

MASTER EN DATA SCIENCE PARA FINANZAS

RIESGO OPERACIONAL

Practica 2.1

Distribuciones de probabilidad

```
##### PRACTICA 2.1 #####
```

```
##### DISTRIBUCIONES DE PROBABILIDAD #####
```

```
#Funciones de la distribuciones de probabilidad  
#raiz:foo (nombre de la distribución)
```

```
#dfoo--función de densidad  
#pfoo--función de distribución (porb.acumulada)  
#qfoo--Función inversa  $F^{-1}(x)$   
#rfoo--generador aleatorio de valores
```

La distribución Gamma, se utiliza para modelar variables que describen *el tiempo hasta que se produce p veces un determinado suceso*. Cuenta con dos parámetros positivos (forma y escala).

```
# DISTRIBUCIONES DE SEVERIDAD
```

```
dgamma(1:2, shape=2, rate=3/2)  
pgamma(1:2, shape=2, rate=3/2)  
qgamma(1/2, shape=2, rate=3/2)  
set.seed(1)  
rgamma(5, shape=2, rate=3/2)
```

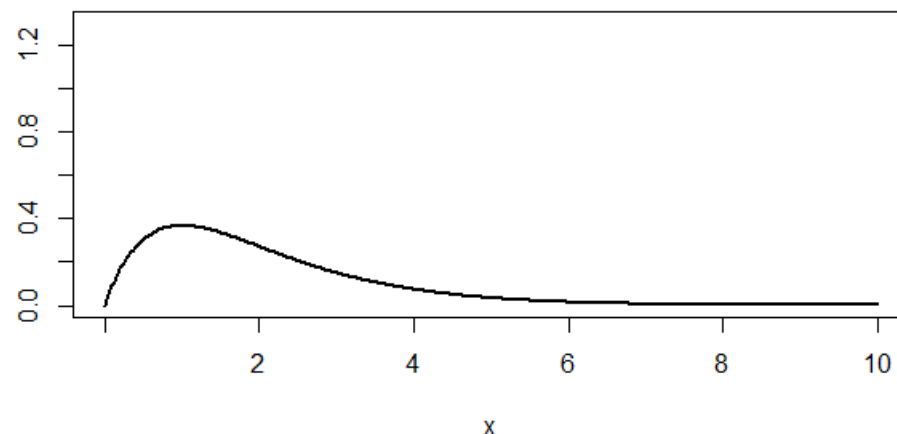
```
> dgamma(1:2, shape=2, rate=3/2)  
[1] 0.5020429 0.2240418  
> pgamma(1:2, shape=2, rate=3/2)  
[1] 0.4421746 0.8008517  
> qgamma(1/2, shape=2, rate=3/2)  
[1] 1.118898  
> set.seed(1)  
> rgamma(5, shape=2, rate=3/2)  
[1] 0.553910 2.380504 2.308780 1.367208 2.590273
```

```
#Transformación de gamma
```

```
f <- function(x) dgamma(x,2)  
f1 <- function(x) f(x-1)  
f2 <- function(x) f(x/2)/2  
f3 <- function(x) 2*x*f(x^2)  
f4 <- function(x) f(1/x)/x^2  
f5 <- function(x) f(exp(x))*exp(x)  
f6 <- function(x) f(log(x))/x
```

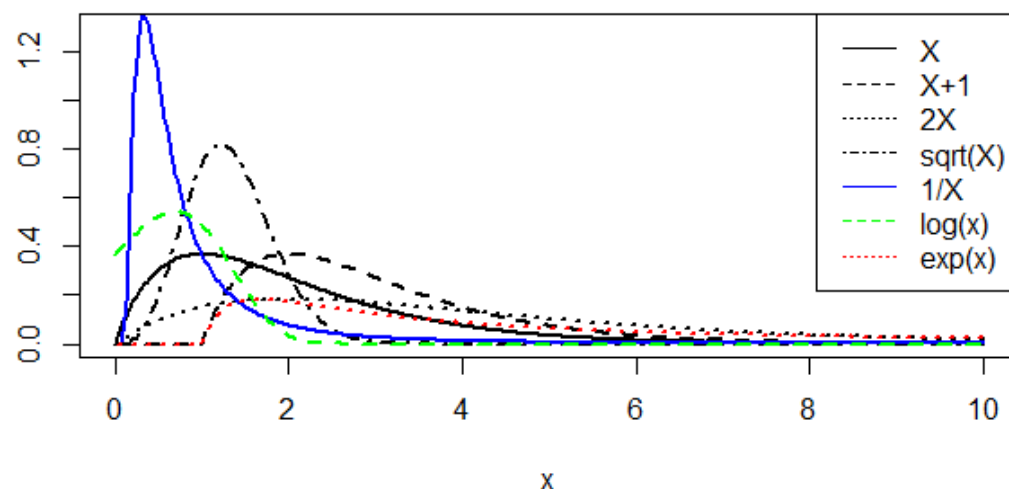
```
x=seq(0,10, by=.025)
plot(x,f(x), ylim=c(0, 1.3), xlim=c(0, 10), main= "Densidades teóricas",
      lwd=2, type="l", xlab="x", ylab="")
```

Densidades teóricas



```
lines(x, f1(x), lty=2, lwd=2)
lines(x, f2(x), lty=3, lwd=2)
lines(x, f3(x), lty=4, lwd=2)
lines(x, f4(x), lty=1, col="blue", lwd=2)
lines(x, f5(x), lty=2, col="green", lwd=2)
lines(x, f6(x), lty=3, col="red", lwd=2)
legend("topright", lty=1:4, col = c(rep("black",4), "blue",
                                     "green", "red"),
      leg=c("X", "X+1", "2X", "sqrt(X)", "1/X", "log(x)", "exp(x)"))
```

