Métodos de Remuestreo

Tema 7. Contrastes de hipótesis basados en remuestreos

basado en

B. Efron, R. Tibshirani (1993). An Introduction to the bootstrap.

O. Kirchkamp (2017). Resampling methods.

Curso 2018/19

Tests de Permutaciones

- Los tests de permutaciones son contrastes de hipótesis basados en métodos computacionales intensivos.
- Fueron introducidos por Fisher en los años 30, aunque solo de una manera teórica por falta de medios computacionales.
- La idea básica es no imponer restricciones de tipo probabilístico a los contrastes de hipótesis, y usar una metodología *semejante* a la usada en los métodos bootstrap.
- Como introducción a estos métodos se puede considerar el ejemplo del contraste para dos muestras.

Tests de Permutaciones

- Se observan dos muestras aleatorias independientes **x** e **y** extraídas de posiblemente dos distribuciones de probabilidad diferentes:

$$\begin{aligned} F &\rightarrow \mathbf{x} = (x_1, x_2, \ldots, x_n) \\ G &\rightarrow \mathbf{y} = (y_1, y_2, \ldots, y_m) \end{aligned}$$

- Se considera la hipótesis nula

$$H_0 : \ F = G$$

- La igualdad $F = G$ significa que F y G asignan igual probabilidad a todos los conjuntos

$$P_F \{A\} = P_G \{A\},$$

donde A es un subconjunto del espacio muestral conjunto de las v.a. x e y.

Tests de Permutaciones

- Si H_0 es cierta, entonces no hay diferencias entre el comportamiento probabilístico de las v.a. **x** e **y**.

- **Ejemplo**: En el caso de los datos de ratones, hay pocas observaciones, de modo que el tamaño muestral del caso tratamiento es $n = 7$ y el del caso del control es $m = 9$.

- La diferencia entre las medias es

$$\widehat{\theta} = \bar{x} - \bar{y} = 30{,}63$$

- Parece indicar que la distribución del tratamiento (F) provoca tiempos de supervivencia mayores que la distribución del control (G).

Tests de Permutaciones

- Pero, si no se puede rechazar categóricamente la posibilidad de que H_0 sea cierta, entonces eso **NO** significa que se haya **demostrado** la hipótesis alternativa.
- En realidad, un test de hipótesis es un método formal que se plantea para decidir si los datos rechazan *decisivamente* la hipótesis nula H_0.
- En el caso de los ratones se calcula el estadístico $\widehat{\theta}$ de modo que, intuitivamente se esperan valores altos si H_0 es falsa.
- Cuanto mayor sea el valor observado de $\widehat{\theta}$ mayor evidencia se tendrá en contra de H_0.

Tests de Permutaciones

- Se define el **nivel de significación alcanzado** (ASL) como la probabilidad de observar al menos un valor tan grande del estadístico cuando la hipótesis nula es cierta

$$ASL = P_{H_0}\left\{\widehat{\theta}^* \geq \widehat{\theta}\right\}$$

- Cuanto menor sea el valor de ASL, mayor es la evidencia en contra de H_0.

- El valor $\widehat{\theta}$ es un valor fijo y observado. La variable $\widehat{\theta}^*$ tiene la distribución que se asume bajo H_0.

- La notación *estrella* diferencia entre la observación real del estadístico $\widehat{\theta}$ y la distribución de $\widehat{\theta}^*$ generada de acuerdo con la hipótesis nula H_0.

Tests de Permutaciones

- La validación de H_0 se hace calculando el valor de ASL para ver si es pequeño en relación con ciertos límites.

- Formalmente si se toma una significación de α, por ejemplo igual a 0,05, entonces se rechaza H_0 si ASL es menor que α.

- En el caso de los ratones, un test tradicional asumiría que F y G se distribuyen como una normal con medias posiblemente diferentes:

$$\begin{aligned} F &= N\left(\mu_x, \sigma^2\right) \\ G &= N\left(\mu_y, \sigma^2\right) \end{aligned}$$

- De este modo, la hipótesis nula H_0 es equivalente a decir que $\mu_x = \mu_y$.

