

Metodos No Paramátricos - Ejercicios Prácticos

*Marina Fragalá
Gonzalo Barrera Borla*

July 6, 2019

Contents

Practica 1	1
Ejercicio 1	1
Ejercicio 2	2
Ejercicio 3	3
Ejercicio 4	5
Ejercicio 6	7
Ejercicio 7	7
Ejercicio 9	10
Practica 2	11
Ejercicio 1	11
Ejercicio 3	11
Ejercicio 4	12
Ejercicio 5	12
Ejercicio 7	12
Ejercicio 8	13
Ejercicio 9	14
Ejercicio 10	16
Ejercicio 11	17
Practica 3	18
Ejercicio 1	18
Ejercicio 2	22
Ejercicio 3	23
Ejercicio 4	23
Ejercicio 5	24
Ejercicio 6	25
Ejercicio 11	26
Practica 4	27
Ejercicio 1	27
Ejercicio 3	29
Ejercicio 4	31
Ejercicio 5	33
Practica 5	34
Ejercicio 1	34

Practica 1

Ejercicio 1

Para un estudio de mercado 277 personas degustan un licor y 69 de ellas desaprueban el nuevo sabor. Construya un intervalo de confianza de nivel asintótico 0.95 para la verdadera proporción p de personas que

aprueban el licor.

```
n <- 277
desaprueba <- 69
alfa <- 1 - 0.95

ic_asint_prop <- function(n, Tobs, alfa) {
  phat <- Tobs/n
  delta <- qnorm(1 - alfa/2) * sqrt(phat*(1-phat)/n)
  return(c(phat - delta, phat + delta))
}

(resultados$p1 <- list(ej1 <- ic_asint_prop(n, n - desaprueba, alfa)))

## [[1]]
## [1] 0.6999712 0.8018338
```

Ejercicio 2

Sea p la probabilidad de que una ambulancia responda a una llamada en 10 minutos. En base a $n = 15$ observaciones independientes, se desea testear

$$H_0 : p = 0.7 \quad \text{vs.} \quad H_1 : p < 0.7$$

Sea X el número de respuestas entre las 15 que ocurren dentro de los 10 minutos, entonces $X \sim Bi(15, p)$.

- Considerando la región de rechazo $R = \{0, 1, 2, 3, 4, 5, 6, 7\}$, calcule el nivel de significación α del test.
- Halle la función de potencia del test, $\pi(p)$, y la probabilidad de error de tipo II, $\beta(p)$ cuando $p = 0.5$ y $p = 0.3$.
- ¿Si se observa $X = 9$ y se utiliza la región de rechazo R , que concluiría Ud.? ¿Qué tipo de error pudo haberse cometido llegando a esa conclusión? Explique.

En nivel de significación de un test y la potencia tienen la misma forma ($Pr_\theta(\Phi(T) \in RR)$), salvo que para la significación, $\alpha = \sup_{\theta \in \Theta_0} Pr_\theta(\Phi(T) \in RR)$, y para la potencia, $\beta = Pr_{\theta \in \Theta_1}(\Phi(T) \in RR)$. Llamemos `prob_rechazo` a la función que calcula $Pr_\theta(\Phi(T) \in RR)$ para un θ y una región de rechazo RR en particular. Luego,

```
n <- 15
p_null <- 0.7
RR <- 0:7
prob_rechazo <- function(n, p, RR) {
  return(sum(dbinom(RR, n, p)))
}
potencia <- prob_rechazo
significacion <- prob_rechazo
prob_etII <- function(n, p_alt, RR) { 1 - potencia(n, p_alt, RR) }
(resultados$p1$ej2 <- list(
  a = significacion(n, p_null, RR),
  b = list(
    potencia = potencia,
    etII.5 = prob_etII(n, 0.5, RR),
    etII.3 = prob_etII(n, 0.3, RR))))

## $a
## [1] 0.05001254
##
## $b
```

```
## $b$potencia
## function(n, p, RR) {
##   return(sum(dbinom(RR, n, p)))
## }
## <bytecode: 0x17454f0>
##
## $b$etII.5
## [1] 0.5
##
## $b$etII.3
## [1] 0.05001254
```

Si se observa $X = 9 \in RR$, debería rechazar la hipótesis nula, y podría haber cometido un error de tipo I: rechazar H_0 cuando esta es verdadera, con probabilidad menor o igual a α .

Ejercicio 3

Una compañía que vende discos de música clásica por correo está tratando de decidir entre dos nuevas grabaciones de la Novena Sinfonía de Beethoven para agregar a su catálogo. Si ambas grabaciones son igualmente atractivas para los suscriptores ambas deben ser ofrecidas, mientras que si una es claramente preferida sobre la otra, entonces será la única ofrecida. Las hipótesis a testear son:

$$H_0 : p = 0.5 \quad vs. H_1 : p \neq 0.5$$

siendo p la proporción de suscriptores que prefieren la grabación A a la B. Se eligen 10 suscriptores al azar y a cada uno se le pide que escuche las dos grabaciones e indique su preferencia. Sea X el número de suscriptores que prefieren la grabación A.

- Calcule el nivel del test que rechaza H_0 si $X \leq 2 \vee X \geq 8$. ¿Es “el mejor test” de nivel α ?
- Calcule $\beta(p)$ para $p = 0.4, 0.6, 0.8$.
- Suponga que se observa $X = 9$. ¿Deberían ofrecerse ambas grabaciones? ¿Qué tipo de error pudo haberse cometido? Calcule el p-valor.
- Repetir los ítems anteriores para un nivel de 0.1.

```
n <- 10
p_null <- 0.5
RR <- c(0,1,2,8,9,10)

(resultados$p1$ej3 <- list(
  a = significacion(n, p_null, RR),
  b = list(
    etII.4 = prob_etII(n, 0.4, RR),
    etII.6 = prob_etII(n, 0.6, RR),
    etII.8 = prob_etII(n, 0.8, RR)))

## $a
## [1] 0.109375
##
## $b
## $b$etII.4
## [1] 0.8204157
##
## $b$etII.6
## [1] 0.8204157
##
## $b$etII.8
## [1] 0.3221225
```

Si se observa $X = 9$, se debe rechazar la hipótesis nula con un nivel de significación de 0.109375, y como en la hipótesis alternativa los clientes prefieren la grabación A a la B, tal vez se pueda ofrecer una sola grabación. Escribamos funciones para obtener el p-valor de un test, y una región de rechazo exacta, con nivel de significación menor o igual a uno deseado, para tests a dos colas. Vale aclarar, que solo se conseguirán RR verdaderamente simétricas bajo $H_0 : p = 0.5$. Si $p \neq 0.5$, un método razonable para obtener una RR, es tomar dos regiones de rechazo, con $\alpha_L, \alpha_U \leq \alpha/2$:

```
Tobs <- 9
p_valor <- function(Tobs, n, p_null) {
  # Solo validos para tests a dos colas
  pizq <- pbinom(Tobs, n, p_null) # P(Bi(n, p_null) <= Tobs)
  pder <- 1 - pbinom(Tobs - 1, n, p_null) # P(Bi(n, p_null) >= Tobs)
  return(2*min(pizq, pder))
}
(resultados$p1$ej3$c <- p_valor(Tobs, n, p_null))

## [1] 0.02148438

# DUDA: Cual es el p-valor si Tobs == p_null*n (0.5*10=5)? Mayor a 1?
# p_valor(5, 10, 0.5) = 1.246094 (!)

region_rechazo <- function(n, alfa, p_null) {
  crit_low <- qbinom(alfa/2, n, p_null) - 1
  crit_upp <- qbinom(1-alfa/2, n, p_null) + 1
  return(c(0:crit_low, crit_upp:n))
}

nueva_RR <- region_rechazo(n, 0.1, p_null)
nueva_significacion <- significacion(n, p_null, nueva_RR)
(resultados$p1$ej3$d <- list(
  nueva_RR = nueva_RR,
  nueva_significacion = nueva_significacion,
  etII.4 = prob_etII(n, 0.4, nueva_RR),
  etII.6 = prob_etII(n, 0.6, nueva_RR),
  etII.8 = prob_etII(n, 0.8, nueva_RR),
  rechazo_9 = 9 %in% nueva_RR,
  nuevo_p_valor = resultados$p1$ej3$c # El p-valor no depende de la RR, sino del Tobs
))

## $nueva_RR
## [1] 0 1 9 10
##
## $nueva_significacion
## [1] 0.02148438
##
## $etII.4
## [1] 0.9519649
##
## $etII.6
## [1] 0.9519649
##
## $etII.8
## [1] 0.6241862
##
## $rechazo_9
## [1] TRUE
```

```
##
## $nuevo_p_valor
## [1] 0.02148438
```

Ejercicio 4

Un empresario de la industria alimenticia asegura que menos del 10% de sus frascos de café instantáneo contiene menos café del que garantiza la etiqueta. Para probar esta afirmación se eligen al azar 15 frascos de café y se pesa su contenido. Su afirmación es aceptada si a lo sumo dos frascos contienen menos café del garantizado.

- ¿Qué hipótesis se deben testear?
- ¿Cuál es el nivel de la regla de decisión planteada? ¿Le parece razonable?
- Encuentre la probabilidad de que la afirmación del empresario sea aceptada cuando el porcentaje real de frascos que contienen menos café del garantizado en la etiqueta es 5%, 10% y 20%.
- Grafique la función de potencia del test planteado inicialmente. Muestre que es insesgado.
- Con el tamaño de muestra dado, ¿es posible obtener un test de nivel 0.05? Hallar el tamaño de muestra mínimo para obtener un test de nivel 0.05, manteniendo la misma región de rechazo que el test anterior.

Sea $X_i = 1$ si el i -ésimo frasco de café tiene menos café del que garantiza la etiqueta, y 0 en caso contrario, de manera que bajo H_0 , $X_i \stackrel{iid}{\sim} \text{Ber}(p = 0.1)$ y $\sum_i X_i = T \sim \text{Bi}(15, 0.1)$. Las hipótesis a testear serán:

$$H_0 : p \geq 0.1 \quad \text{vs.} \quad H_1 : p < 0.1$$

con región de rechazo $RR = \{0, 1, 2\}$. Los puntos (b) y (c) piden el nivel de significación y la probabilidad de error de tipo II, para lo cual ya hemos generado funciones. Para el punto (c), calculamos la función de potencia en una grilla de valores de p , y la graficamos con `ggplot`:

```
n <- 15
p_null <- 0.1 # Usamos el p_null de la igualdad para calcular la significacion
RR <- 0:2
alfa <- significacion(n, p_null, RR)
(resultados$p1$ej4 <- list(
  b = alfa,
  c = list(
    etII.05 = prob_etII(n, 0.05, RR),
    etII.1 = prob_etII(n, 0.1, RR),
    etII.2 = prob_etII(n, 0.2, RR)))
```

```
## $b
## [1] 0.8159389
##
## $c
## $etII.05
## [1] 0.03620024
##
## $etII.1
## [1] 0.1840611
##
## $etII.2
## [1] 0.6019768
```

Notese que para esta RR , como $Pr(\text{Bi}(15, 0.1) \in \{0, 1, 2\}) = 0.816$, cuando $p = 0.1$ rechazaremos la hipótesis

nula alrededor del 81.59% de las veces, lo cual no suena muy razonable. Aun así, el test Φ es insesgado cuando

