

Ejercicio 7

José Antonio Álvarez Ocete

Importamos los paquetes necesarios:

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.3
## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.5      v purrr 0.3.4
## v tibble 3.1.1       v dplyr 1.0.6
## v tidyr 1.1.3        v stringr 1.4.0
## v readr 1.4.0        v forcats 0.5.1
## Warning: package 'tibble' was built under R version 3.6.3
## Warning: package 'tidyr' was built under R version 3.6.3
## Warning: package 'readr' was built under R version 3.6.3
## Warning: package 'purrr' was built under R version 3.6.3
## Warning: package 'dplyr' was built under R version 3.6.3
## Warning: package 'forcats' was built under R version 3.6.3
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(gapminder)

## Warning: package 'gapminder' was built under R version 3.6.3

library(comprehenr)
library(ggplot2)
library(dplyr)
library(ggpubr)

## Warning: package 'ggpubr' was built under R version 3.6.3

defaultW <- getOption("warn")
options(warn = -1)
theme_set(theme_bw())

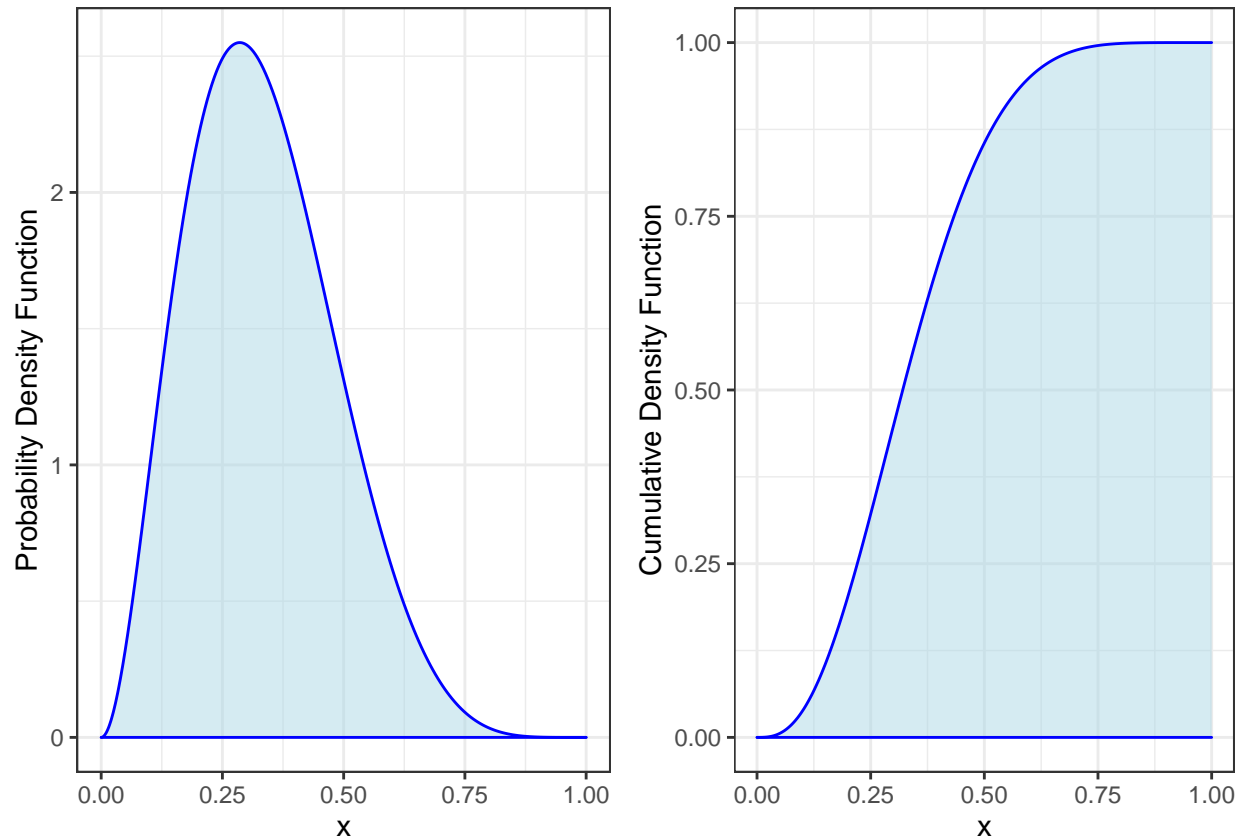
a)

x = seq(0, 1, length=1000)
pdf = dbeta(x, 3, 6)
cdf = pbeta(x, 3, 6)
df <- data.frame(x, pdf, cdf)
```

```
graf1 <- ggplot(df, aes(x=x, y=pdf)) +
  geom_ribbon(aes(ymin=0, ymax=pdf), fill="lightblue", col="blue", alpha=0.5) +
  ylab("Probability Density Function")

graf2 <- ggplot(df, aes(x=x, y=cdf)) +
  geom_ribbon(aes(ymin=0, ymax=cdf), fill="lightblue", col="blue", alpha=0.5) +
  ylab("Cumulative Density Function")

ggarrange(graf1, graf2,
  ncol = 2, nrow = 1)
```



b)

