

Taller #3

Sergio Andrés Díaz Vera

Samuel Ruíz Martínez
Daniel Felipe Cendales G.

Hernan Supelano Vega

Intervalos de confianza

1. Sean Y_1, \dots, Y_n con $Y_i \stackrel{iid}{\sim} \mathcal{N}(\mu_0, \sigma^2)$ para todos $i = 1, \dots, n$

a. Tomemos $\mu_0 = 2, \sigma = \sqrt{2}$

```
# Asignación de parámetros
mu_0 <- 2; sigma <- sqrt(2)

# Fijamos la semilla
set.seed(31415)
```

b. Simulaciones

```
## Metaparámetros
N <- 1000      # Número de simulaciones
n <- 100       # Tamaño de cada muestra
alpha <- 0.05

## Simulaciones
intervalos <- sapply(1:N, function(k){
  x <- rnorm(n = n, mean = mu_0, sd = sigma)
  x_barra <- mean(x)
  lims <- x_barra + c(-1, 1)*qnorm(1 - alpha/2)*sigma/sqrt(n)
  c(lims, x_barra)
})

## Trasponemos los resultados
intervalos <- t(intervalos)

## Función que verifica si se está entre un intervalo
entre <- function(x, valor)
  ifelse(x[1] <= mu_0 & mu_0 <= x[2], 1, 0)

## Contamos la cantidad de intervalos que contienen a mu_0
cantidad <- apply(intervalos, MARGIN = 1,
  FUN = entre, valor = mu_0)
```

c. Cantidad de intervalos que contienen a μ_0

```
## Intervalos que contienen a mu_0
sum(cantidad)
```

```
## [1] 958
```

Que era de esperarse, ya que aproximadamente $1000 * (1 - \alpha)$ de los intervalos deben contener a la media.

2. Sean Y_1, \dots, Y_n con $Y_i \stackrel{iid}{\sim} \mathcal{N}(\mu_0, \sigma^2)$ para todos $i = 1, \dots, n$

a. Tomemos $\mu_0 = 4, \sigma = 3$

```
# Asignación de parámetros
```

```
mu_0 <- 4; sigma <- 3
```

```
# Fijamos la semilla
```

```
set.seed(27182)
```

b. Simulaciones

```
## Metaparámetros
```

```
N <- 1000          # Número de simulaciones
```

```
n <- 100           # Tamaño de cada muestra
```

```
alpha <- 0.05
```

```
## Simulaciones
```

```
muestras <- sapply(1:N, function(k){  
  rnorm(n = n, mean = mu_0, sd = sigma)  
})
```

```
## Función que calcula intervalos y valores P
```

```
cAlculos <- function(y){  
  prueba_t <- t.test(y, alternative = "two.sided",  
                    conf.level = 1 - alpha,  
                    mu = mu_0)  
  c(prueba_t$conf.int, prueba_t$p.value)  
}
```

```
## Aplicamos la función a cada una de las muestras
```

```
resultados <- apply(muestras, MARGIN = 2, FUN = cAlculos)
```

```
## Cálculo de la cantidad de intervalos que contienen a mu_0
```

```
cantidad <- apply(resultados, 2, FUN = entre, valor = mu_0)
```

- Cantidad de intervalos que contienen a μ_0 :

```
## Intervalos que contienen a mu_0
```

```
sum(cantidad)
```

```
## [1] 944
```

Lo cual concuerda con la teoría.

- Conteo de *valores-p* menores a 0.05

```
## Número total de rechazos
```

```
sum(resultados[3, ] < 0.05)
```

```
## [1] 56
```

