MEMORIA PRÁCTICA – 3:

Asociación estadística entre variables categóricas.

Escuela Politécnica Superior - Universidad Autónoma de Madrid

GRADO EN INGENIERÍA BIOMÉDICA

BIOESTADÍSITICA

Versión del documento número 1

Práctica realizada por: *Laura Sánchez Garzón*

Fecha: *24 de marzo de 2023*

Profesorado de la práctica: *Mercedes Sotos Prieto y María Téllez Plaza*

1. ÍNDICE

Contenido

2. INTRODUCCIÓN

Esta tercera práctica de bioestadística tiene por objetivo trabajar con los conceptos teóricos vistos en clase, para asimilarlos y entender cómo funcionan las probabilidades, distribuciones e intervalos de confianza.

RStudio dispone de todo tipo de funciones y gráficos que nos permiten trabajar con dichos conceptos, y en pocas líneas de código se puede extraer de manera inequívoca, sólidas conclusiones. Además, utilizar R resulta mucho más práctico que realizar los ejercicios a mano, dado que los cálculos son rápidos, y no se debe recurrir a las tablas de los valores críticos de la chi-cuadrado (la propia máquina calcula resultados).

3. EJERCICIOS GUIADOS

3.1. Inferencia estadística con variables categóricas

3.1.1. Contraste de una proporción.

La distribución acumulada binomial, se puede aproximar bien con tamaños experimentales grandes mediante una distribución normal con media np y varianza np(1 − p). Se asume que esta aproximación es satisfactoria cuando el número esperado de "exitos" y "fracasos" son ambos mayores de 5. Si denotamos el número observado de "exitos" como x, la hipótesis nula p = p_0 puede contrastarse con el estadístico:

$$u = \frac{x - np_0}{\sqrt{np_0(1 - p_0)}}$$

que distribuye con una distribución normal tipificada (*pnorm()* en R). Nótese que alternativamente, también se puede hacer el contraste con u^2, que se distribuye con una distribución χ^2 con 1 grado de libertad. Siguiendo un procedimiento parecido al que se vio en la Práctica 2, se podría contrastar dicha hipótesis nula utilizando *pbinom()*, *pnorm()*, o incluso, *pchisq()*. Alternativamente, en el Código 1, se enseñan ejemplos de otras maneras en las que se podría evaluar la hipótesis nula sobre igualdad de proporciones utilizando los comandos *prop.test()* y *binom.test()*

3.1.1.1. Código 1 - Funciones para el contraste de una proporción en R

Consideremos un ejemplo (Altman, 1991, p. 230) donde se observa que 39 de 215 pacientes elegidos aleatoriamente tienen asma y se quiere probar la **hipótesis nula** de que la **probabilidad** de que un "paciente cualquiera" de la población que ha originado los datos tenga **asma es 0.15**.

Si queremos utilizar la **aproximación a la distribución normal/ji cuadrado**, esto se puede hacer usando *prop.test()*:

```
prop.test(39,215,.15)
```

```
> prop.test(39,215,.15)

        1-sample proportions test with continuity correction

data:  39 out of 215, null probability 0.15
X-squared = 1.425, df = 1, p-value = 0.2326
alternative hypothesis: true p is not equal to 0.15
95 percent confidence interval:
 0.1335937 0.2408799
sample estimates:
        p
0.1813953
```

Figura 1: Función que calcula la aproximación a la distribución norma/ji cuadrado: prop.test(número de observaciones (x) población total (N), parámetro de probabilidad (teórico) que se quiere probar).

Como salida se obtiene la distribución χ^2 con 1 grado de libertad, y un p-valor de 0.2326.

Se parte de una hipótesis nula (H_0), en la que se cree que los valores ingresados son dependientes. Dado que el p-valor (0.2326) es mayor a la probabilidad, p (0.15), se opta por coger la hipótesis alternativa (H_1), que mantiene que los valores son independientes (menor a 0.05 da un nivel de confianza de H_0 del 95% o mayor). Además, la propia función calcula el rango (entre 0.1335937 y 0.2408799*), con un 95% en el que se puede mover el p-valor, y aún se aceptaría que los valores son independientes, cuya media es 0.1813953.*

En realidad, la cantidad 15 % es un poco artificial, ya que rara vez se da el caso de que uno sepa a priori el valor específico en la población. Por lo general, es más interesante calcular un intervalo de confianza para el parámetro de probabilidad, como se ve en la última parte de la salida (Figura 1).

Si quisiéramos obtener una probabilidad exacta (no basada en aproximación a la normal) podríamos usar *pbinom()* como vimos en la práctica anterior. El procedimiento sería calcular las probabilidades puntuales para todos los valores posibles de x y sumar los que son menores que o iguales a la probabilidad puntual de la x observada.

Alternativamente, existe en R el comando *binom.test()* que devuelve no sólo el p-valor exacto, sino también intervalo de confianza:

```
binom.test(39,215,.15)
```

```
> binom.test(39,215,.15)

        Exact binomial test

data:  39 and 215
number of successes = 39, number of trials = 215, p-value = 0.2135
alternative hypothesis: true probability of success is not equal to 0.15
95 percent confidence interval:
 0.1322842 0.2395223
sample estimates:
probability of success
             0.1813953
```

Figura 2: Función binom.test(): no calcula aproximando a la normal, sino, de forma exacta, la suma de las probabilidades de los valores por debajo o iguales a la probabilidad puntual de la x observada. Aún así se obtiene que los parámetros son independientes.

Sin embargo prop.test() puede hacer más que sólo contrastar proporciones individuales. Por ejemplo, se puede también utilizar para contrastar 2 o más proporciones, como veremos a continuación.

3.1.2. Contrastes de 2 o más proporciones

La función *prop.test()* también se puede utilizar para comparar dos o más proporciones. Para ese propósito, los argumentos deben ser dados como dos vectores, donde el primero contiene el número de éxitos y el segundo el número total de intentos en cada grupo. El procedimiento es similar a la de una sola proporción. Considere la diferencia en las dos proporciones d = (x1/n1)−(x2/n2), que se distribuirá aproximadamente con un normal tipificada de media cero y varianza Vp(d) = (1/n1 + 1/n2) ∗ p(1−p) si los conteos se distribuyen binomialmente con el mismo parámetro p.

Entonces, para contrastar la hipótesis nula de que p1 = p2, se reemplaza el estimador común pˆ = (x1+x2)/(n1+n2) en la fórmula de la varianza y el estadístico de contraste sería $u = d/\sqrt{Vp\hat{}(d)}$, que sigue aproximadamente una distribución normal tipificada, o alternativamente, u^2 , que tiene una distribución aproximada de χ^2 con 1 grado de libertad. El Código 2 muestra ejemplos sobre cómo contrastar múltiples proporciones utilizando *prop.test()* y *fisher.test()*:

3.1.2.1. Código 2 - Funciones para el contraste de múltiples proporciones en R

Se investiga si el ambiente tabáquico en el domicilio es un factor de riesgo de ingreso hospitalario en los pacientes asmáticos. Para ello hemos seleccionado 39 asmáticos, 16 expuestos al tabaco actualmente (16/39 = 41,0%), 12 que estuvieron expuestos al tabaco pero ya no (12/39=30.8%) y 11 no expuestos (11/39 = 28,2%).

Supongamos que obtenemos los siguientes resultados: de los 16 expuestos ingresan en el hospital 14 (87,5%), de los 12 ex-expuestos ingresan 6 (50,0 %) y de los 11 no expuestos ingresan 3 (27,3%).

