

# Entrega 1: Creación de un Estimador de la media poblacional

Daniel Aramburu, Guillermo Palomo, Jorge Salas y Marc Pastor

9 de Octubre de 2020

## 1. Introducción

El objetivo de este trabajo es crear un Estadístico para estimar la media poblacional de una distribución continua determinada (que no sea la Distribución Normal ni la Exponencial) y comparar su rendimiento con la media muestral (que es el estadístico más usado para aproximar la media poblacional). En nuestro caso hemos optado por la distribución Beta (con  $\alpha$  y  $\beta = 2$ ) debido a que su función de densidad es simétrica y centrada, y nos parece que es sencillo encontrar un Estadístico fiable para aproximar su media poblacional.

## 2. Distribución Beta

La Distribución Beta es una distribución continua que depende de dos parámetros ( $\alpha$  y  $\beta$ ) y que toma valores en el intervalo  $[0,1]$ . Debido a que solo está definida en  $[0,1]$  es una distribución muy usada para modelizar la probabilidad de que ocurra un evento, aunque también es usada para describir datos empíricos (debido a la variedad de formas que puede adoptar en función de los valores que tomen sus parámetros) y para modelar la fiabilidad de un sistema. Su función de densidad es distinta de cero solo cuando  $0 < x < 1$  y es la siguiente:

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

También se puede escribir como:

$$f(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

Dónde la función Beta es la siguiente:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha) \Gamma(\beta)}{\Gamma(\alpha + \beta)} = \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx$$

### Propiedades de la función Beta

La función beta cumple las siguientes propiedades (las usaremos para encontrar su función generadora de momentos, etc):

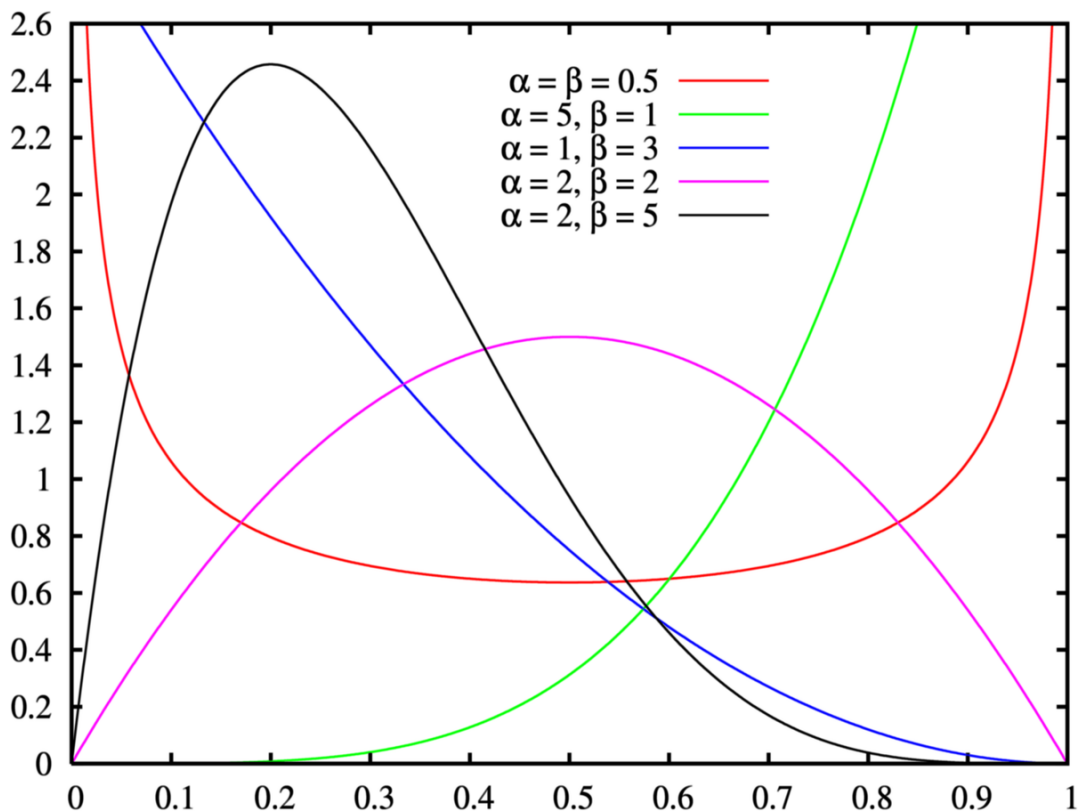
$$B(\alpha, \beta) = B(\beta, \alpha)$$

$$B(\alpha, 1) = \frac{1}{\alpha}$$

$$B(\alpha + 1, \beta) = \frac{\alpha}{(\alpha + \beta)} B(\alpha, \beta)$$

### 2.0 Forma de la distribución Beta

Esta distribución adopta formas muy diversas, en función de los valores de sus parámetros, por ello se utiliza mucho para modelar datos de manera empírica (es decir, observando la forma de la distribución de los datos, se intenta aproximar por la distribución beta con la forma más parecida)



