Proyecto:
Implementación de nuevas distribuciones estadísticas en R

Integrantes del proyecto

Juan Diego Suárez

Karina Garay

Maria Camila Mena

Manuel Gutiérrez

Freddy Hernández

Motivación

Paquetes

https://ousuga.github.io/RelDists/
https://fhernanb.github.io/RealDists/
https://fhernanb.github.io/DiscreteDists/
https://fhernanb.github.io/ZeroOneDists/

Distribuciones implementadas

- Juan Diego:
  Unit Half-Logistic Geometric (UHLG)
  Unit-Power Half-Normal (UPHN)
- Karina:
  Beta Rectangular (BER)
  Beta Rectangular parametrizada (BER2)
- Manuel:
  Exponentiated XLindley (EXL)
  Two-parameter Chris-Jerry (CJ2)
- Maria Camila:
  Discrete power-Ailamujia (DsPA)

Unit Half-Logistic Geometric (UHLG) - Juan Diego

La distribución **Unit Half-Logistic Geometric (UHLG)** es una familia de distribuciones definidas en el intervalo **(0,1)**, desarrollada para modelar datos acotados en este rango. Su construcción combina la distribución **Half-Logistic** con una estructura geométrica, proporcionando flexibilidad en la modelación de datos con tasas de supervivencia, fracciones o proporciones.

El artículo que propone la distribución es:

axioms

Article

A Unit Half-Logistic Geometric Distribution and Its Application in Insurance

Ahmed T. Ramadan [1,†], Ahlam H. Tolba [2,*,†] and Beih S. El-Desouky [2,†]

1 Department of Basic Sciences, High Raya Institute, Damietta 34511, Egypt
2 Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 33516, Egypt
* Correspondence: a.hamdy6@yahoo.com
† These authors contributed equally to this work.

Unit Half-Logistic Geometric (UHLG) - Juan Diego

¿Para qué se usa?

Esta distribución es útil en múltiples áreas, especialmente cuando los datos están restringidos entre **0 y 1**. Algunas aplicaciones incluyen:

- **Supervivencia y confiabilidad**: Modelar la tasa de fallas de productos o la duración de eventos en el tiempo.
- **Ciencias actuariales y seguros**: Estimación de riesgos y pérdidas en modelos de seguros.
- **Economía y finanzas**: Modelar proporciones como retorno sobre activos, tasas de impuestos o indicadores financieros.

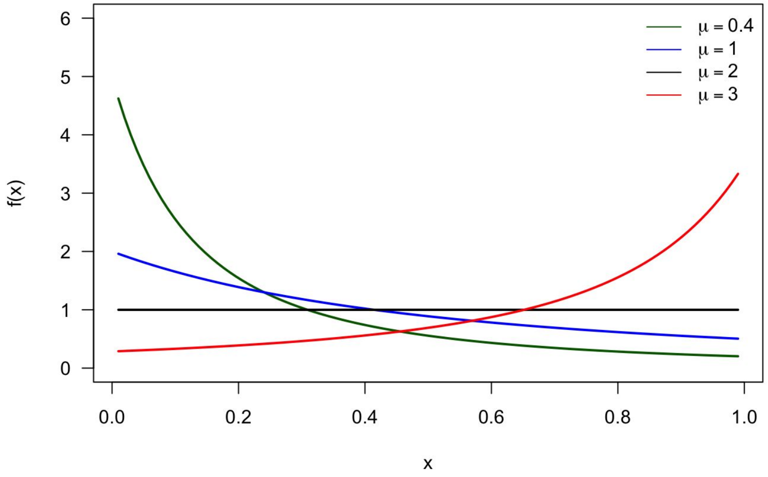
Unit Half-Logistic Geometric (UHLG) - Juan Diego

La función de densidad de la distribución es

$$f(x|\mu) = \frac{2\mu}{(\mu+(2-\mu)x)^2} \quad \text{for } 0 < x < 1 \text{ and } \mu > 0.$$

A continuación la forma de la densidad

para algunas combinaciones de los parámetros.



Unit Half-Logistic Geometric (UHLG) - Juan Diego

¿Cómo se implementó la distribución UHLG en R?