Se trata de saber si la proporción de ingresos hospitalario es distinta entre los expuestos, ex-expuestos y no expuestos al tabaco.

```
asma.ingresos <- c(14,6,3)

asma.total <- c(16,12,11)

prop.test(asma.ingresos,asma.total)
```

```
> asma.ingresos <- c(14,6,3)
> asma.total <- c(16,12,11)
> prop.test(asma.ingresos,asma.total)

        3-sample test for equality of proportions without continuity correction

data:  asma.ingresos out of asma.total
X-squared = 10.35, df = 2, p-value = 0.005657
alternative hypothesis: two.sided
sample estimates:
   prop 1    prop 2    prop 3
0.8750000 0.5000000 0.2727273

Warning message:
In prop.test(asma.ingresos, asma.total) :
  Chi-squared approximation may be incorrect
```

Figura 3: Función prop.test(): Calcula la aproximación de la normal.

asma.ingresos() recoge un vector de aquellos ingresados (de los 16 expuestos ingresan en el hospital 14, de los 12 ex-expuestos ingresan 6 y de los 11 no expuestos ingresan 3.

asma.total() recoge un vector con los totales de enfermos de cada clase (de 39 asmáticos, 16 expuestos al tabaco actualmente, 12 que estuvieron expuestos al tabaco, pero ya no y 11 no expuestos).

prop.test() comparará 3 proporciones. La χ^2=10.35 y cuenta con 2 grados se libertad (se comparan 3 proporciones), y un p-valor de 0.005657 < 0.05 (diferencias serían significativas a un nivel de confianza del 99%).

En este ejemplo R nos ofrece un aviso de atención (Figura 3). Ésto ocurre porque **el *prop.test* es una aproximación**, y que sabemos que falla cuando el número de éxitos es menor que 5 en alguna de las celdas (prop 3 = 0.2727273).

En estos casos, **hace falta un test exacto**. Para obtener test exactos en test de proporciones múltiples se utiliza el *fisher.test*, que requiere que los datos de entrada sean una matriz:

```
matrix(c(14,6,3,2,6, 8),3)

asma.ingresos <- matrix(c(14,6,3,2,6, 8),3)

fisher.test(asma.ingresos)
```

```
> matrix(c(14,6,3,2,6, 8),3)
     [,1] [,2]
[1,]   14    2
[2,]    6    6
[3,]    3    8
> asma.ingresos <- matrix(c(14,6,3,2,6, 8),3)
> fisher.test(asma.ingresos)

        Fisher's Exact Test for Count Data

data:  asma.ingresos
p-value = 0.004765
alternative hypothesis: two.sided
```

Figura 4: Función fisher.test(), utilizada para obtener test exactos en test de proporciones múltiples. Nótese en la matriz que la segunda columna de la tabla tiene que ser el número de fallos, no el número total de ensayos.

Se obtiene un p-valor de 0.004765, y dado que es menor a 0.05, se rechaza la hipótesis nula y se acepta la alternativa (diferencias serían significativas a un nivel de confianza del 99%).

3.1.3. Generalización del estudio de variables categóricas a tablas de dimensión *k p*

En las distribuciones bidimensionales se clasifican las unidades estadísticas de una muestra según las modalidades de dos caracteres X e Y . Con n_{ij} se representa **el número de unidades de la muestra** que presenta simultáneamente la modalidad i-ésima de X y la modalidad j-ésima de Y, i = 1, 2, . . . , k, y j = 1, 2, . . . , p.

Se indica por f_{ij} la correspondiente **frecuencia relativa de cada celda con respecto al total de individuos, n** (es decir n_{ij}/n). La representación tabular se realiza mediante las tablas de contingencia:

	Y_1	Y_2	...	Y_j	...	Y_p	T
X_1	n_{11}	n_{12}	...	n_{1j}	...	n_{1p}	$n_{1\bullet}$
X_2	n_{21}	n_{22}	...	n_{2j}	...	n_{2p}	$n_{2\bullet}$
...							
X_i	n_{i1}	n_{i2}	...	n_{ij}	...	n_{ip}	$n_{i\bullet}$
...							
X_k	n_{k1}	n_{k2}	...	n_{kj}	...	n_{kp}	$n_{k\bullet}$
T	$n_{\bullet 1}$	$n_{\bullet 2}$	...	$n_{\bullet j}$	...	$n_{\bullet p}$	n

Figura 5: Tabla de contingencia.

Donde se ha indicado:

$$n_{i\bullet} = \sum_{j=1}^{p} n_{ij} \qquad n_{\bullet j} = \sum_{i=1}^{k} n_{ij}$$

Figura 6: Cálculo de n_i y n_j.

Mod	Fre
X_1	$n_{1\bullet}$
X_2	$n_{2\bullet}$
...	...
X_i	$n_{i\bullet}$
...	...
X_k	$n_{k\bullet}$
T	n

Mod	Fre
Y_1	$n_{\bullet 1}$
Y_2	$n_{\bullet 2}$
...	...
Y_i	$n_{\bullet j}$
...	...
Y_p	$n_{\bullet p}$
T	n

Figura 7: Las distribuciones marginales corresponden a las distribuciones univariantes.

Mod	Fre
X_1	n_{1j}
X_2	n_{2j}
...	...
X_i	n_{ij}
...	...
X_k	n_{kj}
T	$n_{\bullet j}$

Mod	Fre
Y_1	n_{i1}
Y_2	n_{i2}
...	...
Y_i	n_{ij}
...	...
Y_p	n_{ip}
T	$n_{i\bullet}$

Figura 8: Distribuciones condicionadas.

Las relaciones que pueden darse entre dos variables aleatorias pueden fluctuar entre los extremos que se ilustran seguidamente. Para obtener una intuición sobre cómo funciona el test de χ^2 es fundamental el **concepto de independencia entre filas y columnas** de una tabla de contingencia:

	Y_1	Y_2	Y_3	Y_4	T
X_1	3	5	2	4	14
X_2	6	10	4	8	28
X_3	12	20	8	16	56
T	21	35	14	28	98

Figura 9: Situación de independencia, con filas y columnas proporcionales.

($X_1Y_1 \cdot 2 = X_2Y_1$, $X_2Y_1 \cdot 2 = X_3Y_1$; $X_1Y_2 \cdot 2 = X_2Y_2$, $X_2Y_2 \cdot 2 = X_3Y_2$…)

	Y_1	Y_2	Y_3	T
X_1	3	0	0	3
X_2	0	0	2	2
X_3	0	4	0	4
T	3	4	2	9

Figura 10: Situación de dependencia funcional.

En este caso la dependencia funcional es completamente recíproca porque para cualquier valor de Y automáticamente conocemos el valor de X). La dependencia en este caso es completa.

En general, las tablas de contingencia pueden surgir de varios esquemas de muestreo diferentes, y la noción de "**ninguna relación entre filas y columnas**" puede interpretarse diferentemente de forma correspondiente, como se ilustra a continuación.

Por ejemplo, en un **contraste de independencia** se toma una muestra transversal de la población, es decir, **se selecciona al azar una cierta cantidad de individuos de la población, se observan las dos variables sobre cada uno de ellos, y se contrasta si las probabilidades conjuntas son iguales al producto de las probabilidades marginales de cada variable**. Formalmente, si X e Y son las dos variables, se contrasta si para cada par de posibles valores *x* de X e *y* de Y se tiene que P(X = *x*, Y = *y*) = P(X = *x*)P(Y = *y*) o si por el contrario hay algún par de valores *x, y* para los que esta igualdad sea falsa.

Alternativamente, **el total de cada fila podría fijarse por adelantado y se podría estar interesado en probar si la distribución sobre las columnas es la misma para cada fila, o viceversa** si los totales de las columnas fueran fijos.

