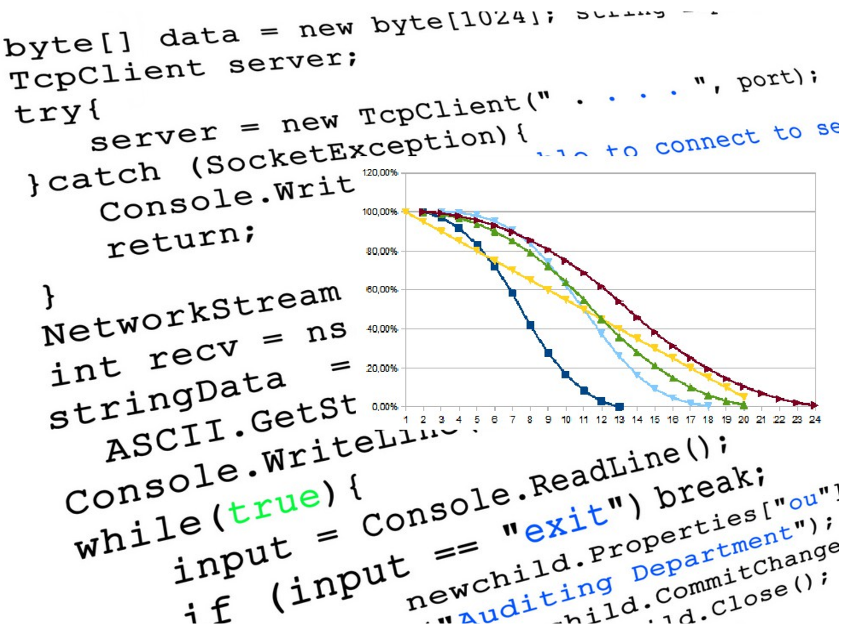
Entornos de población estadísticos y probabilísticos en mIRC Scripting.



Juan Carlos Recio Abad

14/03/2009

juan.carlos.recio.abad@gmail.com

El siguiente documento trata de la implementación de los entornos estadísticos y probabilísticos mediante un lenguaje de programación funcional ligero y a la vez potente, en forma de script conocido como mIRC Scripting.

Importante:

N siempre hace referencia al tamaño de la muestra utilizada, E es la escala.

Los parámetros contadores como i, siempre toman valores desde 0 o 1 (dependendiendo del momento) hasta el tamaño de la muestra N.

El código utiliza un mecanismo de que escala depende de X e Y a la vez, y las probabilidades son siempre < 1, entonces en la implementación de las distribuciones de variables todos los valores de función (eje y) son multiplicados por una variable GENERAL %escalaY (zoom de Y) para ver mejor los resultados, inicializar la variable a 100 por ejemplo con:
//set %escalaY 100
Para cada representación de una distribución convendrá dar un valor a %escalaY adecuado si no se consiguiera visualizar bien la representación que se esté obteniendo.

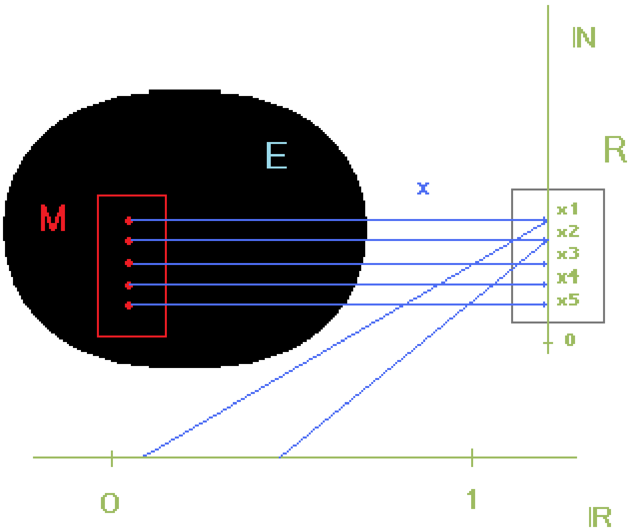
Variables discretas

1.1 Funciones de masa y distribución.

Se considera una variable discreta aleatoria X, que toma los valores de una muestra M de una población o espacio muestral E.

$X: M \to R$

M: $[x_1, x_2, \ldots, x_N]$



con una función de probabilidad $f(x_i) = p[X=x_i]$, conocida como *función de masa*.

$$p(X=k) \leqslant 1$$

para cualquier valor posible de k.

Entonces, dada una muestra M y una función de masa f(x) se tiene que la representación de la función de masa se obtiene con:

//rVADm <M> <f(x)> <E> [-te] (*)

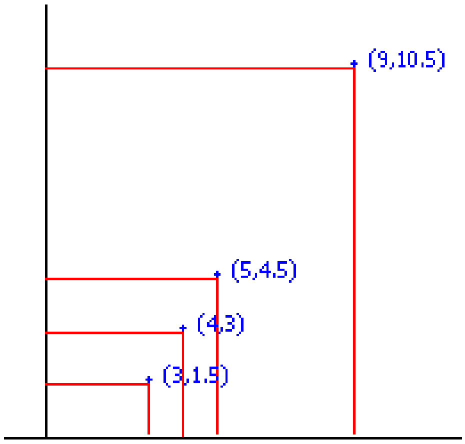
(*) La sintaxis se define generalmente como
//instrucción <parámetro obligatorio> [parámetro opcional] …
E designa la escala elegida para la representación en absolutamente todas las instrucciones de representación.

La muestra M será interpretada por mIRC como x1;x2;...;xN, mientras que la función f(x) será una cadena de carácteres sin espacios, construida por símbolos matemáticos, lógicos y auxiliares (paréntesis, etc) con las variables que se consideren necesarias.

Los parámetros designados por -parámetro1,parámetro2, etc suelen añadir funcionalidades extras.

- t: Los valores de abcisa y función de masa son etiquetados en los ejes correspondientes.
 e: Los puntos son etiquetados.

Ej: //rVADm 3;4;5;9 1.5*x-3 15 -e
(Las funciones de ejemplo no tienen porqué corresponderse con un caso real, tan solo la visualización del funcionamiento, ya que no pueden existir probabilidades > 1)



Implementación:

```
alias rVADm {
        var %a = 1, %b = $numtok($1,59), %c = $1
        while (%b >= %a) {
              %c = $puttok(%c,$+($gettok(%c,%a,59),_,$
                     aplicarFuncion($2,x,$gettok(%c,%a,59))),%a,59)
              if (t isin $4) {
                     drawtext -rn @v $rgb(0,0,255)
                            Tahoma 9 429
                                   $calc(490 -
                                          $calc($aplicarFuncion($2,x,
                                                 $gettok($1,%a,59)) * $3) - 7)
                                          p[X $+ $iif($5 == d,<=,=)
                                          $+ $gettok($1,%a,59) $+ ] $+ = $
                                                 round($aplicarFuncion($2,x,
                                                 $gettok($1,%a,59)),2)

                    drawtext -rn @v $rgb(0,0,255) Tahoma 9
                            $calc(500 + $calc($gettok($1,%a,59) * $3)) 490
                            $gettok($1,%a,59)
              }
              inc %a
       }
       representar -po $rgb(255,0,0) 2 $3 %c
       representar -p $+ $iif(e isin $4,$v1,$chr(32)) $rgb(0,0,255) 2 $3 %c
}
```

Dada una *función de distribución* F(x_i) = p[X <= x_i], función acumuladora de la probabilidad de la función de masa

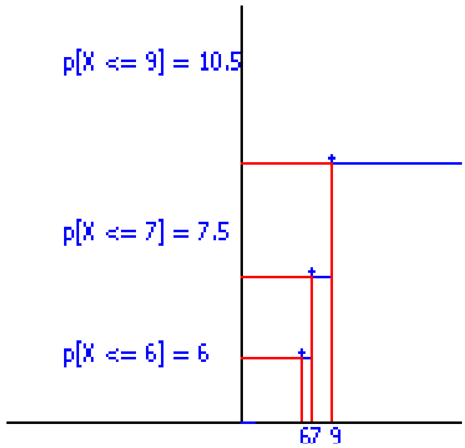
$$F(x) = \sum f(x_i)$$

Entonces, dada una muestra M y una función de distribución F(x), o una función de masa f(x), se tiene que la representación de la función de masa se obtiene indicando cual usar con un parámetro inicial.

