Comparación de distribuciones: test Kolmogorov–Smirnov

Joaquín Amat Rodrigo j.amatrodrigo@gmail.com

Febrero, 2020

Tabla de contenidos

Versión PDF: Github

Más sobre ciencia de datos: cienciadedatos.net o joaquinamatrodrigo.github.io

Introducción

El objetivo de este documento es mostrar varias estrategias que se pueden emplear para comparar distribuciones y detectar si existen diferencias entre ellas. En la práctica, este tipo de comparación puede ser de utilidad en casos como:

- Análisis descriptivo: determinar si existen evidencias de que una misma variable se distribuye de forma distinta entre dos grupos. O bien, si dos variables se distribuyen de forma distinta.

- Monitorización de modelos en producción: en los proyectos que implican la creación y puesta en producción de modelos (predictivos, *clustering*, anomalías…), los modelos suelen entrenarse con datos históricos asumiendo que las variables empleadas por el modelo van a mantener un mismo comportamiento en el futuro cercano. ¿Cómo detectar si esto deja de ser cierto? ¿Si una variable pasa a tener un comportamiento distinto? Detectar estos cambios puede utilizarse a modo de alarmado que indique la necesidad de reentrenar los modelos, bien de forma automatizada o a través de nuevos estudios.

- Encontrar variables con distinto comportamiento entre dos escenarios: en ámbitos industriales, es común tener varias líneas de producción que, supuestamente, realizan exactamente el mismo proceso. Sin embargo, con frecuencia ocurre que alguna de las líneas genera resultados distintos. Una estrategia que puede ayudar a descubrir la razón de dicha diferencia es comparando las variables medidas en cada una de las líneas con el objetivo de identificar cuáles difieren en mayor grado.

Existen múltiples estrategias para dar respuesta a estas preguntas. Una de las aproximaciones más frecuentes es comparar estadísticos de centralidad (media, mediana) o de dispersión (desviación típica, rango intercuartílico). Esta estrategia tiene la ventaja de ser fácilmente interpretable, es sencillo explicar que la media de una variable ha aumentado entre dos años consecutivos. Sin embargo, cada uno de estos estadísticos solo contempla un tipo de diferencia, por lo que, dependiendo de cuál se utilice, se pueden estar ignorando cambios importantes. Por ejemplo, dos distribuciones muy distintas pueden tener la misma media.

Otra aproximación consiste emplear métodos que traten de cuantificar la “distancia” entre distribuciones, por ejemplo el estadístico **Kolmogorov–Smirnov statistic**, y que se ven influenciados tanto por diferencias en la localización como en la forma de la distribución.

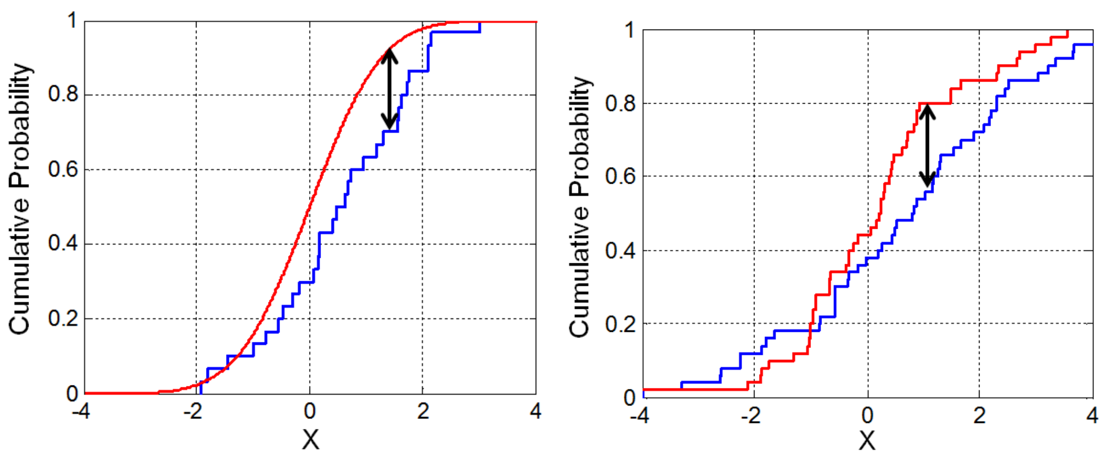
Como en la mayoría de test o métodos estadísticos, no existe uno que supere siempre a los demás. Dependiendo de qué cambio en la distribución sea el más importante de detectar, una estrategia será mejor que otra.