En un **contraste de homogeneidad** se escoge una de las variables y para cada uno de sus posibles valores se toma una muestra aleatoria, de tamaño prefijado, de individuos con ese valor para esa variable. A continuación, se observa sobre cada uno de estos individuos la otra variable. En esta situación contrastamos si la distribución de probabilidades de la segunda variable es la misma en los diferentes estratos definidos por los niveles de la primera variable.

Formalmente, si **Y** es la variable que usamos en primer lugar para **clasificar los individuos de la población y tomar una muestra de cada clase**, con posibles valores $y_1,\ldots,y_p$, y **X** es la variable que **medimos** en segundo lugar sobre los **individuos escogidos**, se contrasta si, para cada posible valor *x* de X, $P(X = x|Y = y_1) = P(X = x|Y = y_2) = \ldots = P(X = x|Y = y_p)$ o si por el contrario existen x, y_i, y_j tales que $P(X = x|Y = y_i) \neq P(X = x|Y = y_j)$.

Sin embargo, en estos dos supuestos, la tabla de contingencia resultante viene a ser la misma, y el análisis de dicha tabla resulta ser también ser el mismo, ya que, en esencia, lo que se está **comprobando** en ambos casos es la presencia de **proporcionalidad entre filas y columnas**.

Si no hay relación entre las variables X e Y (es decir, si se cumple la proporcionalidad entre filas y columnas), entonces los valores esperados en las celdas serían:

$$E_{ij} = \frac{n_{i\bullet} \times n_{\bullet j}}{n}$$

Figura 11: Se cumple la proporcionalidad entre filas y columnas.

Se calcula esta cantidad para cada una de las celdas, y el estadístico de contraste resultante de la prueba:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Figura 12: Estadístico de contraste.

Tiene una distribución aproximada de χ^2 con $(k-1) \times (p-1)$ grados de libertad. Aquí la suma es sobre toda la tabla y los índices *ij* han sido omitidos. O denota los valores observados y E los valores esperados para cada celda como descrito arriba. Nótese que la raíz de la distribución χ^2 con un grado de libertad que se utiliza para tablas 2x2 y para contraste de una proporción es equivalente a una distribución normal tipificada.

El Código 3 muestra varios tipos de problemas que se pueden resolver a partir de tablas de contingencia. **La hipótesis nula en todos los casos se rechaza si se obtiene evidencia de que los valores esperados asumiendo que las filas y columnas son proporcionales se desvían mucho de los observados**. Para el análisis de tablas de contingencia que tienen más de 2 categorías en los dos lados, se puede usar además de la aproximación *chisq.test()*, el *fisher.test()* (test exacto), aunque éste último es computacionalmente más intenso si los conteos totales de las celdas son grandes.

3.1.3.1. Código 3 - Tipos de contraste usando tablas de contingencia de dimensión k x p

```
chisq.test(asma.ingresos)

caff.marital <- matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67),nrow=3,byrow=T)

matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67),nrow=3,byrow=T)

colnames(caff.marital) <- c("0","1-150","151-300",">300")

rownames(caff.marital) <- c("Married","Prev.married","Single")

caff.marital

chisq.test(caff.marital)
```

```
Warning message:
In chisq.test(asma.ingresos) : Chi-squared approximation may be incorrect
> caff.marital <- matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67),nrow=3,byrow=T)
> matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67),nrow=3,byrow=T)
     [,1] [,2] [,3] [,4]
[1,]  652 1537  598  242
[2,]   36   46   38   21
[3,]  218  327  106   67
> colnames(caff.marital) <- c("0","1-150","151-300",">300")
> rownames(caff.marital) <- c("Married","Prev.married","Single")
> caff.marital
              0 1-150 151-300 >300
Married     652  1537     598  242
Prev.married 36    46      38   21
Single      218   327     106   67
> chisq.test(caff.marital)

        Pearson's Chi-squared test

data:  caff.marital
X-squared = 51.656, df = 6, p-value = 2.187e-09
```

Figura 13: Salida del código 3.

Se crea una tabla 3x3 (compara el consumo de cafeína y estado civil), a la que se aplica la función chiq.test(), que devuelve el cálculo de la Ji cuadrado directamente (trabaja con los datos en forma de matriz, de manera parecida a fisher.test).

El resultado obtenido es un estadístico de contraste, $\chi 2=51.656$, 6 grados de libertad y un p-valor ínfimo (2.187e-09 < 0.001, por lo que las diferencias serían significativas a un nivel de confianza del 99.9%, es decir, los datos contradicen la hipótesis de la independencia).

Sin embargo, generalmente gustaría saber la naturaleza de las desviaciones. Para ello, puede consultar algunos componentes adicionales del valor de la salida de chisq.test, que en realidad devuelve más información de lo que comúnmente se imprime:

chisq.test(caff.marital)$expected

chisq.test(caff.marital)$observed

```
> chisq.test(caff.marital)$expected
                     0      1-150   151-300      >300
Married      705.83179 1488.01183 578.06533 257.09105
Prev.married  32.85648   69.26698  26.90895  11.96759
Single       167.31173  352.72119 137.02572  60.94136
> chisq.test(caff.marital)$observed
                0 1-150 151-300 >300
Married       652  1537     598  242
Prev.married   36    46      38   21
Single        218   327     106   67
```

Figura 14: Salida de la Ji-cuadrado, se tienen en cuenta las cantidades esperadas (Figura 11) y observadas (O_{ij}) de las celdas.

```
> E <- chisq.test(caff.marital)$expected
> O <- chisq.test(caff.marital)$observed
> (O-E)^2/E
                      0    1-150   151-300      >300
Married       4.1055981 1.612783 0.6874502 0.8858331
Prev.married  0.3007537 7.815444 4.5713926 6.8171090
Single       15.3563704 1.875645 7.0249243 0.6023355
```

Figura 15: En E y O se guardan las cantidades esperadas y observadas, respectivamente.

La operación final sirve para calcular el estadístico de contraste. Hay algunas contribuciones importantes, particularmente de muchos "no consumidores" solteros, y la distribución entre los previamente casados. Aún así, no es fácil encontrar una descripción simple de la desviación de independencia en estos datos.

3.2. Odds ratio y riesgo relativo

A veces no sólo interesa evaluar si hay diferencias relevantes o no, si no también cuantificar dicha relación.

A partir de las tablas de contingencia 2x2 se pueden obtener medidas de que cuantifican la magnitud de la asociación, como los riesgos relativos o razón de riesgos (**RR**): **cociente entre** la probabilidad de desarrollar una condición en el grupo con el factor de **exposición**, p_e y la probabilidad de desarrollar esa condición en el grupo de referencia, que **no** tiene el factor de **exposición**, p_u; y las odds ratio o razones de odds (**OR**): cociente entre las odds de dichas probabilidades, es decir $\mathbf{p_e/(1-p_e)}$ **y** $\mathbf{p_u/(1-p_u)}$.

3.2.1. Conjunto de datos individuales procesados (conteos)

3.2.1.1. Código 4 - RR y OR (IC 95%) a partir de conteos

Si se tienen los conteos en una tabla de contingencia, como en el ejemplo que se muestra a continuación sobre la hipertensión (HT) y la enfermedad cardiovascular (CV), es decir, no el conjunto de datos individuales sin procesar, se puede recrear la tabla en R y luego calcular la razón de riesgo y sus límites de confianza del 95% usando el comando *riskratio.wald()* en *epitools*.

	No CV	CV	T
HT No	1017	165	1182
HT Sí	2260	992	3252
T	3277	11157	4434

Figura 16: Tabla de contingencia 2x2, que comprobará la dependencia o independencia entre dos factores: la hipertensión (HT) y la enfermedad cardiovascular (CV)

Aquí es donde la orientación de la tabla de contingencia es crítica, es decir, con el grupo no expuesto (referencia) en la primera fila y los sujetos sin ocurrencia del evento resultado en la primera columna; o alternativamente, con el grupo expuesto y los sujetos con ocurrencia del evento en la primera columna.

