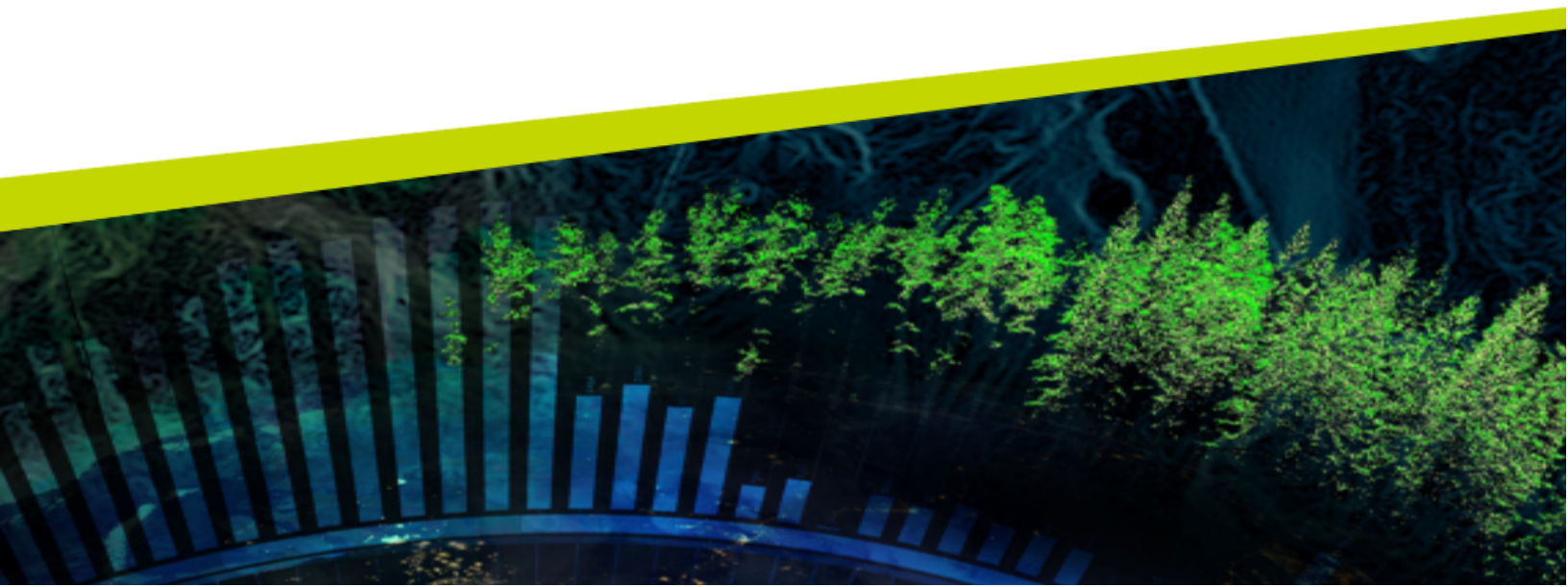
INFERENCIA Y MODELOS ESTADÍSTICOS

Jacqueline Köhler C. y José Luis Jara V.



CAPÍTULO 12. ALTERNATIVAS MÁS RECIENTES PARA ANALIZAR DATOS PROBLEMÁTICOS

Como ya sabemos, muchos procedimientos estadísticos requieren que los datos cumplan con ciertas propiedades o condiciones, lo que no siempre ocurre. En el capítulos anteriores hemos visto que, ante este escenario, podemos intentar analizar datos transformados (sección 11.1) o usar como alternativa algún métodos no paramétrico (capítulo 8 y secciones 11.2 y 11.3). Pero estas no son las únicas opciones como veremos en este capítulo donde abordaremos otras estrategias, que podemos usar cuando necesitamos analizar datos problemáticos, que son más recientes y que requieren cierto poder de cómputo.

12.1 MÉTODOS ROBUSTOS

Pensemos como ejemplo en la prueba t de Student (capitulo 5), que se usa para inferir acerca de la media de una población. Sin embargo, como mencionamos en el capítulo 2, esta medida de tendencia central tiene el problema de ser sensible a la presencia de valores atípicos, a distribuciones asimétricas o a muestras muy pequeñas. En términos generales, el incumplimiento de las condiciones del supuesto de normalidad puede causar diversos problemas:

- Resultados sesgados.
- Intervalos de confianza sub o sobreestimados.
- Reducción del poder estadístico de la prueba.

En el capítulo 2 mencionamos la existencia de estimadores robustos, poco sensibles a asimetrías muestrales o valores atípicos. No obstante, el paradigma estadístico tradicional no suele considerarlos. En esta sección, basada en las ideas expuestas por Mair y Wilcox (2020), aborda pruebas alternativas para muchas de las pruebas estudiadas hasta ahora, disponibles en el paquete `WRS2` de R, basadas en estimadores robustos.

12.1.1 Alternativas robustas a la media

En el capítulo 2 conocimos distintas medidas de tendencia central. Entre ellas vimos que la mediana, correspondiente al valor central (o el promedio de los dos valores centrales) de la muestra ordenada, es una alternativa robusta a la media. No obstante, existen otras opciones que nos pueden ser útiles.