Tests de Permutaciones

- Si se cumple H_0 entonces

$$\widehat{\theta} \sim N\left(0,\ \sigma^2\left(\frac{1}{n}+\frac{1}{m}\right)\right).$$

- Entonces

$$\begin{aligned} ASL &= P_{H_0}\left\{\widehat{\theta}^* \geq \widehat{\theta}\right\} \\ &= P\left\{Z \geq \frac{\widehat{\theta}}{\sigma\sqrt{\frac{1}{n}+\frac{1}{m}}}\right\} \\ &= 1-\Phi\left(\frac{\widehat{\theta}}{\sigma\sqrt{\frac{1}{n}+\frac{1}{m}}}\right) \end{aligned}$$

- De modo que bajo la suposición de normalidad, $\widehat{\theta}$ sigue así una distribución *conocida*.

Tests de Permutaciones

- En el caso habitual se aplica el test de la t de Student de modo que, en el ejemplo,

$$ASL = P\left\{ t_{14} \geq \frac{30{,}63}{54{,}21\sqrt{\frac{1}{9}+\frac{1}{7}}} \right\} = 0{,}141$$

- Con lo que se concluye que este valor no permite rechazar H_0 ni siquiera con una significación $\alpha = 0{,}10$.

- Esta solución solo es válida si se asume normalidad de los datos.

Ejemplo de metodología clásica

```
library(bootstrap)

t.test(mouse.t,mouse.c,alternative="greater",
var.equal=TRUE)
```

```
        Two Sample t-test

data:  mouse.t and mouse.c
t = 1.1214, df = 14, p-value = 0.1405
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -17.48178       Inf
sample estimates:
mean of x mean of y
 86.85714  56.22222
```

Procedimiento de remuestreo

- Se juntan las dos muestras en una sola $(\mathbf{x}, \mathbf{y})$ y de esta nueva muestra se toman aleatoriamente dos nuevas submuestras $\mathbf{x}'$ e $\mathbf{y}'$ de tamaño n y m respectivamente.

- Así se elimina la asociación original de los valores con las distribuciones originales F y G.

- Se generan B submuestras independientes $\mathbf{x}'$ e $\mathbf{y}'$ **SIN reemplazamiento** de la muestra conjunta $(\mathbf{x}, \mathbf{y})$.

Procedimiento de remuestreo

- Se calcula $\widehat{\theta}^*(b)$ para cada muestra.
- Después se calcula

$$\widehat{ASL}_{perm} = \frac{\#\left\{\widehat{\theta}^*(b) \geq \widehat{\theta}\right\}}{B}$$