//rVADd <-m/d> <M> <f(x)/F(x)> <E> [-te]

- m : Recibe como parámetro de representación la función de masa.
  d : Recibe como parámetro de representación la función de distribución.

Ej: //rVADd -m 6;7;9 1.5*x-3 4 -t (Al contrario que el caso anterior, se va utilizando otra serie de parámetros de ejemplo para no sobrecargar los ejemplos y que se intuya mayor claridad, en este caso prescindimos de e y usamos t)



Implementación:

```
alias rVADd {
if ($1 == -m) {
	var %a = 1, %b = $numtok($2,59), %c = $2
	while (%b >= %a) {
		if (%a == 1) {
		%c = $puttok(%c,$+($gettok(%c,1,59),_,$aplicarFuncion($3,x,
			$gettok(%c,1,59))),1,59)
			if (%b > 1) {
				drawrect -rn @v $rgb(0,0,255) 1 500 490
					$calc($calc($aplicarFuncion($3,x,$gettok(%c,2,59)) -
					$gettok($gettok(%c,1,59),1,$asc(_))) * $4) 1
				drawrect -rn @v $rgb(0,0,255) 1 $calc(500 +
					$calc($gettok($2,1,59) * $4))
					$calc(490 - $calc($4 * $aplicarFuncion($3,x,
					$gettok($2,1,59)))) $calc($4 * $calc($gettok($2,2,59) -
					$gettok($2,1,59))) 1
			}
		}
		elseif (%a > 1) {
			 %c = $puttok(%c,$+($gettok(%c,%a,59),_,$calc($aplicarFuncion($3,x,
			$gettok(%c,%a,59)) + $gettok($gettok(%c,$calc(%a – 1),59),2,$asc(_))))),
			%a,59)
			drawrect -rn @v $rgb(0,0,255) 1 $calc(500 + $calc($gettok($gettok(%c,
			%a,59),1,$asc(_)) * $4)) $calc(490 - $calc($4 * $gettok($gettok(%c,
			%a,59),2,$asc(_)))) $iif(%a < %b,$calc($4 * $calc($gettok($gettok(%c,
			$calc(%a + 1),59),1,$asc(_)) - $gettok($gettok(%c,
			%a,59),1,$asc(_)))),1000) 1
		}
		if (t isin $5) {
			drawtext -rn @v $rgb(0,0,255) Tahoma 9 429 $calc(490 -
				$calc($aplicarFuncion($3,x,$gettok($gettok(%c,%a,59),2,$asc(_)))
				* $4) -  7) p[X <=  $gettok($2,%a,59) $+ ] =
				$round($aplicarFuncion($3,x, $gettok($2,%a,59)),2)
			drawtext -rn @v $rgb(0,0,255) Tahoma 9 $calc(500 +
				$calc($gettok($2,%a,59) * $4)) 490 $gettok($2,%a,59)
		}
		 inc %a
		}
	}
elseif ($1 == -d) {
	 rVADm $2- d
	var %a = 1, %b = $numtok($2,59), %c = $2
	while (%b >= %a) {
		if (%a == 1) {
			drawrect -rn @v $rgb(0,0,255) 1 500 490 $calc($4 * $gettok(%c,1,59)) 1
			 drawrect -rn @v $rgb(0,0,255) 1 $calc(500 + $calc($4 *
			$gettok(%c,1,59))) $calc(490 - $calc($4 * $aplicarFuncion($3,x,
			$gettok(%c,1,59)))) $calc($4 * $calc($gettok(%c,2,59) -
```

```
                    $gettok(%c,1,59))) 1
            }
            elseif (%a > 1)
                    drawrect -rn @v $rgb(0,0,255) 1 $calc(500 + $calc($4 * $gettok(%c,
                            %a,59))) $calc(490 - $calc($4 * $aplicarFuncion($3,x,$gettok(%c,
                            %a,59)))) $iif(%a < %b,$calc($4 * $calc($gettok(%c,$calc(%a +
                            1),59) - $gettok(%c,%a,59))),1000) 1
                    drawpic @v
                    inc %a
            }
        }
  if ($1 != -d) {
        representar -po $rgb(255,0,0) 2 $4 %c
         representar -p $+ $iif(e isin $5,$v1,$chr(32)) $rgb(0,0,255) 2 $4 %c
        }
}
```

1.2 Medidas de centralización, posición y dispersión.

Dada una muestra M se tiene que:

La *media* de una muestra se define como:

$$\bar{x} = \frac{1}{N} \sum x_i$$

Representación de la media de una muestra discreta (media muestral)

//mediaD <color> <M> <E>

<color> se designa por $rgb(<0-255>,<0-255>,<0-255>)

alias mediaD pp $1 2 $3 $media($2) 0 µ(X)

Podemos obtener el valor númerico de la media de una muestra M con:

$media(<M>)

Moda de una muestra, el valor más repetido de una muestra (si existe) se representa:

//modaD <color> <M> <E>

alias modaD pp $1 2 $3 $moda($2) 0 Mo(x)

```
alias moda {
	var %a = 2, %b = $numtok($1,59), %moda = $gettok($1,1,59)
	while (%b >= %a) {
		%c = $veces_($1,$gettok($1,%a,59),59)
		 if (%c > %c_) {
			%c_ = %c
			 %moda = $gettok($1,%a,59)
		}
		inc %a
	}
	return %moda
}
alias veces_ {
	var %a = 1, %b = $numtok($1,$3), %c
	 while (%b >= %a) {
		if ($gettok($1,%a,$3) == $2) inc %c
		inc %a
	}
	 return %c
	}
}
```

Los *cuantiles*, valores de la muestra que dejan un % de valores a su izquierda (o derecha):

$$C(X)_w = x_{N*w}$$

Cuantil de orden w (0 < w < 1), donde 100*w representa el %, se representan:

//cuantilD <color> <M> <w> <E>

```
alias cuantilD pp $1 2 $4 $cuantil($2,$3) 0 C_ $+ $3 = %c
alias cuantil {
	var %a = 1, %b = $numtok($1,59), %c
	while (%b >= %a) {
		if ($calc(%a / %b) >= $2) {
			%c = %a
			 inc %a %b
		}
		else inc %a
	return %c
}
```

La *mediana* (cuantil de orden 0.5) es el valor intermedio (de los n valores) de una muestra si n es impar, si es par, la mediana es la media de los 2 valores intermedios.

//mediana <color> <M> <E>

alias mediana cuantilD $1-2 0.5 $3

$$Me(X) = C(X)_{0.5}$$

La *varianza* de una muestra se define como la media de los cuadrados de las desviaciones de sus valores respecto de su media.

$$V(X) = \overline{x^2} - \overline{x}^2$$

y podemos obtener su valor con el siguiente identificador (su representación gráfica no es ahora relevante.)

$varianza(<M>)

```
alias varianza {
	var %media = $media($1), %a = 1, %b = $numtok($1,59), %c = $1
	while (%b >= %a) {
		%d = $calc(%d + $calc($calc($gettok(%c,%a,59) - %media) ^ 2))
		 inc %a
	}
	%d = $calc(%d / %b)
	return %d
}
```

La *desviación típica* muestra un valor más ajustado a los valores de la muestra, ya que obtiene la raíz cuadrada positiva de la varianza (que se eleva al cuadrado por cuestiones de signo).

$$dt(X) = \sqrt{V(X)}$$

$dt(<M>)

alias dt return $sqrt($varianza($1))

El *coeficiente de variación* mide el error relativo, o la varianza real de la muestra respecto de su media.

$$CV(X) = \frac{dt(X)}{\bar{x}}$$

Se representa con:

$cv(<M>)

alias cv return $calc($dt($1) / $media($1))

Las "cuasimedidas" retornan la misma información cambiando n por n-1.