12.1.1.1 Media truncada

La **media truncada** es bastante similar a la media aritmética que ya conocemos, con la diferencia de que se calcula descartando un determinado porcentaje (γ) de los valores en ambos extremos (colas) del conjunto de datos. Tomemos como ejemplo una muestra x con 10 elementos, los cuales han sido ordenados por simplicidad:

$$x = \{5, 20, 37, 38, 40, 43, 43, 45, 87, 91\}$$

Si calculamos la media para la muestra anterior, tenemos que es $\overline{x} = 44{,}9$. No obstante, si observamos la muestra del ejemplo con detención, podemos darnos cuenta de que los valores extremos parecen ser atípicos y, en consecuencia, pueden tener una gran influencia en el valor resultante para la media. Así, podría ser más adecuado calcular la media truncada con $\gamma = 0,2$, es decir, podando el 20 % de los valores más pequeños y el 20 % de los valores más grandes, con lo que obtendremos:

$$\overline{x}_t = \frac{37 + 38 + 40 + 43 + 43 + 45}{6} = 41$$

En R, podemos calcular la media truncada mediante la ya conocida función `mean()` del paquete `base`[1], agregando el argumento adicional `trim` con la proporción γ de los datos extremos a descartar, esto es `mean(x, trim = 0.2)` para el ejemplo. Notemos que si $\gamma = 0{,}5$ (`trim = 0.5`), se obtiene la mediana del conjunto de datos.

12.1.1.2 Media Winsorizada

Un problema de la media truncada es que, al usarla, descartamos muchos datos, lo que puede causar problemas cuando hay pocos datos. Otra opción puede ser, en lugar de descartar los valores extremos en cada cola, reemplazarlos por los valores extremos que no serían descartados al usar la media truncada y luego calcular la media con la muestra modificada. A esta medida se le conoce como **media Winsorizada**. Si retomamos nuestro ejemplo para la media truncada, los valores extremos tras la operación de truncado son 37 y 45. Así, reemplazamos los valores truncados en la muestra original por estos nuevos extremos, con lo que nuestra muestra Winsorizada es:

$$x = \{37, 37, 37, 38, 40, 43, 43, 45, 45, 45\},$$

con lo que la media Winsorizada es:

$$\overline{x}_W = \frac{37 + 37 + 37 + 38 + 40 + 43 + 43 + 45 + 45 + 45}{10} = 41$$

En R, podemos hacer este cálculo mediante la función `winmean(x, tr)` del paquete `WRS2`, donde:

- `x`: vector con los datos originales.
- `tr`: proporción de los datos a Winsorizar en cada extremo.

Así, la llamada para el ejemplo sería `winmean(x, tr = 0.2)`.

12.1.2 Prueba de Yuen para dos muestras independientes

La **prueba de Yuen** es una buena alternativa a la prueba t de Student para muestras independientes cuando las varianzas de ambas muestras son muy diferentes o los tamaños de las muestras son muy dispares. Utiliza las medias truncadas y las medias Winsorizadas, aunque no se recomienda usar esta prueba si las muestras se truncan cerca del nivel de sus medianas ($\gamma \approx 0{,}5$).

Cuando tenemos dos muestras independientes, el estadístico de prueba está dado por la ecuación 12.1, donde $\overline{x}_{ti}$ son las medias truncadas de cada muestra.

$$T_y = \frac{\overline{x}_{t1} - \overline{x}_{t2}}{\sqrt{d_1 + d_2}} \tag{12.1}$$

El denominador de la ecuación 12.1 corresponde al error estándar, donde d_i se calcula como señala la ecuación 12.2, con:

- s_{w1}: desviación estándar de la muestra Winsorizada.
- n_i: tamaño de la muestra original.
- h_i: tamaño de la muestra truncada.

$$d_i = \frac{(n_i - 1)s_{wi}^2}{h_i(h_i - 1)} \tag{12.2}$$

El estadístico T_y sigue una distribución t cuyos grados de libertad se calculan mediante la ecuación 12.3.

[1] Que se carga automáticamente al iniciar una sesión en R.

$$\nu_y = \frac{(d_1 + d_2)^2}{\frac{d_1^2}{h_1 - 1} + \frac{d_2^2}{h_2 - 1}} \tag{12.3}$$

A su vez, la ecuación 12.4 muestra cómo se construye el intervalo de confianza, donde t corresponde al cuantil $1 - \alpha/2$ de la distribución t con ν_y grados de libertad.

$$(\overline{x}_{t1} - \overline{x}_{t2}) \pm t\sqrt{d_1 + d_2} \tag{12.4}$$

Para una prueba bilateral, las hipótesis contrastadas son que las medias truncadas de las poblaciones de origen de las muestras son iguales:

H_0: $\mu_{t1} = \mu_{t2}$
H_A: $\mu_{t1} \neq \mu_{t2}$

En R, podemos aplicar la prueba de Yuen para muestras independientes mediante una llamada a la función `yuen(formula, data, tr)` del paquete `WRS2`, donde:

- `formula`: tiene la forma <variable dependiente>∼<variable independiente>. Note que la variable independiente debe tener dos niveles, a fin de determinar a qué muestra pertenece cada observación de la variable dependiente.
- `data`: matriz de datos.
- `tr`: parámetro γ de la poda.

Para pruebas unilaterales, sin embargo, se recomienda usar una variante que usa una técnica estadística llamada *bootstrapping*, y que estudiaremos más adelante en este capítulo, implementada en la función `yuen(formula, data, tr, nboot)`, donde `nboot` señala la cantidad de repeticiones a realizar mediante bootstrapping.