Para implementar la distribución se crearon 5 funciones:

`UHLG`: Define la distribución con un enlace log, lo que permite estimar el parámetro μ.

`dUHLG`: Calcula la función de densidad de probabilidad (PDF) de la distribución en un punto x.

`pUHLG`: Calcula la función de distribución acumulada (CDF), es decir, la probabilidad de que la variable aleatoria sea menor o igual a q.

`qUHLG`: Calcula el cuantil de la distribución para una probabilidad p.

`rDLD`: Genera valores aleatorios siguiendo la distribución.

Unit Half-Logistic Geometric (UHLG) - Juan Diego

```r
# Example 1
# Generating some random values with
# known mu
y <- rUHLG(n=500, mu=7)

# Fitting the model
library(gamlss)
mod1 <- gamlss(y~1, family=UHLG,
               control=gamlss.control(n.cyc=500, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod1, what="mu"))
#> (Intercept)
#>    7.198641
```

Unit Half-Logistic Geometric (UHLG) - Juan Diego

```
# Example 2
# Generating random values under some model

# A function to simulate a data set with Y ~ UHLG
gendat <- function(n) {
  x1 <- runif(n, min=0.4, max=0.6)
  x2 <- runif(n, min=0.4, max=0.6)
  mu    <- exp(-0.5 + 3*x1 - 2.5*x2)
  y <- rUHLG(n=n, mu=mu)
  data.frame(y=y, x1=x1, x2=x2)
}

datos <- gendat(n=5000)
```

Unit Half-Logistic Geometric (UHLG) - Juan Diego

```r
mod2 <- gamlss(y~x1+x2,
               family=UHLG, data=datos,
               control=gamlss.control(n.cyc=500, trace=TRUE))
#> GAMLSS-RS iteration 1: Global Deviance = -1529.63
#> GAMLSS-RS iteration 2: Global Deviance = -1529.63
summary(mod2)
#> ------------------------------------------------------------------
#> Mu link function:  log
#> Mu Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -0.4923     0.3012  -1.634    0.102
#> x1            2.9557     0.4188   7.057 1.94e-12 ***
#> x2           -2.4772     0.4189  -5.913 3.58e-09 ***
#> ---
#> Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#>
#> ------------------------------------------------------------------
```

Beta Rectangular - Karina

La distribución **Beta Rectangular** es una mezcla de una distribución Beta con una distribución Uniforme y, por lo tanto, es un modelo de mezcla Beta finita. Esta distribución permite distintas cantidades de dispersión y una mayor probabilidad de que se produzcan sucesos más extremos en la cola.

El artículo que propone la distribución es:

Bayesian Analysis (2012) **7**, Number 4, pp. 841–866

A New Robust Regression Model for Proportions

Cristian L. Bayes*, Jorge L. Bazán† and Catalina García‡

Abstract. A new regression model for proportions is presented by considering the Beta rectangular distribution proposed by Hahn (2008). This new model includes the Beta regression model introduced by Ferrari and Cribari-Neto (2004) and the

Beta Rectangular - Karina

¿Para qué se usa?

Esta distribución brinda cierta flexibilidad adicional en comparación con la distribución Beta, permitiendo cantidades variables de dispersión y una mayor probabilidad de eventos de área de cola más extremos.

Algunas aplicaciones incluyen:

- Predicción del porcentaje de grasa corporal (Bfat) de cada atleta teniendo en cuenta su masa corporal magra.
- Estudiar la influencia del Índice de Desarrollo Humano (IDH) de un distrito electoral en la proporción de votos en blanco en las elecciones generales peruanas de 2006.

