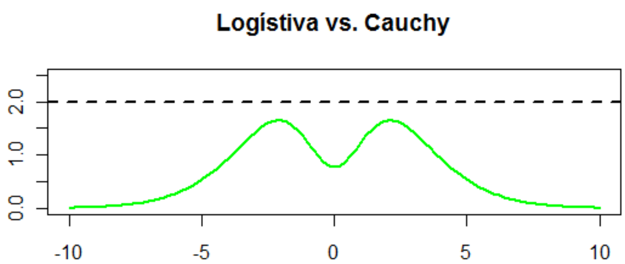
```
# Programa para evaluar la bondad de ajuste
# Establecemos los parámetros
```

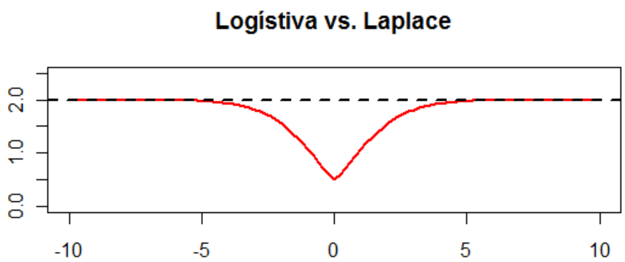
```
alpha=0
beta=1
# Generamos las secuencias de valores
K=100
N=1000
x<-log_inv(N*K,alpha,beta)
cat("Tiempo de generación:",x$tiempo[[3]],
    "segundos.\n")
x<-x$datos
# Convertimos x en una matriz de 100 columnas
dim(x)<-c(N,K)
# Por columnas, realizamos el contraste de KS
res_test<-apply(x,2,TEST_KS,"plogis",alpha,beta)
# Creamos un vector con los p-valores y otro con los
# estadísticos del contraste
(vector_pvalores<-res_test[1,])
(vector_estadistico<-res_test[2,])
# Nos quedamos con los que se rechazan:
rechazados<-vector_pvalores[vector_pvalores<0.05]
# Devolvemos los resultados
(pvalor_estimado<-length(rechazados)/K)
summary(vector_estadistico)
```

Al ejecutar el programa, la estimación del p-valor que nos devuelve está siempre en torno a 0.05, lo cual está en consonancia con el nivel de significación que pretendemos que tenga nuestro generador. En cuanto al valor medio del estadístico, suele estar en torno a 0.00875.

Podemos repetir el experimento para otros parámetros, sin más que variar los valores de *alpha* y *beta* en el código anterior. En cualquier caso, tanto el p-valor como el estadístico del contraste seguirán próximos a 0.05 y 0.00875 respectivamente.

Vamos ahora a generar la distribución logística mediante el método de aceptación-rechazo. En esta ocasión, estudiaremos únicamente el caso particular de la logística de parámetros 0 y 1. El *c* que necesitamos vendrá dado por el máximo del cociente de las funciones de densidad de la distribución logística y las de Cauchy y Laplace. Las gráficas de ambos cocientes son las siguientes:





Así pues, tomaremos *c*=2 en ambos casos. Para la de Cauchy es una aproximación por exceso, pero para la de Laplace es exacta (la recta x=2 es una asíntota horizontal de la curva). Para dibujar las gráficas, incluyendo la función de densidad de Laplace que no está implementada en R, el código ha sido el siguiente:

```
# Función de densidad de Laplace
dlaplace<-function(x,alpha=1){
    dens<-(alpha/2)*exp(-alpha*abs(x))
    return(dens)
}
# Gráficas de los cocientes Logística/Cauchy y
# Logística/Laplace
op<-par(mfrow=c(2,1))
curve(dlogis(x,0,1)/dcauchy(x,0,1),-10,10,
    ylim=c(0,2.5),lwd=2,ylab="",xlab="",col="green",
    main="Logístiva vs. Cauchy")
abline(h=2, lty=2,lwd=2)
curve(dlogis(x,0,1)/dlaplace(x,1),-10,10,
    ylim=c(0,2.5),lwd=2,ylab="",xlab="",col="red",
    main="Logístiva vs. Laplace")
abline(h=2, lty=2,lwd=2)
par(op)
```