El script 12.1 muestra un ejemplo de uso de la prueba de Yuen para dos muestras independientes del tiempo promedio de ejecución (en milisegundos) de dos algoritmos para un conjunto de 70 instancias del problema de igual tamaño seleccionadas aleatoriamente. Estas instancias fueron asignadas al azar a cada uno de los algoritmos, resultando que $n_A = 40$ de ellas fueran resueltas por el algoritmo A y las restantes $n_b = 30$ por el algoritmo B. Se ha establecido para este estudio un nivel de significación $\alpha = 0{,}05$.

Las líneas 19–20 del script 12.1 construyen gráficos Q-Q para comprobar el supuesto de normalidad requerido por la prueba t de Student, obteniéndose como resultado la figura 12.1, donde podemos observar que las muestras obtenidas presentan desviaciones importantes de una distribución normal, especialmente para el caso del algoritmo A.

Para ilustrar los conceptos asociados a la prueba de Yuen, las líneas 28–45 del script 12.1 truncan ambas muestras considerando $\gamma = 0,2$ y construyen gráficos Q-Q para las muestras podadas (figura 12.2). Podemos apreciar que, tras la poda, los datos siguen distribuciones que se aproxima más a la normal.

Finalmente, las líneas 48–49 del script 12.1 efectúan la prueba de Yuen para ambas muestras, obteniéndose como resultado (figura 12.3) una diferencia entre las medias truncadas de 0,246, con intervalo de 95 % de confianza $(-0{,}859; 1{,}351)$ y tamaño del efecto de 0,090. El valor p obtenido, $p = 0{,}653$, no es significativo al nivel de significación establecido, por lo que concluimos con 95 % de confianza que no es posible descartar que ambos algoritmos tienen, en promedio, igual tiempo de ejecución.

Script 12.1: prueba de Yuen para dos muestras independientes.

```
1 library(WRS2)
2 library(ggpubr)
3
4 # Construir data frame.
5 a <- c(25.1, 25.2, 25.3, 25.3, 25.4, 25.4, 25.5, 25.5, 25.6, 25.8, 25.8,
6       25.9, 25.9, 26.0, 26.0, 26.2, 26.2, 26.2, 26.3, 26.4, 26.5, 26.5,
7       26.6, 26.7, 26.7, 26.9, 26.9, 27.0, 27.1, 27.3, 27.8, 28.4, 28.5,
8       29.0, 29.8, 30.2, 31.8, 31.9, 33.3, 33.7)
```

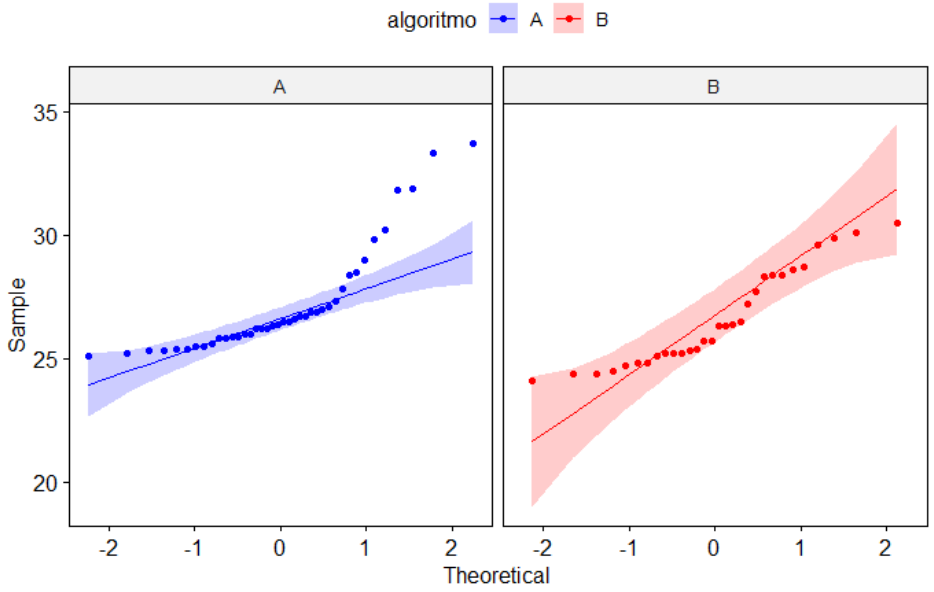


Figura 12.1: gráfico Q-Q de las muestras originales.

```r

b <- c(24.1, 24.4, 24.4, 24.5, 24.7, 24.8, 24.8, 25.1, 25.2, 25.2, 25.2,
       25.3, 25.4, 25.7, 25.7, 26.3, 26.3, 26.4, 26.5, 27.2, 27.7, 28.3,
       28.4, 28.4, 28.6, 28.7, 29.6, 29.9, 30.1, 30.5)

tiempo <- c(a, b)
algoritmo <- c(rep("A", length(a)), rep("B", length(b)))
datos <- data.frame(tiempo, algoritmo)

# Comprobar normalidad.
g <- ggqqplot(datos, x = "tiempo", facet.by = "algoritmo",
              palette = c("blue", "red"), color = "algoritmo")

print(g)

# Establecer nivel de significación.
alfa <- 0.05

# Ver poda del 20%.
gamma <- 0.2
n_a <- length(a)
n_b <- length(b)

poda_a <- n_a * gamma
poda_b <- n_b * gamma

a_truncada <- a[poda_a:(n_a - poda_a)]
b_truncada <- b[poda_b:(n_b - poda_b)]

tiempo <- c(a_truncada, b_truncada)
algoritmo <- c(rep("A", length(a_truncada)), rep("B", length(b_truncada)))
datos_truncados <- data.frame(tiempo, algoritmo)

g <- ggqqplot(datos_truncados, x = "tiempo", facet.by = "algoritmo",
              palette = c("blue", "red"), color = "algoritmo")
```