El Código 4 muestra como se obtendrían las razones de riesgo y de odds, y sus correspondientes intervalos de confianza.

```
install.packages("epitools")

library(epitools)

RRtable<-matrix(c(1017,2260,165,992),nrow = 2, ncol = 2)

RRtable

riskratio.wald(RRtable)

ORtable<-matrix(c(1017,2260,165,992),nrow = 2, ncol = 2)

ORtable oddsratio.wald(ORtable)
```

```
> install.packages("epitools")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and instal
l the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Laura/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/epitools_0.5-10.1.zip'
Content type 'application/zip' length 318567 bytes (311 KB)
downloaded 311 KB

package 'epitools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Laura\AppData\Local\Temp\RtmpULEUk4\downloaded_packages
> library(epitools)
> RRtable<-matrix(c(1017,2260,165,992),nrow = 2, ncol = 2)
> RRtable
     [,1] [,2]
[1,] 1017  165
[2,] 2260  992
```

Figura 17: Se ha instalado el paquete epitools, dado que el "paquete base" en R no tiene un comando para calcular intervalos de confianza para RR, OR. Cada vez que se quiera usar, se ha de cargar con library(epitools). RRtable recoge la tabla de contingencia de la Figura 16.

```
> riskratio.wald(RRtable)
$data
          Outcome
Predictor  Disease1 Disease2 Total
  Exposed1     1017      165  1182
  Exposed2     2260      992  3252
  Total        3277     1157  4434

$measure
          risk ratio with 95% C.I.
Predictor  estimate    lower    upper
  Exposed1 1.000000       NA       NA
  Exposed2 2.185217 1.879441 2.540742

$p.value
          two-sided
Predictor  midp.exact fisher.exact  chi.square
  Exposed1         NA           NA          NA
  Exposed2          0 7.357611e-31 1.35953e-28

$correction
[1] FALSE

attr(,"method")
[1] "Unconditional MLE & normal approximation (Wald) CI"
```

Figura 18: Se calcula el RR y el intervalo de confianza del 95%.

El propio programa calcula el número total de celdas, y calcula el riesgo relativo con una confianza del 95%.

Nos fijamos en $measure: el rango es > 1, por lo que se trata de una relación directa y significativa.

```
> ORtable<-matrix(c(1017,2260,165,992),nrow = 2, ncol = 2)
> ORtable
     [,1] [,2]
[1,] 1017  165
[2,] 2260  992
> oddsratio.wald(ORtable)
$data
          Outcome
Predictor  Disease1 Disease2 Total
  Exposed1     1017      165  1182
  Exposed2     2260      992  3252
  Total        3277     1157  4434

$measure
          odds ratio with 95% C.I.
Predictor  estimate    lower    upper
  Exposed1 1.000000       NA       NA
  Exposed2 2.705455 2.258339 3.241093

$p.value
          two-sided
Predictor  midp.exact fisher.exact  chi.square
  Exposed1         NA           NA          NA
  Exposed2          0 7.357611e-31 1.35953e-28

$correction
[1] FALSE

attr(,"method")
[1] "Unconditional MLE & normal approximation (Wald) CI"
```

Figura 19: Se calcula el OR para realizar estudios de casos y controles. En este caso el procedimiento es muy similar al análisis anterior para RR (Dado que el valor estimado es > 1, se considera que se trata de un factor de riesgo).

3.2.2. Conjunto de datos individuales no procesados

Si se tiene un conjunto de datos sin procesar (base de datos individuales pero no recuentos), calcular las razones de riesgo y las razones de probabilidad y sus correspondientes intervalos de confianza del 95 % es aún más fácil, porque la tabla de contingencia se puede crear usando el comando table() en lugar de la función de matriz.

3.2.2.1. Código 5 - RR y OR (IC 95%) a partir de base de datos sin procesar

Por ejemplo, si tengo datos de un estudio, en el caso del siguiente ejemplo, la encuesta poblacional de EE.UU. National Health and Nutrition Examination Survey (conocida como NHANES) y quiero calcular la razón de riesgo para la asociación entre la diabetes tipo 2 y morir por una enfermedad del corazón, primero uso el comando *table()*.

El Código 5 muestra en primer lugar como se pueden leer datos externos con R, y en segundo lugar como se obtendrían las razones de riesgo y de odds, y sus correspondientes intervalos de confianza a partir de datos sin procesar. **Nótese que, dado que en este caso se trata de un diseño prospectivo con seguimiento, tiene sentido calcular la razón de riesgos, pero también existe la opción de calcular una razón de odds**, mientras que en un estudio de casos y controles sólo se puede calcular una razón de odds. Observe también que, en el ejemplo anterior, se verifica el hecho conocido de que para eventos que no son raros, la razón de odds es un poco más grande en magnitud que la razón de riesgos (para la misma tabla de contigencia).

```
setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-3")

getwd()

data <-read.csv("Mortality_NHANES8894_NonSmokers.csv")

names(data)

dim(data)

data <- data[complete.cases(data),]

dim(data)

attach(data)

table(diab,heart.8yr)

riskratio.wald(table(diab,heart.8yr))

oddsratio.wald(table(diab,heart.8yr))
```

```
> setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-3")
> data <-read.csv("Mortality_NHANES8894_NonSmokers.csv")
> names(data)
 [1] "X"           "seqn"          "race"          "riagendr"      "ridageyr"      "smoking"
 [7] "bmxbmi"      "hbp"           "highchol"      "diab"          "ckd"           "gfr.epi"
[13] "sedent"      "prev.cvd"      "prev.cancer"   "peryr.exm.8yr" "peryr.age.8yr" "mortstat.8yr"
[19] "cancer.8yr"  "heart.8yr"
```

Figura 20: Se lee el fichero ".csv" que viene de "comma separated values", y es un fichero de texto plano donde # los campos están separados por comas.

Se guarda el fichero en el mismo directorio en el que se guardó el csv. a leer.

names(data) devuelve los nombres contenidos en el archivo csv.

```
> dim(data)
[1] 6195   20
> data <- data[complete.cases(data),]
> dim(data)
[1] 5929   20
> attach(data)
```

Figura 21: Ya tenemos los datos leídos, ahora vamos a estudiar la relación entre tener diabetes y morir por enfermedad del corazón en el contexto del análisis de tablas de contigencia.

Cuando veamos modelos de regresión aprenderemos que es buena práctica eliminar individuos que tienen valores perdidos (o, si son muchos, existen métodos de imputación de valores perdidos).

```
> table(diab,heart.8yr)
    heart.8yr
diab    0    1
   0 4543  239
   1 1025  122
> riskratio.wald(table(diab,heart.8yr))
$data
       heart.8yr
diab       0   1 Total
  0     4543 239  4782
  1     1025 122  1147
  Total 5568 361  5929

$measure
    risk ratio with 95% C.I.
diab estimate    lower    upper
   0 1.000000       NA       NA
   1 2.128179 1.727924 2.621148

$p.value
    two-sided
diab   midp.exact fisher.exact   chi.square
   0           NA           NA           NA
   1 2.080092e-11 2.121757e-11 7.391053e-13

$correction
[1] FALSE

attr(,"method")
[1] "Unconditional MLE & normal approximation (Wald) CI"
```

Figura 22: Se crea la tabla de contigencia (table()).