Cuasivarianza

$cuasivarianza(<M>)

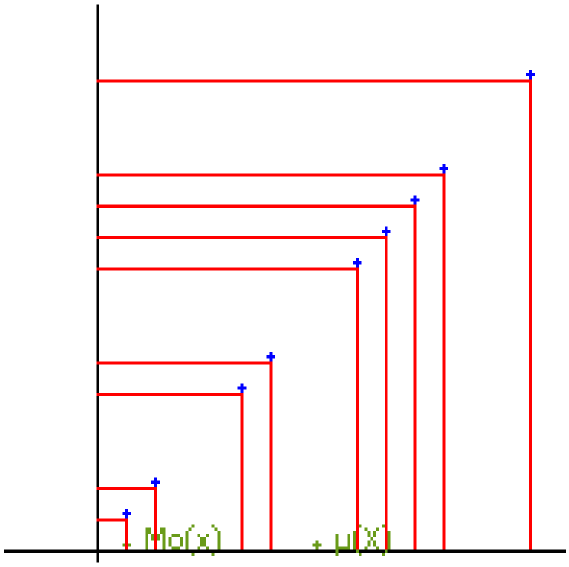
```
alias cuasivarianza {
	var %a = 1, %b = $calc($numtok($1,59) - 1), %c =
	$deltok($puttok($1,$calc($gettok($1,$calc(%b - 1),59) + $gettok($1,%v,59)),%b,59),
	%b,59), %media = $media(%c)
	while (%b >= %a) {
		%c = $puttok(%c,$calc($gettok(%c,%a,59) ^ 2),%a,59)
		inc %a
	}
	%c = $calc($media(%c) - $calc(%media ^ 2))
	return %c
}
```

Cuasidesviación típica

$cdt(<M>)

alias cdt return $sqrt($cuasivarianza($1))

Ejemplo: //ejemplo $rgb(105,155,25) 0;1;2;5;5;6;9;10;11;12;15 10
alias ejemplo {
	mediaD $1-
	modaD $1-
}



1.3 Regresión y correlación

La *covarianza* de dos variables X,Y indica la existencia de dependencia lineal de ambas:

$$Cov(X,Y)=\frac{1}{N}\sum_{1}^{N}(x_i-\bar{x})(y_i-\bar{y})$$

$cov(<x>,<y>)

<x> e <y> tienen que representarse con una muestra de los valores que toman, ambas muestran son de la misma longitud o mismo número de elementos

```
alias cov {
	var %mediax = $media($1), %mediay = $media($2), %a = 1, %b = $numtok($1,59),
	%c = 0
	while (%b >= %a) {
		%c = $calc(%c + $calc($calc($gettok($1,%a,59) - %mediax) *
		$calc($gettok($2,%a,59) - %mediay)))
		inc %a
	}
	%c = $calc(%c / %b)
	return %c
}
```

Coeficiente de correlación lineal de X e Y, así como la covarianza nos indica la existencia de la dependencia lineal, el coeficiente de correlación lineal nos dice cuanto dependen la una de la otra, tomando valores entre -1 y 1 para dependencia inversa o directa, y 0 si son están incorreladas linealmente.

$$r=\frac{Cov(x,y)}{Dt_x\,Dt_y}$$

r => **$crl_(<x>,<y>)**

```
alias crl_ return $calc($cov($1,$2) / $calc($dt($1) * $dt($2)))
```

Dadas dos variables X e Y, *la recta de regresión de Y sobre X*

$$y-\bar{y}=\frac{Cov(x,y)}{V_x}(x-\bar{x})$$

que dada por las muestras de X e Y se representa:

//rRegresion <muestraX> <muestraY> <escala> [-i]

si -i entonces también se representa la recta de regresión de X sobre Y.

```
alias rRegresion {
	var %mediax = $media($1), %mediay = $media($2), %varx = $varianza($1), %cov_xy =
	$cov($1,$2), %fRegresion = %mediay - ( ( %cov_xy * %mediax ) / %varx ) +
	( ( %cov_xy * x ) / %varx ), %a = 1, %b = $numtok($1,59), %c = $1, %d
	while (%b >= %a) {
		%c = $puttok(%c,$+($gettok($1,%a,59),_,$aplicarFuncion(%fRegresion,x,
		$gettok($1,%a,59))),%a,59)
		 %d = $+(%d,$+($gettok($1,%a,59),_,$gettok($2,%a,59)),$chr(59))
		 inc %a
	}
	 representar -pu $rgb(100,150,125) 2 $3 %c
	if (i isin $4) {
		var %vary = $varianza($2), %fRegresion_ = %mediax - ( ( %cov_xy *
		%mediay ) / %vary ) + ( ( %cov_xy * y ) / %vary ), %a = 1, %b =
		$numtok($2,59), %c = $2, %d_
		while (%b >= %a) {
			%c = $puttok(%c,$+($gettok($2,%a,59),_,
			$aplicarFuncion(%fRegresion_,y,$gettok($2,%a,59))),%a,59)
			%d_ = $+(%d_,$+($gettok($2,%a,59),_,$gettok($2,%a,59)),$chr(59))
			 inc %a
		}
	 representar -pu $rgb(100,150,125) 2 $3 %c
}
```

La *varianza residual* de una variable se utiliza para medir el error de la aproximación.

$$VR_y = V(y) - \frac{Cov^2(x,y)}{V(x)}$$

VR(Y) => **$vr(<x>,<Y>)**

(calcula siempre la varianza residual del segundo parámetro, designando a Y)
<x> e <y> son las muestras de ambas variables.

```
alias vrY {
	var %cov_xy  = $cov($1,$2), %varx = $varianza($1), %vary = $varianza($2)
	return $calc(%vary - $calc($calc(%cov_xy ^ 2) / %varx))
}
```

Coeficiente de determinación utilizado para medir el % de acierto de aproximación de la recta de regresión.

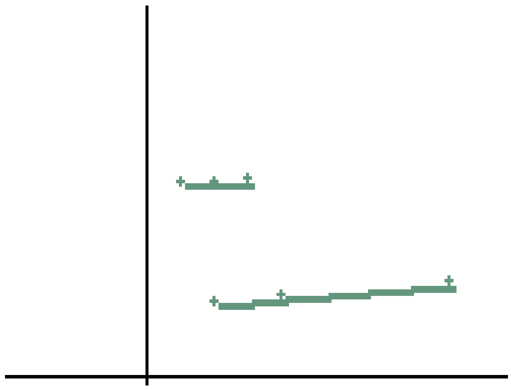
$$R^2 = 1 - \frac{VR_y}{V(y)}$$

R => **$R_d(<x>,<Y>)**

```
alias R_d return $calc(1 - $calc($vr($1,$2) / $varianza($2)))
```

Ejemplo:

```
//rRegresion 1;2;3 2;4;9 11 -i
```



```
> //echo -a $cov(1;2;3,2;4;9)
1

> //echo -a $crl_(1;2;3,2;4;9)
0.13

> //echo -a $vrY(1;2;3,2;4;9)
9.79

> //echo -a $R_d(1;2;3,2;4;9)
1
```

2. Variables continuas

Sea X una v.a. continua en un intervalo [a,b], entonces f(x) es *función de densidad*.

El área bajo la curva de la función de densidad delimitada a un intervalo o subintervalo [ai,aj], nos devuelve la probabilidad de que la variable aleatoria X tome un valor en ese intervalo/subintervalo.

$$p(a \leqslant x \leqslant b) = \int_a^b f(x)\,dx$$

Representación función de densidad de probabilidad:

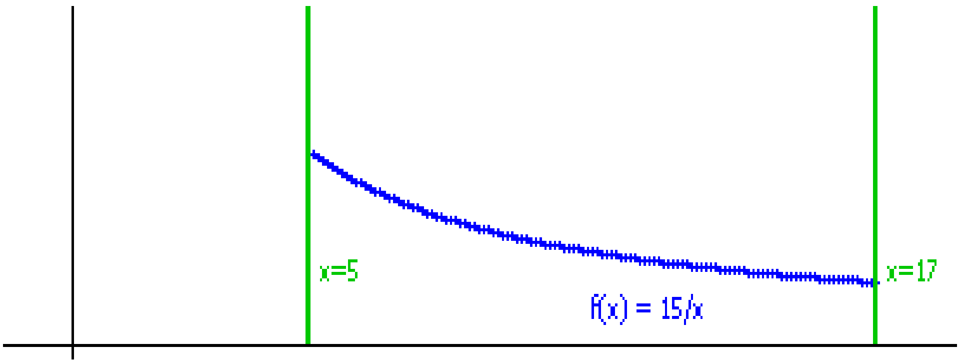
//rFD <-f/d> <f(x)/F1(x);F2(x);...;FN(x)> <E> <a;b>/<a1;b1>_<a2;b2>_..._<aN;bN> [-xeG]

-f: se trabaja directamente con la función de densidad aplicando la definición dada previamente
-d: se trabaja indicando la función de distribución correspondiente al integrar la función de densidad
al trabajar con este parámetro, se indica por trozos (f1(x), f2(x)...)