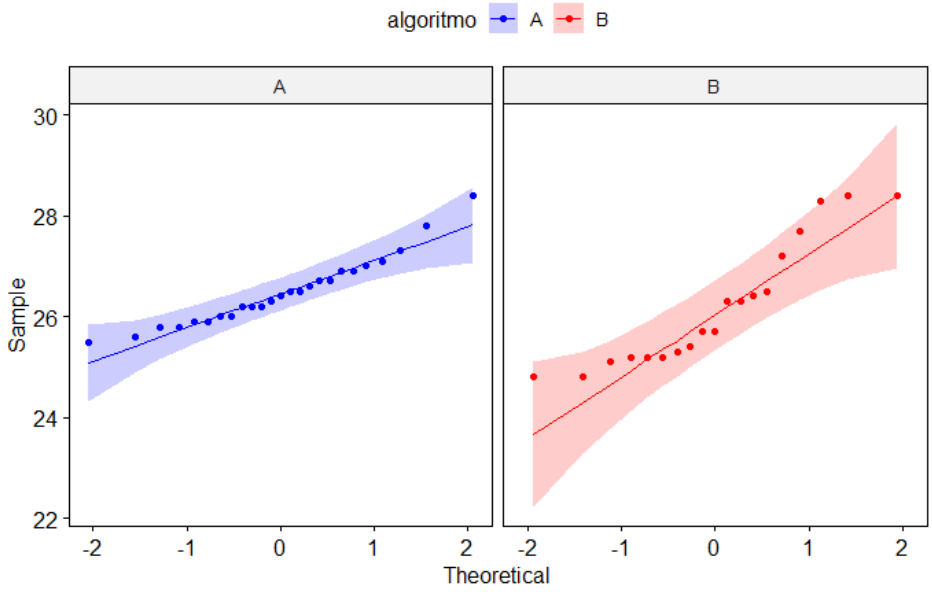


Figura 12.2: gráfico Q-Q de las muestras truncadas.

```
Call:
yuen(formula = tiempo ~ algoritmo, data = datos, tr = gamma)

Test statistic: 0.455 (df = 29.05), p-value = 0.65252

Trimmed mean difference:  0.24583
95 percent confidence interval:
-0.8592     1.3509

Explanatory measure of effect size: 0.09
```

Figura 12.3: resultado de la prueba de Yuen para el ejemplo.

```
44
45 print(g)
46
47 # Aplicar prueba de Yuen.
48 prueba <- yuen(tiempo ~ algoritmo, data = datos, tr = gamma)
49 print(prueba)
```

El paquete `WRS2` incluye también la función `pb2gen(formula, data, est, nboot)`, que usa bootstrapping para aplicar la prueba de Yuen usando otras medidas robustas de tendencia central, donde:

- `formula`: tiene la misma forma descrita para la prueba de Yuen.
- `data`: matriz de datos.
- `est`: medida a emplear. Puede tomar las opciones `"mean"` para la media y `"median"` (mediana), entre otras opciones que escapan a los alcances de este curso.
- `nboot`: cantidad de repeticiones bootstrapping.

El script 12.2 muestra cómo aplicar la prueba de Yuen para el ejemplo, usando como estimadores la media y la mediana, obteniéndose los resultados de la figura 12.4. Podemos ver que, en ambos casos, el valor p obtenido es más bajo que al usar la media podada.

```
Resultado al usar la media como estimador

Call:
pb2gen(formula = tiempo ~ algoritmo, data = datos, est = "mean",
    nboot = bootstrap)

Test statistic: 0.61, p-value = 0.21321
95% confidence interval:
-0.3008    1.5617

Resultado al usar la mediana como estimador

Call:
pb2gen(formula = tiempo ~ algoritmo, data = datos, est = "median",
    nboot = bootstrap)

Test statistic: 0.45, p-value = 0.47147
95% confidence interval:
-0.95    1.35
```

Figura 12.4: resultado de la prueba de Yuen con bootstrapping para el ejemplo, usando como estimadores la media y la mediana.

Script 12.2: prueba de Yuen con bootstrapping para dos muestras independientes usando la media y la mediana.

```r
library(WRS2)

# Construir data frame.
a <- c(25.1, 25.2, 25.3, 25.3, 25.4, 25.4, 25.5, 25.5, 25.6, 25.8, 25.8,
       25.9, 25.9, 26.0, 26.0, 26.2, 26.2, 26.2, 26.3, 26.4, 26.5, 26.5,
       26.6, 26.7, 26.7, 26.9, 26.9, 27.0, 27.1, 27.3, 27.8, 28.4, 28.5,
       29.0, 29.8, 30.2, 31.8, 31.9, 33.3, 33.7)

b <- c(24.1, 24.4, 24.4, 24.5, 24.7, 24.8, 24.8, 25.1, 25.2, 25.2, 25.2,
       25.3, 25.4, 25.7, 25.7, 26.3, 26.3, 26.4, 26.5, 27.2, 27.7, 28.3,
       28.4, 28.4, 28.6, 28.7, 29.6, 29.9, 30.1, 30.5)

tiempo <- c(a, b)
algoritmo <- c(rep("A", length(a)), rep("B", length(b)))
datos <- data.frame(tiempo, algoritmo)

# Establecer nivel de significación y cantidad de muestras a generar
# con bootstrapping.
alfa <- 0.05
bootstrap <- 999

# Aplicar prueba con la media
set.seed(135)

prueba_media <- pb2gen(tiempo ~ algoritmo,
                       data = datos,
                       est = "mean",
                       nboot = bootstrap)

cat("\n\nResultado al usar la media como estimador\n\n")
print(prueba_media)