Mediante el comando riskratio.wald(), el programa calcula la relación de riesgo y los límites de confianza.

```
> oddsratio.wald(table(diab,heart.8yr))
$data
       heart.8yr
diab       0   1 Total
  0     4543 239  4782
  1     1025 122  1147
  Total 5568 361  5929

$measure
    odds ratio with 95% C.I.
diab estimate    lower    upper
   0 1.000000       NA       NA
   1 2.262459 1.800529 2.842899

$p.value
    two-sided
diab   midp.exact fisher.exact   chi.square
   0           NA           NA           NA
   1 2.080092e-11 2.121757e-11 7.391053e-13

$correction
[1] FALSE

attr(,"method")
[1] "Unconditional MLE & normal approximation (Wald) CI"
```

Figura 23: Usando los mismos datos, puedo calcular de manera similar una razón de odds y su intervalo de confianza usando la función oddsratio.wald().

3.2.3. Aproximación al *bootstrap* para obtener intervalos de confianza del riesgo relativo

Se pueden realizar simulaciones para aproximar los resultados a un problema. **El bootstrap es el re-muestreo con reemplazamiento** simulado tomando como partida una muestra real.

Siguiendo el Código 6 utilizaremos *rbinom()* para simular (muchas veces) las tasas de enfermedad en expuestos y poblaciones no expuestas. Luego dividiremos los resultados por el número de simulaciones y utilizaremos la media y el percentil que deja el 2.5% de los valores en los extremos de las colas de la distribución para la estimación puntual y los límites de confianza. Recordemos que la Razón de Riesgo (RR):

$$RR = \frac{R_e}{R_u}$$

donde Re= proporción de casos (R de “riesgo”) en el grupo expuesto (índice e) y Ru= proporción de casos en el grupo no expuesto (o referencia, índice u del inglés “Unexposed”).

3.2.3.1. Código 6 - Simulación de Riesgos Relativos con bootstrap

Hennekens, en 1987 estudia los beneficios protectores de la aspirina, observando 104 infartos de miocardio entre 11.037 personas en el grupo de tratamiento y 189 casos de infarto de miocardio de entre 11.034 personas en el grupo placebo.

```
library(epitools)

asa.tab<- matrix(c(11034-189,11037-104, 189,104),2,2)

epitab(asa.tab, method="riskratio")

set.seed(151)

tx <- rbinom(5000, 11037, 104/11037)

plac <- rbinom(5000, 11034, 189/11034)

set.seed(151)

tx <- rbinom(5000, 11037, 104/11037)

plac <- rbinom(5000, 11034, 189/11034)

rr.sim <- r.tx/r.plac

mean(rr.sim)

quantile(rr.sim, c(0.025, 0.975))

sd(rr.sim)

hist(rr.sim)

hist(log(rr.sim))
```

```
> library(epitools)
> asa.tab<- matrix(c(11034-189,11037-104, 189,104),2,2)
> asa.tab
      [,1] [,2]
[1,] 10845  189
[2,] 10933  104
> epitab(asa.tab, method="riskratio")
$tab
          Outcome
Predictor  Disease1        p0 Disease2         p1 riskratio     lower     upper       p.value
  Exposed1    10845 0.9828711      189 0.01712887  1.000000        NA        NA            NA
  Exposed2    10933 0.9905771      104 0.00942285  0.550115 0.4336731 0.6978217 5.032836e-07

$measure
[1] "wald"

$conf.level
[1] 0.95

$pvalue
[1] "fisher.exact"
```

Figura 24: En asa.tab se guarda la matriz con los valores con los que se va a trabajar.

epitab devuelve las probabilidades, el rango y el p-valor, el RR y sus intervalos de confianza correspondientes con una aproximación log-normal.

```
> set.seed(151)
> tx <- rbinom(5000, 11037, 104/11037)
> tx
   [1] 104 110 107 104  98 111  92  98  91 107  97 107  99 110  92  92  94  96  96 101 121 112 112 104
  [25]  97  89 109 120 100  86 120  92 124 110 110 101 126 122  84  98 106 113 109 103 105 108 120  90
```

...

```
 [985]  96 107 118 110 112 103 120  98 110  95  94  89  98 110  93 111
 [ reached getOption("max.print") -- omitted 4000 entries ]
> plac <- rbinom(5000, 11034, 189/11034)
> plac
   [1] 192 175 180 176 185 167 186 160 170 199 196 166 189 196 178 192 189 183 168 179 197 215 165 169
  [25] 194 163 185 187 168 181 173 205 190 196 199 168 194 189 201 190 214 179 181 195 189 217 197 216
  [49] 195 183 182 194 189 174 208 195 181 194 168 201 166 174 217 187 203 183 182 176 207 196 179 183
```

...

```
[937] 167 208 196 194 183 209 191 179 164 175 202 207 204 199 170 177 184 172 194 182 155 184 214 206
[961] 198 192 189 204 199 193 162 189 187 198 180 191 200 180 176 171 201 191 193 194 185 162 188 196
[985] 177 167 181 187 203 190 165 187 177 192 205 189 187 211 205 183
[ reached getOption("max.print") -- omitted 4000 entries ]
```

Figura 25: Simulación de razones de riesgo (RR) usando rbinom() para repetir 5,000 veces un experimento donde contamos el número de infartos de miocardio (IM) en dos poblaciones el parámetro probabilidad en la población viene definida por los resultados de la estudio de Henneken.

Lo primero es fijar una semilla de aleatorizacion para que los resultados sean reproducibles set.seed(151) (genera un número psudoalealeatorio del 0 al 151).

rbinom(5000, 11037, 104/11037) genera 5000 valores aleatorios de una distribución binomial(11037, 104/11037); se gurda en tx.

En plac se guardan 5000 valores aleatorios de una distribución binomial(11037, 189/11034).

```
> r.tx<-tx/11037
> r.plac<-plac/11034
> r.tx
   [1] 0.009422850 0.009966476 0.009694663 0.009422850 0.008879224 0.010057081 0.008335598 0.008879224
   [9] 0.008244994 0.009694663 0.008788620 0.009694663 0.008969829 0.009966476 0.008335598 0.008335598
  [17] 0.008516807 0.008698016 0.008698016 0.009151037 0.010963124 0.010147685 0.010147685 0.009422850
```

...

```
 [985] 0.008698016 0.009694663 0.010691311 0.009966476 0.010147685 0.009332246 0.010872520 0.008879224
 [993] 0.009966476 0.008607411 0.008516807 0.008063785 0.008879224 0.009966476 0.008426203 0.010057081
 [ reached getOption("max.print") -- omitted 4000 entries ]
> r.plac
   [1] 0.01740076 0.01586007 0.01631321 0.01595070 0.01676636 0.01513504 0.01685699 0.01450063
   [9] 0.01540692 0.01803516 0.01776328 0.01504441 0.01712887 0.01776328 0.01613196 0.01740076
```

...

```
[985] 0.01604133 0.01513504 0.01640384 0.01694762 0.01839768 0.01721950 0.01495378 0.01694762
[993] 0.01604133 0.01740076 0.01857894 0.01712887 0.01694762 0.01912271 0.01857894 0.01658510
[ reached getOption("max.print") -- omitted 4000 entries ]
```

Figura 26: Para cada réplica, se divide el número de resultados por el número de personas en cada población (tx y plac se dividen entre 11037 y 11034 respetivamente, para obtener 5.000 estimaciones de riesgo para cada grupo (tratamiento y placebo).

```
> rr.sim <- r.tx/r.plac
> rr.sim
   [1] 0.5415194 0.6284006 0.5942829 0.5907485 0.5295857 0.6644900 0.4944892 0.6123335 0.5351486
  [10] 0.5375423 0.4947634 0.6444031 0.5236671 0.5610719 0.5167134 0.4790364 0.4972193 0.5244476
  [19] 0.5712732 0.5640924 0.6140462 0.5207886 0.6786034 0.6152173 0.4998641 0.5458639 0.5890290
```

...

```
 [982] 0.5800892 0.4839110 0.5661726 0.5422255 0.6405444 0.6517565 0.5880754 0.5515742 0.5419579
 [991] 0.7270750 0.5239217 0.6213000 0.4946572 0.4584119 0.4707715 0.5239217 0.5211853 0.4535352
[1000] 0.6063925
 [ reached getOption("max.print") -- omitted 4000 entries ]
```

Figura 27: Se calcula el riesgo relativo para cada simulación, dividiendo r.tx/r.plac.

```
> mean(rr.sim)
[1] 0.552865
> quantile(rr.sim, c(0.025, 0.975))
     2.5%     97.5%
0.4298172 0.6949400
> sd(rr.sim)
[1] 0.06748942
> hist(rr.sim)
> hist(log(rr.sim))
```

Figura 28: Se describe la distribución de riesgos relativos resultante (se calculan la media, los cuantiles 2.5 y 97.5, y la desviación típica) y se compara con los resultados de la aproximación log-normal obtenida con epitab(). La media se aproxima a lo que se calificó como riskratio; el cuantil 2.5 al valor inferior, y el 97.5 al superior.