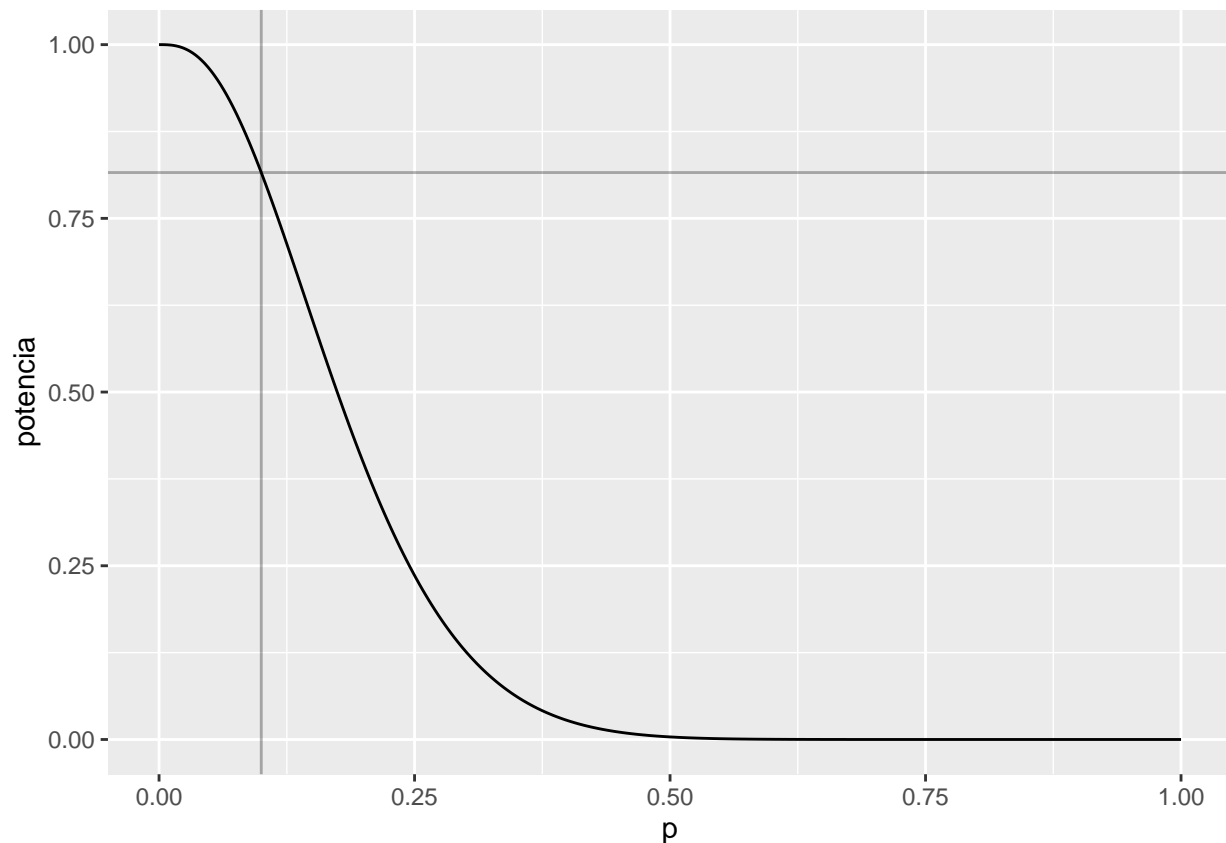
$$\begin{aligned} Pr(\text{rech } H_0 | H_0 \text{ Verdadera}) &< Pr(\text{rech } H_0 | H_0 \text{ Falsa}) \\ Pr(\Phi = 1 | H_0 \text{ V}) &< Pr(\Phi = 1 | H_0 \text{ F}) \\ Pr_{\theta_0}(\Phi = 1) &< Pr_{\theta_1}(\Phi = 1) \quad \forall \quad \theta_0 \in \Theta_0, \theta_1 \in \Theta_1 \end{aligned}$$

y como Φ es una unción indicadora, la probabilidad de que valga 1 es igual a su esperanza. Considerando que la desigualdad se tiene que cumplir para todo par de elementos en la hipótesis nula y la alternativa, concluimos que un test es insesgado cuando:

$$\begin{aligned} \forall \quad \theta_0 \in \Theta_0, \quad Pr_{\theta_0}(\Phi = 1) &\leq \sup_{\theta \in \Theta_0} E_{\theta}(\Phi) \\ \forall \quad \theta_1 \in \Theta_1, \quad Pr_{\theta_1}(\Phi = 1) &\geq \inf_{\theta \in \Theta_1} E_{\theta}(\Phi) \\ \Phi \text{ es insesgado} &\Leftrightarrow \sup_{\theta \in \Theta_0} E_{\theta}(\Phi) < \inf_{\theta \in \Theta_1} E_{\theta}(\Phi) \end{aligned}$$

Y recordemos que la función `prob_rechazo` es, justamente, la esperanza del test, bajo H_0 y H_1 . El siguiente gráfico muestra claramente la insesgadez del test, a pesar de su pesima significación.

```
densidad <- 0.001
graficos <- list()
(graficos$plej4d <- tibble(
  p = seq(0, 1, densidad),
  potencia = map_dbl(p, ~prob_rechazo(n, ., RR))) %>%
  ggplot(aes(p, potencia)) +
  geom_line() +
  geom_vline(xintercept = p_null, alpha = 0.3) + # Referencia: p_null
  geom_hline(yintercept = alfa, alpha = 0.3)) # Referencia: signif. de la RR
```



Manteniendo la RR propuesta, el punto (e) nos pide $\min n : Pr(Bi(n, 0.1) \in \{0, 1, 2\}) \leq 0.05$. Escribamos la funcion que lo busca:

```
n_requerido <- function(RR, alfa, p_null, max_n = 10000) {
  n_req <- 0
  while (n_req < max_n) {
    if (significacion(n_req, p_null, RR) <= alfa) {
      return(n_req)
    } else {
      n_req <- n_req + 1
    }
  }
  return(NA) # Devuelve NA si no encuentra n antes de max_n
}
(resultados$p1$ej4$e <- n_requerido(RR, 0.05, p_null))
```

```
## [1] 61
```

Ejercicio 6

En un juego se tiró 180 veces un par de dados y 38 veces se obtuvo suma de puntos igual a 7. Halle un intervalo de confianza de nivel asintótico 0.95 para $P(X = 7)$ siendo X la suma de puntos. ¿Hay razones para creer que los dados no están equilibrados?

Sean X_1, X_2 los puntajes de los dos dados, independientes y balanceados bajo la hipotesis nula. En ese caso, $X = X_1 + X_2$ y $Pr(X = 7 | X_1 = x) = Pr(X_2 = 7 - x) = 1/6 \forall x \in \{1 : 6\}$. Luego, si $Y_i = I(X=7)$ en la i -ésima tirada, $Y_i \stackrel{iid}{\sim} Ber(p = 1/6) \forall i \in \{1 : n\}$. Cualquier \bar{Y}_n observado significativamente distinto a $1/6$ debería hacernos dudar de lo equilibrado de los dados.

H_0 : Los dados estan equilibrados vs. H_1 : Los dados estan sesgados

$H_0 : p = 1/6$ vs. $H_1 : p \neq 1/6$

```
n <- 180
Tobs <- 38
p_null <- 1/6
alfa <- 1 - 0.95
(resultados$p1$e6 <- list(
  ic_asintotico = ic_asint_prop(n, Tobs, alfa),
  p_valor = p_valor(Tobs, n, p_null))) # p-valor exacto, no asintotico
```

```
## $ic_asintotico
## [1] 0.1514934 0.2707288
##
## $p_valor
## [1] 0.1399347
```

Veamos que $p_0 = 1/6$ pertenece al IC asintotico obtenido, lo cual nos hace suponer que no hay razones para creer que los dados estan sesgados. Alternativamente, podemos calcular el p-valor exacto del estadistico observado (38), que nos da 0.1399347, mayor al nivel de significacion planteado, de manera que no rechazamos H_0 .

Ejercicio 7

En la siguiente tabla se presentan 20 observaciones independientes correspondientes a una v.a. X con distribución desconocida $F(x)$:

- a. Halle un intervalo de confianza de nivel 0.95 para $F(100)$. Compararlo con el que quedaría para muestras grandes.

Sea $F(k) = P(X \leq k) = p$. Luego, necesitamos un IC “exacto” para p . Si $T(k) = \sum_{i=1}^n I(X_i \leq k) \Rightarrow T(k) \sim Bi(n, p = F(k))$. El IC exacto (metodo “A” de los apuntes) se puede escribir como el conjunto de valores de p tales que dado T^{obs} , no rechazariamos ninguno de los dos tests unilaterales con nivel de significacion $1 - \alpha$. Sean:

$$\begin{aligned} \sup S_{\leq} &= \hat{p}_{upp} : T \sim Bi(n, \hat{p}_{upp}) \Rightarrow Pr(T \leq T^{obs}) = F_T(T^{obs}) = \alpha/2 \\ \inf S_{\geq} &= \hat{p}_{low} : T \sim Bi(n, \hat{p}_{low}) \Rightarrow Pr(T \geq T^{obs}) = 1 - F_T(T^{obs}) = \alpha/2 \\ &\Rightarrow IC(T^{obs}, \alpha) = (\hat{p}_{low}, \hat{p}_{upp}) \end{aligned}$$

```
ic_exacto_prop <- function(n, Tob, alfa, densidad = 1e-3) {
  # Adaptado de https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval,
  # seccion "Clopper-Pearson interval"
  # Busco el supremo del conjunto  $S_{\{leq\}}$ , de izquierda a derecha
  sup_leq <- 0
  while (significacion(n, sup_leq, 0:Tobs) > alfa/2) { sup_leq <- sup_leq + densidad }
  # Busco el infimo del conjunto  $S_{\{geq\}}$ , de derecha a izquierda
  inf_geq <- 1
  while (significacion(n, inf_geq, Tob:n) > alfa/2) { inf_geq <- inf_geq - densidad }
  return(c(inf_geq, sup_leq))
}

# En el caso del IC para un cuantil  $F(k)$ , en vez de proveer n y Tob directamente,
# proveemos la muestra aleatoria X y el cuantil k, sea este asintotico o exacto:
ic_cuantil <- function(X, k, alfa, metodo = 'exacto', ...) {
  n <- length(X)
  Tob <- sum(obs <= k)
  if (metodo=="exacto") {
    return(ic_exacto_prop(n, Tob, alfa, ...))
  } else if (metodo=="asintotico") {
    return(ic_asint_prop(n, Tob, alfa))
  } else {
    stop("`metodo` no soportado. Pruebe con 'exaco' o 'asintotico'")
  }
}

k <- 100
alfa <- 1 - 0.95
(resultados$p1$ej7$a <- list(
  ic_exacto = ic_cuantil(obs, k, alfa, "exacto"),
  ic_asint = ic_cuantil(obs, k, alfa, "asintotico")))
```

```
## $ic_exacto
## [1] 0.057 0.437
##
## $ic_asint
## [1] 0.02469549 0.37530451
```

- b. Testee la hipótesis de que la mediana es 103.