Cuando se comparan estadísticos entre dos muestras (media, mediana, varianza…) la pregunta inmediata tras calcular la diferencia es determinar si esta existe en realidad o si se debe únicamente a variaciones aleatorias de los datos utilizados, es decir, realizar inferencia. Para algunos de estos estadísticos, por ejemplo la media (t-test), existen test analíticos con los que se puede calcular la probabilidad exacta de observar diferencias de esa magnitud o superior si realmente no existiese una diferencia real, a esta probabilidad se le conoce como *p-value*. Sin embargo, para que sus resultados sean válidos, muchos de estos test requieren que se cumplan unas condiciones que, en la práctica, raramente se dan. Una alternativa a estos test analíticos son los métodos de resampling (permutación y bootstrapping) que, si bien son más lentos, no requiere de apenas asunciones.

Es importante recordar que el *p-value* se emplea como forma de cuantificar la seguridad que se tiene a la hora de aceptar que la diferencia es real, pero no dice nada sobre cómo de grande es esta. Por lo tanto, a la hora de tomar decisiones, se debe de tener en cuenta ambos valores de forma conjunta: magnitud de la diferencia y *p-value*.

Distancia Kolmogorov–Smirnov

El estadístico de Kolmogorov–Smirnov, también conocido como *distancia Kolmogorov–Smirnov (K-S)*, se define como la distancia vertical máxima entre las funciones de distribución acumulada empíricas de dos muestras, o entre una función de distribución empírica y una función de distribución acumulada teórica de referencia. La ventaja principal de este estadístico es que es sensible a diferencias tanto en la localización como en la forma de la función de distribución acumulada.

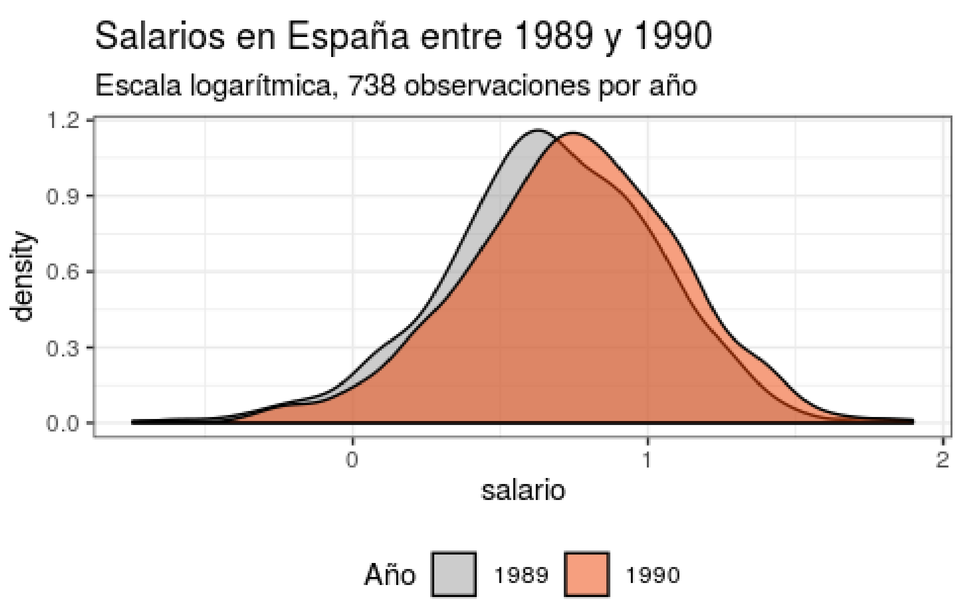


Izquierda: *Ilustración del estadístico de Kolmogorov–Smirnov. La línea roja muestra la función de distribución acumulada teórica, la azul la función de distribución acumulada empírica, y la flecha negra es el estadístico K-S. Fuente: Wikipedia* **Derecha:** *Ilustración del estadístico de Kolmogorov–Smirnov entre dos muestras. Las líneas roja y azul muestran la función de distribución acumulada empírica de dos muestras, y la flecha negra es el estadístico K-S. Fuente: Wikipedia*

Ejemplo

El set de datos `Snmesp` del paquete `plm` contiene una muestra de los salarios (en escala logarítmica) pagados en España durante los años 1983 a 1990 (783 observaciones por año). Se pretende determinar si la distribución de los salarios cambió entre 1989 y 1993. Para ello se calcula la distancia de *Kolmogorov–Smirnov*.

```
library(tidyverse)
data(Snmesp, package = "plm")

Snmesp <- Snmesp %>%
          filter(year %in% c(1989, 1990)) %>%
          mutate(year = as.factor(year)) %>%
          select(year, salario = w)
Snmesp %>%
    ggplot(aes(x = salario, fill = year)) +
    geom_density(alpha = 0.5) +
    scale_fill_manual(values = c("gray60", "orangered2")) +
    labs(title = "Salarios en España entre 1989 y 1990",
         subtitle = "Escala logarítmica, 738 observaciones por año",
         fill = "Año") +
    theme_bw() +
    theme(legend.position = "bottom")
```

