

UNIVERSIDAD NACIONAL DE ROSARIO
Facultad de Ciencias Económicas y Estadística



"Metropolis-Hastings"

Estadística Bayesiana - Trabajo Práctico N°2

Alumnas: Agustina Mac Kay, Ailén Salas y Rocio Canteros

Año 2024

```
#Librerías  
library(ggplot2)  
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
set.seed(394)
```

```
cant_saltos <- 0

#Punto 1:
sample_mh <- function(d_objetivo, r_propuesta, d_propuesta, p_inicial, n) {
  muestras <- matrix(nrow = n, ncol = length(p_inicial))
  muestras[1, ] <- p_inicial

  for(i in 2:n) {
    p_actual <- muestras[i-1,]
    p_nuevo <- r_propuesta(p_actual)

    f_nuevo <- d_objetivo(p_nuevo)
    f_actual <- d_objetivo(p_actual)

    q_actual <- d_propuesta(p_actual, mean = p_nuevo)
    q_nuevo <- d_propuesta(p_nuevo, mean = p_actual)

    alpha <- min(1, (f_nuevo/f_actual)*(q_actual/q_nuevo))
    aceptar <- rbinom(1, 1, alpha)

    if(aceptar) {
      muestras[i,] <- p_nuevo
      cant_saltos <- cant_saltos + 1
    } else {
      muestras[i,] <- p_actual
    }
  }

  if (ncol(muestras) == 1) {
    muestras <- as.vector(muestras)
  }
  return(list(muestras=muestras,cant_saltos=cant_saltos))
}
```

```
#Punto 2
# Crear grilla para los valores de "x"
grid_n <- 200
x_grid <- seq(0, 1, length.out = grid_n)
```

```

kumaraswamy <- function(x, a, b){
  a*b*(x^(a-1))*((1-(x^a))^(b-1))
}

a <- c(0.2, 3, 4, 10, 1)
b <- c(0.2, 3, 9, 5, 7)

#Creamos un data frame para graficar la distribución de Kumaraswamy:

data1 <- data.frame(
  Funcion = as.factor(rep(1:5, each = grid_n)),
  Densidad = numeric(5 * grid_n),
  Grilla = rep(x_grid, times = 5)
)

#Completamos el data frame con las densidades:
for(i in 1:5) {
  indices <- seq(from = 1 + (i - 1) * 200, to = 200 + (i - 1) * 200)
  data1$Densidad[indices] <- kumaraswamy(x_grid, a[i], b[i])
}

# Definimos etiquetas personalizadas para cada función
# etiq <- c(
#   "1" = "alpha = 0.2 , beta = 0.2",
#   "2" = "alpha = 3 , beta = 3",
#   "3" = "alpha = 4 , beta = 9",
#   "4" = "alpha = 10, beta = 5",
#   "5" = "alpha = 1, beta = 7"
# )

# Quiero que se vean alpha y beta como simbolos pero no me sale
# etiq <- c(
#   "1" = expression(alpha ~ " = 0.2," ~ beta ~ "= 0.2"),
#   "2" = expression(alpha ~ " = 3," ~ beta ~ "= 3"),
#   "3" = expression(alpha ~ " = 4," ~ beta ~ "= 9"),
#   "4" = expression(alpha ~ " = 10," ~ beta ~ "= 5"),
#   "5" = expression(alpha ~ " = 1," ~ beta ~ "= 7")
# )

# tibble(x = c(1,1,1,1),
#         y = c(1,1,1,1),
#         f = factor(c("a","b","c","d"),
#                   labels = c(expression(alpha),
#                               expression(beta),
#                               expression(gamma),
#                               expression(delta)))) %>%
#   ggplot() +
#   geom_point(aes(x=x, y=y)) +
#   facet_wrap(~f, labeller = label_parsed)

levels(data1$Funcion) <- c(

```

```

expression(alpha ~ " = 0.2," ~ beta ~ "= 0.2"),
expression(alpha ~ " = 3," ~ beta ~ "= 3"),
expression(alpha ~ " = 4," ~ beta ~ "= 9"),
expression(alpha ~ " = 10," ~ beta ~ "= 5"),
expression(alpha ~ " = 1," ~ beta ~ "= 7")
)
ggplot(data = data1, aes(x = Grilla, y = Densidad)) +
  geom_line(size = 0.55) +
  facet_wrap(~Funcion, labeller = label_parsed) +
  theme_bw() +
  labs(x = "x",
       caption = "Gráfico 1: distribución Kumaraswamy con distintos parámetros") +
  theme(
    strip.background = element_rect(fill = "olivedrab3"),
    plot.caption = element_text(hjust = 0.5)
  )

```

Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.

Warning: Please use 'linewidth' instead.