(Guía 2, p 24) Una forma de testear esta hipótesis, es plantear dos tests simultáneos de nivel $\alpha_1, \alpha_2 : \alpha_1 + \alpha_2 \leq \alpha$. Para simplificar, tomamos ambos iguales a $\alpha/2$. “Testear” es sinónimo de “calcular el p-valor y comparar con el nivel de significación, así que tendremos que reescribir nuestra función `p_valor` para que permita testear alternativas unilaterales:

```
p_valor <- function(Tobs, n, p_null, alternativa="dos colas") {
  pizq <- pbinom(Tobs, n, p_null) #  $P(Bi(n, p_{null}) \leq Tobs)$ 
  pder <- 1 - pbinom(Tobs - 1, n, p_null) #  $P(Bi(n, p_{null}) \geq Tobs)$ 
  if (alternativa=="dos colas") { return(2*min(pizq, pder)) }
  } else if (alternativa=="mayor") { return(pder) }
  } else if (alternativa=="menor") { return(pizq) }
  } else {stop("`alternativa` debe ser una de 'dos colas', 'mayor' o 'menor'") }
}
```

```
test_puntual_cuantil <- function(obs, k, p_null = 0.5, alfa = 0.05) {
  # Testea si `k` es el cuantil `p_null`, a dos colas.
  n <- length(obs)
  T1 <- sum(obs <= k) # estadístico para 'p' en los apuntes,
  T2 <- sum(obs < k) # estadístico para 'p tilde' en los apuntes.
  p_valor_T1 <- p_valor(T1, n, p_null, alternativa='menor')
  p_valor_T2 <- p_valor(T2, n, p_null, alternativa='mayor')
  return (list(
    p_valor_T1 = p_valor_T1,
    p_valor_T2 = p_valor_T2,
    # Rechazo la hipótesis conjunta si rechazo alguna de las individuales
    rechazo_H0 = any(p_valor_T1 < alfa/2, p_valor_T2 < alfa/2)))
}
(resultados$p1$ej7$b <- test_puntual_cuantil(obs, 103, 0.5, alfa = 0.05))
```

```
## $p_valor_T1
## [1] 0.05765915
##
## $p_valor_T2
## [1] 0.9987116
##
## $rechazo_H0
## [1] FALSE
```

- c. Encuentre un intervalo de nivel 0.90 para la mediana. Compararlo con el que quedaría para muestras grandes. Sigo sin entender verdaderamente este tema. Aplicando métodos como un buen marinero,

```
alfa <- 0.1
n <- 20

ic_percentil <- function(obs, perc, alfa, metodo = "exacto") {
  n <- length(obs)
  obs_ordenadas <- sort(obs)
  if (metodo=="exacto") {
    r <- 0
    # Si r+1 aun no acumula alfa/2, considerarlo
    while (pbinom(r, n, perc) < alfa/2) { r <- r + 1 }
    s <- 0
    while (pbinom(s-1, n, perc) < 1 - alfa/2) { s <- s + 1 }
  } else if (metodo=="asintotico") {
    media <- n*perc
```

```

varianza <- n*perc*(1-perc)
r <- ceiling(qnorm(alfa/2, media, sqrt(varianza)))
s <- ceiling(qnorm(1-alfa/2, media, sqrt(varianza)))
} else { stop("`metodo` debe ser 'exacto' o 'asintotico'") }

return(list(
  r = r, s = s, #Ambos tienen el "+1" incluido
  ic = obs_ordenadas[c(r, s)]))
}

percentil <- 0.5 # mediana
alfa <- 1 - 0.9
(resultados$p1$ej7$c <- list(
  ic_exacto = ic_percentil(obs, percentil, alfa, "exacto"),
  ic_asintotico = ic_percentil(obs, percentil, alfa, "asintotico")))

## $ic_exacto
## $ic_exacto$r
## [1] 6
##
## $ic_exacto$s
## [1] 15
##
## $ic_exacto$ic
## [1] 103 142
##
##
## $ic_asintotico
## $ic_asintotico$r
## [1] 7
##
## $ic_asintotico$s
## [1] 14
##
## $ic_asintotico$ic
## [1] 117 137

```

Ejercicio 9

¿Cuál debe ser el tamaño de la muestra para tener un 90% de seguridad de que el rango muestral incluye al menos al 95% de la población? a. Use una tabla exacta. b. Use la aproximación.

Empecemos por la aproximación que es fácil de computar:

```

tol_n_aprox <- function(r, m, q, alfa) {
  crit <- qchisq(1-alfa, 2*(r+m))
  return( crit/4 * (1+q)/(1-q) + (r+m-1)/2 )
}
tol_n_aprox(1, 1, 0.95, 0.1)

```

```
## [1] 76.34954
```

Los ejercicios 10 y 11 son idénticos.

Practica 2

Ejercicio 1

```
# a: ara obtener el k critico
n <- 10
p_null <- 0.5
pbinom(0:n, n, p_null)

## [1] 0.0009765625 0.0107421875 0.0546875000 0.1718750000 0.3769531250
## [6] 0.6230468750 0.8281250000 0.9453125000 0.9892578125 0.9990234375
## [11] 1.0000000000

RR <- 7:n
# b: para calcular la potencia bajo la alternativa N(1,1)
p_alt <- 1 - pnorm(0, 1, 1) # Pr(X_i > 0) bajo N(1,1),
# b.i: exacta
potencia(n, p_alt, RR)

## [1] 0.940255

# c aproximada
1 - pnorm(7-1/2, n*p_alt, sqrt(n*p_alt*(1-p_alt)))

## [1] 0.9511553

# b.ii
# La alternativa parece una doble exponencial con media/mediana en 1
p_alt <- 1 - exp(-1)/2 # Pr(X_i > 0) bajo DobleExp(mediana=1)
potencia(n, p_alt, RR)

## [1] 0.9056837
```

Ejercicio 3

```
obs <- read.csv(file.path(DATA_DIR, "p2e3.csv"), header = FALSE)[[1]]
n <- length(obs)
# Testeo tita = 90 versus distinto, asi que
tita <- 90
Tobs <- sum(obs - tita > 0)
# Para el test del signo, p_null siempre sera 0.5
# p valor exacto
p_valor(Tobs, n, p_null = 0.5, alternativa = "dos colas")

## [1] 0.2668457

# p valor aproximado
pnorm(Tobs, n/2, sqrt(n/4)) * 2

## [1] 0.1655179

# Corregido por continuidad
pnorm(Tobs + 1/2, n/2, sqrt(n/4)) * 2

## [1] 0.2672575

# estimador puntual
median(obs)
```

```
## [1] 73
# IC 90%
ic_percentil(obs, 0.5, 0.1)

## $r
## [1] 4
##
## $s
## [1] 10
##
## $ic
## [1] 43 105
```

Ejercicio 4

```
Tmas <- 33
Tmenos <- 16
Tcerro <- 12
# Contando no-cruces como casos negativos
p_valor(Tmas, (Tmas + Tmenos + Tcerro), 0.5, alternativa="mayor")

## [1] 0.3044604
# Descartando no-cruces (AKA "empates", en G3p33 EM recomienda este metodo)
p_valor(Tmas, (Tmas + Tmenos), 0.5, alternativa="mayor")

## [1] 0.01064706
```

Ejercicio 5

```
obs <- read.csv(file.path(DATA_DIR, "p2e5.csv"))
D <- obs$despues - obs$antes
n <- length(D)
Tobs <- sum(D > 0) # expresado en terminos de Tmas
p_valor_p2e5 <- p_valor(Tobs, n, 0.5, alternativa = "menor")
alfa <- 0.05
p_valor_p2e5 < alfa

## [1] FALSE
# Aun con 1 positivo en 6, el p-valor es demasiado alto, no rechazo.
```

Ejercicio 7

```
# Parte A: 30 de las 37 finalmente en contra arrancan a favor
X0Y1 <- 30 # "b" en los apuntes y Conover
X1Y0 <- 36 # "c" en los apuntes y Conover
T1 <- (X0Y1 - X1Y0)^2 / (X0Y1 + X1Y0)
T2 <- X0Y1
# T2 ('b' o 'X0Y1') distribuye Bi(b + c, 1/2).
# Se recomienda para b+c<=20 pero lo computo igual para practicar
(p_valor_T2 <- p_valor(Tobs = X0Y1, n =(X0Y1 + X1Y0), p_null = 0.5,
  alternativa = "dos colas"))

## [1] 0.5385828
```

```
# T1 tiene distribucion chisq(df=1), asi que calculamos el p-valor a mano:
(p_valor_T1 <- 1 - pchisq(T1, 1))
```

```
## [1] 0.4601809
```

Bajo cualquier α razonable, el cambio no es significativo.

```
# Parte B: las 37 finalmente en contra estaban originalmente en contra
X0Y1 <- 0 # "b" en los apuntes y Conover
X1Y0 <- 6 # "c" en los apuntes y Conover
T1 <- (X0Y1 - X1Y0)^2/(X0Y1 + X1Y0)
T2 <- X0Y1
(p_valor_T2 <- p_valor(Tobs = X0Y1, n =(X0Y1 + X1Y0), p_null = 0.5,
                        alternativa = "dos colas"))
```

```
## [1] 0.03125
```

```
# T1 se recomienda para b+c>20 pero lo computo igual para practicar
(p_valor_T1 <- 1 - pchisq(T1, 1))
```

```
## [1] 0.01430588
```

Bajo $\alpha = 0.05$, ahora el cambio en el numero de personas opuestas si es significativo.

DUDA: Bajo H_0 , se supone que *entre quienes cambian de opinion*, es igual de probable que cambien de 0 a 1, que de 1 a 0, pero la direccion del cambio, *entre los que cambiaron*, esta determinada por la opinion inicial. Luego, esta hipotesis seria equivalente a decir que “la probabilidad de un cambio de opinion, es inversamente proporcional a la prevalencia original de esa opinion”?

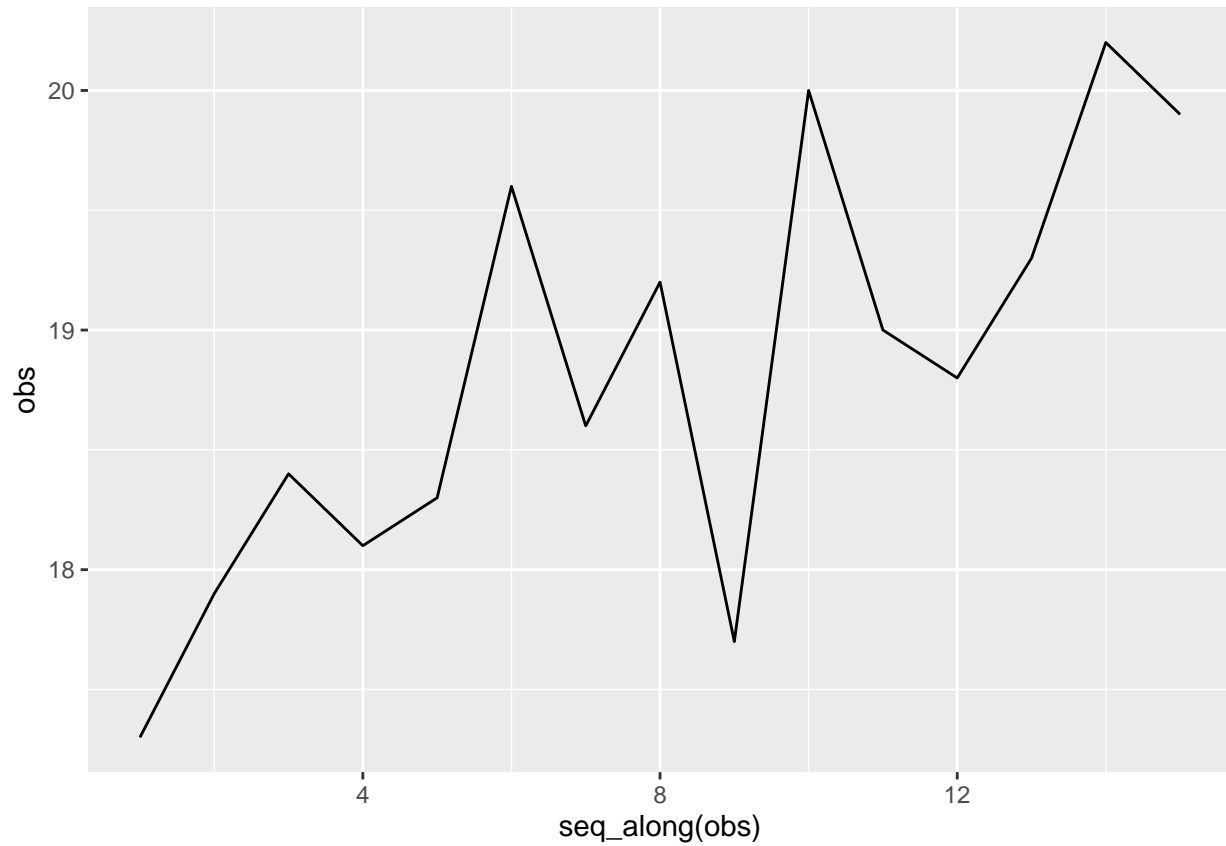
Ejercicio 8

```
obs <- read.delim(file.path(DATA_DIR, "p2e8.csv"), header = FALSE)[[1]]
cox_stuart <- function(obs, alternativa="dos colas") {
  n_obs <- length(obs)
  n <- floor(n_obs/2) # cantidad de pares
  D <- obs[(n_obs-n+1):n_obs] - obs[1:n]
  Tobs <- sum(D>0)
  return(list(
    p_valor = p_valor(Tobs, n, 0.5, alternativa),
    D = D, n = n, Tobs = Tobs))
}
cox_stuart(obs, "mayor")
```

```
## $p_valor
## [1] 0.0078125
##
## $D
## [1] 0.4 2.1 0.6 0.7 1.0 0.6 1.3
##
## $n
## [1] 7
##
## $Tobs
## [1] 7
```