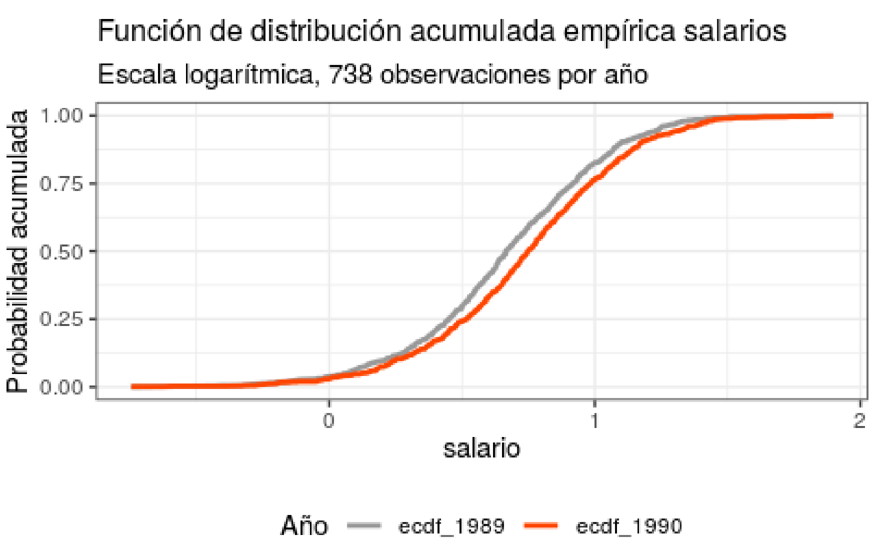


Cálculo de la función de distribución acumulada empírica

La función `ecdf()` permite ajustar la función de distribución acumulada empírica a partir de una muestra. El resultado de esta función es un objeto `ecdf` que se comporta de forma similar a un modelo predictivo, recibe un vector de observaciones y devuelve su probabilidad acumulada.

```
# Se ajustan las funciones ecdf con cada muestra.
ecdf_1989 <- ecdf(Snmesp %>% filter(year== 1989) %>% pull (salario))
ecdf_1990 <- ecdf(Snmesp %>% filter(year== 1990) %>% pull (salario))
```

```
# Se calcula la probabilidad acumulada de cada valor de salario observado con cada
# una de las funciones ecdf.
grid_salario <- unique(Snmesp %>% pull(salario))
prob_acumulada_ecdf_1989 <- ecdf_1989(v = grid_salario)
prob_acumulada_ecdf_1990 <- ecdf_1990(v = grid_salario)

# Se unen los valores calculados en un dataframe.
df_ecdf <- data.frame(
            salario = grid_salario,
            ecdf_1989 = prob_acumulada_ecdf_1989,
            ecdf_1990 = prob_acumulada_ecdf_1990
           ) %>%
          pivot_longer(
            cols = c(ecdf_1989, ecdf_1990),
            names_to = "year",
            values_to = "ecdf"
          )
grafico_ecdf <- ggplot(data = df_ecdf,
                       aes(x = salario, y = ecdf, color = year)) +
                geom_line(size = 1) +
                scale_color_manual(values = c("gray60", "orangered1")) +
                labs(
                 title = "Función de distribución acumulada empírica salarios",
                 subtitle = "Escala logarítmica, 738 observaciones por año",
                 color = "Año",
                 y = "Probabilidad acumulada"
                ) +
                theme_bw() +
                theme(legend.position = "bottom",
                      plot.title = element_text(size=12))
grafico_ecdf
```



Este mismo gráfico puede generarse directamente con `stat_ecdf()` de `ggplot2`.

```
Snmesp %>%
    ggplot(aes(x = salario, color = year)) +
    stat_ecdf(geom = "step") +
    labs(title = "Función de distribución acumulada empírica salarios",
         subtitle = "Escala logarítmica, 738 observaciones por año",
         color = "Año") +
    theme_bw() +
    theme(legend.position = "bottom",
          plot.title = element_text(size=12))
```