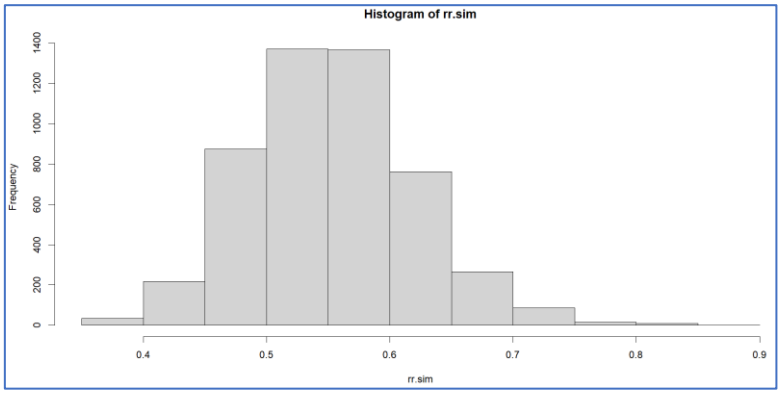

Figura 29: Histograma de rr.sim.

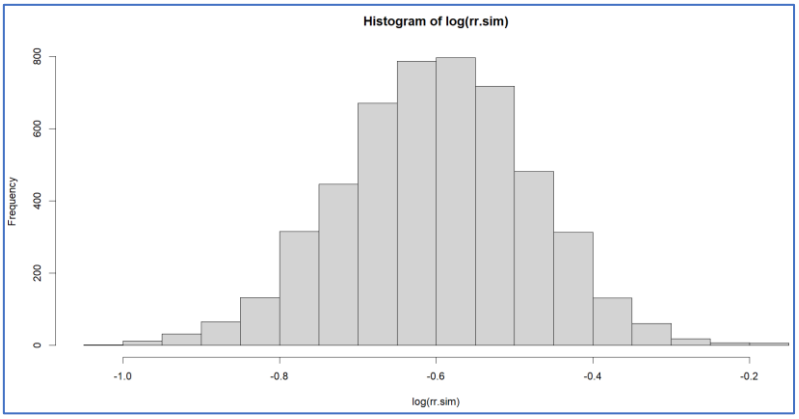

Figura 30: Histograma del logaritmo de rr.sim.

4. EJERCICIOS PROPUESTOS

4.1. Ejercicio 1

Utilice los mismos números del ejemplo anterior para simular razones de odds en lugar de riesgos relativos. Muestre su código. Compare los estimadores e intervalos de confianza obtenidos empíricamente con el bootstrap con los obtenidos a partir de la aproximación de la distribución log-normal.

#Ejercicio 1

library(epitools) #para poder utilizar el método "oddsratio"

asa.tab<- matrix(c(11034-189,11037-104, 189,104),2,2) #misma matriz utilizada para el código 6

epitab(asa.tab, method="oddsratio") #estimadores e intervalos de confianza obtenidos empíricamente con el bootstrap

set.seed(151) #se fija una semilla de aleatorización para que los resultados sean reproducibles

#rbinom(5000)--> repetir 5000 veces un experimento donde contamos el número de infartos de miocardio en 2 poblaciones

tx <- rbinom(5000, 11037, 104/11037) #104 infartos de miocardio entre 11.037 personas en el grupo de tratamiento

plac <- rbinom(5000, 11034, 189/11034) #189 casos de infarto de miocardio de entre 11.034 personas en el grupo placebo.

#odds ratio: número de individuos que tienen una característica entre el número de quienes no la tienen.

o.tx<-tx/(11037-tx) #nº de personas con IM, entre el número de personas que no lo sufrieron (en el grupo de tratamiento)

o.plac<-plac/(11034-plac) #nº de personas con IM, entre el número de personas que no lo sufrieron (en el grupo de placebo)

or.sim <- o.tx/o.plac #se calcula el ratio de odds para cada simulación

mean(or.sim) #se obtiene una media de todos los ratios de odds obtenidos

quantile(or.sim, c(0.025, 0.975)) #los cuantiles teóricamente declaran el rango en el que se mueven los valores

sd(or.sim) #se calcula la desviación típica

hist(or.sim,main = "Histograma del Odds Ratio", xlab = "Simulaciones", ylab = "Frecuencia", col = "slategray1") #se crea un histograma

hist(log(or.sim),main = "Histograma del logaritmo del Odds Ratio", xlab = "Simulaciones", ylab = "Frecuencia", col = "slategray2") #en este caso, al aplicar logaritmos nos aproximamos más a una distribución normal

```
> library(epitools) #para poder utilizaar el método "oddsratio"
> asa.tab<- matrix(c(11034-189,11037-104, 189,104),2,2) #misma matriz utilizada para el código 6
> epitab(asa.tab, method="oddsratio") #estimadores e intervalos de confianza obtenidos empíricamente
$tab
          Outcome
Predictor  Disease1        p0 Disease2        p1 oddsratio    lower    upper      p.value
  Exposed1    10845 0.4979796      189 0.6450512 1.0000000       NA       NA           NA
  Exposed2    10933 0.5020204      104 0.3549488 0.5458355 0.429041 0.694424 5.032836e-07

$measure
[1] "wald"

$conf.level
[1] 0.95

$pvalue
[1] "fisher.exact"
```

Figura 31: estimadores e intervalos de confianza obtenidos empíricamente con el bootstrap.

```
> set.seed(151) #se fija una semilla de aleatorizacion para que los resultados sean reproducibles
> #rbinom(5000)--> repetir 5000 veces un experimento donde contamos el número de infartos de miocardio en 2 poblaciones
> tx <- rbinom(5000, 11037, 104/11037) #104 infartos de miocardio entre 11.037 personas en el grupo de tratamiento
> plac <- rbinom(5000, 11034, 189/11034) #189 casos de infato de miocardio de entre 11.034 personas en el grupo placebo.
> #odds ratio: número de individuos que tienen una característica entre el número de quienes no la tienen.
> o.tx<-tx/(11037-tx) #nº de personas con IM, entre el número de personas que no lo sufrieron (en el grupo de tratamiento)
> o.plac<-plac/(11034-plac) #nº de personas con IM, entre el número de personas que no lo sufrieron (en el grupo de placebo)
> or.sim <- o.tx/o.plac #se calcula el ratio de odds para cada simulación
> or.sim
   [1] 0.5371581 0.6246598 0.5903111 0.5868555 0.5253714 0.6610815 0.4902401 0.6088605 0.5312841 0.5330150 0.4902837
  [12] 0.6409220 0.5193558 0.5566533 0.5126511 0.4746574 0.4929004 0.5202749 0.5675115 0.5600666 0.6097681 0.5158759
  [23] 0.6753085 0.6115571 0.4954296 0.5421720 0.5849299 0.6375966 0.5913740 0.4708861 0.6900835 0.4440241 0.6485052
  [34] 0.5566533 0.5481098 0.5973423 0.6452582 0.6413629 0.4133319 0.5113101 0.4902973 0.6272975 0.5980769 0.5236158
```