Hay bases para rechazar la hipotesis nula “La tasa de accidentes no aumenta” ($\approx \theta \leq 0$). Veamos los datos en un grafico para convencernos:

```
ggplot(mapping=aes(seq_along(obs), obs)) + geom_line()
```

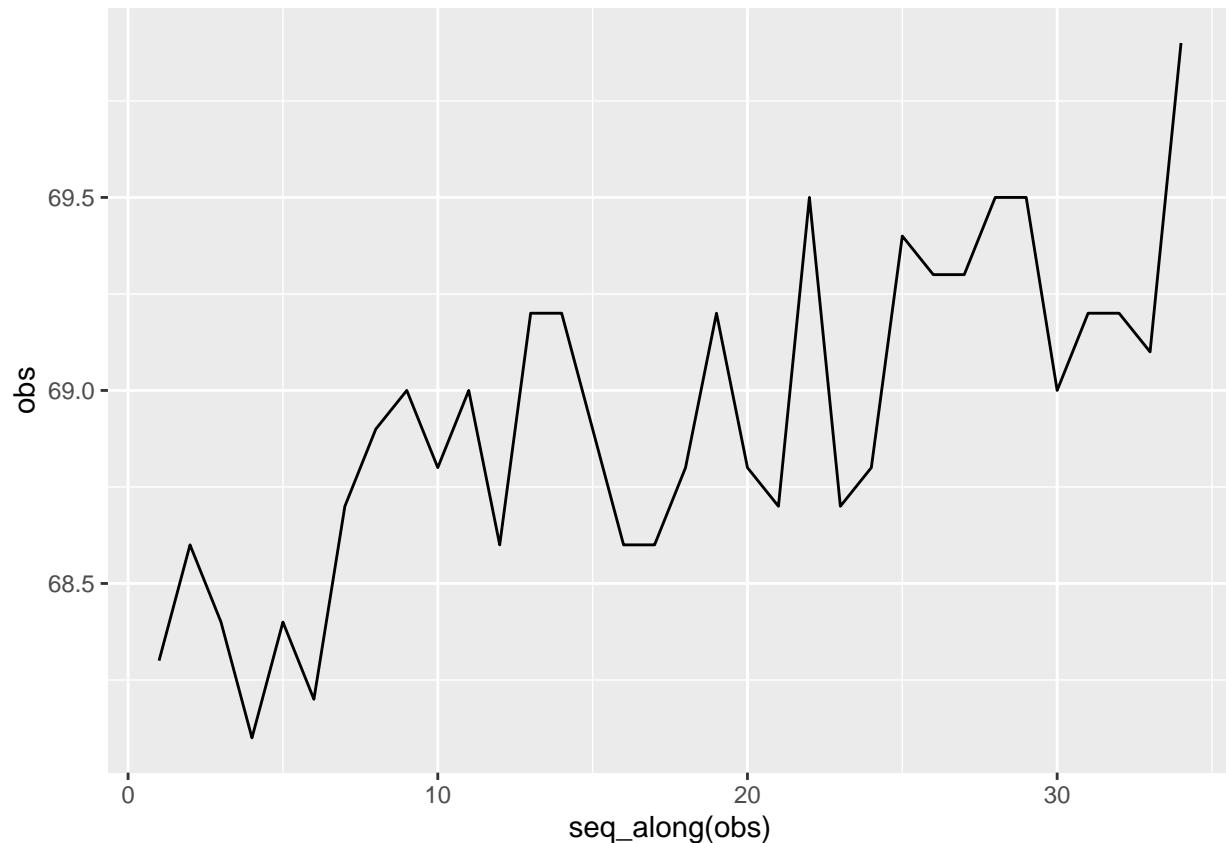


Ejercicio 9

```
obs <- read.delim(file.path(DATA_DIR, "p2e9.csv"), header = FALSE)[[1]]
cox_stuart(obs, "mayor")
```

```
## $p_valor
## [1] 0.001174927
##
## $D
## [1] 0.5 0.6 0.4 0.6 1.1 0.5 0.1 0.5 0.3 0.5 0.5 0.9 -0.2 0.0
## [15] 0.3 0.5 1.3
##
## $n
## [1] 17
##
## $Tobs
## [1] 15
```

```
ggplot(mapping=aes(seq_along(obs), obs)) + geom_line()
```

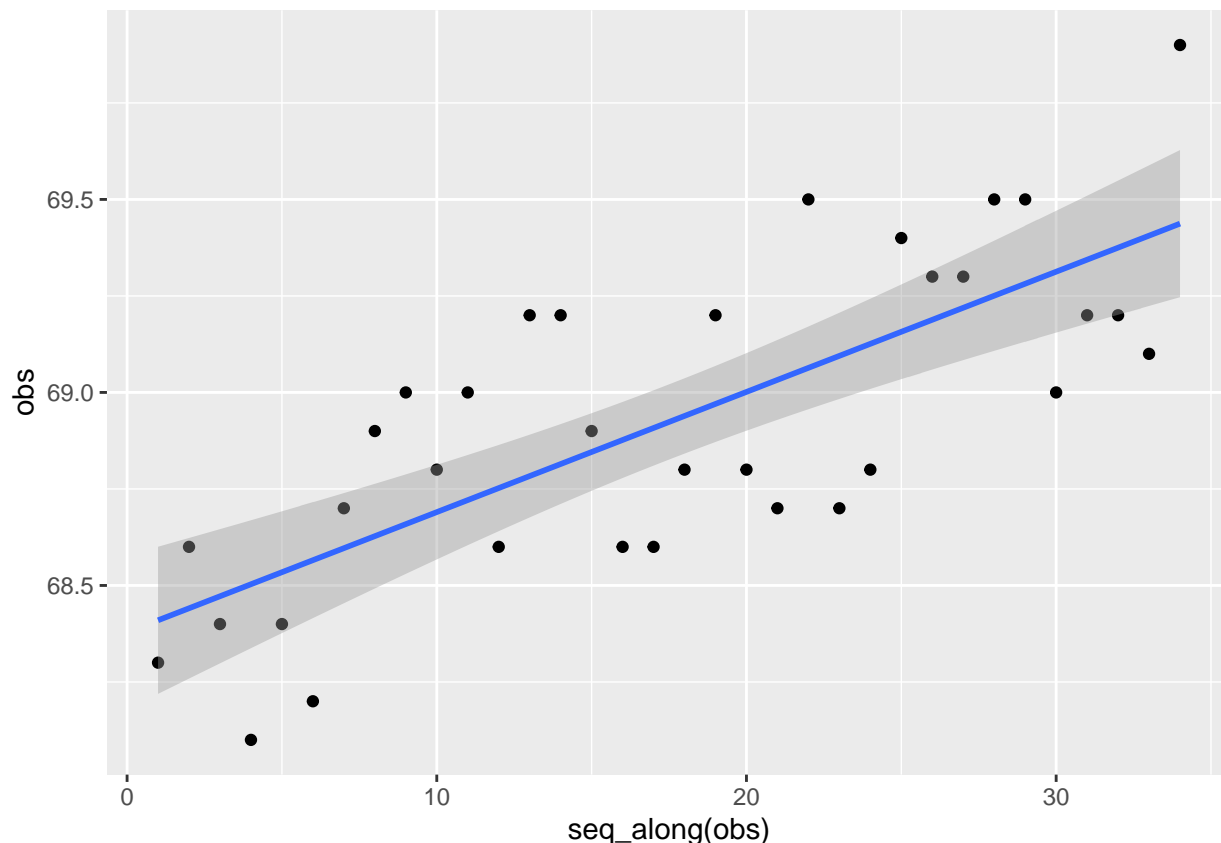


Alternativamente, podemos usar un test parametrico para testear la presencia de una tendencia *lineal* en los datos, que da un aumento interanual de ~ 0.03 pulgadas/año:

```
lm(obs ~ seq_along(obs)) %>% summary

##
## Call:
## lm(formula = obs ~ seq_along(obs))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40316 -0.25796 -0.00885  0.23964  0.46269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  68.378610   0.097793  699.215 < 2e-16 ***
## seq_along(obs)  0.031138   0.004874   6.388 3.55e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2788 on 32 degrees of freedom
## Multiple R-squared:  0.5605, Adjusted R-squared:  0.5467
## F-statistic: 40.81 on 1 and 32 DF,  p-value: 3.546e-07

ggplot(mapping=aes(seq_along(obs), obs)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Ejercicio 10

El enunciado es confuso, pero lo razonable es suponer que *el sujeto A es el complice*, y deseamos testear si hay correlacion positiva entre la sugestion de A y la respuesta genuina de B. En la G4p49, E.M. sugiere ordenar los pares a partir de la variable con menos empates (aunque no queda claro si se refiere a “mayor cantidad de valores unicos” o “menor cantidad de ‘grupos de valores’ empatados”), y “usar aquel ordenamiento que rechaza H_0 con menos probabilidad”.

En este caso, A tiene 17 valores unicos, y B solo 15, asi que deberiamos ordenar los pares segun A. Sin embargo, en este ejemplo considero razonable ordenar los pares (A, B) *siempre* segun A, porque el experimentador esta *explicitamente* interesado en saber si lo que dice A (controlado) afecta/sugestiona a B (lo que se testea).

Las hipotesis a testear son;

$$H_0 : \text{No existe correlacion positiva} \quad \text{vs.} \quad H_1 : \text{Existe correlacion positiva}$$

Que en terminos de Cox-Stuart, es equivalente a suponer que los pares $X_{i+c} - X_i = D_i \sim F_D(t - \theta)$, y testeamos $H_0 : \theta \leq 0$ vs. $H_1 : \theta > 0$. No es evidente cual es el ordenamiento de los B_i que corresponden a empates en A_i que menos rechaza H_0 , pero dado que estamos buscando una tendencia positiva (creciente), una heuristica razonable es ordenarlos de manera opuesta (i.e., decreciente):

```
df <- read.csv(file.path(DATA_DIR, "p2e10.csv"))
valores_unicos <- df %>%
  summarise(
    unicos.A = n_distinct(Sujeto.A),
    unicos.B = n_distinct(Sujeto.B))

(cox_stuart_desc <- df %>%
  arrange(Sujeto.A, desc(Sujeto.B)) %>%
```



```

pull("Sujeto.B") %>%
cox_stuart(alternativa="mayor"))

## $p_valor
## [1] 0.01074219
##
## $D
## [1] 0.0 1.4 1.1 0.9 1.4 1.4 0.9 0.5 1.6 0.2
##
## $n
## [1] 10
##
## $Tobs
## [1] 9

(cox_stuart_asc <- df %>%
  arrange(Sujeto.A, Sujeto.B) %>%
  pull("Sujeto.B") %>%
  cox_stuart(alternativa="mayor"))

## $p_valor
## [1] 0.0009765625
##
## $D
## [1] 0.6 0.8 0.9 1.1 1.4 1.4 0.9 0.5 1.6 0.2
##
## $n
## [1] 10
##
## $Tobs
## [1] 10

```

Notese que en el primer test (con los B ordenados contra la tendencia), el p-valor ronda 0.01, mientras que con los B ordenados a favor de la tendencia, el p-valor es aproximadamente 0.01, 10 veces (!) mas bajo que antes. En ambos casos se rechaza la hipotesis nula para $\alpha = 0.05$, pero si fuésemos exigentes y usamos $\alpha = 0.01$, el test mas conservador ya no rechaza, y el menos conservador si.

DUDA: Entiendo que los elementos de D (pares de la serie a testear por correlacion) se conservan aun cuando son cero, ya que tales valores son evidencia de una no-tendencia. O se espera que eliminemos pares?

Ejercicio 11

El diseño del experimento sugiere la conveniencia de usar el test del signo para observaciones apareadas, antes y despues de la inmunizacion, y testear:

H_0 : La inmunizacion NO aumenta la concentracion de anticuerpos vs. H_1 : Si lo hace

$$X_i^{post} - X_i^{pre} = D_i \sim F_D(t - \theta) \Rightarrow H_0 : \theta \leq 0 \quad \text{vs.} \quad H_1 : \theta > 0$$