Densidades teóricas



```
# A través de simulaciones con visualización de densidades kernel

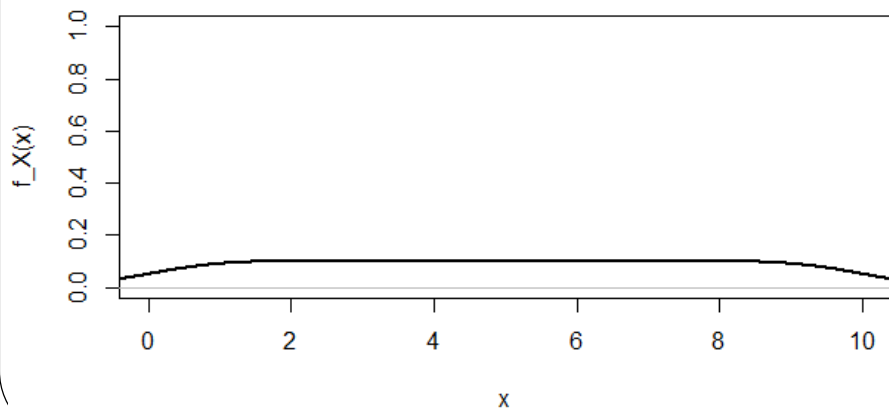
set.seed(123)

x <- rgamma(100,2) #Si omitimos el parámetro escala, se asume que es 1
x
x1 <- (x+1)
x2 <- 2*x
x3 <- sqrt(x)
x4 <- 1/x
x5 <- log(x)
x6 <- exp(x)

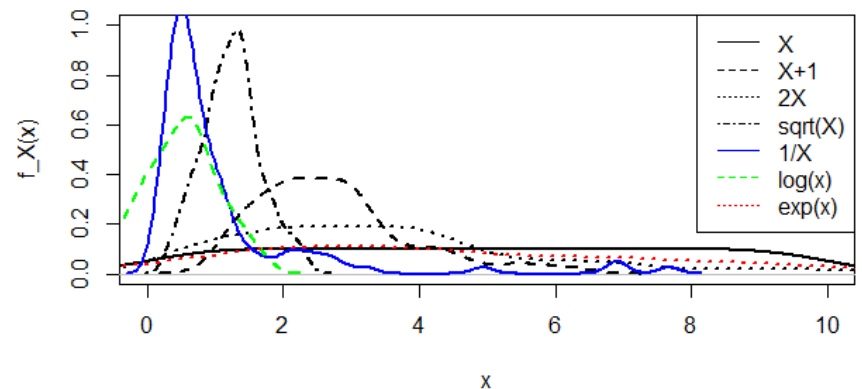
x=seq(0,10, by=.025)
plot(density(x), ylim=c(0, 1), xlim=c(0, 10), main="Densidades empíricas",
     lwd=2, xlab="x", ylab="f_X(x)")

lines(density(x1), lty=2, lwd=2)
lines(density(x2), lty=3, lwd=2)
lines(density(x3), lty=4, lwd=2)
lines(density(x4), lty=1, col="blue", lwd=2)
lines(density(x5), lty=2, col="green", lwd=2)
lines(density(x6), lty=3, col="red", lwd=2)
legend("topright", lty=1:4, col = c(rep("black",4), "blue","green", "red"),
     leg=c("x", "x+1", "2x", "sqrt(x)", "1/x", "log(x)", "exp(x)"))
```

Densidades empíricas



Densidades empíricas



#DISTRIBUCION DE FRECUENCIA

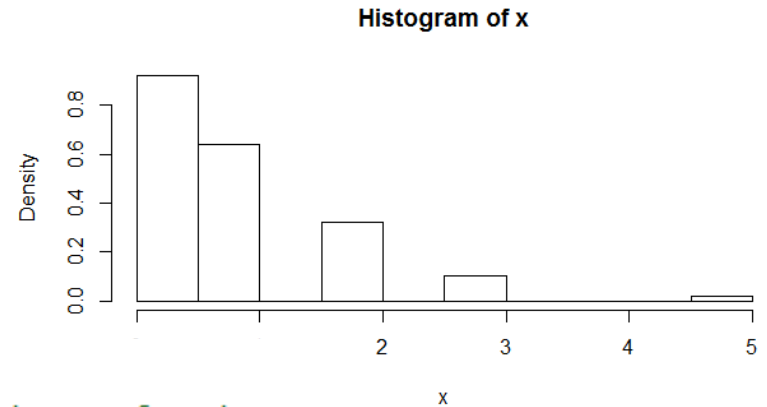
#DISTRIBUCION POISSON

#Generar muestra de n=100 Lambda=0.9

```
x= rpois(100, 0.9)
```

```
rpois(100,0.9)
```

```
hist(x,probability=T) #Histograma
```



#Transformaciones de la Poisson

#Se suelen implementar en paquetes especiales, pero tambien con function:

```
dpoisZM <- function(x, prob, lambda)
```

```
prob*(x == 0) + (1-prob)*(x > 0)*dpois(x-1, lambda)
```

```
ppoisZM <- function(q, prob, lambda)
```

```
prob*(q >= 0) + (1-prob)*(q > 0)*ppois(q-1, lambda)
```

```
qpoisZM <- function(p, prob, lambda)
```

```
ifelse(p <= prob, 0, a+qpois((p-prob)/(1-prob), lamda))
```

```
rpoisZM <- function(n, prob, lambda)
```

F. distribucion Poisson truncada