Beta Rectangular - Karina

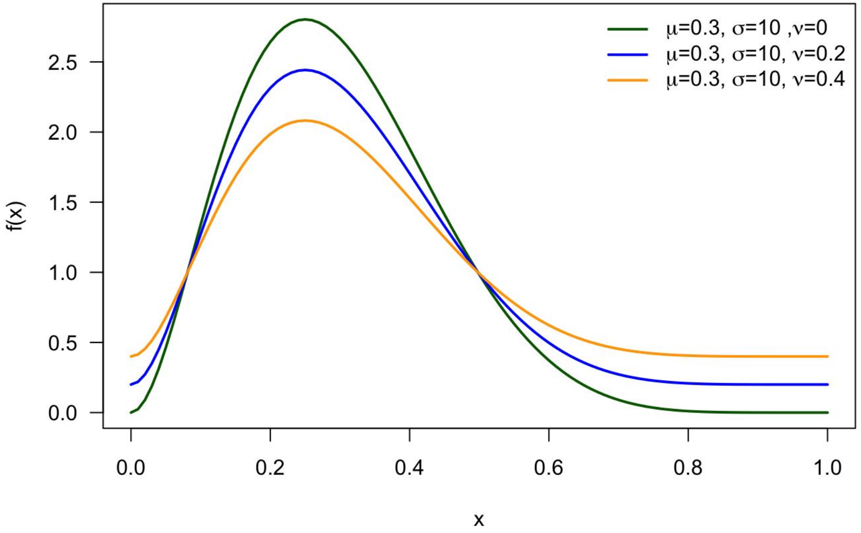
La función de densidad de la distribución es

$$f(x|\mu, \sigma, \nu) = \nu + (1 - \nu)b(x|\mu, \sigma)$$

for $0 < x < 1$, $0 < \mu < 1$, $\sigma > 0$ and $0 < \nu < 1$.

A continuación la forma de la densidad

para algunas combinaciones de los parámetros.



Beta Rectangular - Karina

¿Cómo se implementó la distribución Beta Rectangular en R?

Para implementar la distribución se crearon 5 funciones:

`BER`: Define la distribución con un enlace logit y log, lo que permite estimar los parámetros μ, σ y ν.

`dBER`: Calcula la función de densidad de probabilidad (PDF) de la distribución en un punto x.

`pBER`: Calcula la función de distribución acumulada (CDF), es decir, la probabilidad de que la variable aleatoria sea menor o igual a q.

`qBER`: Calcula el cuantil de la distribución para una probabilidad p.

`rBER`: Genera valores aleatorios siguiendo la distribución.

Beta Rectangular - Karina

```r
# Example 1
# Generating some random values with
# known mu and sigma
y <- rBER(n=500, mu=0.5, sigma=10, nu=0.5)

# Fitting the model
library(gamlss)
mod1 <- gamlss(y~1, family=BER)
```

Beta Rectangular - Karina

```r
# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
inv_logit <- function(x) 1/(1 + exp(-x))

inv_logit(coef(mod1, what="mu"))
#> (Intercept)
#>   0.5208244
exp(coef(mod1, what="sigma"))
#> (Intercept)
#>    10.35817
inv_logit(coef(mod1, what="nu"))
#> (Intercept)
#>   0.5628579
```

Beta Rectangular - Karina

```r
# Example 2
# Generating random values under some model

# A function to simulate a data set with Y ~ BER
gendat <- function(n) {
  x1 <- runif(n, min=0.4, max=0.6)
  x2 <- runif(n, min=0.4, max=0.6)
  x3 <- runif(n, min=0.4, max=0.6)
  mu    <- inv_logit(-0.5 + 1*x1)
  sigma <- exp(-1 + 4.8*x2)
  nu    <- inv_logit(-1 + 0.5*x3)
  y <- rBER(n=n, mu=mu, sigma=sigma, nu=nu)
  data.frame(y=y, x1=x1, x2=x2, x3=x3)
}

set.seed(1234)
datos <- gendat(n=500)
```

Beta Rectangular - Karina

```
mod2 <- gamlss(y~x1, sigma.fo=~x2, nu.fo=~x3,
               family=BER, data=datos,
               control=gamlss.control(n.cyc=500, trace=TRUE))
summary(mod2)
```

Beta Rectangular - Karina

```
#> ------------------------------------------------------------------
#> Mu link function:  logit
#> Mu Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -1.4240     0.4567  -3.118  0.00193 **
#> x1            2.7233     0.8992   3.029  0.00258 **
#> ---
#> Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#>
#> ------------------------------------------------------------------
#> Sigma link function:  log
#> Sigma Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -1.2053     0.7012  -1.719 0.086252 .
#> x2            5.3047     1.4059   3.773 0.000181 ***
#> ---
#> Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#>
#> ------------------------------------------------------------------
#> Nu link function:  logit
#> Nu Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    3.724      3.187   1.168    0.243
#> x3           -10.308      6.640  -1.552    0.121
#>
#> ------------------------------------------------------------------
```

The DsPA distribution - Maria Camila

La distribución discreta power-Ailamujia (DsPA distribution) es una nueva familia de distribuciones derivada mediante la **discretización de la función de supervivencia**.