```

df <- read.csv(file.path(DATA_DIR, "p2e11.csv"))
# Compruebo que las diferencias esten bien calculadas, con tolerancia numerica
stopifnot(abs(df$Diferencia - (df$X4.semanas.después - df$Antes)) < 1e-7)
D <- df$Diferencia
Tobs <- sum(D > 0) # expresado en terminos de Tmas
(p_valor_con_empates <- p_valor(Tobs, n = length(D), 0.5, alternativa = "mayor"))

```

```
## [1] 0.4119015
(p_valor_sin_empates <- p_valor(Tobs, n = sum(D!=0), 0.5, alternativa = "mayor"))

## [1] 0.003173828
# Comparo con el test t parametrico para la media de las diferencias, que no es
# significativamente distinta de cero, como reportaron los autores.
t.test(D)

##
## One Sample t-test
##
## data: D
## t = 1.8498, df = 19, p-value = 0.07996
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1551563 2.5151563
## sample estimates:
## mean of x
## 1.18
```

En resumen, $p\text{-valor}_{s/emp} < \alpha = 0.05 < p\text{-valor}_t < p\text{-valor}_{c/emp}$. Si seguimos la metodología habitual de "descartar los empates (diferencias iguales a cero), nos queda una muestra de 12 elementos $c/11$ positivos, y el p-valor resultante justifica rechazar la hipótesis nula y declarar el tratamiento significativo.

Como estamos testeando la eficacia de un tratamiento de inmunización, parece poco sabio descartar los casos en que la inmunización no produjo cambio alguno, y decidir en base al $p\text{-valor}_{c/emp}$, que se condice con los autores de la investigación.

DUDA: Existen casos donde sea razonable *no rechazar* con test paramétrico y *si rechazar* con uno no paramétrico? No deberían ser siempre los no-paramétricos más conservadores?

Practica 3

Ejercicio 1

Halle la distribución bajo H_0 del estadístico T^+ del test de rangos signados de Wilcoxon para $n = 5$.

Hallemos su distribución en general, que para algo existen las computadoras:

```
crear_memoria <- function(n) { matrix(nrow=n*(n+1)/2, ncol=n) }
memTmas <- crear_memoria(200)
prob_Tmas <- function(k, n) {
  if (n < 0) stop("`n` debe ser >=0")
  if (k < 0) return(0)
  if (n == 0) return(if (k == 0) 1 else 0)
  if (k == 0) return(2^-n)
  if (!is.na(memTmas[k,n])) return(memTmas[k,n])
  valor <- if (k > n*(n+1)/2) { prob_Tmas(k=(n*(n+1)/2 - k), n=n)
  } else { (prob_Tmas(k=k, n=n-1) + prob_Tmas(k=k-n, n=n-1))/2 }
  memTmas[k,n] <- valor
  return(valor)
}

dTmas <- function(x, n) {
  # Devuelve Pr(Tmas = xi, n=n) para cada xi en x
  return(map_dbl(x, prob_Tmas, n = n))
}
```

```

}

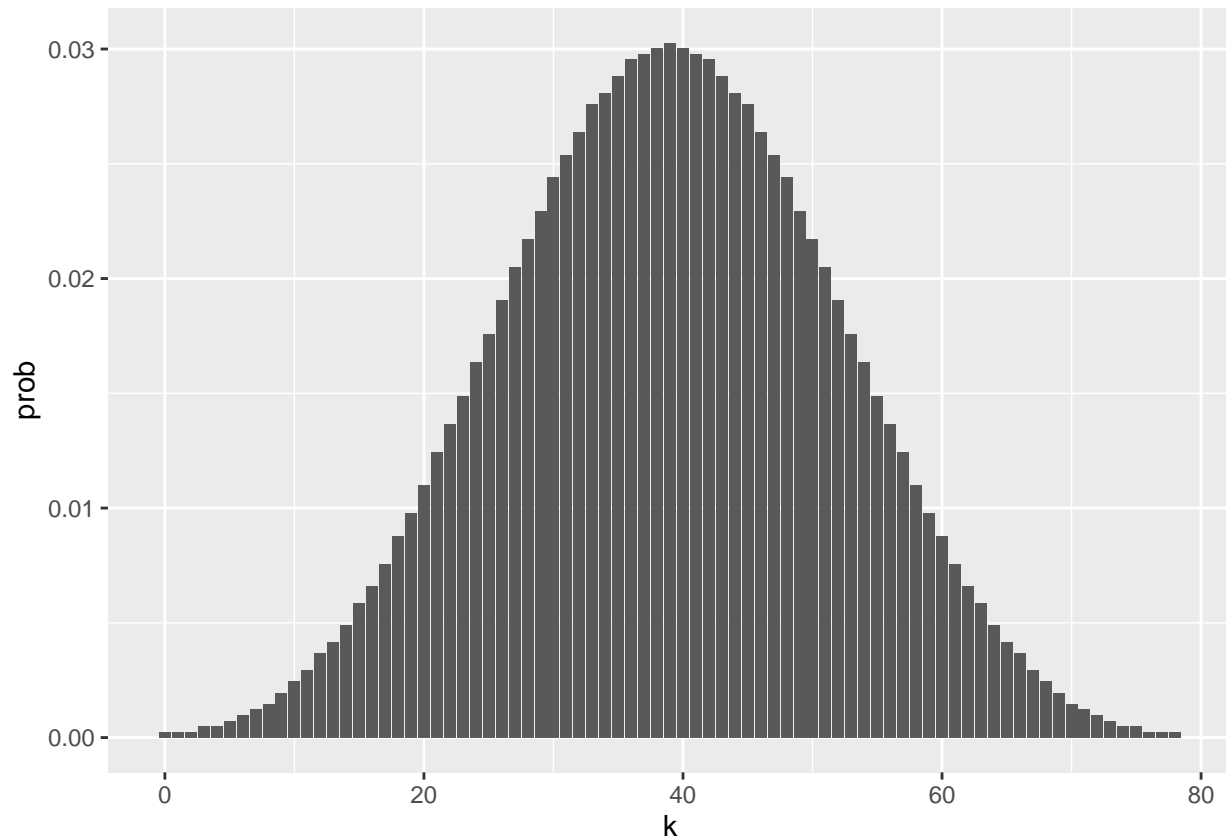
pTmas <- function(q, n) {
  # Devuelve Pr(Tmas <= k) para n = n, y k desde 0 hasta n*(n+1)/2
  max_q <- max(q)
  pT <- dTmas(0:max_q, n)
  res <- vector("double", length(q))
  for (i in seq_along(q)) {
    res[i] <- sum(pT[1:(q[[i]]+1)])
  }
  return(res)
}

# La distribucion no es mas que la acumulada sobre todo el rango de Tmas(5)
distr_Tmas <- function(n) { pTmas(0:((n*(n+1)/2)), n) }
distr_Tmas(5)

## [1] 0.03125 0.06250 0.09375 0.15625 0.21875 0.31250 0.40625 0.50000
## [9] 0.59375 0.68750 0.78125 0.84375 0.90625 0.93750 0.96875 1.00000

# Pero divirtamosnos un poquito
n <- 12
# Y un histograma para aprovechar
tibble(
  k = 0:(n*(n+1)/2),
  prob = dTmas(k, n)) %>%
  ggplot(aes(k, prob)) + geom_col()

```



Antes de encarar los ejercicios de la guía, vamos con algunas funciones caseras para implementar un test equivalente al `wilcox.test` de R base.

```

rangos_signados <- function(obs) {
  # Con el compositor %>%, seria
  # obs %>% abs %>% rank %>% subset(obs > 0)
  # No es mas claro y bonito ?
  rank(abs(obs))[obs > 0]
}

suma_rangos_signados <- function(obs) {
  # Con el compositor %>%, seria
  # obs %>% abs %>% rank %>% subset(obs > 0) %>% sum
  # No es mas claro y bonito ?
  sum(rangos_signados(obs))
}

p_valor.wilcoxon <- function(Tobs, n, alternativa="dos colas") {
  pizq <- sum(dTmas(0:Tobs, n))
  pder <- sum(dTmas(Tobs:(n*(n+1)/2), n))
  if (alternativa=="dos colas") { return(2*min(pizq, pder)) }
  } else if (alternativa=="mayor") { return(pder) }
  } else if (alternativa=="menor") { return(pizq) }
  } else {stop("`alternativa` debe ser una de 'dos colas', 'mayor' o 'menor'")}
}

p_valor.normal <- function(Tobs, mu=0, sigma=1, alternativa="dos colas",
                           corregir=FALSE) {
  cc <- if (corregir) 1/2 else 0
  pizq <- pnorm(Tobs + cc, mu, sigma)
  pder <- 1 - pnorm(Tobs - cc, mu, sigma)
  if (alternativa=="dos colas") { return(2*min(pizq, pder)) }
  } else if (alternativa=="mayor") { return(pder) }
  } else if (alternativa=="menor") { return(pizq) }
  } else {stop("`alternativa` debe ser una de 'dos colas', 'mayor' o 'menor'")}
}

p_valor.binom <- p_valor

# Calcula los promedios de walsh, necesarios para el estimador HL para tita
walsh <- function(x) {
  medias <- apply(combn(x, 2), 2, mean) # promedios con i != j
  return(c(x, medias))
}

test.wilcoxon <- function(
  x, y = NULL, alternativa="dos colas", mu=0, apareadas=FALSE, exacto=TRUE,
  corregir=TRUE, int_conf=FALSE, alfa = 0.05, sin_ceros=TRUE) {
  obs <- if (is.null(y)) {
    x - mu
  } else if (apareadas) {
    x - y
  } else {
    # Todos los pares de diferencias xi - yi
    expand.grid(x = x, y = y) %>% mutate(d = x - y) %>% pluck("d")
  }
}

```

```

}

if (any(obs == 0)) {
  warning("No se puede calcular el p-valor exacto con ceros")
  exacto <- FALSE
}
if (length(obs) != length(unique(obs))) {
  warning("No se puede calcular el p-valor exacto con empates")
  exacto <- FALSE
}
if (sin_ceros) { obs <- obs[obs!=0] }
n <- length(obs)
rangos <- rangos_signados(obs)
Tobs <- if (apareadas | is.null(y)) {
  suma_rangos_signados(obs)
} else {
  # Mann-Whitney-Wilcoxon's  $W = \sum(i, j) s(X_i - Y_j)$ 
  sum(obs>0)
}

p_valor <- if (exacto) {
  p_valor.wilcoxon(Tobs, n, alternativa) }
else {
  muT <- n*(n+1)/4
  sigmaT <- sqrt(n*(n+1)*(2*n+1)/24)
  p_valor.normal(Tobs, muT, sigmaT, alternativa, corregir)
}
if (int_conf) {
  w <- sort(walsh(obs))
  estimado <- median(w)
  if (alternativa=="dos colas") {
    if (exacto) {
      a <- 0
      while (pTmas(a + 1, n) <= alfa/2) { a <- a + 1 }
    } else {
      a <- qnorm(alfa/2, muT - 1/2, sigmaT)
    }
    intervalo <- w[c(a+1, n*(n+1)/2-a)]
  } else {
    if (exacto) {
      a <- 0
      while (pTmas(a + 1, n) <= alfa) { a <- a + 1 }
    } else {
      a <- qnorm(alfa, muT - 1/2, sigmaT)
    }
    intervalo <- if (alternativa=="menor") {
      c(-Inf, w[n*(n+1)/2-a])
    } else if (alternativa=="mayor") {
      c(w[a+1], Inf)
    } else { stop("`alternativa` no reconocida.") }
  }
}
return(list(

```