```

```r
# Aplicar prueba con la mediana
set.seed(135)

prueba_mediana <- pb2gen(tiempo ~ algoritmo,
                         data = datos,
                         est = "median",
                         nboot = bootstrap)

cat("Resultado al usar la mediana como estimador\n\n")
print(prueba_mediana)
```

12.1.3 Prueba de Yuen para dos muestras pareadas

Para el caso de dos muestras apareadas, podemos generalizar el estadístico de la prueba de Yuen con medias truncadas como muestra la ecuación 12.5, donde el nuevo término d_{12} está dado por la ecuación 12.6. En este caso, ambas muestras tienen igual tamaño n y h corresponde al tamaño de la muestra combinada tras la poda.

$$T_y = \frac{\overline{x}_{t1} - \overline{x}_{t2}}{\sqrt{d_1 + d_2 - 2\,d_{12}}} \tag{12.5}$$

$$d_{12} = \frac{1}{h(h-1)} \sum_{i=1}^{n} (x_{i1} - \overline{x}_1) \cdot (x_{i2} - \overline{x}_2) \tag{12.6}$$

Supongamos ahora que queremos comparar el rendimiento de dos algoritmos X e Y, para lo cual hemos seleccionado aleatoriamente 25 instancias del problema y registrado su tiempo de ejecución en milisegundos con cada uno de los algoritmos. Considerando una prueba bilateral, las hipótesis contrastadas son que la media truncada del tiempo de ejecución del algoritmo X es igual a la media truncada del tiempo de ejecución promedio del algoritmo Y:

H_0: $\mu_{t1} = \mu_{t2}$
H_A: $\mu_{t1} \neq \mu_{t2}$

Debemos notar que estas son las mismas hipótesis contrastadas por la prueba de Yuen para muestras independientes.

El script 12.3 ilustra el uso de la función `yuend(x, y, tr)` del paquete `WRS2`, que compara las medias truncadas, obteniéndose, para este ejemplo, el resultado que muestra la figura 12.5.

```
Call:
yuend(x = x, y = y, tr = gamma)

Test statistic: -4.5915 (df = 14), p-value = 0.00042

Trimmed mean difference:  -0.76
95 percent confidence interval:
-1.115     -0.405

Explanatory measure of effect size: 0.44
```

Figura 12.5: resultado de la prueba de Yuen para las muestras apareadas del ejemplo, usando como estimador la media truncada.

Script 12.3: prueba de Yuen para dos muestras pareadas.

```
library(WRS2)

# Construir data frame.
x <- c(32.0, 32.0, 32.0, 32.0, 32.1, 32.1, 32.1, 32.2, 32.3, 32.3, 32.5,
       32.7, 32.7, 32.7, 33.1, 33.4, 33.9, 34.1, 34.2, 34.5, 36.0, 36.6,
       36.7, 37.2, 38.0)

y <- c(33.0, 33.0, 33.0, 33.0, 33.0, 33.0, 33.3, 33.3, 33.3, 33.3, 33.5,
       33.6, 33.7, 33.9, 33.9, 34.2, 34.2, 34.3, 34.3, 34.4, 34.5, 34.6,
       36.4, 38.9, 40.2)

# Fijar nivel de significación.
alfa <- 0.05

# Aplicar prueba de Yuen para muestras pareadas.
gamma <- 0.2
prueba <- yuend(x = x, y = y, tr = gamma)
print(prueba)
```

Puesto que el valor p obtenido, $p < 0{,}001$, es menor que el nivel de significación, la evidencia es suficientemente fuerte como para rechazar la hipótesis nula en favor de la hipótesis alternativa. En consecuencia, podemos afirmar con 95 % de confianza que existe una diferencia estadísticamente significativa en el desempeño de ambos algoritmos, siendo el algoritmo X el más eficiente (puesto que la diferencia estimada entre las medias tiene signo negativo).

12.1.4 Comparaciones de una vía para múltiples grupos independientes

El paquete `WRS2` ofrece diferentes alternativas a ANOVA de una vía para muestras independientes que podemos usar cuando los tamaños muestrales son muy diferentes o no se cumple la condición de homocedasticidad.

La función `t1way(formula, data, tr, alpha)` efectúa un procedimiento similar a ANOVA usando medias truncadas. A su vez, la función `lincon(formula, data, tr, alpha)` permite realizar el procedimiento post-hoc correspondiente.

De manera similar, `t1waybt(formula, data, tr, nboot)` realiza un procedimiento análogo al anterior incorporando bootstrapping. En este caso, el procedimiento post-hoc puede realizarse mediante la función `mcppb20(formula, data, tr, nboot)`.