Una vez ya hemos elegido el *c* para ambos casos, vamos a crear el generador. Este admitirá como parámetros el número de valores a devolver (*n*) y la distribución con que trabajar (*op* = “cauchy” ó “laplace”). Hay que tener en cuenta que previamente deberemos generar también las distribuciones de Cauchy mediante el método de inversión y de Laplace mediante el de convolución a partir de la exponencial (generada también mediante el método de inversión)

```
# Generar Cauchy por el método de inversión
cauchy<-function(n,sigma){
    x<-runif(n,0,1)
    x<-sigma*tan(pi*(x-0.5))
    return(x)
}
# Generar Laplace por el método de convolución
laplace<-function(n,alpha=1){
    x<-runif(n,0,1)
    x<-(-1/alpha)*log(x)
    u<-runif(n)
    x[u>0.5]<-(-1.0)*x[u>0.5]
    return(x)
}
# Generar logística
log_ar<-function(n,op){
    x<-numeric(n)
    int<-0
    ngen<-0
    t<-proc.time()
```

```
        if (op=="cauchy"){
            c<-2
            while(ngen<n){
                int<-int+1
                v<-cauchy(1,1)
                u<-runif(1)*c*dcauchy(v,0,1)
                if (u<dlogis(v,0,1)) {
                     ngen<-ngen+1
                     x[ngen]<-v
                }
            }
        }
        else if (op=="laplace") {
            c<-2
            while(ngen<n){
                int<-int+1
                v<-laplace(1,1)
                u<-runif(1)*c*dlaplace(v,1)
                if (u<dlogis(v)) {
                     ngen<-ngen+1
                     x[ngen]<-v
                }
            }
        }
        t<-proc.time()-t
        return(list("datos"=x,"intentos"=int,"tiempo"=t))
    }
```

Finalmente, al igual que con el método de inversión, vamos a generar 100 secuencias de 1000 valores y realizar un contraste de Kolmogorov-Smirnov con cada una de ellas, con el fin de comprobar la bondad de ajuste de ambos generadores. El código es idéntico al del método de inversión, cambiando únicamente las funciones de generación usadas.

```
# Generamos las secuencias de valores
K=100
N=1000
x1<-log_ar(N*K,"cauchy")
x2<-log_ar(N*K,"laplace")
cat("Tiempo de generación con Cauchy:",x1$tiempo[[3]],
    "segundos.\n","Tiempo de generación con Laplace:",
    x2$tiempo[[3]],"segundos.\n")
x1<-x1$datos
x2<-x2$datos
# Convertimos x en una matriz de 100 columnas
dim(x1)<-c(N,K)
dim(x2)<-c(N,K)
# Por columnas, realizamos el contraste de KS
res_test1<-apply(x1,2,TEST_KS,"plogis",0,1)
res_test2<-apply(x2,2,TEST_KS,"plogis",0,1)
# Creamos un vector con los p-valores
# y otro con los estadísticos del contraste
(vector_pvalores1<-res_test1[1,])
```

```
(vector_pvalores2<-res_test2[1,])
(vector_estadistico1<-res_test1[2,])
(vector_estadistico2<-res_test2[2,])
# Nos quedamos con los que se rechazan:
rechazados1<-vector_pvalores1[vector_pvalores1<0.05]
rechazados2<-vector_pvalores2[vector_pvalores2<0.05]
# Devolvemos los resultados
(pvalor_estimado1<-length(rechazados1)/K)
(pvalor_estimado2<-length(rechazados2)/K)
summary(vector_estadistico1)
summary(vector_estadistico2)
```

Nuevamente, las estimaciones del p-valor nos devuelven valores en torno a 0.05. Además, los estadísticos de los contraste tienen, de media, valores en aproximados a 0.025. Cabe señalar que, puesto que este último método requiere de un mayor número de operaciones, el tiempo que tarda en generar valores es notablemente mayor que el que usa el método de inversión (menos de un segundo el de inversión frente a los casi 8 del de aceptación-rechazo con Cauchy o casi 12 con Laplace).