```

estadistico = Tobs,
p_valor = p_valor,
tita_nulo = mu,
alternativa = alternativa,
estimado = if (int_conf) estimado else NULL,
int_conf = if (int_conf) intervalo else NULL,
parametros = list(
  obs = obs,
  n = n,
  ceros = sum(obs == 0),
  sin_ceros = sin_ceros,
  apareadas = apareadas,
  exacto = exacto,
  corregir = corregir,
  a = if (int_conf) a else NULL,
  alfa = alfa)
))
}

```

Ejercicio 2

Una muestra aleatoria de 20 personas que manejan fue seleccionada para ver si el alcohol afectaba el tiempo de reacción. Cada tiempo de reacción fue medido en el laboratorio antes y después de beber determinada cantidad de alcohol. ¿Hay evidencias de que el alcohol afecta el tiempo de reacción?

H_0 : El alcohol NO afecta el tiempo de reaccion vs. H_1 : Si lo hace

$$X_i^{post} - X_i^{pre} = D_i \sim F_D(t - \theta) \Rightarrow H_0 : \theta = 0 \text{ vs. } H_1 : \theta \neq 0$$

Usaremos nuestro test y el de R base para asegurarnos que los resultados sean similares:

```

df <- read.csv(file.path(DATA_DIR, "p3e2.csv")) %>%
  mutate(dif = Despues - Antes)
wcx.R <- wilcox.test(df$dif, exact=TRUE, alternative="two.sided")

## Warning in wilcox.test.default(df$dif, exact = TRUE, alternative =
## "two.sided"): cannot compute exact p-value with ties

## Warning in wilcox.test.default(df$dif, exact = TRUE, alternative =
## "two.sided"): cannot compute exact p-value with zeroes

wcx.yo <- test.wilcoxon(df$dif, exacto = TRUE, alternativa = "dos colas")

## Warning in test.wilcoxon(df$dif, exacto = TRUE, alternativa = "dos colas"):
## No se puede calcular el p-valor exacto con ceros

## Warning in test.wilcoxon(df$dif, exacto = TRUE, alternativa = "dos colas"):
## No se puede calcular el p-valor exacto con empates

str(wcx.R)

## List of 7
## $ statistic : Named num 154
## .. attr(*, "names")= chr "V"
## $ parameter : NULL
## $ p.value : num 0.00302

```

```
## $ null.value : Named num 0
##   .. attr(*, "names")= chr "location"
## $ alternative: chr "two.sided"
## $ method      : chr "Wilcoxon signed rank test with continuity correction"
## $ data.name    : chr "df$dif"
## - attr(*, "class")= chr "htest"
```

```
str(wcx.yo)
```

```
## List of 7
## $ estadistico: num 154
## $ p_valor    : num 0.00306
## $ tita_nulo   : num 0
## $ alternativa: chr "dos colas"
## $ estimado    : NULL
## $ int_conf    : NULL
## $ parametros :List of 9
##   ..$ obs      : num [1:18] 0.05 -0.02 -0.02 0.1 0.1 ...
##   ..$ n        : int 18
##   ..$ ceros     : int 0
##   ..$ sin_ceros: logi TRUE
##   ..$ apareadas: logi FALSE
##   ..$ exacto    : logi FALSE
##   ..$ corregir  : logi TRUE
##   ..$ a         : NULL
##   ..$ alfa      : num 0.05
```

Con un p-valor de 0.0030622, es razonable rechazar la hipótesis nula y determinar que el alcohol afecta el tiempo de respuesta. Un test un poco mas razonable, es postular en la alternativa que el tiempo de reaccion *aumenta* despues de beber, y para dicho test, el p-valor sera exactamente la mitad.

Ejercicio 3

Con el propósito de saber si la mediana del número de ítems vendidos en cada operación de venta es igual a 10, un empresario registró los números de ítems vendidos en 12 operaciones de ventas independientes. Testee, usando el test de Wilcoxon, las siguientes hipótesis:

$$H_0 : \theta = 10 \quad \text{vs.} \quad H_1 : \theta \neq 10$$

```
obs <- read.csv(file.path(DATA_DIR, "p3e3.csv"))[["No..items"]]
wcx.R <- wilcox.test(obs, mu = 10, exact = TRUE)
```

```
## Warning in wilcox.test.default(obs, mu = 10, exact = TRUE): cannot compute
## exact p-value with ties
```

```
wcx.yo <- test.wilcoxon(obs, mu=10, exacto = FALSE)
```

Usando $\alpha = 0.05$ un p-valor de 0.2895874 determina que rechazo la hipótesis nula.

Ejercicio 4

Un candidato a elecciones legislativas considera que aumentará el caudal de votos a su favor si adopta la posición mediana de los ciudadanos. Para ello diseñó un cuestionario y lo distribuyó entre 15 votantes (muestra aleatoria), asignando scores de 0 a 10 a los resultados obtenidos.

Halle un intervalo de confianza para el score mediano. En base a los procedimientos usados en esta práctica, ¿cuál es la estimación puntual de la mediana?

Vamos a tener que expresar T^+ como la suma de los promedios de walsh, y luego el estimador puntual de Hodges Lehmann para la mediana sera la mediana de los promedios de Walsh.

```
obs <-read.csv(file.path(DATA_DIR, "p3e4.csv"))$Score
w <- walsh(obs)
n <- length(obs)
(tita <- median(w))
```

```
## [1] 6.5
```

```
# IC exacto: al haber empates no es adecuado
```

```
a <- 0
while (pTmas(a + 1, n) <= alfa/2) { a <- a + 1 }
(intervalo <- sort(w)[c(a+1, n*(n+1)/2-a)])
```

```
## [1] 5.15 7.70
```

```
# IC asintotico con correccion por continuidad
```

```
a <- qnorm(alfa/2, mean = (n*(n+1)/4 - 1/2), sd = sqrt(n*(n+1)*(2*n+1)/24))
(intervalo <- sort(w)[c(a+1, n*(n+1)/2-a)])
```

```
## [1] 5.15 7.70
```

Para mantener la linea con `wilcox.test`, agregamos la funcionalidad de estimacion puntual y por intervalo a nuestra funcion `test.wilcoxon`. Basta con pasarle `int_conf=TRUE` y `alfa` para darle nivel de confianza $1 - \alpha$ al IC:

```
test.wilcoxon(obs, int_conf = TRUE, alfa = 0.05)[c("estimado", "int_conf")]
```

```
## Warning in test.wilcoxon(obs, int_conf = TRUE, alfa = 0.05): No se puede
## calcular el p-valor exacto con empates
```

```
## $estimado
## [1] 6.5
##
## $int_conf
## [1] 5.15 7.70
```

```
wilcox.test(obs, conf.int = TRUE)[c("estimate", "conf.int")]
```

```
## Warning in wilcox.test.default(obs, conf.int = TRUE): cannot compute exact
## p-value with ties
```

```
## Warning in wilcox.test.default(obs, conf.int = TRUE): cannot compute exact
## confidence interval with ties
```

```
## $estimate
## (pseudo)median
##      6.499938
##
## $conf.int
## [1] 5.150027 7.699953
## attr(,"conf.level")
## [1] 0.95
```

Ejercicio 5

Resuelva el ejercicio 2 usando scores normales en vez de rangos y compare los resultados obtenidos por ambos métodos.


```
df <- read.csv(file.path(DATA_DIR, "p3e2.csv")) %>%
  mutate(dif = Despues - Antes)

x <- df$dif[df$dif != 0]
alfa <- .05
n <- length(x)
rangos <- x %>% abs %>% rank
A <- qnorm((rangos + n + 1)/(2*(n+1)))
Asq <- A^2
Vbar <- sum(A[x > 0])/n
muVbar <- sum(A)/(2*n)
sigmaVbar <- sqrt(sum(Asq)/(4*n^2))
crit <- qnorm(c(alfa/2, 1-alfa/2), muVbar, sigmaVbar)
```

Como $\bar{V} = 0.7042488$ y el valor critico superior es 0.170079, 0.5975269, rechazamos la hipotesis nula. Tengamos en cuenta que al ser un test a dos colas, rechazamos siempre y cuando $|\bar{V}| > \text{Crit}$. Al igual que se comento en el Ejercicio 2, probablemente seria mas razonable en este caso hacer un test a una cola con alternativa “mayor”.

Ejercicio 6

En un ejemplo visto en clase como aplicación del test de Cox-Stuart, se registraba el caudal de un río (en pies cúbicos por segundo) durante 24 meses. Testee la hipótesis de que existe una tendencia positiva en los caudales en base al test de Wilcoxon de rangos signados.

$$H_0 : \text{No hay tendencia creciente en el caudal} \quad \text{vs.} \quad H_1 : \text{Si la hay}$$

$$X_i^{Año2} - X_i^{Año1} = D_i \sim F_D(t - \theta) \Rightarrow H_0 : \theta \leq 0 \quad \text{vs.} \quad H_1 : \theta > 0$$

```
df <- read.csv(file.path(DATA_DIR, "p3e6.csv"))
df <- mutate(df, dif = A.o.2 - A.o.1)

filter(df, !near(dif, Diferencia))
```

```
##           Mes A.o.1 A.o.2 Diferencia dif
## 1      Julio  92.8  88.1      -14.0 -4.7
## 2 noviembre  29.3  27.1       -1.8 -2.2
```

Hay diferencias interanuales mal calculadas para Julio y Noviembre! Usamos las correctas calculadas por R, `df$dif`.

Con el `test.wilcoxon` bien implementada, hacer una prueba de muestras apareadas es trivial. Basta con ampliar la definicion de las observaciones a testear:

```
obs <- if (is.null(y)) { x - mu
} else if (apareadas) { x - y
} else { stop("MWW no implementado aun") }
```

```
wcx.yo <- test.wilcoxon(x=df$A.o.2, y=df$A.o.1, apareadas=TRUE, alternativa = "mayor")
str(wcx.yo)
```

```
## List of 7
## $ estadistico: num 39
## $ p_valor    : num 0.515
## $ tita_nulo  : num 0
## $ alternativa: chr "mayor"
```

```
## $ estimado : NULL
## $ int_conf : NULL
## $ parametros :List of 9
## ..$ obs : num [1:12] -0.4 -1.7 19 -30 28 ...
## ..$ n : int 12
## ..$ ceros : int 0
## ..$ sin_ceros: logi TRUE
## ..$ apareadas: logi TRUE
## ..$ exacto : logi TRUE
## ..$ corregir : logi TRUE
## ..$ a : NULL
## ..$ alfa : num 0.05
```

Notese que como $T^{obs} = 39$, exactamente la mitad del maximo valor posible (si $n = 12 \Rightarrow T^+ \in [0 : 78]$), si hicieramos un test a dos colas, el p-valor daria nominalmente mayor a 1!

Las siguientes 3 llamadas son equivalentes:

Ejercicio 11

Cierto estudio muestra que drogas para la hipertensión arterial, tales como el propranolol, podrían aliviar los síntomas de pánico escénico (Time Magazine, jul 1982, pag. 58). Para testear esta hipótesis, profesionales y estudiantes dieron 2 recitales como solistas ante una audiencia de críticos y miembros de la Universidad. 90 minutos antes de cada recital se les suministró propranolol o un placebo. El pulso cardíaco se les midió mediante un monitoreo electrocardiográfico remoto durante la representación. El pulso normal en reposo es de 70 pulsaciones por minuto. Los datos correspondientes a 8 ejecutantes son los siguientes.

Sea θ la mediana de la distribución de las diferencias: Placebo - Droga. Use el procedimiento de rangos signados de Wilcoxon para testear las hipótesis:

$$H_0 : \theta = 0 \quad \text{vs.} \quad H_1 : \theta > 0$$

a nivel $\alpha = 0.05$. Obtenga un estimador puntual de θ y construya un intervalo de confianza de nivel aproximado 0.95 para θ .

Explicitemos un poco mas las hipotesis:

$$H_0 : \text{La droga no disminuye el pulso respecto al placebo} \quad \text{vs.} \quad H_1 : \text{Si lo hace}$$

$$X_i^{\text{Placebo}} - X_i^{\text{Droga}} = D_i \sim F_D(t - \theta) \Rightarrow H_0 : \theta = 0 \quad \text{vs.} \quad H_1 : \theta > 0$$

Realizamos un tipico test de muestras apareadas y pedimos el estimador puntual e IC de la mediana de las diferencias. Vale aclarar que en estos casos, los IC de confianza seran unilaterales, desde/hasta -/+Inf segun corresponda.

