

---

Curso OpenCourseWare

**Aprendizaje del Software Estadístico R: un entorno  
para simulación y computación estadística**

Alberto Muñoz García

---

**11. Repaso de diversos contrastes estadísticos**



Dada la heterogeneidad de los alumnos del curso, y teniendo en cuenta que el curso no es un curso de aprendizaje de estadística, sino más bien de aplicación de técnicas estadísticas, que se suponen en general conocidas, utilizando un determinado paquete estadístico, no vamos a explicar aquí los fundamentos de los contrastes, sino tan sólo vamos a dar algunas funciones útiles para realizar contrastes en R.

Por si a alguien le resultase de utilidad, vamos a dar algunas páginas con material didáctico sobre contraste de hipótesis, en general, accesibles por internet:

<https://labs.la.utexas.edu/gilden/files/2016/05/Statistics-Text.pdf> ("Statistical Methods for Psychology" textbook by David C. Howell)

<http://davidmlane.com/hyperstat/> (Hyperstat Online Statistical Textbook, David Lane, Rice University)

[http://pse.cs.vt.edu/SoSci/converted/Hypoth\\_I/activity.html](http://pse.cs.vt.edu/SoSci/converted/Hypoth_I/activity.html) (tutoriales estadísticos, Virginia Tech).

Existe en R una librería de funciones para realizar contrastes estadísticos, llamada ctest. Para cargarla, escribiremos:

```
> library(ctest)
```

Dicha librería contiene bastantes funciones, pero sólo nos vamos a fijar en algunas que implementan contrastes muy comúnmente utilizados. Las funciones que vamos a ver son las siguientes:

<b>binom.test</b>	Test exacto sobre el parámetro de una binomial
<b>cor.test</b>	Test de asociación entre muestras apareadas
<b>wilcox.test</b>	Test de suma de rangos de Wilcoxon para una y dos muestras
<b>prop.test</b>	Test de igualdad de proporciones
<b>chisq.test</b>	Test de la chi-cuadrado para datos de conteo
<b>fisher.test</b>	Test exacto de Fisher para datos de conteo
<b>ks.test</b>	Test de Kolmogorov-Smirnov para ajuste de datos a distribuciones dadas
<b>shapiro.test</b>	Test de Shapiro para comprobar ajuste de datos a una distribución normal

<b>oneway.test</b>	Test para comprobar la igualdad de medias entre varios grupos de datos
<b>var.test</b>	Test para comprobar la igualdad de varianzas entre dos grupos de datos

Vamos a poner ejemplos de cada contraste, para ver qué tipo de problemas resuelven y tener una base para su aplicación.

### **binom.test**

Lleva a cabo un contraste exacto (no aproximado) sobre el valor de la probabilidad de éxito en un experimento de Bernoulli.

Por ejemplo, tiramos una moneda al aire 600 veces, y nos salen 300 caras. Y queremos contrastar la hipótesis de que la moneda está equilibrada, esto es, la probabilidad de que  $p=0.5$ . Para ello diremos cuántas veces salió cara y cuántas veces salió cruz:

**> binom.test(c(300,300),p=0.5)**

Exact binomial test

data: 300 and 600

number of successes = 300, number of trials = 600, p-value = 1

alternative hypothesis: true probability of success is not equal to 0.5

95 percent confidence interval:

0.4592437 0.5407563

sample estimates:

probability of success

0.5

Como el p-valor ha salido 1 (es lo más alto posible), no rechazaremos la hipótesis nula. Esto es, daremos por bueno que la moneda estaba equilibrada. Observamos que también obtenemos un intervalo de confianza al 95% para el verdadero valor de la probabilidad de obtener cara. Dicho intervalo es (0.459, 0.540) en nuestro ejemplo.