-x: se marcan con asíntotas verticales los límites de la función densidad si son finitos
-e: se etiqueta la función de densidad
-G: se marcan los limites en cada trozo de la función que se defina diferente (la de distribución)

Ejemplo:

//rfD -f 15/x 20 5 17 -xeG



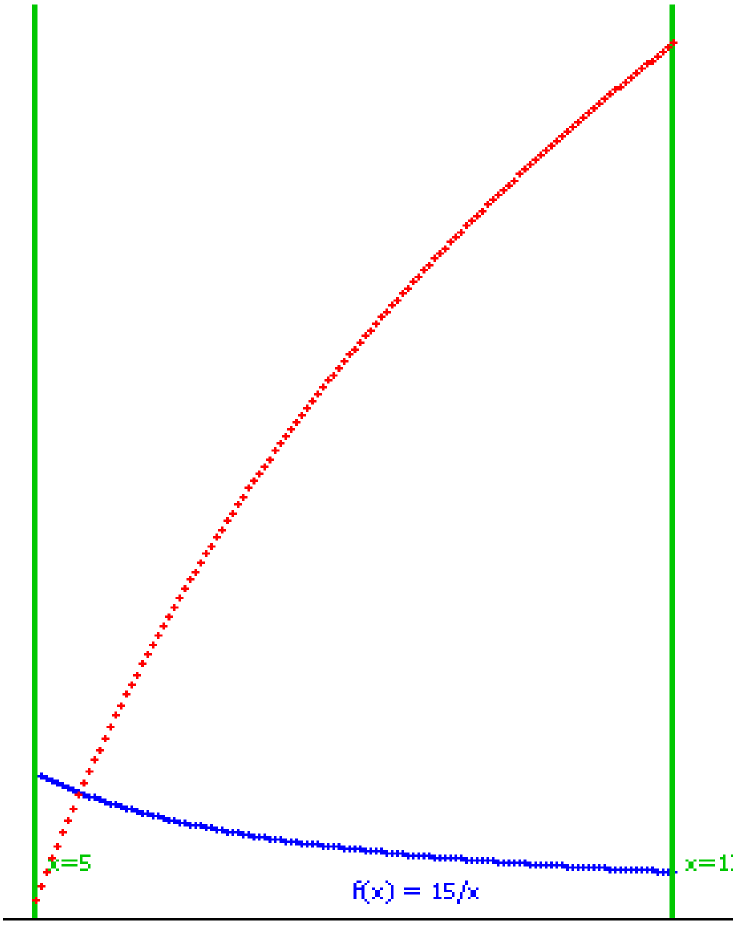
<u>Implementación:</u>

```
alias rFD {
    if ($1 == -f) {
        representar -f $rgb(0,0,255) 2 $3 $4 $+ $chr(59) $+ $5 $+ $chr(59) $+ 0.1_ $+ $2
         if (x isin $6) {
            asintotaV +p $rgb(0,200,0) $3 $4 -e
            asintotaV +p $rgb(0,200,0) $3 $5 -e
        }
        if (e isin $6) drawtext -rn @v $rgb(0,0,255) Tahoma 10 $calc(500 + $calc($3 *
            $4) + $calc($calc($calc($5 - $4) * $3) / 2)) $calc(490 - $calc($3 *
            $aplicarFuncion($2,x,$5))) f(x) = $2
        }
        elseif ($1 == -d) {
            var %a = 1, %b = $numtok($2,59)
            asintotaV +p $rgb(0,200,0) $3 $gettok($gettok($4,1,$asc(_)),1,59)
            asintotaV +p $rgb(0,200,0) $3 $gettok($gettok($4,%b,$asc(_)),2,59)
            while (%b >= %a) {
                representarDerivada $rgb(0,0,255) $gettok($2,%a,59) $3
                $gettok($4,%a,$asc(_))
                if (G isin $5) asintotaV +p $rgb(0,200,0) $3
                    $gettok($gettok($4,%a,$asc(_)),1,59)
                if (e isin $5) drawtext -rn @v $rgb(0,0,255) Tahoma 10 $calc(500
                    + $calc($3 * $gettok($gettok($4,%a,$asc(_)),1,59)) +
                    $calc($calc($calc($gettok($gettok($4,%a,$asc(_)),2,59) -
                    $gettok($gettok($4,%a,$asc(_)),1,59)) * $3) / 2)) $calc(490
                    - $calc($3 * $aplicarFuncion($gettok($2,%a,59),x,
                    $gettok($gettok($4,%a,$asc(_)),1,59)))) f $+ %a $+ (x)
                 inc %a
        }
    }
    drawpic @v
}
```

La *función de distribución* de una variable aleatoria continua se obtiene integrando la función de densidad entre menos infinito y el valor cuya probabilidad acumulada se desea obtener.

$$p(X \leqslant x) = F(x) = \int_{-\infty}^{x} f(x)\,dx$$

Ej:
//rfD -f 15/x 20 5 17 -xeG
//representarIntegral $rgb(255,0,0) 15/x 20 5;17



<u>3. Modelos de distribución estadísticas de variable discreta</u>

Las funciones implementadas se basan en recibir la muestra como parámetro, con unas funciones nuevas se puede implementar rápidamente para un X=? o X<=? sin pasar como parámetro la muestra de una variable.

<u>3.1 Distribución uniforme discreta</u>: *U(a,b)*

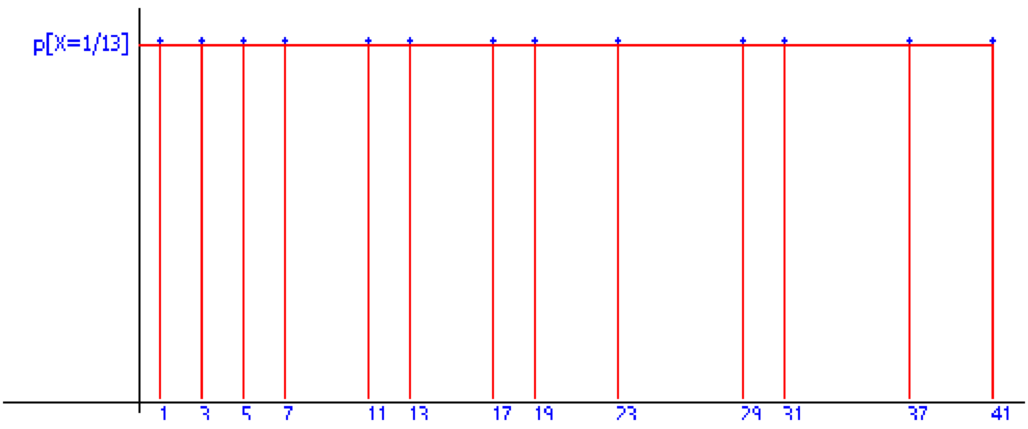
<u>Función de masa:</u>

$$p(X=x_i)=\frac{1}{N}$$

//disUniformeM <M> <E> [-t]
-t: etiqueta los valores en los ejes

Ejemplo:

//set %escalaY 200
//disUniformeM 1;3;5;7;11;13;17;19;23;29;31;37;41 10 -t



__Función de distribución:__

$$0 \qquad si\, x < x_1$$

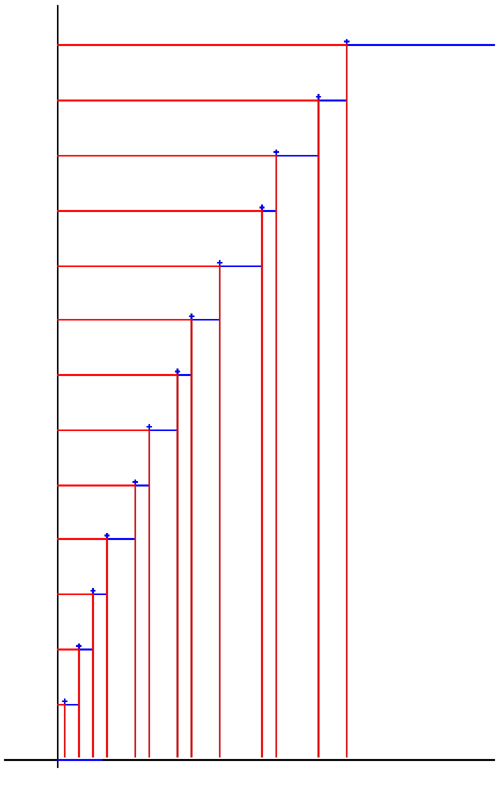
$$\frac{1}{N}\, si\, x_1 \leqslant x < x_2$$

$$p(X \leqslant x) = F(X) = \{ \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