Sus dos parámetros pueden estimarse por máxima verosimilitud y cuenta con propiedades útiles en análisis de confiabilidad, simulación y datos reales. Surge como una extensión de otras distribuciones discretas previamente desarrolladas, como la Poisson–Ailamujia, la Burr–XII discreta y la Rayleigh discreta.

AIMS *Mathematics*

AIMS Mathematics, 7(5): 8344–8360.
DOI: 10.3934/math.2022465
Received: 12 January 2022
Revised: 07 February 2022
Accepted: 22 February 2022
Published: 28 February 2022

http://www.aimspress.com/journal/Math

Research article

The discrete power-Ailamujia distribution: properties, inference, and applications

Abdulaziz S. Alghamdi[1], Muhammad Ahsan-ul-Haq[2,3], Ayesha Babar[4], Hassan M. Aljohani[5] and Ahmed Z. Afify[6,*]

The DsPA distribution - Maria Camila

¿Cuál es la utilidad de esta distribución?

Su objetivo es ofrecer mayor flexibilidad al **modelar datos de conteo** y conjunto de datos sobredispersos como subdispersos.

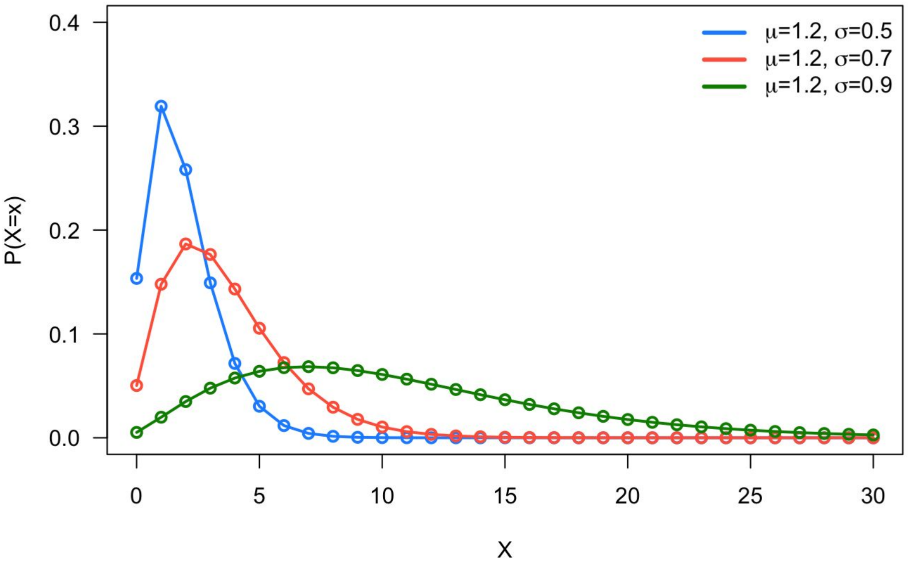
Función de masa

$$f(x|\mu,\sigma) = (\sigma^{x})^{\mu}(1 - x^{\mu}\log(\lambda)) - (\sigma^{(x+1)})^{\mu}(1 - (x+1)^{\mu}\log(\lambda))$$

$$x = 0, 1, 2, \ldots, \quad \mu > 0, \quad 0 < \sigma < 1$$

The DsPA distribution - Maria Camila

Formas de la distribución



The DsPA distribution - Maria Camila

¿Cómo se implementó la distribución en R?

Para implementar la distribución se crearon 5 funciones:

`DsPA`: Define la distribución con un enlace log y logit, lo que permite estimar los parámetros μ y σ.