Otro ejemplo: tiramos la moneda al aire y obtenemos 30 caras y 100 cruces:

**> binom.test(c(30,100),p=0.5)**

### Exact binomial test

data: 30 and 130

number of successes = 30, number of trials = 130, p-value = 5.421e-10

alternative hypothesis: true probability of success is not equal to 0.5

95 percent confidence interval:

0.1614375 0.3127614

sample estimates:

probability of success

0.2307692

En este caso, el p-valor es prácticamente cero, lo que quiere decir que debemos rechazar la hipótesis de que la moneda está equilibrada. Como el intervalo de confianza resulta ser (0.16, 0.31), concluiremos que la probabilidad de obtener cara para la moneda utilizada está probablemente dentro de dicho intervalo.

### **prop.test**

Con esta función contrastaremos si las proporciones en varios grupos son las mismas, o bien que dichas proporciones equivalen a unas determinadas.

Para empezar, tiraremos una moneda al aire 100 veces, y nos preguntaremos si la proporción de caras puede ser considerada el 50% :

```
> caras = rbinom(1, size=100, pr = .5)
```

```
> caras
```

```
[1] 51
```

```
> prop.test(caras,100)
```

### 1-sample proportions test with continuity correction

data: caras out of 100, null probability 0.5

X-squared = 0.01, df = 1, p-value = 0.9203

alternative hypothesis: true p is not equal to 0.5

95 percent confidence interval:

0.4086512 0.6105719

sample estimates:

p

0.51

Como el p-valor es muy alto, 0.92, no rechazaremos la hipótesis nula de que  $p=0.5$ . En este caso, el test funciona como en el caso de `binom.test`.

Supongamos ahora que queremos saber si dos muestras han salido de la misma población.

Por ejemplo, queremos saber si las mujeres estudian un grupo de carreras diferentes en la misma proporción. Para ello miramos la lista de matriculados en cuatro carreras y comprobamos si las proporciones de mujeres en las mismas coinciden:

```
> estudiantes = c(100,200,50,500)
```

```
> names(estudiantes) = c("quimica","derecho","matematicas","economia")
```

```
> estudiantes
```

```
quimica  derecho  matematicas  economia
  100     200      50      500
```

```
> estudiantes.mujeres = c(60,120,10,200)
```

```
> prop.test(estudiantes.mujeres,estudiantes)
```

```
4-sample test for equality of proportions without continuity
correction
```

```
data: estudiantes.mujeres out of estudiantes
```

```
X-squared = 44.5373, df = 3, p-value = 1.160e-09
```

```
alternative hypothesis: two.sided
```

```
sample estimates:
```

```
prop 1 prop 2 prop 3 prop 4
```

```
0.6  0.6  0.2  0.4
```

La conclusion es que no. Las mujeres, en este ejemplo, se matriculan de las carreras en distinta proporcion. Es aparente que se matriculan menos en Matematicas, en este ejemplo, pero el uso del contraste nos asegura con un alto grado de seguridad que esas diferencias observadas no son debidas al azar, sino que son estructurales.

### **chisq.test**

Este contraste se utiliza para tablas de contingencia. El caso habitual será disponer de una tabla de contingencia que cruza dos variables, contando el número de coincidencias según las categorías de ambas variables, y se trata de ver si las variables son o no son independientes.

Para poner un ejemplo, supongamos que tenemos una muestra de personas en las que hemos observado dos variables: el color de ojos, que puede ser "claro" y "oscuro", y el color de pelo, que puede ser "rubio" ó "moreno". Entonces hacemos una tabla de contingencia para cruzar dichas características, y aplicamos el test para ver si el "color de ojos" y el "color de pelo" son variables independientes:

Suponemos que los datos han sido recogidos previamente en una variable llamada 'datos':

```
> datos
```

```
  claro oscuro
```

```
rubio  20   2
```

```
moreno  3  18
```

```
> chisq.test(datos)
```

```
Pearson's Chi-squared test with Yates' continuity correction
```

```
data: datos
```

```
X-squared = 22.3693, df = 1, p-value = 2.249e-06
```

Como el p-valor es prácticamente cero, concluiremos que hay que rechazar la hipótesis nula, que es que hay independencia entre las variables observadas: el color de ojos y el color de pelo guardan correlación en estos datos.