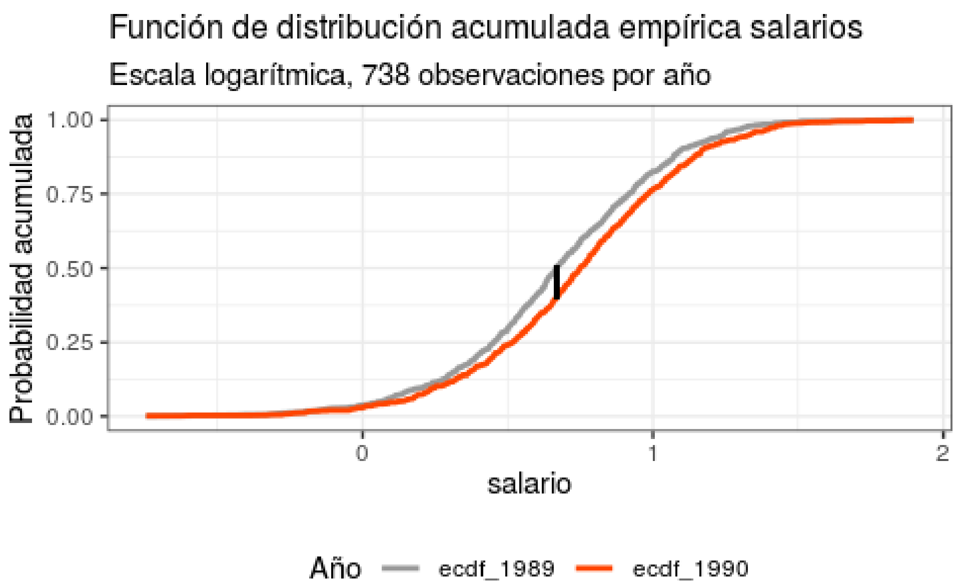
Cálculo de la distancia Kolmogorov–Smirnov

```
# Se calcula la diferencia absoluta entre las probabilidades acumuladas de cada
# función.
abs_dif <-  abs(prob_acumulada_ecdf_1989 - prob_acumulada_ecdf_1990)

# La distancia Kolmogorov-Smirnov es el máximo de las distancias absolutas.
distancia_ks <- max(abs_dif)
paste("Distancia Kolmogorov-Smirnov:", distancia_ks)
```

```
## [1] "Distancia Kolmogorov-Smirnov: 0.105691056910569"
```

Se añade al gráfico anterior un segmento que representa la distancia *Kolmogorov–Smirnov*.

```
indice_ks <- which.max(abs_dif)

grafico_ecdf +
  geom_segment(aes(
                x = grid_salario[indice_ks],
                xend = grid_salario[indice_ks],
                y = prob_acumulada_ecdf_1989[indice_ks],
                yend = prob_acumulada_ecdf_1990[indice_ks]
               ),
               color = "black")
```



La función `ks.test()` del paquete `stats` calcula, como parte del test, la distancia de *Kolmogorov–Smirnov*. Se comprueba, a modo de validación, que ambos valores coinciden.

```
test <- ks.test(
        x = Snmesp %>% filter(year== 1989) %>% pull (salario),
        y = Snmesp %>% filter(year== 1990) %>% pull (salario)
      )
test$statistic
```

```
##         D
## 0.1056911
```

Test de Kolmogorov–Smirnov

Una vez calculada la distancia de *Kolmogorov–Smirnov*, hay que determinar si el valor de esta distancia es suficientemente grande, teniendo en cuenta las muestras disponibles, como para considerar que las dos distribuciones son distintas (*p-value*). Esto puede conseguirse calculando la probabilidad de observar distancias iguales o mayores si ambas muestras procediesen de la misma distribución, es decir, que las dos distribuciones son la misma.

Para el estadístico de *Kolmogorov–Smirnov* existen dos tipos de solución:

- Solución analítica (exacta): si se cumple que las muestras son grandes y que no hay ligaduras, esta solución es mucho más rápida y genera *p-values* exactos. Esta solución está implementada en la función `ks.test()` del paquete `stats`.

- Mediante un test de resampling: consiste en simular, mediante permutaciones o *bootstrapping*, las distancias de *Kolmogorov–Smirnov* que se obtendrían si ambas muestras procediesen de la misma distribución. Una vez obtenidas las simulaciones, se calcula el porcentaje de distancias iguales o mayores a la observada.