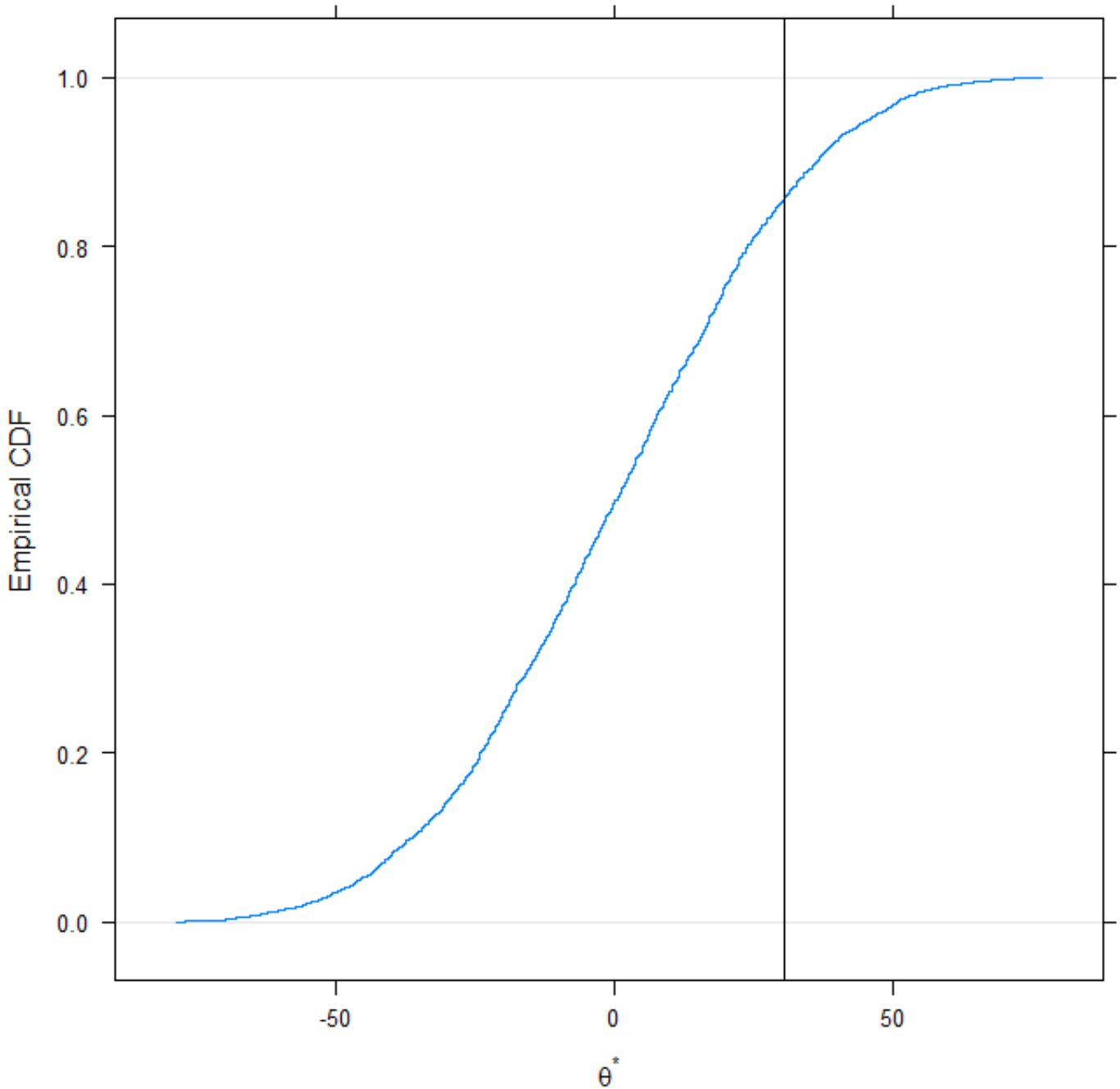
- De manera ideal se deberían tomar $B = \binom{N}{n}$ diferentes submuestras para calcular el valor *exacto* de ASL.

Ejemplo

```
thetaHat = mean(mouse.t) - mean(mouse.c)

xy = c(mouse.c,mouse.t)
n = length(mouse.c)
N = length(xy)

# OJO: SIN reemplazamiento
thetaStar = replicate(5000,{ind = sample(1:N,n);
mean(xy[ind])-mean(xy[-ind])})

mean(thetaStar>thetaHat)
```

```
[1] 0.1414
```

```
library(latticeExtra)
ecdfplot(thetaStar,xlab=expression(theta^{"*"}))+
layer(panel.abline(v=thetaHat))
```


Empirical CDF
1.0
0.8
0.6
0.4
0.2
0.0
-50
0
50
θ*

Ejemplo tablas de contingencia

	Creencias Religiosas			
Educación	*Fundamentalista*	*Moderada*	*Liberal*	*Total*
$<$ *Secundaria*	178	138	108	424
Secundaria	570	648	442	1660
Graduado	138	252	252	642
Total	886	1038	802	2726

```
tabla = as.table(rbind(c(178, 138, 108),
        c(570, 648, 442), c(138,252, 252)))

(res = chisq.test(tabla))
```

Ejemplo tablas de contingencia

```
        Pearson's Chi-squared test

data:  tabla.array
X-squared = 69.1568, df = 4, p-value = 3.42e-14
```

```r
# Recuentos esperados
res$expected
```

```
             Grado
Religiosidad      <HS  HS o JH Graduado
        Fund 137.8078 539.5304 208.6618
        Mod  161.4497 632.0910 244.4593
        Lib  124.7425 488.3786 188.8789
```

Ejemplo tablas de contingencia

- Con la función `chisq.test` también se pueden hacer contrastes por simulación Montecarlo, es decir, calculando el estadístico de la chi cuadrado para todas las posibles tablas con las mismas sumas marginales por filas y columnas de la tabla original.

```
chisq.test(tabla, sim=T, B=2000)
```

```
    Pearson's Chi-squared test with simulated p-value
    (based on 2000 replicates)

data:  tabla.array
X-squared = 69.1568, df = NA, p-value = 0.0004998
```