//disUniformeD <M> <E> [-t]

Ejemplo:

//set %escala 200
//disUniformeD 1;3;5;7;11;13;17;19;23;29;31;37;41 10 -t



Implementación:

```
alias disUniformeM {
	var %b = $numtok($1,59)
	rVADm $1 $calc(%escalaY * $+(1/,%b)) $2 $iif($3 == -t,$v1,$chr(32))
	drawrect -rfn @v $rgb(255,255,255) 1 420 $calc(490 - $calc(%escalaY * $2 * $
		+(1/,%b)) - 7) 80 12
	drawtext -rn @v $rgb(0,0,255) Tahoma 10 450 $calc(490 - $calc(%escalaY * $2 * $
	+(1/,%b)) - 7) p[X=1/ $+ %b $+ ]
	drawpic @v
}
```

$$\frac{i}{N}\ si\ x_i \leqslant x < x_{i+1}$$

$$1 \qquad si\ x_N \leqslant x$$

3.2 Distribución binomial:

$$\beta(k, p)$$

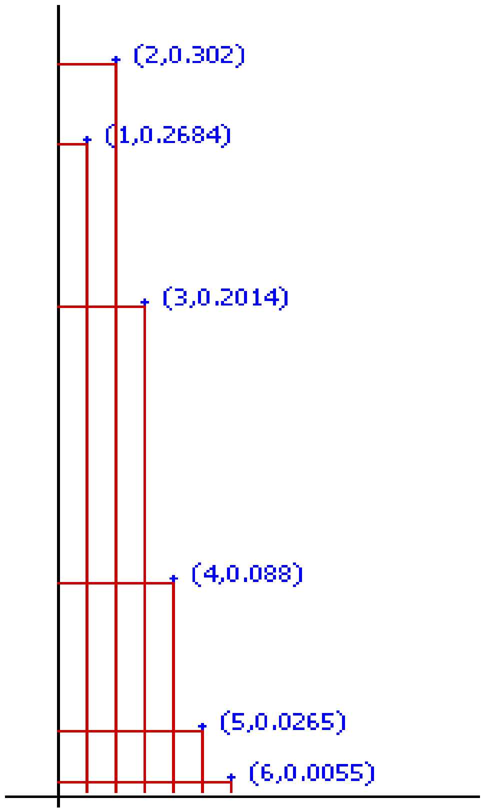
La distribución binomial nos informa acerca de la probabilidad de que ocurra un cierto suceso un determinado un número de veces a (a éxitos) en k intentos (k pruebas de Bernoulli), siendo p la probabilidad de que ocurra el suceso una sola vez.

Función de masa

$$p(X=a)=\binom{k}{a} p^a (1-p)^{(k-a)}$$

//distribucionBinomialM <M> <E> <k> <p> [-to]

Ejemplo: //distribucionBinomialM 1;2;3;4;5;6 10 10 0.2 -to



La distribución binomial utiliza números combinatorios para expresar sus funciones matemáticas, la implementación de las combinaciones es la siguiente, pero antes vemos su definición formal:

<u>Combinaciones:</u> son las diferentes agrupaciones de a elementos desordenadas tomadas de un grupo de k elementos.

$$\binom{k}{a} = C(k,a) = \frac{k!}{a!(k-a)!}$$

$combinatorio_(<k>,<a>)

```
alias combinatorio_
	return $calc($factorial($1) / $calc($factorial($2) * $factorial($calc($1 - $2))))
```

A su vez, los números combinatorios se ayudan de números factoriales que se definen como el producto de todos los números naturales desde 1 hasta el propio número, tomando como abuso de notación 0! = 1.

Factorial de un número:

$$k\,! = \prod_{(i=k)}^{0} (k-i)$$

$factorial(<n>)

```
alias factorial {
	if ($1 == 0) return 1
	else return $calc($1 * $factorial_($calc($1 - 1)))
}
alias factorial_ return $factorial($1)
```

Implementación:

```
alias distribucionBinomialM {
	var %a = 1, %b = $numtok($1,59), %fdensidadBinomial = $ $+ combinatorio_( $+ $3,x)
	* ( $+ $4 $+ ^ x $+ ) * $calc(1 - $4) $+ ^ $+ ( $+ $3 - x)
	while (%b >= %a) {
		pp $rgb(0,0,255) 2 $2 $gettok($1,%a,59) $calc(%escalaY *
			$calc($eval($gettok($replace(%fdensidadBinomial,x,
			$gettok($1,%a,59)),1,$asc(*)),2) *
			$calc($gettok($replace(%fdensidadBinomial,x,$gettok($1,%a,59)),2-,
			$asc(*))))) $iif(t isin $5,( $+ $gettok($1,%a,59) $+ $chr(44) $+
			$round($calc($calc($eval($gettok($replace(%fdensidadBinomial,x,
			$gettok($1,%a,59)),1,$asc(*)),2) *
			$calc($gettok($replace(%fdensidadBinomial,x,$gettok($1,%a,59)),2-,
			$asc(*))))),4) $+ ),$chr(32))
		if (o isin $5) dibujarCoordenadas $rgb(205,0,0) $2 $gettok($1,%a,59)
				$calc(%escalaY *
				$calc($eval($gettok($replace(%fdensidadBinomial,x,
				$gettok($1,%a,59)),1,$asc(*)),2) *
				$calc($gettok($replace(%fdensidadBinomial,x,
				$gettok($1,%a,59)),2-,$asc(*)))))
		inc %a
	}
	drawpic @v
}
```

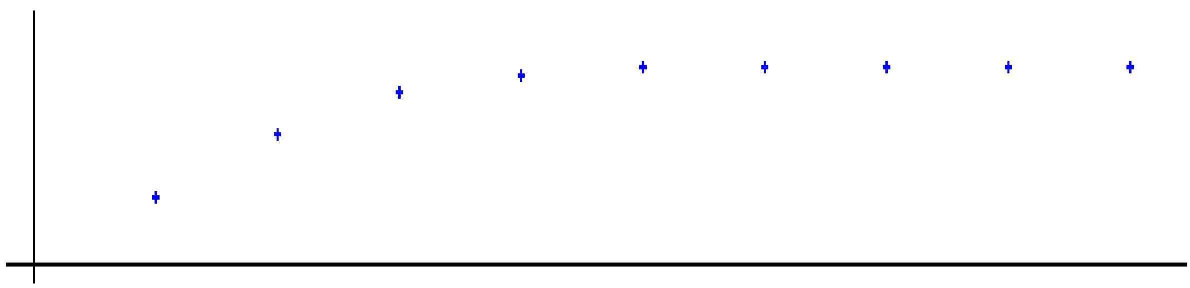
Función de distribución

$$F(x)=\begin{cases} 0 & si\ x<0 \\ \sum_{(i=1)}^{|x|}\left(\frac{k}{i}\right)p^{i}(1-p)^{(k-i)} & \end{cases}$$

//distribucionBinomialD <M> <E> <k> <p> [-to]

-t: representación de cada punto
-o: dibujar coordenadas de cada punto

Ejemplo: //distribucionBinomialD 1;2;3;4;5;6;7;8;9 50 10 0.2



Implementación:

```
alias distribucionBinomialD {
	var %a = 1, %b = $numtok($1,59), %fdensidadBinomial = $ $+ combinatorio_( $+ $3,x)
	* ( $+ $4 $+ ^ $+ x) * $calc(1 - $4) $+ ^ $+ ( $+ $3 - x), %c = 0
	while (%b >= %a) {
		%c = $calc(%c + $calc($calc($eval($gettok($replace(%fdensidadBinomial,x,
			$gettok($1,%a,59)),1,$asc(*)),2) *
			$calc($gettok($replace(%fdensidadBinomial,x,$gettok($1,%a,59)),2-,
			$asc(*))))))
		pp $rgb(0,0,255) 2 $2 $gettok($1,%a,59) %c $iif(t isin $5, ( $+
			$gettok($1,%a,59) $+ $chr(44) $+ $round($calc(%c / %escalaY),4) $+ ),
			$chr(32))
		if (o isin $5) dibujarCoordenadas $rgb(205,0,0) $2 $gettok($1,%a,59) %c
		inc %a
	}
	drawpic @v
}
```