```
alfa <- 0.05
df <- read.csv(file.path(DATA_DIR, "p3e11.csv"))
wcx.R <- wilcox.test(df$Placebo, df$Droga, paired=TRUE, alternative="greater",
                    conf.int = TRUE, conf.level = 1 - alfa)
```

```
## Warning in wilcox.test.default(df$Placebo, df$Droga, paired = TRUE,
## alternative = "greater", : cannot compute exact p-value with ties

## Warning in wilcox.test.default(df$Placebo, df$Droga, paired = TRUE,
## alternative = "greater", : cannot compute exact confidence interval with
## ties
```

```
wcx.yo <- test.wilcoxon(df$Placebo, df$Droga, apareadas=TRUE,
                       alternativa="mayor", int_conf = TRUE, alfa=alfa)

## Warning in test.wilcoxon(df$Placebo, df$Droga, apareadas = TRUE,
## alternativa = "mayor", : No se puede calcular el p-valor exacto con empates

str(wcx.R)

## List of 9
## $ statistic : Named num 35
## .. attr(*, "names")= chr "V"
## $ parameter : NULL
## $ p.value : num 0.0104
## $ null.value : Named num 0
## .. attr(*, "names")= chr "location shift"
## $ alternative: chr "greater"
## $ method : chr "Wilcoxon signed rank test with continuity correction"
## $ data.name : chr "df$Placebo and df$Droga"
## $ conf.int : num [1:2] 18.5 Inf
## .. attr(*, "conf.level")= num 0.95
## $ estimate : Named num 34.1
## .. attr(*, "names")= chr "(pseudo)median"
## - attr(*, "class")= chr "htest"

str(wcx.yo)

## List of 7
## $ estadistico: num 35
## $ p_valor : num 0.0104
## $ tita_nulo : num 0
## $ alternativa: chr "mayor"
## $ estimado : num 34
## $ int_conf : num [1:2] 18.5 Inf
## $ parametros :List of 9
## ..$ obs : int [1:8] 41 33 26 26 36 51 -4 56
## ..$ n : int 8
## ..$ ceros : int 0
## ..$ sin_ceros: logi TRUE
## ..$ apareadas: logi TRUE
## ..$ exacto : logi FALSE
## ..$ corregir : logi TRUE
## ..$ a : num 5.75
## ..$ alfa : num 0.05
```

Practica 4

Ejercicio 1

Halle la distribucion exacta del estadistico $T = \sum_{j=1}^n R(X_j)$ para $n = 3, m = 2$ y compare con la tabla A7 del Conover.

Al igual que para T^+ , podemos usar una formula recursiva y una memoria para calcular la distribucion exacta bajo la hipotesis nula de manera eficiente:

```
max_n <- 50
max_m <- 50
```

```

max_k <- sum( (max_m+1):(max_n+max_m) )
memW <- array(dim = c(max_n, max_n, max_k))

A <- function(n, m ,k) {
  # Devolver valores limites
  if (k < 0) { return(0) }
  if (k == 0 | n == 0 | m == 0) {
    if (n == 0 & k == 0) { return(1)
    } else if (m == 0 & (k==n*(n+1)/2)) { return(1)
    } else { return (0) }
  }
  # De estar precomputado, devolver el valor
  if (!is.na(memW[n,m,k])) { return(memW[n,m,k]) }
  # Si no, calcularlo, cachearlo y devolverlo
  valor <- A(n-1, m, k-(m+n)) + A(n, m-1, k)
  memW[n,m,k] <- valor
  return(valor)
}

dW <- function(x, n, m) {
  # Devuelve Pr(W = xi | n, m) para cada xi en x
  posibles <- map_dbl(x, A, n=n, m=m)
  totales <- choose(n+m, n)
  return(posibles / totales)
}

pW <- function(q, n, m) {
  # Devuelve Pr(W <= qi | n, m)
  max_q <- max(q)
  probs <- dW(0:max_q, n, m)
  res <- vector("double", length(q))
  for (i in seq_along(q)) {
    res[i] <- sum(probs[1:q[[i]]+1])
  }
  return(res)
}

n <- 3
m <- 2
N <- n + m
# Dados n, m, W esta siempre en el rango [n*(n+1)/2, nm + n*(n+1)/2]
rango <- (n*(n+1)/2):(n*m + n*(n+1)/2)
probs <- dW(rango, n = 3, m = 2)
acums <- pW(rango, n = 3, m = 2)
tibble(rango, probs, acums)

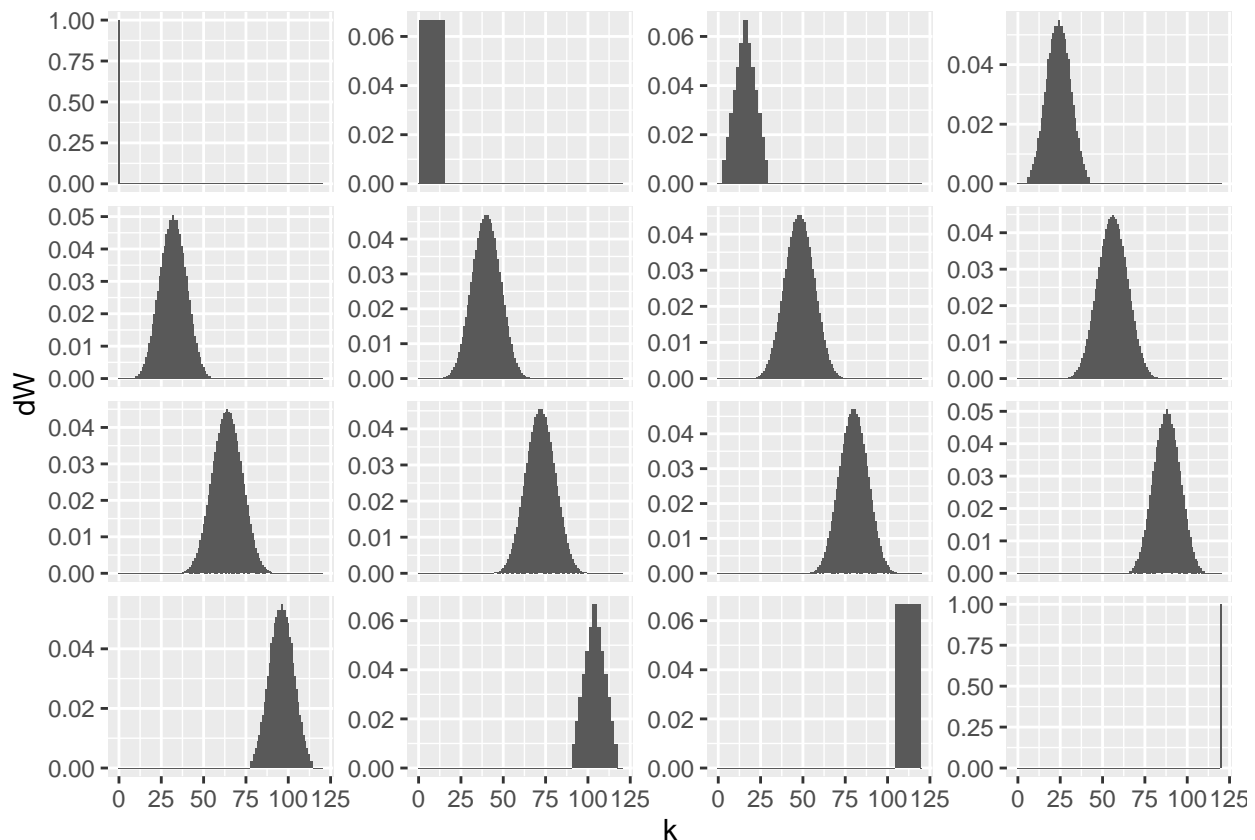
## # A tibble: 7 x 3
##   rango probs acums
##   <int> <dbl> <dbl>
## 1     6  0.1  0.1
## 2     7  0.1  0.2
## 3     8  0.2  0.4
## 4     9  0.2  0.6
## 5    10  0.2  0.8

```

```
## 6    11    0.1    0.9
## 7    12    0.1    1
```

Podemos jugar un poco, y ver como cambia la distribucion de W si dejamos fijo $n + m = N = 15$, y hacemos variar la cantidad de elementos de la primera muestra, n . De izquierda a derecha y arriba a abajo, cada panel muestra la distribucion de W para $n = 0 : 15$; $m = 15 - n$. La densidad se va “corriendo hacia la derecha”, pero sin dejar de ser siempre simetrica:

```
tibble(
  n = 0:15,
  m = 15-n,
  k = list(0:120),
  dW = pmap(list(k, n, m), dW)) %>%
  tidyr::unnest() %>%
  ggplot(aes(k, dW)) +
  geom_col() +
  facet_wrap(~n, scales = "free_y") +
  theme(
    strip.background = element_blank(),
    strip.text.x = element_blank()
  )
```



Ejercicio 3

Testee usando el conjunto de datos presentado en la Tabla, si la temperatura media máxima (en grados Fahrenheit) de la ciudad de Des Moines es mayor que la temperatura media máxima de Spokane.

De tener observaciones apareadas, podríamos usar Wilcoxon para datos apareados. A falta de ello, usaremos MWW para muestras independientes.

Asumimos que la temperatura en Des Moines (X_i) y Spokane (Y_j) tienen la misma distribución, salvo por a lo sumo un corrimiento Δ . Las hipótesis serán:

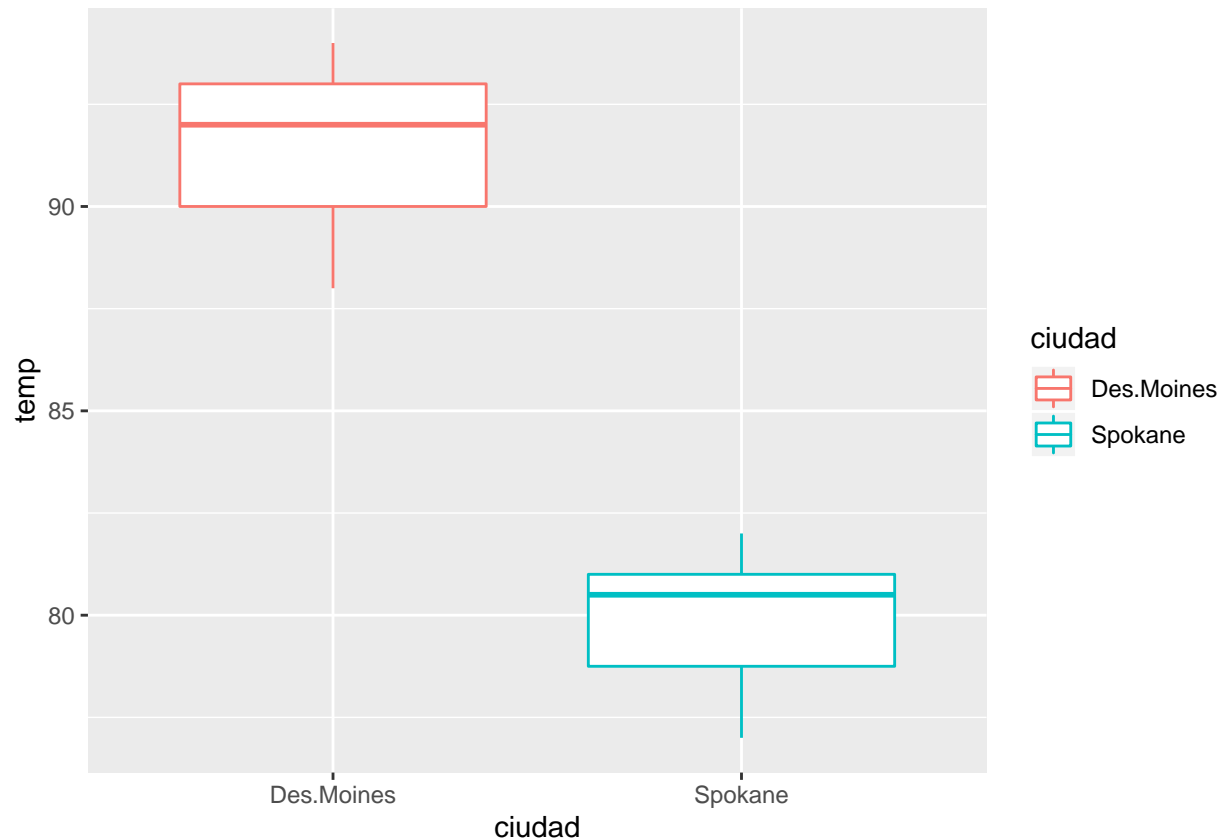
H_0 : La temp. máxima media en Des Moines no es mayor que en Spokane
vs. H_1 : Si la es

$$X_i \stackrel{iid}{\sim} F(t - \theta_x) \quad Y_j \stackrel{iid}{\sim} F(t - \theta_y) \quad \theta_y = \theta_x + \Delta$$

$\approx H_0 : \theta_x = \theta_y$ vs. $H_1 : \theta_x > \theta_y$
 $\approx H_0 : \Delta = 0$ vs. $H_1 : \Delta < 0$