-

Algoritmo mediante permutación

1. Calcular la distancia *Kolmogorov–Smirnov* entre las dos muestras. Ha esta distancia se llama distancia observada.

2. Combinar todas las observaciones en un mismo vector.

3. Dividir aleatoriamente las observaciones en dos conjuntos del mismo tamaño que las muestras iniciales.

4. Calcular la distancia de *Kolmogorov–Smirnov* entre los dos nuevos conjuntos.

5. Repetir los pasos del 1 al 3 N veces.

6. Calcular la fracción de las N distancias simuladas iguales o superiores a la distancia observada. Este valor se corresponde con la estimación empírica del *p-value*.

La función `ks.boot()` de la librería `Matching` implementa la solución por *resampling* utilizando bootstrapping.

Nota: siendo estrictos en simular acorde a la hipótesis nula de que ambas muestras proceden de la misma distribución, la estrategia de permutación es más adecuada.

Ejemplo

Empleando los mismos datos del ejemplo anterior se aplica el test de *Kolmogorov–Smirnov* para responder a la pregunta de si la distribución del salario ha cambiado entre 1989 y 1990.

Solución por bootstrapping

```
set.seed(123)
test_ks_boot <- Matching::ks.boot(
                  Tr = Snmesp %>% filter(year== 1989) %>% pull (salario),
                  Co = Snmesp %>% filter(year== 1990) %>% pull (salario),
                  alternative = "two.sided",
                  nboots = 5000
                )

test_ks_boot$ks.boot.pvalue
```

```
## [1] 8e-04
```

Solución analítica

```
test_ks <- ks.test(
        x = Snmesp %>% filter(year== 1989) %>% pull (salario),
        y = Snmesp %>% filter(year== 1990) %>% pull (salario)
      )
test_ks
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Snmesp %>% filter(year == 1989) %>% pull(salario) and Snmesp %>%
filter(year == 1990) %>% pull(salario)
## D = 0.10569, p-value = 0.0005257
## alternative hypothesis: two-sided
```

En ambos casos, existen evidencias suficientes para considerar que la distribución de salarios ha variado de un año a otro.

Otros estadísticos

Tal y como se ha explicado en la introducción, el estadístico más adecuado depende de qué tipo de diferencia se esté interesado en detectar. A continuación, se muestra una función `calcular_diferencias_permutacion()` con la que detectar la diferencia observada y significancia estadística (calculada mediante permutaciones) para la media, mediana, desviación típica e intervalo intercuartílico.

```
# Funciones auxiliares que calculan el estadístico para cada permutación (split)
#-------------------------------------------------------------------------------
calcular_dif_media <- function(split) {
  grupo_1 <- rsample::analysis(x = split)
  grupo_2 <- rsample::assessment(x = split)
  dif <- mean(grupo_1[[1]]) - mean(grupo_2[[1]])
  dif_abs <- abs(dif)
  return(dif_abs)
}

calcular_dif_mediana <- function(split) {
  grupo_1 <- rsample::analysis(x = split)
  grupo_2 <- rsample::assessment(x = split)
  dif <- median(grupo_1[[1]]) - median(grupo_2[[1]])
  dif_abs <- abs(dif)
  return(dif_abs)
}

calcular_dif_sd <- function(split) {
  grupo_1 <- rsample::analysis(x = split)
  grupo_2 <- rsample::assessment(x = split)
  dif <- sd(grupo_1[[1]]) - sd(grupo_2[[1]])
  dif_abs <- abs(dif)
  return(dif_abs)
}


calcular_dif_iqr <- function(split) {
  grupo_1 <- rsample::analysis(x = split)
  grupo_2 <- rsample::assessment(x = split)
  dif <- IQR(grupo_1[[1]]) - IQR(grupo_2[[1]])
  dif_abs <- abs(dif)
  return(dif_abs)
}
```

```r
# Función para calcular la diferencia observada y su significancia estadística
#--------------------------------------------------------------------------------
calcular_diferencias_permutacion <- function(sample_1,
                                             sample_2,
                                             n_permut = 1000,
                                             seed     = NULL) {
  # Se calcula la longitud de cada sample para que el posterior reparto sea
  # proporcional.
  len_sample_1 <- length(sample_1)
  len_sample_2 <- length(sample_2)
  len_pull_samples <- len_sample_1 + len_sample_2

  # Se almacenan las observaciones en un dataframe
  df  <- tibble(
          sample = rep(c("sample_1", "sample_2"),
                   times = c(len_sample_1, len_sample_2)),
          valor  = c(sample_1, sample_2)
         )
  # Cálculo del valor observado de cada estadístico y su diferencia.
  valor_observado <- df %>%
                    group_by(sample) %>%
                    summarise(
                      media = mean(valor, na.rm = TRUE),
                      mediana = median(valor, na.rm = TRUE),
                      sd = sd(valor, na.rm = TRUE),
                      iqr = IQR(valor, na.rm = TRUE)
                    ) %>%
                   pivot_longer(
                     cols = -sample,
                     names_to = "estadistico",
                     values_to = "valor"
                   ) %>%
                   pivot_wider(
                     names_from = sample,
                     values_from = valor) %>%
                   mutate(
                     diferencia = sample_1 - sample_2,
                     diferencia_abs = abs(diferencia)
                   )

  # Se generan las permutaciones con la función rsample::mc_cv().
  if(! is.null(seed)) {
    set.seed(seed = seed)
  }
  resamples <- rsample::mc_cv(
                 data = df %>% select(valor),
                 prop = len_sample_1/len_pull_samples,
                 times = n_permut
                )
```

```
  # Se paraleliza el cálculo de cada permutación.
  future::plan(strategy = "multiprocess")
  permut_media   <- furrr::future_map_dbl(.x = resamples$splits,
                                          .f = calcular_dif_media)
  permut_mediana <- furrr::future_map_dbl(.x = resamples$splits,
                                          .f = calcular_dif_mediana)
  permut_sd      <- furrr::future_map_dbl(.x = resamples$splits,
                                          .f = calcular_dif_sd)
  permut_iqr     <- furrr::future_map_dbl(.x = resamples$splits,
                                          .f = calcular_dif_iqr)

  # Cálculo de significancia
  dif_observada_abs <- valor_observado %>% pull(diferencia_abs)
  names(dif_observada_abs) <- valor_observado %>% pull(estadistico)

  pvalue_media   <- mean(permut_media > dif_observada_abs[["media"]])
  pvalue_mediana <- mean(permut_mediana > dif_observada_abs[["mediana"]])
  pvalue_sd      <- mean(permut_sd > dif_observada_abs[["sd"]])
  pvalue_iqr     <- mean(permut_iqr > dif_observada_abs[["iqr"]])

  resultados <- valor_observado %>%
                    mutate(p_value = c(pvalue_media, pvalue_mediana,
                                       pvalue_sd,pvalue_iqr)
                          )
 return(resultados)
}
result_permutacion <- calcular_diferencias_permutacion(
                        sample_1 = Snmesp %>% filter(year== 1989) %>% pull
(salario),
                        sample_2 = Snmesp %>% filter(year== 1990) %>% pull
(salario),
                        n_permut = 5000,
                        seed = 123
                      )

result_permutacion
```