## 2.1 Momentos

## 2.2 Función Generadora de Momentos

# 3. Creación del Estadístico para estimar la Media Poblacional

En primer lugar hemos creado la función *muestreo*. Esta función es la base de toda la práctica, ya que nos permite crear muestras aleatorias, calcular nuestro Estadístico, y la media muestral en cada una de esas 40 muestras, etc.

```
library(tidyverse)
library(e1071)
library(moments)
library(gt)
library(ggpubr)
library(ggplotify)
library(grid)

muestreo <- function(alpha, beta){
  lista <- list()
  tabla_muestras <- matrix(ncol = 40, nrow = 10) %>% as.data.frame()
  lista[[6]] <- matrix(ncol = 40, nrow = 1) %>% as.data.frame()
  media_poblacional <- alpha / (alpha + beta)
  varianza_poblacional <- alpha * beta / (((alpha + beta) ^ 2) * (alpha + beta + 1))
  for(j in 1:40){
    set.seed(j)
    tabla_muestras[, j] <- rbeta(10, shape1 = alpha, shape2 = beta)
    colnames(tabla_muestras)[j] <- as.numeric(gsub("V", "",
                                                    colnames(tabla_muestras)[j]))
    colnames(lista[[6]])[j] <- as.numeric(gsub("V", "",
                                                    colnames(tabla_muestras)[j]))
    lista[[6]][, j] <- 0.5*(quantile(tabla_muestras[, j],
                                    probs = c(0.6)) +
                           quantile(tabla_muestras[, j],
                                    probs = c(0.4)))
  }
  lista[[1]] <- tabla_muestras
  lista[[2]] <- media_poblacional
  lista[[3]] <- varianza_poblacional
  lista[[4]] <- data.frame(Muestra = names(colMeans(tabla_muestras)),
                           mediamuestral = unname(colMeans(tabla_muestras)))
  lista[[5]] <- data.frame(Estimador = "Media Muestral",
                           Media = mean(lista[[4]][, 2]),
                           Mediana = median(lista[[4]][, 2]),
                           SD = sd(lista[[4]][, 2]),
                           IQR = IQR(lista[[4]][, 2]),
                           MAD = mad(lista[[4]][, 2]),
                           Curtosis = moments::kurtosis(lista[[4]][, 2]),
                           Asimetría = e1071::skewness(lista[[4]][, 2]))
  lista[[6]] <- gather(as.data.frame(lista[[6]]))
  colnames(lista[[6]]) <- c("Muestra", "Estadístico")
  lista[[7]] <- data.frame(Estimador = "Estadístico",
                           Media = mean(lista[[6]][, 2]),
                           Mediana = median(lista[[6]][, 2]),
                           SD = sd(lista[[6]][, 2]),
```

```

        IQR = IQR(lista[[6]][, 2]),
        MAD = mad(lista[[6]][, 2]),
        Curtosis = moments::kurtosis(lista[[6]][, 2]),
        Asimetría = e1071::skewness(lista[[6]][, 2]))
lista[[8]] <- rbind(lista[[5]], lista[[7]])
names(lista) <- c("tabla_muestras", "media_poblacional",
                 "varianza_poblacional", "media_muestral",
                 "medidas_media_muestral", "estadistico",
                 "medidas_estadistico", "tabla_comparacion")

    lista
}

resultados <- muestreo(alpha = 2, beta = 2)

datosgrafico <- inner_join(resultados$media_muestral , resultados$estadistico)

```

## 4. Evaluación de nuestro Estadístico

4.1 Tabla de Medidas (falta pasar a formato leible por un pdf, es decir eliminar lo de gt)

## 4.2 Histogramas

```
grafico_estadistico_histogram <- ggplot(data = datosgrafico,
                                         aes(x = Estadístico)) +
  geom_histogram(fill = "lightblue", binwidth = 0.05, color = "black") +
  ggtitle("Histograma de nuestro Estadístico") +
  geom_vline(aes(xintercept = resultados[[7]]$Media, col = "Estimador"),
            linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = resultados$media_poblacional,
                  col = "Media Poblacional"), size = 1) +
  scale_color_manual(name = "",
                    values = c("Media Poblacional" = "black",
                              Estimador = "red")) +
  xlab("x") + ylab("f(x)") +
  theme(legend.position = "top",
        legend.text = element_text(size = 14),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        axis.title = element_text(size = 13, face = "bold"),
        plot.title = element_text(size = 18, face = "bold",
                                   hjust = 0.5)) +
  xlim(0, 1)
grafico_estadistico_histogram
```

### Histograma de nuestro Estadístico