...

```
[4962] 0.5692745 0.6036191 0.6104759 0.4754575 0.4297823 0.5776430 0.5521676 0.5843091 0.5221665 0.5128887 0.4642539
[4973] 0.5564341 0.5418854 0.5163093 0.5063649 0.5903111 0.6424177 0.4479050 0.4712705 0.5380457 0.4477517 0.5893945
[4984] 0.4953374 0.5311079 0.6153511 0.4748259 0.5436084 0.5723393 0.5973423 0.4647741 0.5913740 0.6160821 0.5487894
[4995] 0.5469770 0.5841238 0.5038782 0.4980425 0.4595298 0.5448297
```

Figura 32: se calculan por separado las muestras de la población de tratamiento y placebo, sus respetivas razones (personas que padecieron infarto de miocardio entre las que no). En or.sim se guarda los propios ratios (tratamiento entre placebo) de cada simulación.

En realidad esto se puede ver más fácilmente aplicando la fórmula de OR:

$$OR = \frac{Odds\ expuestos}{Odds\ no\ expuestos} = \frac{a/b}{c/d} = \frac{a*d}{c*b}$$

	Casos	Controles
Expuestos	*a*	*b*
No expuestos	*c*	*d*

Donde a es tx (casos expuestos), b es 11037-tx (controles expuestos), c es plac (casos no expuestos), y d es 11034-plac (controles no expuestos).

```
> mean(or.sim) #se obtiene una media de todos los ratios de odds obtenidos
[1] 0.5486656
> quantile(or.sim, c(0.025, 0.975)) #los cuantiles teóricamente declaran el rango en el que se mueven los valores
     2.5%     97.5%
0.4251793 0.6917513
> sd(or.sim) #se calcula la desviación típica
[1] 0.06780828
> hist(or.sim,main = "Histograma del Odds Ratio", xlab = "Simulaciones", ylab = "Frecuencia", col = "slategray1") #se crea un hi
stograma
> hist(log(or.sim),main = "Histograma del logaritmo del Odds Ratio", xlab = "Simulaciones", ylab = "Frecuencia", col = "slategra
y2") #en este caso, al aplicar logaritmos nos aproximamos más a una distribución normal
```

Figura 33: Se obtienen la media de las muestras simuladas, los cuantiles y la desviación típica.

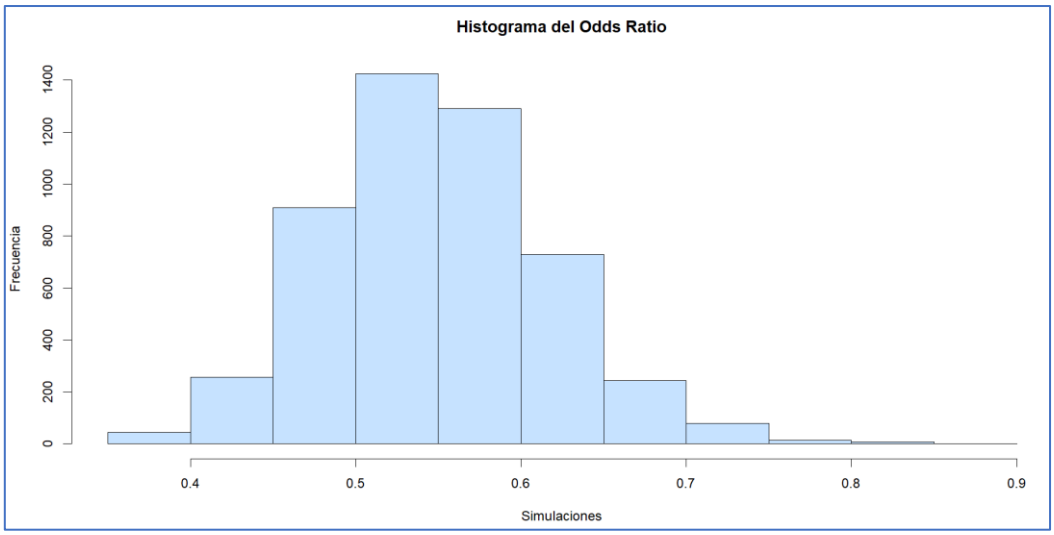

Figura 34: Histograma del Odds ratio creado con las muestras de la simulación. Observar que la forma no resulta muy similar a la de una distribución normal.

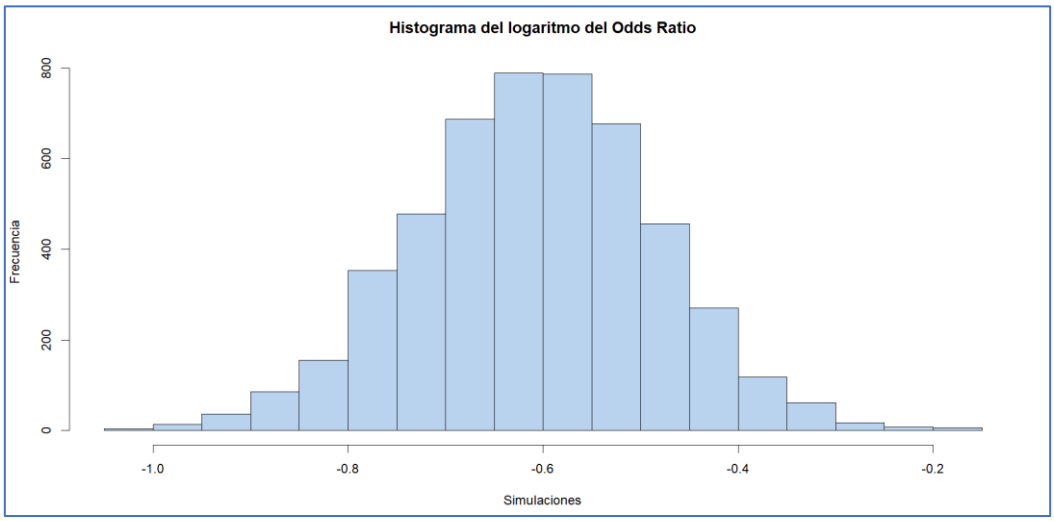

Figura 35: Histograma del logaritmo del Odds ratio creado con las muestras de la simulación. Observar que la forma es muy similar a la de una distribución normal.

Si se comparan los resultados de los estimadores e intervalos de confianza obtenidos empíricamente con el bootstrap (Figura 31) con los obtenidos a partir de la aproximación de la distribución log-normal (Figura 33), se puede observar que la aproximación es bastante correcta: La media aproximada (0.5486656) es similar al oddsratio (0.5458355). El rango real oscila entre 0.429041 y 0.694424, mientras que los cuantiles 2.5 y 97.5% resultan ser 0.4251793 0.6917513, respectivamente. Por último, la desviación típica es baja, próxima a 0, lo que explica la forma de campana del histograma.