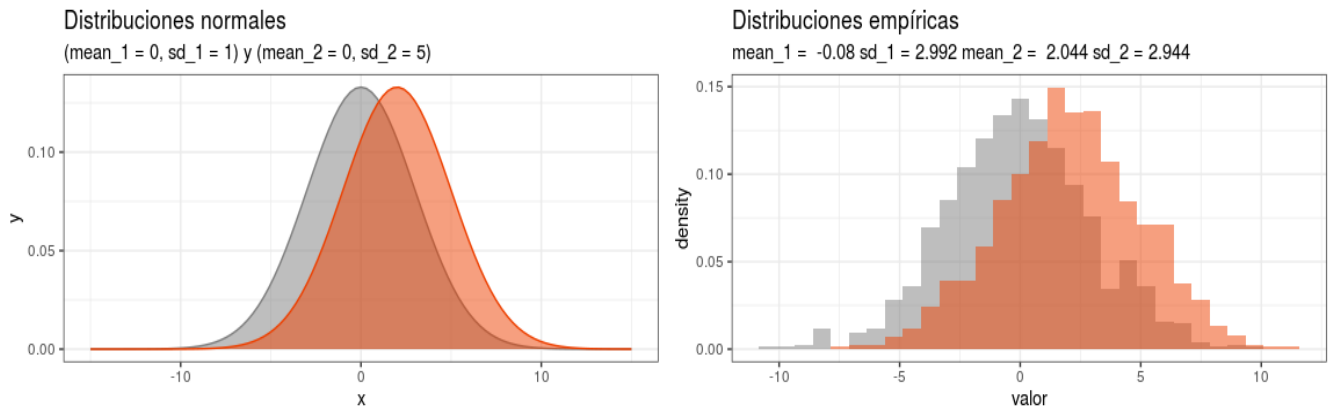
```
## # A tibble: 4 x 6
##   estadistico sample_1 sample_2 diferencia diferencia_abs p_value
##   <chr>          <dbl>    <dbl>      <dbl>          <dbl>   <dbl>
## 1 media          0.669    0.738   -0.0696         0.0696   0.0002
## 2 mediana        0.665    0.748   -0.0828         0.0828   0
## 3 sd             0.359    0.368   -0.00860        0.00860  0.554
## 4 iqr            0.464    0.465   -0.00109        0.00109  0.973
```

Otros ejemplos

Se explora las capacidades de cada una de estas estrategias para diferenciar las siguientes distribuciones.