```
set.seed(123)

x = seq(0, 1, length=1000)
pdf = dbeta(x, 3, 6)
cdf = pbeta(x, 3, 6)
df <- data.frame(x, pdf, cdf)

muestra <- rbeta(20, 3, 6)
estimador_nucleo <- density(muestra)
df_estimator <- data.frame("x"=estimador_nucleo$x, "y"=estimador_nucleo$y)

graf1 <- ggplot() +
  geom_ribbon(data=df, aes(x=x, y=pdf, ymin=0, ymax=pdf),
```

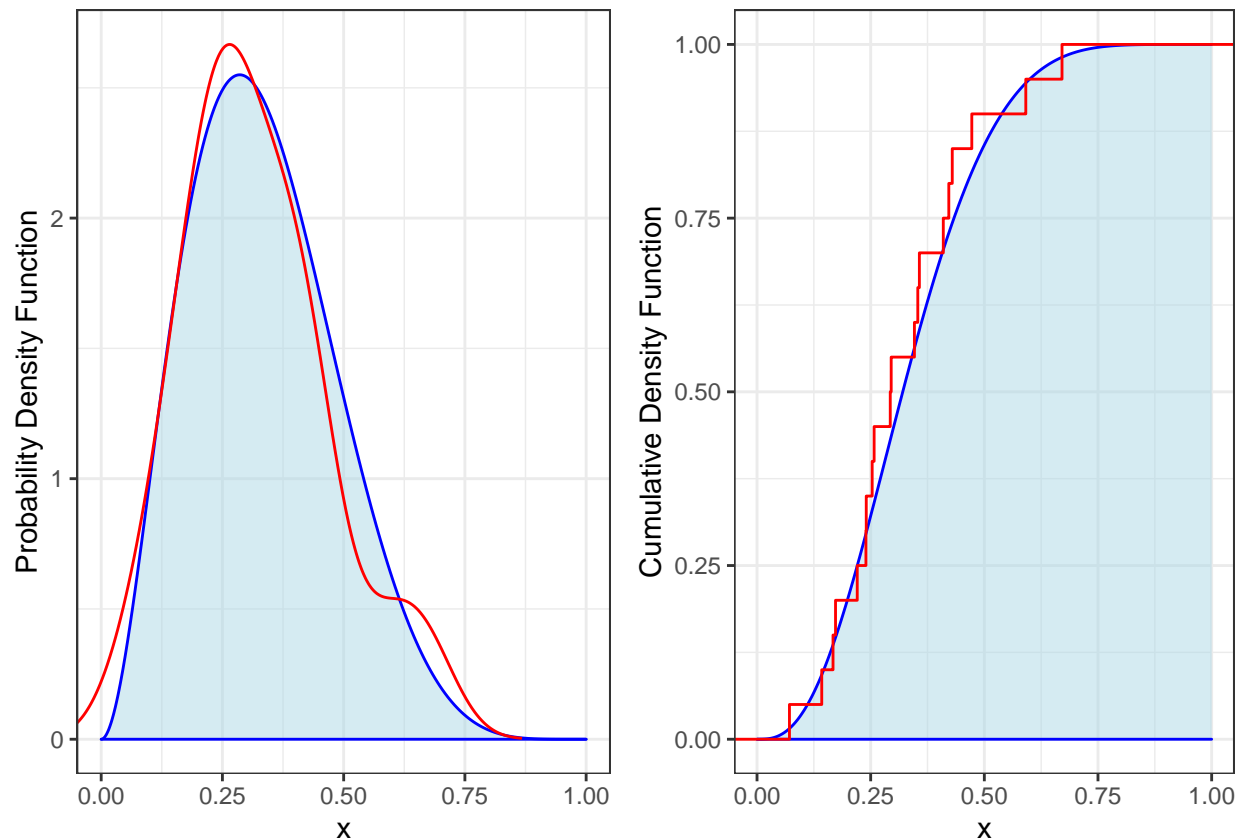
```

        fill="lightblue", col="blue", alpha=0.5) +
    geom_line(data=df_estimator, aes(x=x, y=y), col="red") +
    ylab("Probability Density Function") +
    coord_cartesian(xlim = c(0, 1))

graf2 <- ggplot() +
    geom_ribbon(data=df, aes(x=x, y=cdf, ymin=0, ymax=cdf),
              fill="lightblue", col="blue", alpha=0.5) +
    stat_ecdf(data=data.frame(muestra), aes(x=muestra), color="red", geom="step") +
    ylab("Cumulative Density Function") +
    coord_cartesian(xlim = c(0, 1))

ggarrange(graf1, graf2, ncol = 2, nrow = 1)

```



c)

```

set.seed(123)

n <- 20
m <- 200
alpha <- 3
beta <- 6

errors_pdf <- NULL
errors_cdf <- NULL
p_values_pdf <- NULL

```

```

p_values_cdf <- NULL

for (i in 1:m){
  muestra <- rbeta(n, alpha, beta)

  estimador_nucleo <- density(muestra)
  theoric_pdf_ys <- dbeta(estimador_nucleo$x, alpha, beta)
  ks_pdf <- ks.test(estimador_nucleo$y, theoric_pdf_ys)

  ecdf_estimada <- ecdf(muestra)
  theoric_cdf_ys <- pbeta(muestra, alpha, beta)
  ks_cdf <- ks.test(ecdf_estimada(muestra), "pbeta", alpha, beta)

  errors_pdf <- c(errors_pdf, ks_pdf$statistic)
  p_values_pdf <- c(p_values_pdf, ks_pdf$p.value)
  errors_cdf <- c(errors_cdf, ks_cdf$statistic)
  p_values_cdf <- c(errors_cdf, ks_cdf$p.value)
}

cat("Mean error in cdf: ", mean(errors_cdf), "\n")

## Mean error in cdf: 0.4115441

cat("Mean p-value for cdf: ", mean(p_values_cdf), "\n")

## Mean p-value for cdf: 0.4095036

cat("Mean error in pdf: ", mean(errors_pdf), "\n")

## Mean error in pdf: 0.1869434

cat("Mean p-value for pdf: ", mean(p_values_pdf), "\n")

## Mean p-value for pdf: 0.0017868

Sys.sleep(1)

```