Media y varianza de una variable que sigue una distribución binomial.

$$\mu(X)=\sum_{(i=1)}^{k}\mu(X_i)=kp$$

$$\text{ya que cada } X_i \text{ sigue una distribución}$$
$$\beta(1,p) \text{ lo que implica } \mu(X_i)=p$$

$mediaBinomial(<k>,<p>)

alias mediaBinomial return $calc($1 * $2)

De forma análoga, se sigue un proceso similar con la varianza para obtener:

$$V(X)=\delta^2(X)=\sum_{(i=1)}^{k}\delta^2(X_i)=\sum_{(i=1)}^{k}(p-p^2)$$
$$=kp(1-p)$$

$$\delta^2(X)=p-p^2$$

$varianzaBinomial(<p>)

alias varianzaBinomial return $calc($1 - $calc($1 ^ 2))

3.3 Distribución geométrica: $G(p)$

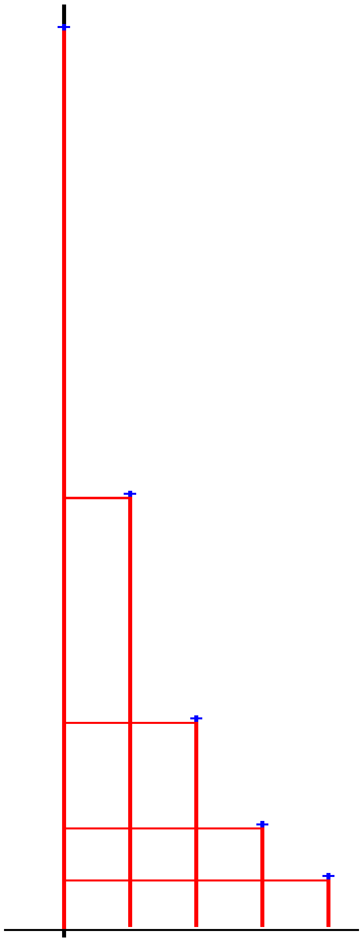
La distribución geométrica nos informa de la probabilidad de que un suceso tenga éxito por primera vez en un determinado número de intentos, siendo p la probabilidad de éxito.

Función de masa

$$p(X=x)=(1-p)^x\, p$$

//distribucionGeometricaM <M> <E> <p> [-to]

Ejemplo:
//set %escalaY 50
//distribucionGeometricaM 0;1;2;3;4 10 0.92 -to



<u>Implementación:</u>

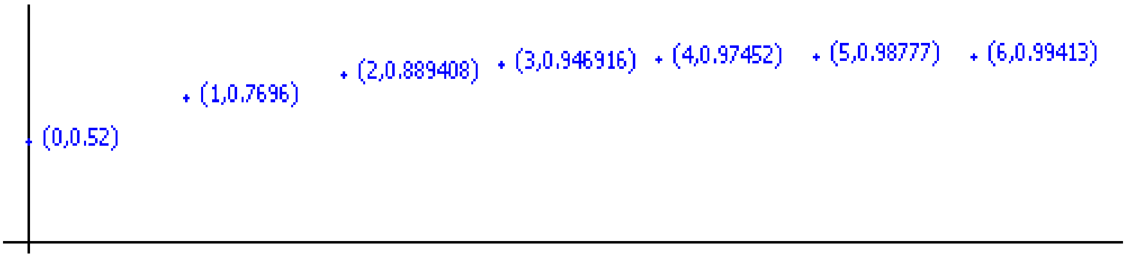
```
alias distribucionGeometricaM {
	var %a = 1, %b = $numtok($1,59), %c
	while (%b >= %a) {
		%c = $addtok(%c,$+($gettok($1,%a,59),_,$calc(%escalaY * $calc($calc($calc(1 -
		$3) ^ $gettok($1,%a,59)) * $3))),59)
		if (o isin $4) dibujarCoordenadas $rgb(255,0,0) $2 $gettok($1,%a,59)
				$calc(%escalaY * $calc($calc($calc(1 - $3) ^ $gettok($1,%a,59)) *
				$3)))
		inc %a
	}
	representar -p $+ $iif(t isin $4,e,$chr(32)) $rgb(0,0,255) 2 $2 %c
}
```

Función de distribución

$$p(X \leqslant x) = F(x) = 1 - (1 - p)^{(x+1)}$$

//distribucionGeometricaD <M> <E> <p> [-to]

Ejemplo:
//set %escalaY 50
//distribucionGeometricaM 0;1;2;3;4 10 0.92 -to



Implementación:

```
alias distribucionGeometricaD {
	var %a = 1, %b = $numtok($1,59), %c
	while (%b >= %a) {
		%c = $addtok(%c,$+($gettok($1,%a,59),_,$calc(1 * $calc($gettok($gettok(%c,
		$calc(%a - 1),59),2,$asc(_)) + $calc($calc($calc(1 - $3) ^ $gettok($1,%a,59)) *
		$3)))),59)
		if (o isin $4) dibujarCoordenadas $rgb(255,0,0) $2 $gettok($gettok(%c,
			$numtok(%c,59),59),1,$asc(_)) $calc(%escalaY * $gettok($gettok(%c,
			$numtok(%c,59),59),2,$asc(_)))
		inc %a
	}
	representar -p $+ $iif(t isin $4,e,$chr(32)) $rgb(0,0,255) 2 $2 %c
}
```

Media y *varianza* de una distribución geométrica:

$$\mu(X)=\frac{1-p}{p}$$

$mediaGeo(<p>)

```
alias mediaGeo return $calc($calc(1 - $1) / $1)
```

$$\delta^2(X)=\frac{1-p}{p^2}$$

$varianzaGeo(<p>)

```
alias varianzaGeo return $calc($calc(1 / $1) * $mediaGeo($1))
```

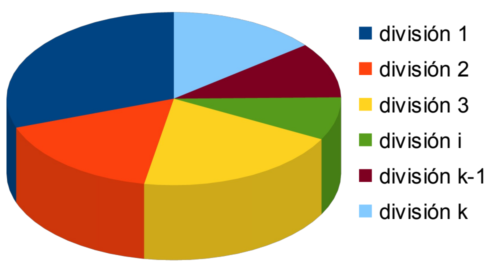
En la implementación de la varianza se ve que podemos relacionar ambas directamente:

$$\delta^2(X)=\frac{1}{p}\mu(X)$$

3.4 Distribución de Poisson: $Poi(\lambda)$

Probabilidad de que ocurran k sucesos sabiendo que ocurren con un promedio λ por unidad de alguna determinada magnitud (tiempo, superficie, etc).

Ejemplo gráfico.



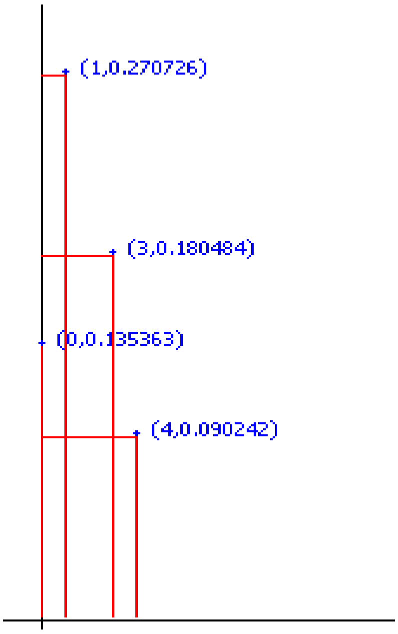
k divisiones de un disco duro
λ número de errores esperados por unidad de superficie del disco

Función de masa:

$$p(X=x)=f(x)=e^{-\lambda}\frac{\lambda^{x}}{x!}$$

//disPoissonM <M> <E> <λ> [-to]

Ejemplo: //disPoissonM 0;1;3;4 10 2 -to



Implementación:

```
alias disPoissonM {
	var %a = 1, %b = $numtok($1,59), %c
	while (%b >= %a) {
		%c  = $calc($calc(2.718 ^ $calc(-1 * $3)) * $calc($calc($3 ^ $gettok($1,%a,59)) /
		$factorial($gettok($1,%a,59))))
		if (o isin $4) dibujarCoordenadas $rgb(255,0,0) $2 $gettok($1,%a,59)
					$calc(%escalaY * %c)
		pp $rgb(0,0,255) 2 $2 $gettok($1,%a,59) $calc(%escalaY * %c) $iif(t isin $4, ( $
			+ $gettok($1,%a,59) $+ $chr(44) $+ %c $+ ),$chr(32))
		inc %a
	}
}
```