Distribuciones normales con misma varianza y distinta media

```
# Distribuciones teóricas
p1 <- ggplot(data = data.frame(x=c(-15,15)), aes(x = x)) +
        stat_function(fun=dnorm, args=list(mean = 0, sd = 3),
                      geom = "line", color = "gray50") +
        stat_function(fun=dnorm, args=list(mean = 0, sd = 3),
                    geom = "area", fill = "gray50", alpha = 0.5) +
        stat_function(fun=dnorm, args=list(mean = 2, sd = 3),
                    geom = "line", color = "orangered2") +
        stat_function(fun=dnorm, args=list(mean = 2, sd = 3),
                    geom = "area", fill = "orangered2", alpha = 0.5) +
        labs(title = "Distribuciones normales",
             subtitle = "(mean_1 = 0, sd_1 = 1) y (mean_2 = 0, sd_2 = 5)") +
        theme_bw()

# Simulación de datos extraídos de dos distribuciones
set.seed(1234)
sample_normal_1 <- rnorm(n = 1000, mean = 0, sd = 3)
sample_normal_2 <- rnorm(n = 1000, mean = 2, sd = 3)

# Distribuciones de las muestras simuladas
p2 <- ggplot() +
      geom_histogram(data = tibble(valor = sample_normal_1),
                     aes(x = valor, stat(density)),
                     fill = "gray50",
                     alpha = 0.5) +
      geom_histogram(data = tibble(valor = sample_normal_2),
                     aes(x = valor, stat(density)),
                     fill = "orangered2",
                     alpha = 0.5) +
        labs(title = "Distribuciones empíricas",
             subtitle = paste(
                        "mean_1 = ", round(mean(sample_normal_1), 3),
                        "sd_1 =", round(sd(sample_normal_1), 3),
                        "mean_2 = ", round(mean(sample_normal_2), 3),
                        "sd_2 =", round(sd(sample_normal_2), 3)
                       )
        ) +
        theme_bw()

ggpubr::ggarrange(p1, p2, nrow = 1)
```



Se analizan las diferencias con el test de *Kolmogorov–Smirnov* y con los estadísticos de centralidad y dispersión.

```
set.seed(123)
test_ks_boot <- Matching::ks.boot(
                    Tr = sample_normal_1,
                    Co = sample_normal_2,
                    alternative = "two.sided",
                    nboots = 5000
                  )

test_ks_boot$ks.boot.pvalue
```

```
## [1] 0
```

```
result_permutacion <- calcular_diferencias_permutacion(
                          sample_1 = sample_normal_1,
                          sample_2 = sample_normal_2,
                          n_permut = 5000,
                          seed = 123
                        )

result_permutacion
```

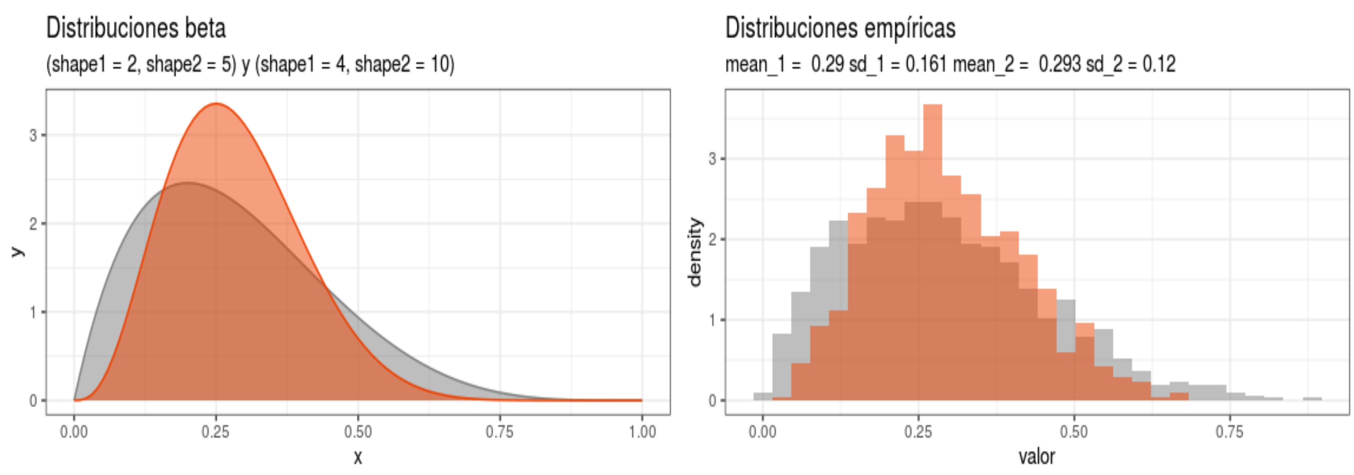
```
## # A tibble: 4 x 6
##   estadistico sample_1 sample_2 diferencia diferencia_abs p_value
##   <chr>          <dbl>    <dbl>      <dbl>          <dbl>   <dbl>
## 1 media        -0.0798     2.04   -2.12           2.12      0
## 2 mediana      -0.119      2.04   -2.16           2.16      0
## 3 sd            2.99       2.94    0.0484         0.0484    0.624
## 4 iqr           3.87       3.87   -0.00493        0.00493   0.984
```