`dDsPA`: Calcula la probabilidad (PDF) de la distribución en un punto x.

`pDsPA`: Calcula la función de distribución acumulada (CDF), es decir, la probabilidad de que la variable aleatoria sea menor o igual a q.

`qDsPA`: Calcula el cuantil de la distribución para una probabilidad p.

`rDsPA`: Genera valores aleatorios de la distribución.

The DsPA distribution - Maria Camila

```r
# Example 1
# Generating some random values with
# known mu and sigma
y <- rDsPA(n=500, mu=1.2, sigma=0.5)

# Fitting the model
library(gamlss)
mod1 <- gamlss(y~1, family=DsPA,
               control=gamlss.control(n.cyc=500, trace=FALSE))

# Extracting the fitted values for mu and sigma
# using the inverse link function
inv_logit <- function(x) 1/(1 + exp(-x))

exp(coef(mod1, what="mu"))
#> (Intercept)
#>    1.262877
inv_logit(coef(mod1, what="sigma"))
#> (Intercept)
#>   0.5182921
```

The DsPA distribution - Maria Camila

```
# Example 2
# Generating random values under some model

# A function to simulate a data set with Y ~ GGEO
gendat <- function(n) {
  x1 <- runif(n)
  x2 <- runif(n)
  x3 <- runif(n)
  x4 <- runif(n)
  mu    <- exp(1 + 1.2*x1 + 0.2*x2)
  sigma <- inv_logit(2 + 1.5*x3 + 1.5*x4)
  y <- rDsPA(n=n, mu=mu, sigma=sigma)
  data.frame(y=y, x1=x1, x2=x2, x3=x3, x4=x4)
}

set.seed(16494786)
datos <- gendat(n=500)
```

The DsPA distribution - Maria Camila

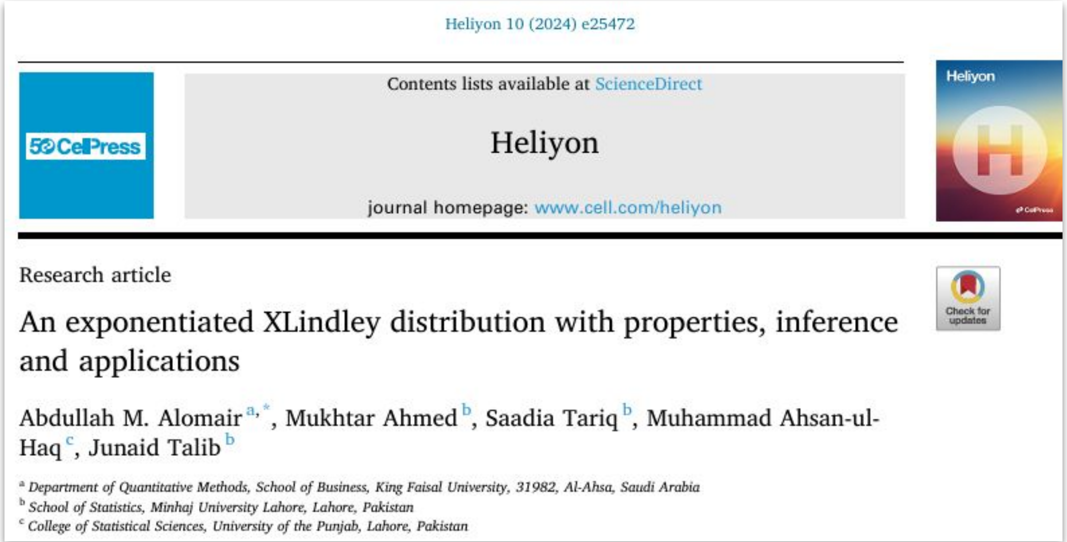
```
mod2 <- gamlss(y~x1+x2, sigma.fo=~x3+x4, family=DsPA, data=datos,
               control=gamlss.control(n.cyc=500, trace=TRUE))

summary(mod2)
```

```
#> ------------------------------------------------------------------
#> Mu link function:  log
#> Mu Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.95811    0.02332   41.09  < 2e-16 ***
#> x1           1.18156    0.04467   26.45  < 2e-16 ***
#> x2           0.22221    0.03637    6.11 2.01e-09 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> ------------------------------------------------------------------
#> Sigma link function:  logit
#> Sigma Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   1.9423     0.1264  15.370   <2e-16 ***
#> x3            1.3462     0.1583   8.503   <2e-16 ***
#> x4            1.4494     0.1667   8.697   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> ------------------------------------------------------------------
```