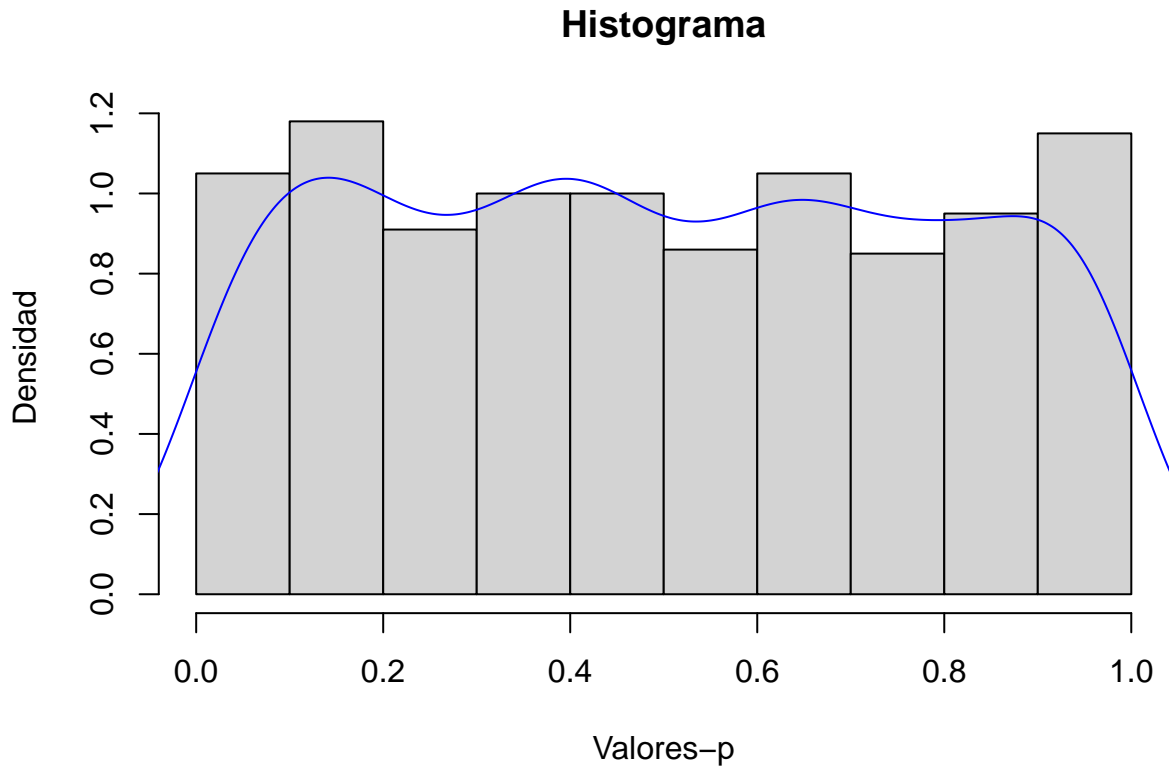
- Distribución empírica (y teórica) del *valor-p*

```
## Histograma y densidad ajustada
```

```
hist(resultados[3, ], main = "Histograma",  
     xlab = "Valores-p", ylab = "Densidad",  
     freq = FALSE)
```

```
# Curva suavizada
```

```
lines(density(resultados[3, ]), col = "blue")
```



Que era de esperarse, ya que aproximadamente $1000 * (1 - \alpha)$ de los intervalos deben contener a la media.

Pruebas No Paramétricas

3. Cuadro de resumen

4. Ejercicios libro *Hollander y Wolfe*

a. Ejercicios 9 y 11

(9.) Supongamos que $n = 5$ y hemos observado $Z_1 = -1.3, Z_2 = 2.4, Z_3 = 1.3, Z_4 = 1.3$ y $Z_5 = 2.4$

Para calcular todos los posibles valores que puede tomar el estadístico, podemos generar las 2^n tuplas de 1's y 0's de longitud n , hacer el producto punto con los rangos y contruir una tabla de frecuencias.

Evitamos poner el código que nos genera la distribución del estadístico.

```
## Datos
z <- c(-1.3, 2.4, 1.3, 1.3, 2.4)      # Observaciones
r1 <- rank(abs(z))                    # Rangos con empates
r2 <- order(abs(z))                   # Rangos sin empates
phi <- ifelse(z > 0, yes = 1, 0)

## Cálculo del estadístico con empates
t_1 <- sum(r1 * phi)
t_2 <- sum(r2 * phi)

## Distribución
distrib_wilc(r1)

## # A tibble: 12 x 3
##       t `P[T>=t]` `p-val`
##   <dbl> <chr>      <dbl>
```

```
## 1 15 1/32 0.0312
## 2 13 4/32 0.125
## 3 11 7/32 0.219
## 4 10.5 9/32 0.281
## 5 9 10/32 0.312
## 6 8.5 16/32 0.5
## 7 6.5 22/32 0.688
## 8 6 23/32 0.719
## 9 4.5 25/32 0.781
## 10 4 28/32 0.875
## 11 2 31/32 0.969
## 12 0 32/32 1
```

A continuación podemos el *valor-p* asociado a cada test usando (y evitando) los empates.

```
## Valor P asociado
subset(distrib_wilc(r1), t == t_1)
```

```
## # A tibble: 1 x 3
##       t `P[T>=t]` `p-val`
##   <dbl> <chr>      <dbl>
## 1    13 4/32      0.125
```

```
## Valor P asociado
subset(distrib_wilc(r2), t == t_2)
```

```
## # A tibble: 1 x 3
##       t `P[T>=t]` `p-val`
##   <dbl> <chr>      <dbl>
## 1    14 2/32      0.0625
```

Podemos ver que, usando la distribución errónea, el *valor-p* asociado es de 0.0625. Sin embargo, usando el test y distribución apropiados, el *valor-p* es de 0.125. Lo que implica que se hubiese tomado decisiones diferentes si el nivel de significancia del test hubiese sido del 1%.