Función de distribución

$$p(X=x)=F(x)=\sum_{i=0}^{x} e^{-\lambda}\frac{\lambda^{i}}{i!}$$

//disPoissonD <M> <E> <λ> [-to]

```
alias disPoissonD {
	var %a = 1, %b = $numtok($1,59), %c
	while (%b >= %a) {
		%c  = $calc(%c + $calc($calc(2.718 ^ $calc(-1 * $3)) * $calc($calc($3 ^
		$gettok($1,%a,59)) / $factorial($gettok($1,%a,59))))))
		if (o isin $4) dibujarCoordenadas $rgb(255,0,0) $2 $gettok($1,%a,59)
		$calc(%escalaY * %c)
		pp $rgb(0,0,255) 2 $2 $gettok($1,%a,59) $calc(%escalaY * %c) $iif(t isin $4, ( $
		+ $gettok($1,%a,59) $+ $chr(44) $+ %c $+ ),$chr(32))
		inc %a
	}
}
```

4. **Modelos de distribución estadísticas de variable continua**

4,1 Distribución uniforme continua. *U(a,b)*

Función de densidad

$$f(x)=\begin{cases}\frac{1}{b-a} & si\; a<x<b\\ 0 & en\,otro\,caso\end{cases}$$

con la función de densidad, podemos calcular la probabilidad por el método ya definido en el apartado 2.

$$p(a<X<b)=\int_a^b f(x)\,dx$$
$$p(X=x)=0\ \forall\ x\in\mathbb{R}$$

//uniformeVdensidad <E> <a;b> [-xeG]

alias uniformeVdensidad rFD -f $calc(%escalaY / $calc($gettok($2,2,59) - $gettok($2,1,59))) $remove($1-,;-xeG;)

Función de distribución

$$F(x)=\begin{cases} 0\ si\ x<a \\ \dfrac{x-a}{b-a}\ si\ a\leqslant x<b \\ 1\ si\ b\leqslant x \end{cases}$$

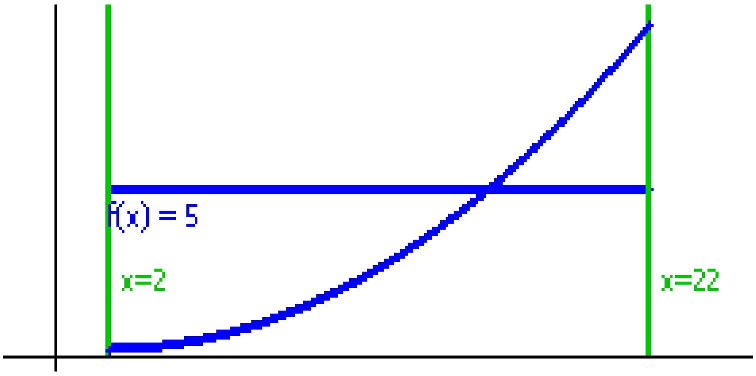
//uniformeVdistribucion <E> <a;b> [-xeG]

```
alias uniformeVdistribucion {
        representarIntegral $rgb(0,0,255) ((x- $+ $gettok($2,1,59) $+ )/( $+ $gettok($2,2,59) $
                + - $+ $gettok($2,1,59) $+ )) $1 $2
}
```

Ejemplo:

//uniformeVdensidad 10 2;22 -xeG

//uniformeVdistribucion 10 2;22 -xeG



4.2 Distribución exponencial: $Exp(B)$

Nos indica cuando la probabilidad de que una variable aumente o disminuya se da de forma exponencial a la propia variación de la variable.

Función de densidad:

$$f(x)=\begin{cases} 0 \; si \; x<0 \\ \beta e^{-\beta x} \; si \; x>0 \end{cases}$$

//disExponencialM <a;b> <E> <B> [-xeG]
alias disExponencialM rfd -f ($+ $3 $+ *2.71^($+ (- $+ $3 $+ *x))) $2 $1 $4-

//disExponencialD <muestra> <escala> <B> [-xeG]
alias disExponencialD rfd -f (1- $+ 2.71^($+ (- $+ $3 $+ *x))) $2 $gettok($1,1,59) $+ $chr(59) $+ $gettok($1,$numtok($1,59),59) $4-

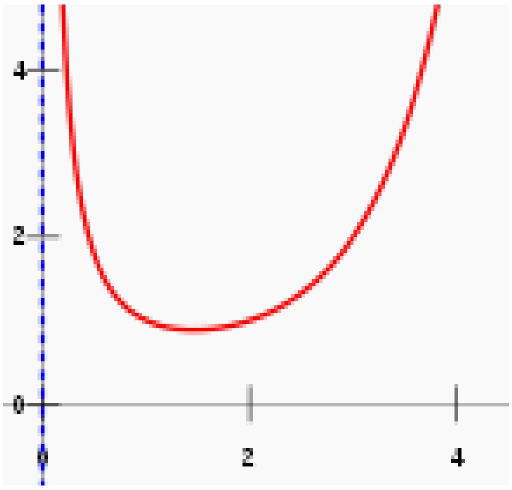
//disExponencial <-md> <muestra> <escala> <B> [-xeGm]

```
alias disExponencial {
        if (m isin $1) disExponencialM $2-
        if (d isin $1) disExponencialD $2-
        if (m isin $5) pp $rgb(0,200,0) 2 $3 $4 0 media(X)
}
```

4.3 Distribución normal: $N(\mu, \delta)$

En estadística y probabilidad se llama distribución normal a una de las distribuciones de probabilidad de variable continua que con más frecuencia aparece en fenómenos reales, como el peso de una persona, la altura de un piso, la envergadura de una libélula, etc

Previo definimos la función gamma de Euler en el dominio de espacios probabilísticos, ya que la función de densidad de la distribución normal la incluye en su definición.



$$\Gamma(x)=\int_0^{\infty} t^{x-1} e^{-t} dt$$

(Esta representación no ha sido generada por mIRC Scripting)

Función de densidad de una distribución normal ESTÁNDAR

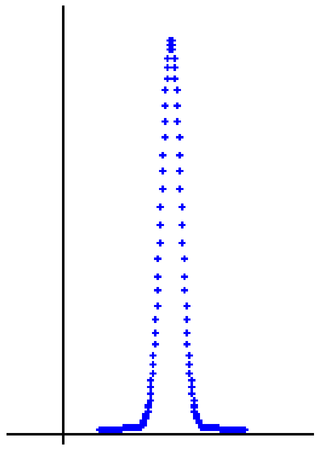
$$X \rightarrow N(0,1)$$

//fNormal <escala> <a;b>

```
alias fNormal {
    var %a = $gettok($2,1,59), %b = $calc($gettok($2,2,59) - $gettok($2,1,59)), %c
    while (%b >= %a) {
        %c = $calc($calc($sqrt(2.71) * $calc(2.71 ^ $calc(-1 * $calc(%a ^ 2)))) /
            $sqrt(6.2832))
        pp $rgb(0,0,255) 2 $1 %a $calc(%escalaY * %c)
        inc %a 0.1
    }
}
```

$$f(x) = \begin{cases} 0 \; si \; x \leqslant 0 \\ \dfrac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \; si \; x > 0 \end{cases}$$

//fGamma <E> <µ> <ძ> <a;b>



Ejemplo: //fGamma 3 15 3 5;30

Implementación:

```
alias fGamma {
	var %a = $gettok($4,1,59), %b = $calc($gettok($4,2,59) - $gettok($4,1,59)), %c
	while (%b >= %a) {
		%c = $calc($calc(2.71^ $calc($calc(-1 * $calc($calc(%a - $2) ^ 2)) / $calc(2 *
			$sqrt($3))))) / $calc($sqrt($3) * $sqrt($calc(2 * 3.1416))))
		pp $rgb(0,0,255) 2 $1 %a $calc(%escalaY * %c)
		inc %a 0.1
	}
}
```

Función de densidad de una distribución normal ESTÁNDAR

$$X \rightarrow N(0,1)$$

```
//fNormal <escala> <a;b>
alias fNormal {
	var %a = $gettok($2,1,59), %b = $calc($gettok($2,2,59) - $gettok($2,1,59)), %c
	while (%b >= %a) {
		%c = $calc($calc($sqrt(2.71) * $calc(2.71 ^ $calc(-1 * $calc(%a ^ 2)))) /
		$sqrt(6.2832))
		pp $rgb(0,0,255) 2 $1 %a $calc(%escalaY * %c)
		inc %a 0.1
	}
}
```