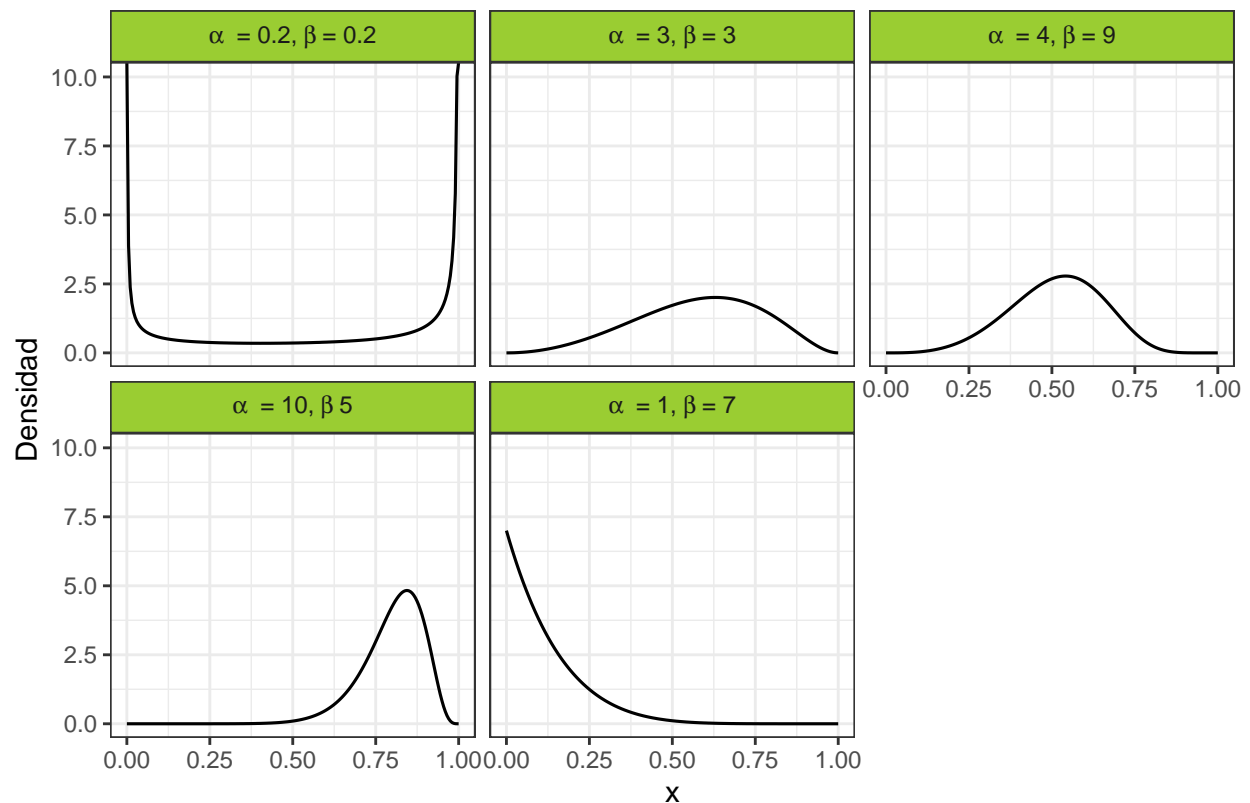


Gráfico 1: distribución Kumaraswamy con distintos parámetros

```

#Punto 3
concentracion_3 <- c(4,10,20)
tabla_3 <- data.frame (
  concentracion = rep(concentracion_3, each = 5000),

```

```

    muestra = numeric(15000)
  )
tasa_3 <- numeric(3)

for (i in 1:3) {
  #Funciones a usar
  d_objetivo <- function(x) kumaraswamy(x, 6, 2)
  d_propuesta <- function(x, mean) dbeta(x, shape1 = mean * concentracion_3[i], shape2 = (1-mean) * concentracion_3[i])
  r_propuesta <- function(x) rbeta(1, shape1 = x * concentracion_3[i], shape2 = (1-x) * concentracion_3[i])
  #Donde x hace referencia a mu
  n_3 <- 5000
  p_inicial <- rbeta(1,shape1=2,shape2=2)
  funcion_3 <- sample_mh(d_objetivo, r_propuesta, d_propuesta, p_inicial, n_3)
  indices <- seq(from = 1 + (i - 1) * 5000, to = 5000 + (i - 1) * 5000)
  tabla_3$muestra[indices] <- funcion_3$muestras

  tasa_3[i] <- funcion_3$cant_saltos/n_3
}

# Agrego una columna con el orden de las muestras
tabla_3$orden <- rep(1:5000, times = 3)
tabla_3$concentracion <- as.factor(tabla_3$concentracion)

# Gráfico con múltiples líneas
ggplot(data = tabla_3, aes(x = orden, y = muestra, group = concentracion, color = concentracion)) +
  geom_line() +
  theme_bw() +
  labs(x = "x", y = "y", color = "Group") +
  theme(legend.position = "top")

```