```
df <- read.csv(file.path(DATA_DIR, "p4e3.csv")) %>%
  gather("ciudad", "temp") %>%
  drop_na

df %>%
  ggplot(aes(x = ciudad, y = temp, color=ciudad)) +
  geom_boxplot()
```



```
# Usando T = suma de rangos de la muestra X
x <- df[df$ciudad=="Des.Moines",]$temp
n <- length(x)
y <- df[df$ciudad=="Spokane",]$temp
m <- length(y)
N <- n + m
rgs <- rank(c(x, y))
Tobs <- sum(rgs[1:n])
```

```

# Hay empates, así que calculo la esperanza, varianza y corrección por empates
muT <- n*(N+1)/2
varT <- (m*n/(N*(N+1))*sum(rgs^2)) - (m*n*(N+1)^2)/(4*(N-1))
# p-valor aproximación normal
p_valor.normal(Tobs, muT, sqrt(varT), "mayor", TRUE)

## [1] 8.101201e-07

# p-valor exacto
1 - pW(Tobs - 1, n, m)

## [1] 4.113534e-05

wilcox.test(x, y, paired = FALSE)

## Warning in wilcox.test.default(x, y, paired = FALSE): cannot compute exact
## p-value with ties

##
## Wilcoxon rank sum test with continuity correction
##
## data: x and y
## W = 72, p-value = 0.0006067
## alternative hypothesis: true location shift is not equal to 0

Se debe rechazar la hipótesis nula.

```

Ejercicio 4

Ejercicio 4: En un experimento controlado de laboratorio, 10 hombres y 10 mujeres son puestos a prueba para determinar cuál es la temperatura (en grados Fahrenheit) que consideran más confortable. Los resultados se presentan a continuación. Suponiendo que los datos constituyen muestras aleatorias de las respectivas poblaciones, testeé si la temperatura más confortable es la misma para hombres y mujeres.

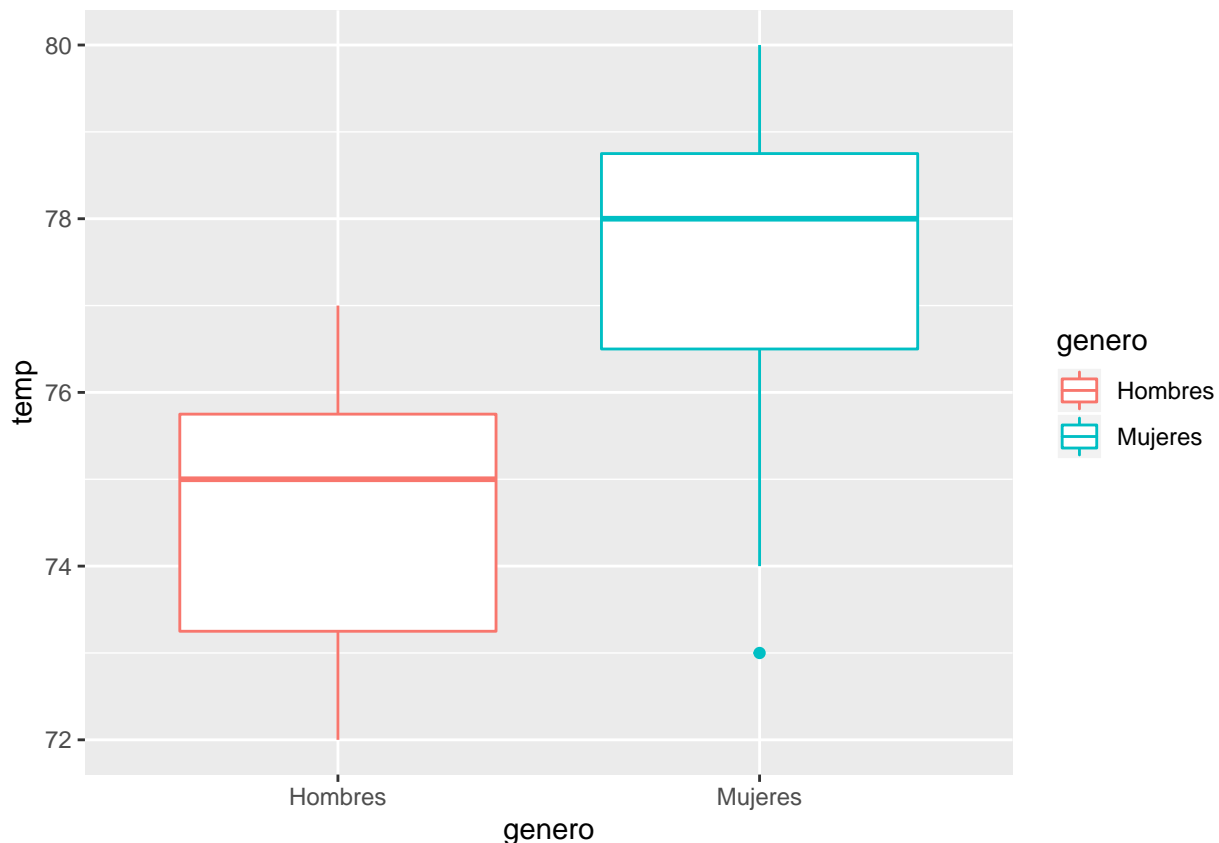
Testeamos igual contra distinto, con alternativa a dos colas:

```

df <- read.csv(file.path(DATA_DIR, "p4e4.csv")) %>%
  gather("genero", "temp") %>%
  drop_na

df %>%
  ggplot(aes(x = genero, y = temp, color=genero)) +
  geom_boxplot() # Agrega ruido para "desolapar" obs repetidas

```



```
# Usando T = suma de rangos de la muestra X
x <- df[df$genero=="Hombres",]$temp
n <- length(x)
y <- df[df$genero=="Mujeres",]$temp
m <- length(y)
N <- n + m
rgs <- rank(c(x, y))
Tobs <- sum(rgs[1:n])

# Hay empates, asi que calculo la esperanza, varianza y correccion por empates
muT <- n*(N+1)/2
varT <- (m*n/(N*(N+1))*sum(rgs^2)) - (m*n*(N+1)^2)/(4*(N-1))
# p-valor aproximacion normal
p_valor.normal(Tobs, muT, sqrt(varT), "dos colas", TRUE)
```

```
## [1] 0.000979009
```

```
# p-valor exacto
2*min(1 - pW(Tobs - 1, n, m), pW(Tobs, n, m))
```

```
## [1] 0.008930698
```

```
wilcox.test(x, y, paired = FALSE)
```

```
## Warning in wilcox.test.default(x, y, paired = FALSE): cannot compute exact
## p-value with ties
```

```
##
## Wilcoxon rank sum test with continuity correction
```



```
##
## data:  x and y
## W = 16.5, p-value = 0.01182
## alternative hypothesis: true location shift is not equal to 0
```

Ejercicio 5

Un grupo de siete alumnos recibieron clases de álgebra con un método de enseñanza tradicional y otro grupo de seis alumnos con un nuevo método. Al finalizar el curso, ambos grupos fueron sometidos al mismo examen y los puntajes obtenidos fueron los siguientes. Encuentre un intervalo de confianza del 90% para la diferencia de los puntajes medianos.

```
df <- read.csv(file.path(DATA_DIR, "p4e5.csv"))

tradi <- df$Método.tradicional
nuevo <- df$Método.nuevo
nuevo <- nuevo[!is.na(nuevo)]

d <- expand.grid(tradi = tradi, nuevo = nuevo) %>%
  mutate(dif = nuevo - tradi) %>%
  pluck("dif") %>%
  sort()

median(d)
```

```
## [1] -8
```

```
ic_percentil(d, 0.5, .1, "exacto")
```

```
## $r
## [1] 16
##
## $s
## [1] 27
##
## $ic
## [1] -9 -6
```

```
ic_percentil(d, 0.5, .1, "asintotico")
```

```
## $r
## [1] 16
##
## $s
## [1] 27
##
## $ic
## [1] -9 -6
```

```
wilcox.test(nuevo, tradi, conf.int = TRUE, conf.level = 0.90)
```

```
## Warning in wilcox.test.default(nuevo, tradi, conf.int = TRUE, conf.level =
## 0.9): cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(nuevo, tradi, conf.int = TRUE, conf.level =
## 0.9): cannot compute exact confidence intervals with ties
```

```
##
```

```
## Wilcoxon rank sum test with continuity correction
##
## data:  nuevo and tradi
## W = 9, p-value = 0.09948
## alternative hypothesis: true location shift is not equal to 0
## 90 percent confidence interval:
##  -1.399995e+01  8.696493e-05
## sample estimates:
## difference in location
##                -7.999973
dwilcox(0:9, 3, 2)

## [1] 0.1 0.1 0.2 0.2 0.2 0.1 0.1 0.0 0.0 0.0
```

Practica 5

Ejercicio 1

```
df <- read.csv("data/p5e1.csv") %>%
  gather(clase, duracion) %>%
  drop_na()
df

##   clase duracion
## 1     A        73
## 2     A        64
## 3     A        67
## 4     A        62
## 5     A        70
## 6     B        84
## 7     B        80
## 8     B        81
## 9     B        77
## 11    C        82
## 12    C        79
## 13    C        71
## 14    C        75

kruskal.test(df$duracion, df$clase)

##
## Kruskal-Wallis rank sum test
##
## data:  df$duracion and df$clase
## Kruskal-Wallis chi-squared = 8.4033, df = 2, p-value = 0.01497

k <- df %>%
  mutate(rg = rank(duracion)) %>%
  group_by(clase) %>%
  summarise(
    Ridot = sum(rg),
    ni = length(duracion)) %>%
  mutate(k = Ridot^2/ni) %>%
  pluck("k")
```

```
N <- length(df$duracion)
S2 <- N*(N+1)/2
Tcon <- (1/S2) * (sum(k) - N*(N+1)/4)
Tsin <- 12/(N*(N+1))*sum(k) - 3*(N+1)
pchisq(Tsin, length(k)-1, lower.tail = FALSE)
```

```
## [1] 0.01497088
```