Ejemplo test permutaciones usando coin

```
library(coin)

# Simulas unos datos
niveles = c(40, 57, 45, 55, 58, 57, 64, 55, 62, 65)
trata = factor(c(rep("A",5), rep("B",5)))
eso = data.frame(trata, niveles)

# Test clasico de t Student
t.test(niveles~trata, data=eso, var.equal=TRUE)
```

Ejemplo usando la librería coin

```
        Two Sample t-test

data:  niveles by trata
t = -2.345, df = 8, p-value = 0.04705
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -19.0405455  -0.1594545
sample estimates:
mean in group A mean in group B
           51.0            60.6
```

Ejemplo usando la librería coin

```
# Numero posible de permutaciones
choose(10,5)
```

```
[1] 252
```

```
oneway_test(niveles~trata, data=eso, distribution="exact")
```

```
        Exact 2-Sample Permutation Test

data:  niveles by trata (A, B)
Z = -1.9147, p-value = 0.07143
alternative hypothesis: true mu is not equal to 0
```

Ejemplo test permutaciones usando coin

```
# Difusion de agua via membrana fetal

agua_entra = data.frame(
pd = c(0.80, 0.83, 1.89, 1.04, 1.45, 1.38, 1.91, 1.64,
0.73, 1.46, 1.15, 0.88, 0.90, 0.74, 1.21),
edad=factor(c(rep("A termino", 10),
rep("12-26 Semanas", 5))))

# Test de permutaciones
eso = oneway_test(pd ~ edad, data=agua_entra,
distribution=approximate(B=1000))
```

Ejemplo test permutaciones usando coin

```
print(eso)
```

```
        Approximative 2-Sample Permutation Test

data:  pd by edad (12-26 Semanas, A termino)
Z = -1.5225, p-value = 0.121
alternative hypothesis: true mu is not equal to 0
```

```
pvalue(eso)
```

```
[1] 0.121

99 percent confidence interval:
 0.09579856 0.14989053
```

Ejemplo ANOVA usando la librería coin

```
library(coin)
# ANOVA unifactorial no parametrico

# Longitud de sardinas en funcion de factores ambientales
peces = data.frame(
longi=c(46, 28, 46, 37, 32, 41, 42, 45, 38, 44, 42, 60,
32, 42, 45, 58, 27, 51, 42, 52, 38, 33, 26, 25, 28, 28,
26, 27, 27, 27,31, 30, 27, 29, 30, 25, 25, 24, 27, 30),
sitio = factor(c(rep("I", 10), rep("II", 10),
rep("III", 10), rep("IV", 10))))

# test de Kruskal-Wallis
kwa = kruskal_test(longi ~ sitio,
data=peces, distribution=approximate(B=10000))
```

Ejemplo ANOVA usando la librería coin

```
print(kwa)
```

```
        Approximative Kruskal-Wallis Test

data:  longi by sitio (I, II, III, IV)
chi-squared = 22.8524, p-value < 2.2e-16
```

```
pvalue(kwa)
```

```
[1] 0
99 percent confidence interval:
 0.0000000000 0.0005296914
```