Una tercera opción es la función `med1way(formula, data, iter)`, que emplea la mediana y sigue un proceso iterativo. No obstante, en este caso el paquete no ofrece funciones que permitan realizar el procedimiento post-hoc.

Los argumentos asociados a las funciones mencionadas en los párrafos anteriores son:

- `formula`: de la forma <variable dependiente>∼<variable independiente>.
- `data`: matriz de datos.
- `tr`: parámetro γ de la poda.
- `alpha`: nivel de significación.
- `nboot`: cantidad de repeticiones bootstrapping.
- `iter`: cantidad de iteraciones a realizar.

El script 12.4 ilustra el funcionamiento de algunas de las funciones descritas en los párrafos precedentes, suponiendo que ahora deseamos comparar el tiempo promedio de ejecución (en milisegundos) de tres algoritmos, contando con $n_a = 40$ observaciones para el algoritmo A, $n_b = 30$ observaciones para el algoritmo B y $n_c = 35$ observaciones para el algoritmo C. Se ha establecido para este estudio un nivel de significación $\alpha = 0{,}05$. Podemos ver en las figuras 12.6 y 12.7 que las funciones `t1way()` y `t1waybt()` arrojan el mismo resultado. Puesto que el valor p obtenido, $p < 0{,}001$, es menor que el nivel de significación, rechazamos la

hipótesis nula en favor de la hipótesis alternativa. Concluimos, entonces, con 95 % de confianza, que existe una diferencia estadísticamente significativa entre los tiempos promedio de ejecución de los algoritmos.

Al efectuar los procedimientos post-hoc respectivos, los valores p obtenidos son ligeramente diferentes para ambos métodos (figuras 12.6 y 12.7). No obstante, en ambos casos podemos concluir que el algoritmo C presenta un tiempo de ejecución promedio diferente.

Si examinamos las medias de cada grupo, tenemos que $\overline{x}_A = 27{,}19$ [ms], $\overline{x}_B = 26{,}58$ [ms] y $\overline{x}_C = 25{,}52$ [ms], por lo que el algoritmo C es más rápido.

```
Comparación entre grupos usando medias truncadas

Call:
t1way(formula = tiempo ~ algoritmo, data = datos, tr = gamma,
    alpha = alfa)

Test statistic: F = 10.9813
Degrees of freedom 1: 2
Degrees of freedom 2: 34.39
p-value: 0.00021

Explanatory measure of effect size: 0.48
Bootstrap CI: [0.28; 0.68]


Procedimiento post-hoc

Call:
lincon(formula = tiempo ~ algoritmo, data = datos, tr = gamma,
    alpha = alfa)

         psihat ci.lower ci.upper p.value
A vs. B 0.24583 -1.11867  1.61033 0.65252
A vs. C 1.49583  0.65571  2.33596 0.00022
B vs. C 1.25000 -0.02757  2.52757 0.03909
```

Figura 12.6: resultado de la comparación entre múltiples grupos independientes usando medias truncadas.

Script 12.4: alternativas robustas para comparar entre múltiples grupos independientes.

```r
library(WRS2)

# Construir data frame.
a <- c(25.1, 25.2, 25.3, 25.3, 25.4, 25.4, 25.5, 25.5, 25.6, 25.8, 25.8,
       25.9, 25.9, 26.0, 26.0, 26.2, 26.2, 26.2, 26.3, 26.4, 26.5, 26.5,
       26.6, 26.7, 26.7, 26.9, 26.9, 27.0, 27.1, 27.3, 27.8, 28.4, 28.5,
       29.0, 29.8, 30.2, 31.8, 31.9, 33.3, 33.7)

b <- c(24.1, 24.4, 24.4, 24.5, 24.7, 24.8, 24.8, 25.1, 25.2, 25.2, 25.2,
       25.3, 25.4, 25.7, 25.7, 26.3, 26.3, 26.4, 26.5, 27.2, 27.7, 28.3,
       28.4, 28.4, 28.6, 28.7, 29.6, 29.9, 30.1, 30.5)

c <- c(24.5, 24.5, 24.5, 24.5, 24.5, 24.5, 24.6, 24.6, 24.6, 24.6, 24.6,
       24.6, 24.7, 24.7, 24.7, 24.7, 24.8, 25.0, 25.0, 25.0, 25.2, 25.2,
       25.2, 25.2, 25.5, 25.7, 25.9, 26.2, 26.5, 26.5, 26.7, 27.0, 29.2,
       29.9, 30.1)

tiempo <- c(a, b, c)
```

```
Comparación entre grupos usando bootstrap

Call:
t1way(formula = tiempo ~ algoritmo, data = datos, tr = gamma,
    alpha = alfa)

Test statistic: F = 10.9813
Degrees of freedom 1: 2
Degrees of freedom 2: 34.39
p-value: 0.00021

Explanatory measure of effect size: 0.48
Bootstrap CI: [0.28; 0.68]


Procedimiento post-hoc

Call:
mcppb20(formula = tiempo ~ algoritmo, data = datos, tr = gamma,
    nboot = muestras)

         psihat ci.lower ci.upper p-value
A vs. B 0.24583 -0.91667  1.61389 0.55656
A vs. C 1.49583  0.73690  2.44464 0.00000
B vs. C 1.25000  0.16270  2.34206 0.00801
```