Repitamos el ejemplo, construyendo la tabla de contingencia para repasar cómo se hace, para un conjunto de datos distinto y más pequeño que el anterior:

```

> color.ojos = c("claro","oscuro","oscuro","oscuro","claro","claro","claro",
"oscuro","oscuro","claro","claro","claro","claro","oscuro","oscuro")
> color.pelo = c("rubio","moreno","moreno","rubio","moreno","rubio","rubio",
"moreno","rubio","moreno","rubio","rubio","rubio","moreno","moreno")
> table(color.ojos,color.pelo)

      color.pelo
color.ojos moreno rubio
claro      2      6
oscuro     5      2
> chisq.test(table(color.ojos,color.pelo))

```

Pearson's Chi-squared test with Yates' continuity correction

```

data: table(color.ojos, color.pelo)
X-squared = 1.637, df = 1, p-value = 0.2007

```

Warning message:

Chi-squared approximation may be incorrect in: `chisq.test(table(color.ojos, color.pelo))`

En este caso, el p-valor es 0.2, que es muy alto: no rechazaremos la hipótesis de que las variables color de ojos y color de pelo son independientes. El mensaje de aviso que da el test es porque el número de casos utilizado es muy pequeño.

**cor.test**

**wilcox.test**

Son dos tests para ver si hay asociación entre muestras de datos. Por ejemplo, una serie de alumnos pueden ser sometidos a dos test de inteligencia distintos. Si hay 100 alumnos, tendríamos dos vectores de longitud 100: el primero sería la puntuación alcanzada por los alumnos en el primer test, y el segundo la puntuación de dichos alumnos para el segundo test. Los contrastes `cor.test` y `wilcox.test` servirían para ver si hay correlación entre ambos tests.

Como advertimos al principio del tema, se supone que el fundamento del contraste se conoce. De todos modos, digamos que el test de Wilcoxon es no paramétrico, y tiene más en cuenta el orden correlativo de los datos que el valor de los datos mismos. Para apreciar las diferencias, pongamos unos ejemplos:

Primero generamos una secuencia de datos, y a partir de la misma, otras dos, relacionadas de alguna manera, y veremos

qué dicen los tests al respecto:

```
> x = seq(-5,5)
```

```
> x
```

```
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5
```

```
> y = x^2
```

```
> y
```

```
[1] 25 16 9 4 1 0 1 4 9 16 25
```

```
# Vemos que entre x e y hay una relación, aunque sea no lineal
```

```
> z = x^3
```

```
> z
```

```
[1] -125 -64 -27 -8 -1 0 1 8 27 64 125
```

```
>
```

```
# Vemos que entre x y z también hay relación
```

```
> cor.test(x,y)
```

```
Pearson's product-moment correlation
```

```
data: x and y
```

```
t = 0, df = 9, p-value = 1
```

```
alternative hypothesis: true correlation is not equal to 0
```

```
sample estimates:
```

```
cor
```

```
0
```

```
# La relación entre x e y no es detectada por el test cor.test: el p-valor ha salido 1.
```

```
> wilcox.test(x,y)
```



### Wilcoxon rank sum test with continuity correction

```
data: x and y
W = 17.5, p-value = 0.005106
alternative hypothesis: true mu is not equal to 0

Warning message:
Cannot compute exact p-value with ties in: wilcox.test.default(x, y)
>
# La relación entre x e y sí ha sido detectada por wilcox.test.
> cor.test(x,z)
```

### Pearson's product-moment correlation

```
data: x and z
t = 7.1257, df = 9, p-value = 5.51e-05
alternative hypothesis: true correlation is not equal to 0
sample estimates:
  cor
0.921649
# En este caso, cor.test sí detecta la relación, al mantenerse el orden de los datos.
```