; Funciones para la construcción gráfica del programa

```
; //v
; abre ventana de trabajo o la limpia
alias v {
	if (!$window(@v)) window -apkB +d @v 5 5 1000 500
	else clear @v
	drawrect -rp @v $rgb(1,1,1) 1 0 0 1000 500
	drawrect -rp @v $rgb(1,1,1) 1 500 5 1 485
	drawrect -rp @v $rgb(1,1,1) 1 10 490 980 1
	drawpic @v
}
```

; //representar -<p/f>[e/o/u]

```
; -p: por puntos //representar -p <color $rgb> <tamaño> <escala> x1_fx1;x2_fx2;..;xN_fxN
; si se añade e se etiquetan los puntos: //representar -pe <color $rgb> <tamaño> <escala>
x1_fx1;x2_fx2;..;xN_fxN
; -f: por funcion //representar -f <color $rgb> <tamaño> <escala> a;b;(bi - ai)_f(xi)
; bi-ai representa cada que subintervalo (variación X) aplicar la función f en el intervalo [a,b]

; si se añade o (-foN / -p[e]o) se señalan los puntos de ordenada y abcisa correspondiente,
para -f se hace cada N veces el subintervalo
//representar -peo <color $rgb> <tamaño> <escala> x1_fx1;x2_fx2;..;xN_fxN => con etiqueta y
señales
//representar -fo3 <color $rgb> <tamaño> <escala> a;b;(bi - ai)_f(xi) => con señales cada 3
veces el subintervalo

; si se añade u se unen los puntos para la sensación de continuidad (se admite esta opción por
puntos xD)
//representar -p[e/o]u <color $rgb> <tamaño> <escala> x1_fx1;x2_fx2;..;xN_fxN
//representar -f[oN]u <color $rgb> <tamaño> <escala> a;b;(bi - ai)_f(xi)



alias representar {
  if ($left($1,2) == -p) { var %a = 1, %b = $numtok($5-,59)
    while (%b >= %a) { pp $2 $3 $4 $gettok($gettok($5-,%a,59),1,$asc(_))
$gettok($gettok($5-,%a,59),2,$asc(_)) $iif(e isin $1,( $+ $eval($gettok($gettok($5-,
%a,59),1,$asc(_))) $+ , $+ $eval($gettok($gettok($5-,%a,59),2,$asc(_))) $+ ),$chr(32))
      if (o isin $1) dibujarCoordenadas $2 $4 $gettok($gettok($5-,%a,59),1,$asc(_))
$gettok($gettok($5-,%a,59),2,$asc(_))
      if (u isin $1) && (%a > 1) drawline -rn @v $2 $calc($3 - 0) $calc(502 +
$calc($gettok($gettok($5-,%a,59),1,$asc(_)) * $4)) $calc(490 -
$calc($gettok($gettok($5-,%a,59),2,$asc(_)) * $4)) $calc(502 +
$calc($gettok($gettok($5-,$calc(%a - 1),59),1,$asc(_)) * $4)) $calc(490 -
$calc($gettok($gettok($5-,$calc(%a - 1),59),2,$asc(_)) * $4))
      inc %a } }
  elseif ($left($1,2) == -f) { var %a = 1, %b =
$calc($calc($gettok($gettok($5-,1,$asc(_)),2,59) -
$gettok($gettok($5-,1,$asc(_)),1,59)) / $gettok($gettok($5-,1,$asc(_)),3,59)), %x, %x_
    var %t = $right($left($1,$calc($pos($1,o,1)+1)),1), %t_ = 0
    while (%b >= %a) { %x = $calc($gettok($gettok($5-,1,$asc(_)),1,59) + $calc(%a *
$gettok($gettok($5-,1,$asc(_)),3,59)))
      %x_ = $calc($gettok($gettok($5-,1,$asc(_)),1,59) + $calc($calc(%a - 1) *
$gettok($gettok($5-,1,$asc(_)),3,59)))
      pp $2 $3 $4 %x $aplicarFuncion($gettok($5-,2,$asc(_)),x,%x)
      if (u isin $1) && (%a > 1) drawline -rn @v $2 $calc($3 - 0) $calc(502 + $calc(%x_ *
$4)) $calc(490 - $calc($aplicarFuncion($gettok($5-,2,$asc(_)),x,%x_) * $4) - 10)
$calc(502 + $calc(%x * $4)) $calc(490 - $calc($aplicarFuncion($gettok($5-,2,$asc(_)),x,
%x) * $4) - 10)
      if (o isin $1) && (%t isnum) && (%t_ <= %t) { if (%t_ == %t) { dibujarCoordenadas
$2 $4 %x $aplicarFuncion($gettok($5-,2,$asc(_)),x,%x)
        %t_ = 0 }
```

```
      else inc %t_ }
    inc %a } }
  drawpic @v }

; poner punto: //pp <color $rgb> <tamaño> <escala> <x> <y> [etiqueta]
alias pp { drawdot -rn @v $1 $2 $calc(502 + $calc($4 * $3)) $calc(490 - $calc($5 * $3))
  if ($6) drawtext -rn @v $1 Tahoma 10 $calc(502 + $calc($4 * $3) + 5) $calc(490 -
$calc($5 * $3) - 10) $6-
  drawpic @v }

; aplicar función en un punto: $aplicarFuncion(f,<x>,x)

alias aplicarFuncion return $calc($replace($replace($replace($1,sin(x),
$sin($3)),cos(x),$cos($3)),$2,$3))

; aplicar función de varias variables en un punto: $aplicarFuncionM(f,<x> <y> <z> ...
<t>,x y z ... t)

alias aplicarFuncionM { var %a = 1, %b = $numtok($2,32), %c = $1
  while (%b >= %a) { %c = $replace(%c,$gettok($2,%a,32),$gettok($3,%a,32))
    inc %a }
  return $calc(%c) }

; dibujar coordenadas de un punto
; dibujarCoordenadas <color> <escala> <x> <y>
alias dibujarCoordenadas { drawrect -rn @v $1 1 500 $calc(490 - $calc($4 * $2))
$calc($3 * $2) 1
  drawrect -rn @v $1 1 $calc(500 + $calc($3 * $2)) $calc(490 - $calc($4 * $2)) 1
$calc($4 * $2) }



; dibujar asíntotas
```

; asíntota vertical:

```
;// asintotaV <+p/-n/t> <color> <escala> <x> [-e]
; -e etiqueta la asíntota
; +p y -n se utilizan para colocar solo media asíntota (por encima del eje x +p, por debajo del
eje x -n)
; t imprime toda
alias asintotaV {
  drawrect -rn @v $2 2 $calc(500 + $calc($3 * $4)) $iif(($1 == +p) || ($1 == t),0,490) 0
  $iif($1 == t,1000,491)
  if ($5 == -e) drawtext -rn @v $2 Tahoma 10 $calc(505 + $calc($3 * $4)) $iif($1 ==
  +p,5,495) x= $+ $3
  drawpic @v
}
```

; asíntota horizontal:

; //asintotaH <+p/-n/t> <color> <escala> <y> [-e]
; misma sintaxis que para asintotaV

```
alias asintotaH {
  drawrect -rn @v $2 2 $iif($1 == +p,500,0) $calc(490 - $calc($3 * $4)) $iif($1 ==
  -n,500,1000) 1
  if ($5 == -e) drawtext -rn @v $2 Tahoma 10 $iif($1 == +p, $calc($window(@v).w - 30),
  5) $calc(490 - $calc($3 * $4)) y= $+ $4
  drawpic @v
}
```

; //representarDerivada <color> <F(x)> <escala> <a;b>

; si la función no es derivable en un punto no se representa, tendrán que hacerse manualmente los puntos de discontinuidad (obvio) con //representar

```
alias representarDerivada { var %a = $gettok($4,1,59), %b = $gettok($4,2,59), %f'
 while (%b >= %a) { %f' = $calc($calc($aplicarFuncion($2,x,$calc(%a + 0.1)) -
$aplicarFuncion($2,x,%a)) / 0.1)
  pp $1 2 $3 %a %f'
  inc %a 0.1 } }
```