Exponentiated XLindley (EXL) - Manuel

La distribución **Exponentiated XLindley (EXL)** es una familia de distribuciones continuas definida en el intervalo $(0,\infty)$, desarrollada como una generalización flexible del modelo XLindley. Su construcción se basa en aplicar la técnica de exponenciación a la función de distribución acumulada original, lo cual permite incorporar un parámetro adicional que modifica la forma de la densidad y la función de riesgo. El artículo que propone la distribución es:


Heliyon 10 (2024) e25472

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

An exponentiated XLindley distribution with properties, inference and applications

Abdullah M. Alomair [a,*], Mukhtar Ahmed [b], Saadia Tariq [b], Muhammad Ahsan-ul-Haq [c], Junaid Talib [b]

[a] Department of Quantitative Methods, School of Business, King Faisal University, 31982, Al-Ahsa, Saudi Arabia
[b] School of Statistics, Minhaj University Lahore, Lahore, Pakistan
[c] College of Statistical Sciences, University of the Punjab, Lahore, Pakistan

Exponentiated XLindley (EXL) - Manuel

¿De donde viene?

La distribución **Exponentiated XLindley (EXL)** es una extensión de la distribución XLindley, que a su vez generaliza la distribución de Lindley. Esta progresión añade parámetros que incrementan la flexibilidad del modelo para ajustarse a diversos conjuntos de datos en análisis de confiabilidad y estudios de supervivencia.

Distribución Lindley

$$f(x) = \frac{\theta^2}{\theta+1}(1+x)\, \mathrm{e}^{-\theta x}, \quad x > 0, \theta > 0$$

Distribución xLindley

$$f(x;\alpha,\beta) = \frac{\alpha\beta^2}{(\alpha+\beta)^2}(1+x)e^{-\beta x}, \quad x > 0,\ \alpha,\beta > 0$$

Distribución EXL

$$f(x) = \frac{\sigma\mu^2(2+\mu+x)\exp(-\mu x)}{(1+\mu)^2}\left[1-\left(1+\frac{\mu x}{(1+\mu)^2}\right)\exp(-\mu x)\right]^{\sigma-1}$$

for $x \geq 0$, $\mu \geq 0$ and $\sigma \geq 0$.

Exponentiated XLindley (EXL) - Manuel

¿Para qué se usa?

La distribución **Exponentiated XLindley (EXL)** es una herramienta estadística versátil utilizada en diversas áreas para modelar datos positivos. Algunas de sus aplicaciones incluyen:

- **Análisis de supervivencia y confiabilidad**: Modela tiempos de falla de productos o duración de eventos en el tiempo, gracias a su capacidad para representar tasas de riesgo variables.

- **Meteorología**: Analiza datos de precipitaciones, permitiendo una mejor comprensión de patrones climáticos y eventos extremos.

- **Salud pública**: Estudia tasas de mortalidad, como en análisis relacionados con la COVID-19, proporcionando insights sobre la evolución de pandemias.

- **Ingeniería**: Evalúa tiempos de falla en sistemas reparables, contribuyendo al diseño y mantenimiento de sistemas más eficientes.