(11.) Supongamos que tenemos n observaciones y queremos juzgar el par de hipótesis $H_0 : \theta = 0$ vs. $H_a : \theta \neq 0$

Supongamos que la región de confianza viene dada por el conjunto de valores $\left\{0, 1, \frac{n(n+1)}{2} - 1, \frac{n(n+1)}{2}\right\}$.

Notemos que T^+ puede reescribirse como un producto punto entre 2 vectores:

$$T^+ = \begin{bmatrix} 1 & 2 & \cdots & n \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix} = [1 : n] \psi$$

Cada posible vector ψ tiene una probabilidad de $\frac{1}{2^n}$. Para que T^+ tome el valor 0, todos los componentes de ψ deben ser 0 y para que tome el valor 1 el primer componente de ψ debe ser 1. De igual forma, para que tome el valor $\frac{n(n+1)}{2}$ todos los componentes de ψ deben ser 1 y para que tome el valor $\frac{n(n+1)}{2} - 1$, el primer componente de ψ debe ser 0 y el resto 1.

Por ende $\alpha = P\left[T \leq 1 \text{ o } T \geq \frac{n(n+1)}{2} - 1\right] = P\left[T \in \left\{0, 1, \frac{n(n+1)}{2} - 1, \frac{n(n+1)}{2}\right\}\right] = \frac{4}{2^n} = \frac{1}{2^{n-2}}$

b. Ejercicios 47 y 51

(47.) Supongamos que $F_1 = \cdots = F_{20} = F$. Además se tiene que $F(0) = 0.3$. Queremos comparar el par de hipótesis $H_0 : \theta = 0$ vs. $H_a : \theta > 0$

Bajo H_0 se tiene que $B \sim \text{binom}(n = 20, p = 0.5)$, luego el valor que nos da la región de confianza viene dado por:

```
## Ajuste de hiperparámetros
alpha <- 0.0577
n <- 20
p0 <- 0.5
pa <- 1 - 0.3

## Cálculo del cuantil
t_c <- qbinom(prob = p0, size = n, p = alpha,
              lower.tail = FALSE) + 1
t_c
```

```
## [1] 14
```

Como $F(0) = 0.3$ entonces, bajo H_a , $B \sim \text{pbinom}(n = 20, p = 1 - 0.3)$ y por ende el cálculo de la potencia viene dado por $P_a[B \geq 14]$

```
pbinom(q = t_c - 1, prob = pa, size = n, lower.tail = FALSE)
```

```
## [1] 0.6080098
```

(51.) Hacemos las mismas suposiciones que en el punto anterior. Basándonos en la aproximación, tenemos que el cuantil que nos da la región de rechazo viene dado por:

```
## Cuantil que nos da la región de rechazo
(z_q <- qnorm(alpha, lower.tail = FALSE) )
```

```
## [1] 1.574378
```

Denotemos por z_q al cuantil anterior y por p_a el valor de la hipótesis alterna, es decir $p_a = P[X - \theta > 0]$. Recordemos que el estadístico construido (bajo H_0) viene dado por:

$$Z = \frac{B - np_0}{\sqrt{np_0(1 - p_0)}}$$

Por ende, la potencia, que es rechazar la hipótesis nula dado que es falsa viene dada por:

$$\begin{aligned} 1 - \beta &= P[Z > z_q] \\ &= P\left[\frac{B - np_0}{\sqrt{np_0(1 - p_0)}} > z_q\right] \\ &= P\left[B > z_q \sqrt{np_0(1 - p_0)} + np_0\right] \\ &= P\left[B - np_a > z_q \sqrt{np_0(1 - p_0)} + n(p_0 - p_a)\right] \\ &= P\left[\frac{B - np_a}{\sqrt{np_a(1 - p_a)}} > \frac{z_q \sqrt{np_0(1 - p_0)} + n(p_0 - p_a)}{\sqrt{np_a(1 - p_a)}}\right] \\ &= 1 - \Phi\left(\frac{z_q \sqrt{np_0(1 - p_0)} + n(p_0 - p_a)}{\sqrt{np_a(1 - p_a)}}\right) \end{aligned}$$

Numéricamente:

```
1 - pnorm((z_q*sqrt(n*p0*(1 - p0)) + n*(p0 - pa)) /
          sqrt(n*pa*(1 - pa)))
```

```
## [1] 0.5925124
```

Al comparar, vemos que las potencias en ambos casos son muy parecidas, mas o menos del 60%.