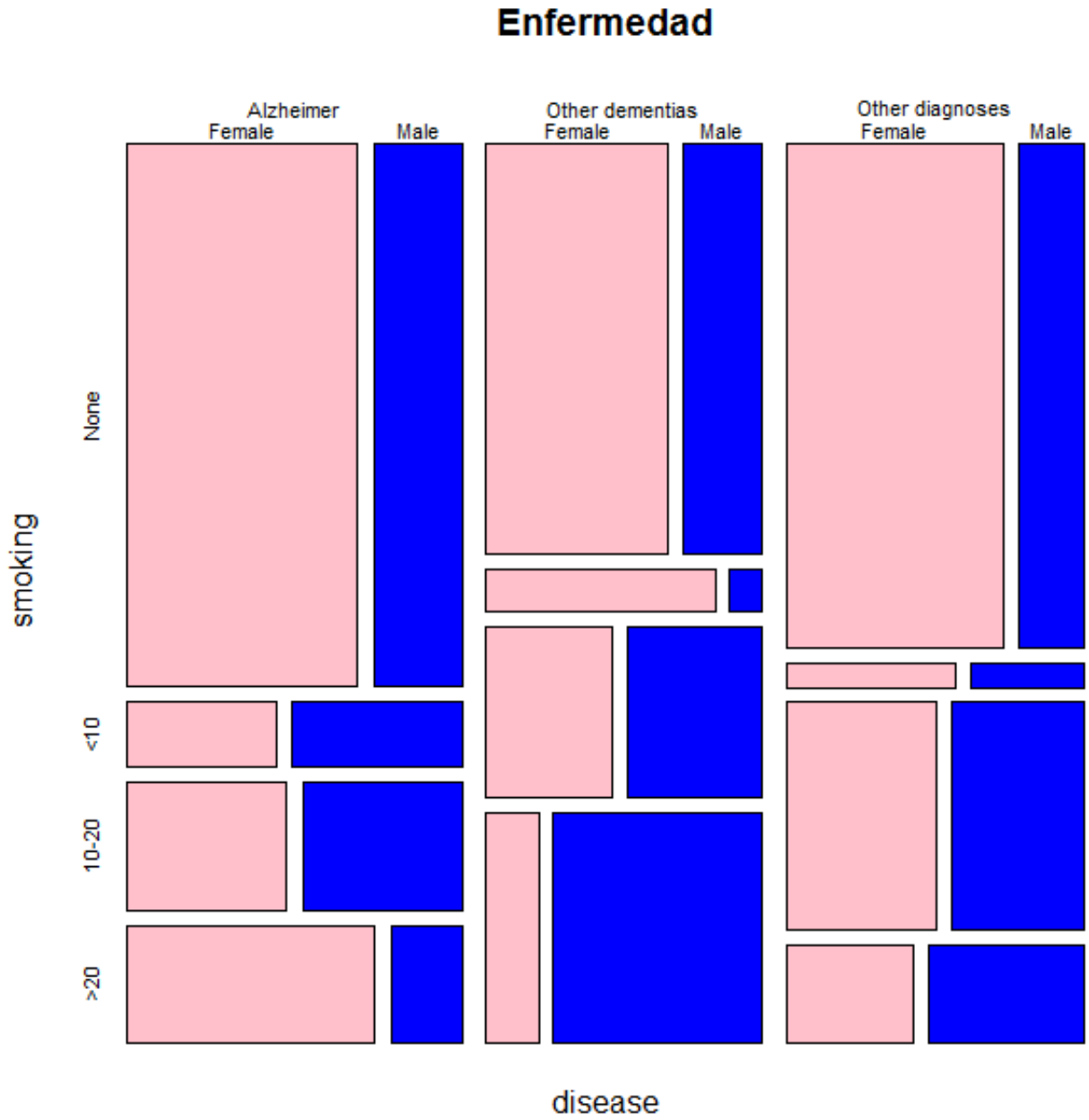
Ejemplo tablas de contingencia usando la librería coin

- Se consideran los datos de relación entre enfermedades degenerativas cerebrales y el consumo de tabaco

```
library(coin)
print(alzheimer)
```

```
    smoking          disease gender
1      None        Alzheimer Female
2      None        Alzheimer Female
.....................................
537     >20 Other diagnoses    Male
538     >20 Other diagnoses    Male
```

```
mosaicplot(~disease + smoking + gender,
data=alzheimer, main="Enfermedad",
col=c("pink","blue"), off=c(5,5,5,5)))
```


Enfermedad
Alzheimer
Female
Male
Other dementias
Female
Male
Other diagnoses
Female
Male
smoking
None
<10
10-20
>20
disease

Ejemplo tablas de contingencia usando la librería coin

```
it_alz = independence_test(disease ~ smoking | gender,
data=alzheimer, distribution="approximate")

print(it_alz)
```

```
        Approximative General Independence Test

data:  disease by
         smoking (None, <10, 10-20, >20)
         stratified by gender
maxT = 3.5106, p-value = 0.0055
alternative hypothesis: two.sided
```