Figura 12.7: resultado de la comparación entre múltiples grupos independientes usando medias truncadas con bootstrapping.

```r
algoritmo <- c(rep("A", length(a)), rep("B", length(b)), rep("C", length(c)))
datos <- data.frame(tiempo, algoritmo)

# Fijar nivel de significación.
alfa <- 0.05

# Comparar los diferentes algoritmos usando medias truncadas.
cat("Comparación entre grupos usando medias truncadas\n\n")
gamma <- 0.2

set.seed(666)

medias_truncadas <- t1way(tiempo ~ algoritmo, data = datos, tr = gamma,
                          alpha = alfa)

print(medias_truncadas)

if(medias_truncadas$p.value < alfa) {
  cat("\nProcedimiento post-hoc\n\n")

  set.seed(666)

  post_hoc <- lincon(tiempo ~ algoritmo, data = datos, tr = gamma,
                     alpha = alfa)

  print(post_hoc)
}
```

```r

# Comparar los diferentes algoritmos usando bootstrap.
cat("Comparación entre grupos usando bootstrap\n\n")
muestras <- 999

set.seed(666)

bootstrap <- t1waybt(tiempo ~ algoritmo, data = datos, tr = gamma,
                     nboot = muestras)

print(medias_truncadas)

if(medias_truncadas$p.value < alfa) {
  cat("\nProcedimiento post-hoc\n\n")

  set.seed(666)

  post_hoc <- mcppb20(tiempo ~ algoritmo, data = datos, tr = gamma,
                      nboot = muestras)

  print(post_hoc)
}
```

12.1.5 Comparaciones de una vía para múltiples grupos correlacionados

Desde luego, el paquete `WRS2` también ofrece opciones robustas para reemplazar el procedimiento ANOVA de una vía para muestras correlacionadas, que podemos usar cuando los datos disponibles violan la condición de esfericidad.

La función `rmanova(y, groups, blocks, tr)` efectúa un procedimiento similar a ANOVA usando medias truncadas, mientras que la función `rmmcp(y, groups, blocks, tr, alpha)` implementa el procedimiento post-hoc para dicha prueba. Por otra parte, `rmanovab(y, groups, blocks, tr, nboot)` realiza la misma tarea que `rmanova()`, incorporando bootstrapping. En este caso, el procedimiento post-hoc está dado por la función `pairdepb(y, groups, blocks, tr, nboot)`. Los argumentos para esta familia de funciones son:

- `formula`: de la forma <variable dependiente>∼<variable independiente>.
- `y`: vector con la variable dependiente.
- `groups`: vector que indica los grupos.
- `blocks`: vector que identifica los sujetos o bloques.
- `tr`: parámetro γ de la poda.
- `alpha`: nivel de significación.
- `nboot`: cantidad de repeticiones bootstrapping.

El script 12.5 muestra el uso de las funciones robustas sin bootstrapping para ANOVA de una vía con muestras correlacionadas. El ejemplo presentado aborda, una vez más, la comparación del desempeño de tres algoritmos, X, Y y Z. Para ello, se han seleccionado aleatoriamente 25 instancias del problema y se registra su tiempo de ejecución (en milisegundos) con cada uno de los algoritmos. Para este estudio consideraremos un nivel de significación $\alpha = 0{,}05$. Podemos ver los resultados obtenidos en la figura 12.8.

El valor p resultante, $p < 0{,}001$, indica que la evidencia es suficientemente fuerte para rechazar la hipótesis nula en favor de la hipótesis alternativa, por lo que realizamos el procedimiento post-hoc correspondiente. La conclusión, con 95 % de confianza, es que no todos los algoritmos tienen el mismo rendimiento promedio, siendo el algoritmo X más eficiente que los algoritmos Y y Z.

Script 12.5: alternativa robusta para comparar entre múltiples grupos correlacionados.

```r
library(WRS2)
library(tidyverse)
```

```
Call:
rmanova(y = tiempo, groups = algoritmo, blocks = instancia, tr = gamma)

Test statistic: F = 24.1706
Degrees of freedom 1: 1.5
Degrees of freedom 2: 20.96
p-value: 1e-05


Procedimiento post-hoc

Call:
rmmcp(y = tiempo, groups = algoritmo, blocks = instancia, tr = gamma,
    alpha = alfa)

          psihat ci.lower ci.upper p.value p.crit   sig
X vs. Y -0.85333 -1.16837 -0.53830 0.00000 0.0169  TRUE
X vs. Z -0.68667 -0.98245 -0.39089 0.00002 0.0250  TRUE
Y vs. Z -0.00667 -0.26776  0.25443 0.94566 0.0500 FALSE
```

Figura 12.8: resultado de las alternativa robusta para comparar entre múltiples grupos correlacionados.

```r

# Construir data frame.
X <- c(32.0, 32.0, 32.0, 32.0, 32.1, 32.1, 32.1, 32.2, 32.3, 32.3, 32.5,
       32.7, 32.7, 32.7, 33.1, 33.4, 33.9, 34.1, 34.2, 34.5, 36.0, 36.6,
       36.7, 37.2, 38.0)

Y <- c(33.0, 33.0, 33.0, 33.0, 33.0, 33.0, 33.3, 33.3, 33.3, 33.3, 33.5,
       33.6, 33.7, 33.9, 33.9, 34.2, 34.2, 34.3, 34.3, 34.4, 34.5, 34.6,
       36.4, 38.9, 40.2)

Z <- c(32.0, 32.2, 32.5, 32.6, 32.7, 32.7, 32.7, 33.0, 33.2, 33.4, 33.6,
       33.6, 33.9, 34.1, 34.2, 34.4, 34.4, 34.5, 34.6, 34.7, 36.3, 36.6,
       36.7, 38.9, 39.2)

instancia <- 1:length(X)
datos <- data.frame(instancia, X, Y, Z)

# Llevar data frame a formato largo.
datos <- datos %>% pivot_longer(c("X", "Y", "Z"), names_to = "algoritmo",
                                values_to = "tiempo")

datos[["algoritmo"]] <- factor(datos[["algoritmo"]])

# Fijar nivel de significación.
alfa <- 0.05

# Aplicar alternativa robusta para ANOVA de una vía con
# muestras correlacionadas.
gamma <- 0.2

prueba <- rmanova(y = datos[["tiempo"]], groups = datos[["algoritmo"]],
                  blocks = datos[["instancia"]], tr = gamma)

print(prueba)