Exponentiated XLindley (EXL) - Manuel

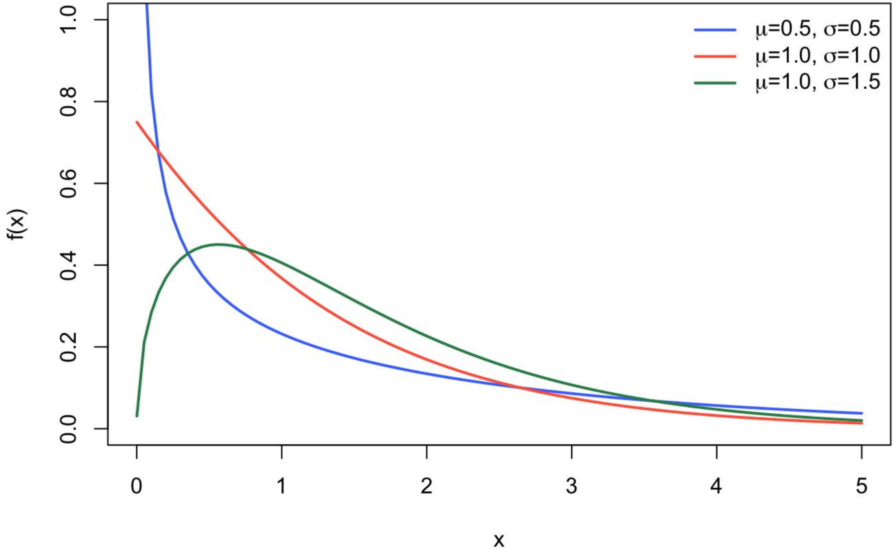
La función de densidad de la distribución es

$$f(x) = \frac{\sigma\mu^2(2+\mu+x)\exp(-\mu x)}{(1+\mu)^2}\left[1-\left(1+\frac{\mu x}{(1+\mu)^2}\right)\exp(-\mu x)\right]^{\sigma-1}$$

for $x \geq 0$, $\mu \geq 0$ and $\sigma \geq 0$.

A continuación la forma de la densidad

para algunas combinaciones de los parámetros.



Exponentiated XLindley (EXL) - Manuel

¿Cómo se implementó la distribución EXL en R?

Para implementar la distribución se crearon 5 funciones:

`EXL`: Define la distribución con enlaces log, lo que permite estimar el parámetro μ ó σ.

`dEXL`: Calcula la función de densidad de probabilidad (PDF) de la distribución en un punto x.

`pEXL`: Calcula la función de distribución acumulada (CDF), es decir, la probabilidad de que la variable aleatoria sea menor o igual a un valor q dado.

`qEXL`: Calcula el cuantil de la distribución para una probabilidad p.

`rEXL`: Genera valores aleatorios siguiendo la distribución.

Exponentiated XLindley (EXL) - Manuel

```r
# Example 1
# Generating some random values with
# known mu and sigma
require(lamW)
y <- rEXL(n=1000, mu=0.75, sigma=1.3)

# Fitting the model
require(gamlss)
mod <- gamlss(y~1, sigma.fo=~1, family="EXL")


# Extracting the fitted values for mu and sigma
# using the inverse link function
exp(coef(mod, what="mu"))
# (Intercept)
# 0.7747697
exp(coef(mod, what="sigma"))
# (Intercept)
# 1.295132
```

Exponentiated XLindley (EXL) - Manuel

```
# Example 2
# Generating random values under some model
n <- 1000
x1 <- runif(n)
x2 <- runif(n)
mu <- exp(1.45 - 3 * x1)
sigma <- exp(2 - 1.5 * x2)
y <- rEXL(n=n, mu=mu, sigma=sigma)

mod <- gamlss(y~x1, sigma.fo=~x2, family=EXL)

coef(mod, what="mu")
# (Intercept)          x1
# 1.459424    -3.007707
coef(mod, what="sigma")
# (Intercept)          x2
# 2.008845    -1.402916
```

Exponentiated XLindley (EXL) - Manuel

```
summary(mod)
# ------------------------------------------------------------------
#    Mu link function:   log
# Mu Coefficients:
#    Estimate Std. Error t value Pr(>|t|)
# (Intercept)  1.45942    0.04174   34.97   <2e-16
# x1          -3.00771    0.05340  -56.33   <2e-16
#

# ------------------------------------------------------------------
#    Sigma link function:   log
# Sigma Coefficients:
#    Estimate Std. Error t value Pr(>|t|)
# (Intercept)    2.0088     0.0904   22.22   <2e-16
# x2            -1.4029     0.1130  -12.41   <2e-16
#
```

Preguntas

Gracias