Ejemplo tablas de contingencia usando la librería coin

```
females = alzheimer$gender == "Female"
males = alzheimer$gender == "Male"

pvalue(independence_test(disease ~ smoking, data=alzheimer,
subset=females, distribution="approximate"))
```

```
[1] 0.2522
99 percent confidence interval:
 0.2410842 0.2635530
```

```
pvalue(independence_test(disease ~ smoking,
data=alzheimer, subset=males, distribution="approximate"))
```

```
[1] 0
99 percent confidence interval:
 0.0000000000 0.0005296914
```

Contrastes de hipótesis Bootstrap

- Contraste de hipótesis de dos muestras bootstrap:

  1. Se eligen B submuestras independientes $\mathbf{x}'$ e $\mathbf{y}'$ tomadas **CON reemplazamiento** de la distribución conjunta $(\mathbf{x}, \mathbf{y})$.

  2. Se calcula $\widehat{\theta}^*(b)$ para cada muestra.

  3. se calcula el nivel de significación alcanzado (ASL) (equivalente al *p-valor*)

$$ASL = \frac{\#\left\{\widehat{\theta}^*(b) > \widehat{\theta}\right\}}{B}$$

Contrastes Bootstrap-t

- Consideramos, por ejemplo, la H_0 de que las medias y varianzas son iguales en ambas poblaciones:

```
library(bootstrap)

thetaHat = mean(mouse.t) - mean(mouse.c)
xy = c(mouse.c,mouse.t)
n = length(mouse.t)
m = length(mouse.c)

tStat = replicate(5000, {
xx = sample(xy, n, replace=TRUE);
yy = sample(xy, m, replace=TRUE);
(mean(xx)-mean(yy))/(sd(c(xx,yy)*sqrt(1/n+1/m)))})

(ASLt = mean(tStat > thetaHat/(sd(xy)*sqrt(1/n+1/m)))
```

```
[1] 0.1336
```

Contrastes de hipótesis BCa

- Los intervalos *BCa* resultan ser bastante eficientes para calcular intervalos de confianza.
- De este modo, la mejor opción es adaptarlos al caso de los contrastes de hipótesis.
- Se trata de calcular el valor de *ASL* a partir de intervalos de confianza para una probabilidad de recubrimiento cualquiera α.
- Se tiene que encontrar un nivel α tal que los extremos superior e inferior del intervalo de confianza coincidan con el α_1 o el α_2 observados.

Contrastes de hipótesis BCa

- Se tenía que los intervalos *BCa* se definían como

$$BCa = \left[\widehat{\theta}^*_{(\alpha_1)};\ \widehat{\theta}^*_{(\alpha_2)}\right]$$

- con

$$\alpha_2 = \Phi\left(\widehat{z}_0 + \frac{\widehat{z}_0 + z_{\frac{\alpha}{2}}}{1 - \widehat{a}\left(\widehat{z}_0 + z_{\frac{\alpha}{2}}\right)}\right)$$

- Si al hacer los remuestreos se obtiene el valor α_2 correspondiente al estadístico observado $\widehat{\theta}$, entonces despejando el valor de $z_{\alpha/2}$ se obtiene que

$$ASL_{BCa} = \Phi\left(\frac{\Phi^{-1}(\alpha_2) - \widehat{z}_0}{1 - \widehat{a}\left(\Phi^{-1}(\alpha_2) - \widehat{z}_0\right)} - \widehat{z}_0\right)$$

Contrastes de hipótesis BCa

```
library(bootstrap)

# Se simulan dos columnas de datos
x = rnorm(100)
y = rnorm(100)

# Se calcula la correlacion
(teta.hat = cor(x, y))
```

```
[1] 0.05065355
```

```
corre = function(k) cor(x[k], y[k])

bca.out = bcanon(seq(along = x), nboot = 10000,
    theta=corre, alpha=seq(0.001, 0.999, 0.001))
```

Contrastes de hipótesis BCa

```
# plot(density(bca.out$confpoints[,2]), main="")

# Test de una cola inferior: p-valor
# para para teta = teta.hat
ltpv = approx(bca.out$confpoints[,2],
bca.out$confpoints[,1], xout=teta.hat)$y

# Test de una cola superior para teta = teta.hat
1 - ltpv
```

```
[1] 0.5034074
```

```
# Test de dos colas para teta = 0
2 * min(ltpv, 1 - ltpv)
```

```
[1] 0.9931851
```

Varianza de los estimadores bootstrap

- La varianza de los estimadores bootstrap tiene dos causas o fuentes de variación:
  1. Una procede del muestreo aleatorio simple de tamaño n de la población cuyo parámetro queremos estimar.
  2. La otra procede del remuestreo bootstrap de tamaño B.
- Se trata de diseñar un método para estimar la varianza de los estimadores bootstrap, que se denomina *jackknife-after-bootstrap*.
- Se basa en utilizar aproximaciones jackknife en la estimación de la varianza de un estimador bootstrap.