```

```r
if(prueba$p.value < alfa) {
  cat("\nProcedimiento post-hoc\n\n")

  post_hoc <- rmmcp(y = datos[["tiempo"]], groups = datos[["algoritmo"]],
                    blocks = datos[["instancia"]], tr = gamma, alpha = alfa)

  print(post_hoc)
}
```

Nota importante

Debemos saber que el paquete `WRS2` implementa muchas más estimadores robustos y pruebas de hipótesis basadas en estos estimadores. Entre los no mencionados en esta sección, hay medidas y procedimientos robustos para obtener intervalos de confianza a partir de una muestra, alternativas para ANOVA de más de una vía, para modelos mixtos y para analizar regresión. También hay una familia de funciones que permite comparar variables aleatorias discretas (proporciones).

No es posible discutir las bases teóricas de todos estos métodos en esta sección, pero los desarrollos matemáticos y teoremas asociados a sus distribuciones muestrales se presentan en detalle en Wilcox (2012). Este libro se ha convertido en un referente de la estadística robusta, y presenta muchos más conceptos y métodos que los implementados en `WRS2`.

R. R. Wilcox no es el único investigador en esta área, como puede verse en la página de R dedicada al tema (Maechler, 2014).

12.1.6 Ejercicios propuestos para la sección 12.1

1. El conjunto de datos `diet` del paquete `WRS2` contiene datos de la pérdida de peso conseguida por tres tipos de dietas. Determina si la pérdida de peso conseguida por las mujeres con las dietas A y C es la misma.
2. El conjunto de datos `essays` del paquete `WRS2` se compone de datos recolectados por un estudio de los efectos de dos formas de retroalimentación sobre la calidad de la escritura académica producida por estudiantes universitarios/as de inglés como lengua extranjera. Tres de grupos de estudiantes, dos de tratamiento (las dos formas de retroalimentación) y uno de control, se formaron de forma aleatoria. Cada estudiante escribió cuatro ensayos: uno antes del tratamiento, uno de práctica durante el tratamiento, uno terminado el tratamiento y el último un mes después del tratamiento. Obviamente, estudiantes del grupo de control realizaron las tareas de escritura pero no recibieron retroalimentación. Determina si una de las formas de retroalimentación estudiadas (directa o indirecta) es mejor que la otra (considera el ensayo 3 realizado al finalizar la intervención para este análisis).
3. Considera el conjunto de datos `essays` descrito en la pregunta 2. Determina si las y los estudiantes del grupo de control pudieron mejorar la tasa de errores cometidos en el tercer ensayo respecto del segundo.
4. Considera el conjunto de datos `essays` descrito en la pregunta 2. Determina si las y los estudiantes que recibieron retroalimentación directa mantuvieron la misma tasa de errores en el tercer y cuarto ensayo.
5. El conjunto de datos `bush` del paquete `WRS2` contiene datos de un "reality" australiano en que los participantes que comer palotes, ojos de pescado, testículos de canguro y larvas de una polilla. Determina si el tiempo que se tarda una persona en tener arcadas al comer estas cosas es el mismo usando métodos robustos.
6. Considera el conjunto de datos `essays` descrito en la pregunta 2. Determina si la tasa de errores cometidos en el tercer ensayo son las mismas para cada uno de los tres grupos de estudiantes.
7. Considera el conjunto de datos `essays` descrito en la pregunta 2. Determina si las tasas de errores mejoran al aplicar la retroalimentación indirecta y, si existe, esta se mantiene un mes después de la intervención.

12.6 BIBLIOGRAFÍA DEL CAPÍTULO

Amat Rodrigo, J. (2016). *Resampling: test de permutación, simulación de Monte Carlo y Bootstrapping.* Consultado el 31 de mayo de 2021, desde https://www.cienciadedatos.net/documentos/23_resampling_test_permutacion_simulacion_de_monte_carlo_bootstrapping

Hesterberg, T., Monaghan, S., Moore, D. S., Clipson, A., & Epstein, R. (2003). *Bootstrap Methods and Permutation Tests.* Consultado el 3 de junio de 2021, desde https://statweb.stanford.edu/~tibs/stat315a/Supplements/bootstrap.pdf

Maechler, M. (2014). *CRAN task view: Robust statistical methods.* https://cran.r-project.org/web/views/Robust.html

Mair, P., & Wilcox, R. (2020). Robust statistical methods in R using the WRS2 package. *Behavior Research Methods*, *52*(2), 464-488.

Wilcox, R. R. (2012). *Introduction to robust estimation and hypothesis testing* (3.ª ed.). Academic Press.