### **fisher.test**

Este test tiene el mismo fin que el test `chisq.test`, pero otro fundamento, y es útil para muestras pequeñas, caso para el cual `chisq.test` no es adecuado. El ejemplo que trae la librería es muy adecuado. Cierta dama fina (inglesa, por supuesto), era aficionada a tomar te con leche (¡cómo no!), y presumía de ser capaz de distinguir cuando su criada le echaba primero la leche y luego el té, o bien echaba primero el té y después la leche. Tal habilidad era mirada con incredulidad por sus allegados. Para salir de dudas, Fisher le dio a probar a la dama una serie de tazas de té, en las que unas veces se había echado primero la leche y otras primero el té. Fisher apuntó los aciertos de la dama y con la tabla de contingencia así construida, veamos lo que pasó....

```
> TeaTasting =  
matrix(c(3, 1, 1, 3),  
       nr = 2,  
       dimnames = list(Guess = c("Milk", "Tea"),  
                       Truth = c("Milk", "Tea")))  
> fisher.test(TeaTasting, alternative = "greater")
```

Fisher's Exact Test for Count Data

```
data: TeaTasting  
p-value = 0.2429  
alternative hypothesis: true odds ratio is greater than 1  
95 percent confidence interval:  
0.3135693    Inf  
sample estimates:  
odds ratio  
6.408309
```

Dado el p-valor obtenido, el test no mostró evidencia suficiente a favor de la habilidad de la dama. Sin embargo, vemos que de 4 veces que se le dio primero la leche, acertó 3, y lo mismo para el té.

Otro ejemplo, con los datos de color de pelo y ojos que hemos utilizado más arriba:

```
> fisher.test(table(color.ojos,color.pelo))
```

Fisher's Exact Test for Count Data

```
data: table(color.ojos, color.pelo)  
p-value = 0.1319  
alternative hypothesis: true odds ratio is not equal to 1
```

95 percent confidence interval:

0.007811653 1.922966759

sample estimates:

odds ratio

0.1563219

En este caso, tampoco somos concluyentes, como no lo éramos al utilizar la función `chisq.test`. A pesar de todo, para el tamaño de la muestra, que es pequeño, este test es más fiable.

### **ks.test**

Utilizaremos este contraste para comprobar si dos conjuntos de datos siguen la misma distribución, o bien para ver si un determinado conjunto de datos se ajusta a una distribución determinada. Las muestras no necesitan ser del mismo tamaño.

Por ejemplo, vamos a generar 50 datos de una distribución normal, y 30 de una uniforme, a ver si el test es capaz de advertirnos que las distribuciones de origen no son las mismas:

```
> x = rnorm(50)
```

```
> y = runif(30)
```

```
> ks.test(x, y)
```

Two-sample Kolmogorov-Smirnov test

data: x and y

D = 0.52, p-value = 2.903e-05

alternative hypothesis: two.sided

Como el p-valor es prácticamente cero, rechazaremos la hipótesis de que los datos vienen de la misma distribución.

### **shapiro.test**

Es como `ks.test`, pero está especializado en la distribución normal. Por ejemplo, con los mismos datos `x` generados en el ejemplo anterior, comprobemos si el test detecta que provienen de una distribución normal:

```
> shapiro.test(x)
```

```
Shapiro-Wilk normality test
```

```
data: x
```

```
W = 0.9758, p-value = 0.3905
```

En efecto, el p-valor es grande (0.39), y no rechazaremos la hipótesis nula de que los datos `x` provienen de una normal.

### **oneway.test**

Este es el contraste de igualdad de medias que se estudia en los modelos ANOVA. Se supone que hay varios grupos de datos, que cada grupo sigue una distribución normal (no necesariamente con la misma varianza), y se trata de ver si las medias son todas iguales o no. Este test aparece mucho en procesos de calidad, de diseño de experimentos, en los que se fabrica un producto de varias maneras distintas, y se trata de ver si los procedimientos conducen al mismo resultado o no.