Distribuciones beta con misma media y distinta varianza

$$E(X) = \frac{\alpha}{\alpha + \beta}$$

$$V(X) = \frac{\alpha\beta}{(\alpha + \beta + 1)(\alpha + \beta)^2}$$

```
# Distribuciones teóricas
p1 <- ggplot(data = data.frame(x=c(0,1)), aes(x = x)) +
      stat_function(fun=dbeta, args=list(shape1 = 2, shape2 = 5),
                    geom = "line", color = "gray50") +
      stat_function(fun=dbeta, args=list(shape1 = 2, shape2 = 5),
                    geom = "area", fill = "gray50", alpha = 0.5) +
      stat_function(fun=dbeta, args=list(shape1 = 4, shape2 = 10),
                    geom = "line", color = "orangered2") +
      stat_function(fun=dbeta, args=list(shape1 = 4, shape2 = 10),
                    geom = "area", fill = "orangered2", alpha = 0.5) +
      labs(title = "Distribuciones beta",
           subtitle = "(shape1 = 2, shape2 = 5) y (shape1 = 4, shape2 = 10)") +
      theme_bw()


# Simulación de datos extraídos de dos distribuciones
set.seed(123)
sample_beta_1 <- rbeta(n = 1000, shape1 = 2, shape2 = 5)
sample_beta_2 <- rbeta(n = 1000, shape1 = 4, shape2 = 10)

# Distribuciones de las muestras simuladas
p2 <- ggplot() +
      geom_histogram(data = tibble(valor = sample_beta_1),
                     aes(x = valor, stat(density)),
                     fill = "gray50",
                     alpha = 0.5) +
      geom_histogram(data = tibble(valor = sample_beta_2),
                     aes(x = valor, stat(density)),
                     fill = "orangered2",
                     alpha = 0.5) +
        labs(title = "Distribuciones empíricas",
             subtitle = paste(
                          "mean_1 = ", round(mean(sample_beta_1), 3),
                          "sd_1 =", round(sd(sample_beta_1), 3),
                          "mean_2 = ", round(mean(sample_beta_2), 3),
                          "sd_2 =", round(sd(sample_beta_2), 3)
                        )
        ) +
        theme_bw()

ggpubr::ggarrange(p1, p2, nrow = 1)
```



Se analizan las diferencias con el test de *Kolmogorov–Smirnov* y con los estadísticos de centralidad y dispersión.

```r
set.seed(123)
test_ks_boot <- Matching::ks.boot(
                    Tr = sample_beta_1,
                    Co = sample_beta_2,
                    alternative = "two.sided",
                    nboots = 5000
                )
test_ks_boot$ks.boot.pvalue
```

```
## [1] 0
```

```r
result_permutacion <- calcular_diferencias_permutacion(
                          sample_1 = sample_beta_1,
                          sample_2 = sample_beta_2,
                          n_permut = 5000,
                          seed = 123
                      )
result_permutacion
```

```
## # A tibble: 4 x 6
##   estadistico sample_1 sample_2 diferencia diferencia_abs p_value
##   <chr>          <dbl>    <dbl>      <dbl>          <dbl>   <dbl>
## 1 media          0.290    0.293   -0.00364        0.00364   0.558
## 2 mediana        0.269    0.279   -0.0102         0.0102    0.138
## 3 sd             0.161    0.120    0.0411         0.0411    0
## 4 iqr            0.224    0.169    0.0552         0.0552    0
```