Varianza de los estimadores bootstrap

- Tomamos el ejemplo de la estimación de $Var(\widehat{se}_B)$, la varianza del estimador bootstrap del error estándar de un estimador $\hat{\theta}$.
- El método *jackknife-after-bootstrap* consiste en eliminar primero uno de los i valores muestrales, $i = 1, \ldots, n$ y luego recalcular $\widehat{se}_B$ con las muestras bootstrap obtenidas, denominando a este valor $\widehat{se}_{B(i)}$.
- El *jackknife-after-bootstrap* se define como

$$Var_{jack}(\widehat{se}_B) = \frac{n-1}{n} \sum_{i=1}^{n} \left(\widehat{se}_{B(i)} - \widehat{se}_{B(\cdot)}\right)^2$$

donde

$$\widehat{se}_{B(\cdot)} = \frac{1}{n} \sum_{i=1}^{n} \widehat{se}_{B(i)}$$

Varianza de los estimadores bootstrap

- Este método presenta la dificultad del cálculo de $\widehat{se}_{B(i)}$, ya que éste requiere un conjunto de muestras completamente nuevas para cada i.
- No obstante, podemos evitar este exceso de cálculo de la siguiente manera.
- Para cada dato x_i de los valores observados de la muestra, existen muestras bootstrap en las que no aparece x_i, pudiendo utilizarse estas muestras para estimar $\widehat{se}_{B(i)}$
- Por ejemplo, en este caso, lo estimaremos con la desviación típica de las muestras bootstrap que **no** contengan el dato x_i.

Varianza de los estimadores bootstrap

- Si C_i es el conjunto de muestras bootstrap que **no** contienen a x_i, el cual contiene $B_i \leq B$ elementos, utilizaremos el $\widehat{se}_{B(i)}$ siguiente

$$\widehat{se}_{B(i)} = \sqrt{\frac{1}{B_i} \sum_{b \in C_i} \left(s(\mathbf{x}^{*b}) - \overline{s}_i\right)^2}$$

donde

$$\overline{s}_i = \frac{1}{B_i} \sum_{b \in C_i} s(\mathbf{x}^{*b})$$

- Este método se puede aplicar, no solo en la estimación de la varianza del estimador bootstrap del error de muestreo, sino en todos los estimadores bootstrap.
- Esto falla si todas las B muestras bootstrap contienen algún dato x_i.
- Así, el número de muestras bootstrap que se debe utilizar es bastante elevado.

Varianza estimadores bootstrap

```
# Datos de parches
data(patch, package = "bootstrap")
n = nrow(patch)
y = patch$y
z = patch$z
B = 2000
theta.b = numeric(B)
indices = matrix(0, nrow = B, ncol = n)

# jackknife-after-bootstrap
# Paso 1: hacer el bootstrap
for (b in 1:B) {
     i = sample(1:n, size=n, replace=TRUE)
     y = patch$y[i]
     z = patch$z[i]
     theta.b[b] = mean(y) / mean(z)
     # guardas losindices para el jackknife
     indices[b, ] = i
    }
```

Varianza estimadores bootstrap

```
# jackknife-after-bootstrap
# Paso 2: estimacion de se(se)
se.jack = numeric(n)

for (i in 1:n) {
    # En la i-esima replica omites
    # todas las muestras con x[i]
    guarda = (1:B)[apply(indices, MARGIN=1,
                    FUN = function(k) {!any(k == i)})]
    se.jack[i] = sd(theta.b[guarda])
 }
```

Varianza estimadores bootstrap

```
print(sd(theta.b))
```

```
[1] 0.1034732
```

```
print(sqrt((n-1) * mean((se.jack - mean(se.jack))^2)))
```

```
[1] 0.03425426
```