Para poner un ejemplo, consideraremos los datos de fabricación de camisas de trabajo sacados del libro de Diseño y Analisis de Experimentos de Montgomery, del grupo Editorial Iberoamericana. Se trata de fabricar camisas de trabajo que tengan la mayor resistencia posible a la rotura. Para ellos se fabrican 5 tipos de camisas, donde en cada grupo se da una proporción determinada de algodón en la composición de la camisa. Las camisas del grupo 1 tienen un 15% de algodón, las del grupo 2 un 20%, las del grupo 3 un 25%, las del grupo 4 un 30%, y las del grupo 5 un 35% de algodón. Se fabrican 5 camisas de cada tipo, se rompen y se mide su resistencia. Se trata de determinar si la resistencia media de las camisas de cada grupo es la misma o no. Vamos allá:

```
> algodón = scan() # introducimos los datos por teclado
```

```
1: 7
```

```
2: 7
```

```
3: 15
```

4: 11

5: 9

6: 12

7: 17

8: 12

9: 18

10: 18

11: 14

12: 18

13: 18

14: 19

15: 19

16: 19

17: 25

18: 22

19: 19

20: 23

21: 7

22: 10

23: 11

24: 15

25: 11

26:

Read 25 items

```
> porcentaje.algodon = c(15,15,15,15,15,20,20,20,20,20,25,25,25,25,25,30,30,  
30,30,30,35,35,35,35,35)
```

```
> datos.algodon = cbind(algodon,porcentaje.algodon)
```

```
> datos.algodon
```

```
algodon porcentaje.algodon
```

```
[1,] 7 15
```

```
[2,] 7 15
```

[3,]	15	15
[4,]	11	15
[5,]	9	15
[6,]	12	20
[7,]	17	20
[8,]	12	20
[9,]	18	20
[10,]	18	20
[11,]	14	25
[12,]	18	25
[13,]	18	25
[14,]	19	25
[15,]	19	25
[16,]	19	30
[17,]	25	30
[18,]	22	30
[19,]	19	30
[20,]	23	30
[21,]	7	35
[22,]	10	35
[23,]	11	35
[24,]	15	35
[25,]	11	35

>

Hasta aquí tenemos los datos de resistencia de las 25 camisas, junto con la proporción de algodón que hay en cada una.

Ahora aplicaremos el test de igualdad de medias. En la función se especifica como parámetro `algodon~porcentaje.algodon` lo que significa que estudiaremos la variable `algodon` (su resistencia), dividiéndola en grupos según el porcentaje de algodón presente.

```
> oneway.test(algodon~porcentaje.algodon,data=datos.algodon)
```

## One-way analysis of means (not assuming equal variances)

data: algodón and porcentaje.algodón

F = 12.4507, num df = 4.000, denom df = 9.916, p-value = 0.0006987

Como vemos, el p-valor es prácticamente cero. Se rechaza, pues, la hipótesis de que todos los grupos son similares. Esto es, variar el porcentaje de algodón tiene consecuencias en la resistencia de la camisa. De hecho, podemos hacer un diagrama de caja (un boxplot) por grupos, para ver cómo evoluciona la resistencia en función del porcentaje de algodón:

```
> boxplot(algodón~porcentaje.algodón)
```

### **var.test**

Este es un contraste para comprobar la igualdad de varianzas entre dos grupos de datos. Nos puede ser útil en multitud de situaciones, por ejemplo, cuando tenemos dos máquinas que fabrican lo mismo (por ejemplo, rodamientos), y queremos saber si lo hacen con la misma variabilidad. Para poner un ejemplo, generemos datos de dos distribuciones normales con distinta varianza y veamos si el test detecta esta situación:

```
> x = rnorm(50, mean = 0, sd = 2)
```

```
> y = rnorm(30, mean = 1, sd = 1)
```

```
> var.test(x, y)
```

F test to compare two variances

data: x and y

F = 5.3488, num df = 49, denom df = 29, p-value = 7.546e-06

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

2.687373 10.063368

sample estimates:

ratio of variances

5.348825

Como vemos, el p-valor sale prácticamente cero: rechazaremos la hipótesis de que los conjuntos de datos  $x$  e  $y$  provienen de distribuciones con la misma varianza.