4.2. Ejercicio 2

Los sitios CpG o sitios CG son regiones de ADN en las que un nucleótido de citosina va seguido de un nucleótido de guanina en la secuencia lineal de bases a lo largo de su dirección 5 '→ 3' . Los sitios CpG ocurren con alta frecuencia en regiones genómicas llamadas islas CpG (o islas CG). Las citosinas en los dinucleótidos CpG se pueden metilar para formar 5-metilcitosinas . Metilar la citosina dentro de un gen puede cambiar su expresión, un mecanismo que forma parte de un campo más amplio de la ciencia que estudia la regulación genética que se llama epigenética. En los seres humanos, aproximadamente el 70% de los promotores ubicados cerca del sitio de inicio de la transcripción de un gen (promotores proximales) contienen una isla CpG. Se ha simulado una muestra aleatoria de cadenas formadas por las bases "a", "c", "g" y "t". En concreto, hemos generado cadenas de longitud 100 de tres tipos en lugar de la localización genómica: A (Regiones promotoras de genes), B (Cuerpo de genes) y C (Regiones intergénicas). Estos tipos se distinguen por los vectores de probabilidades que han determinado las frecuencias de las cuatro bases en las secuencias. Queremos investigar si hay relación entre el tipo (A, B o C) de una cadena, y las bases de frecuencia máxima en ella. Ya vimos en la Práctica 1 como trabajar descriptivamente con tablas de contingencia. En este ejercicio vamos a repasar y ampliar algunas de las funciones de R para el manejo de esta clase de tablas. Realice las siguientes tareas:

4.2.1. Apartado a)

Lea la base de datos "MuestraTotalBases.csv" y genere un objeto de R que se llamen region.

#Ejercicio 2

#Apartado a

```
setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-3")
```

```
region <- read.csv("MuestraTotalBases.csv", sep=",") #se abre el fichero csv.
```

```
> #Ejercicio 2
> #Apartado a
> setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-3")
> region <- read.csv("MuestraTotalBases.csv", sep=",") #se abre el fichero csv.
```

Figura 36: Salida del apartado a del ejercicio 2.

Region devuelve las columnas leídas en el fichero csv. Importante haber leído el csv. teniendo en cuenta la separación de datos mediante comas (,).

4.2.2. Apartado b)

Explore la base de datos y compruebe que hay 1000 secuencias, y cada una de ella tiene almacenada la información tipo y max.frec. Puede ver en pantalla el cabecero de la tabla con la funcion head().

```
#Apartado b

options(max.print=999999)

region

head(region)
```

```
> region
   tipo max.frec
1     C        t
2     C        a
3     A        g
4     A        c
5     A        c
...
9997     A        c
9998     C        t
9999     B        t
10000    C        a
> head(region)
  tipo max.frec
1    C        t
2    C        a
3    A        g
4    A        c
5    A        c
6    C        t
```

Figura 37: Salida del apartado b del ejercicio 2.

Region devuelve las columnas leídas en el fichero csv.

Observar las 10.000 secuencias en la Figura 36 (options(max.print=999999) sirve para asegurar obtener todas las filas que se desean, sino el programa lee únicamente las 1.000 primeras filas).

Head(region) devuelve el cabecero de la tabla.

4.2.3. Apartado c)

Genere un objeto tabla de contingencia de frecuencias absolutas conjuntas de las dos variables del data frame region. Si se quisiera añadir los márgenes a la tabla puede utilizar addmargins(tabla). La tabla de frecuencia absolutas y márgenes correspondientes se puede obtener mediante prop.table(tabla) y addmargins(prop.table(tabla)).

#Apartado c

tabla=table(region)

addmargins(tabla)

prop.table(tabla)

addmargins(prop.table(tabla))

```
> tabla=table(region)
> addmargins(tabla)
     max.frec
tipo      a     c     g     t   Sum
  A       0   774   746     0  1520
  B     724     0     0   738  1462
  C    3525     0     0  3493  7018
  Sum  4249   774   746  4231 10000
```

```
> prop.table(tabla)
    max.frec
tipo      a      c      g      t
   A 0.0000 0.0774 0.0746 0.0000
   B 0.0724 0.0000 0.0000 0.0738
   C 0.3525 0.0000 0.0000 0.3493
> addmargins(prop.table(tabla))
     max.frec
tipo       a      c      g      t    Sum
  A   0.0000 0.0774 0.0746 0.0000 0.1520
  B   0.0724 0.0000 0.0000 0.0738 0.1462
  C   0.3525 0.0000 0.0000 0.3493 0.7018
  Sum 0.4249 0.0774 0.0746 0.4231 1.0000
```

Figura 38: Salida del apartado c del ejercicio 2.

En tabla se guarda table(region), que transforma los datos de las columnas en una matriz, cuyas columnas son las bases, y sus filas, los tipo; y se rellena la matriz con las frecuencias absolutas según cuántas bases haya de cada tipo. Además el propio programa calcula las sumas totales de filas y columnas, y devuelve la población total (n=10.000).

addmargins() sirve para devolver una tabla con márgenes, de manera que quede más vistosa.

prop.table(tabla) devuelve una tabla con las frecuencias relativas, y addmargins(prop.table(tabla)) la devuelve con márgenes en cada celda

4.2.4. Apartado d)

Ahora queremos decidir si podemos aceptar que las variables tipo y max.frec son independientes o si por el contrario hay evidencia de que la distribución de las bases de máxima frecuencia depende del tipo de cadena. Vamos a extraer una muestra aleatoria simple de la población de regiones y observar los valores de las dos variables. En concreto, seleccionaremos una muestra transversal de 150 filas, al azar y con reposición, de entre las 10000 filas del data frame region. Realice un test de independencia. Interprete los resultados.

#Apartado d

set.seed(42) # fijamos la semilla de aleatorización para que sea reproducible

n=150

indices.muestra=sample(1:10000, size=n, replace=TRUE)

muestra.test.indep= region[indices.muestra,]

tabla.indep=table(muestra.test.indep$tipo, muestra.test.indep$max.frec)

test.indep=chisq.test(tabla.indep)

test.indep

```
> set.seed(42) # fijamos la semilla de aleatorización para que sea reproducible
> n=150
> indices.muestra=sample(1:10000, size=n, replace=TRUE)
> muestra.test.indep= region[indices.muestra, ]
> indices.muestra
  [1] 2369 5273 9290 1252 8826  356 7700 3954 9091 5403  932 9189 5637 4002 9052  259 5434  481 7326 8491 2454 9028 9174
 [24] 7789 5468 6341 9732 2274 2552  727  945  626 4358 6534 1396 5123 2818 9207  517 8225  945  103 5348 7146 9545 1693
 [47] 4172 5897 5611 3619   16 5445  988 7810 8274 9106 2344  149 7140 5689  100 2346 2450   91 6925 7349 5174 6995 4816
 [70] 5878 9501 6782 8619 6586 5232 4680 7453  951 8102 9729 5261 1925 4407 2859 9770 4856 1889 9467 1693 1907 8895 3310
 [93] 6670 2310 4395 7614 8946 4895 6050 2910 1248 7382 4390  607 2831 8738 2474 2236 8446 8460 4736 3482 2089 7361 8856
[116] 1122 6681  162  636 2754 9823 5664 3995 5817 4892 6774 3109 9354 2821 4174 4622 3740 3169 6627 3546 1151 3565 5675
[139] 7860 3578 4433 4027 4635 3252 9803 5521 1293 2015 1406 3196
> muestra.test.indep
     tipo max.frec
2369    A        g
5273    C        a
9290    C        t
1252    C        t
8826    C        t
```

...

```
1293       C         a
2015       C         a
1406       C         t
3196       A         g
> tabla.indep=table(muestra.test.indep$tipo, muestra.test.indep$max.frec)
> test.indep=chisq.test(tabla.indep)
Warning message:
In chisq.test(tabla.indep) : Chi-squared approximation may be incorrect
> tabla.indep

    a  c  g  t
  A 0  6 13  0
  B 9  0  0  7
  C 64 0  0 51
> test.indep

        Pearson's Chi-squared test

data:  tabla.indep
X-squared = 150, df = 6, p-value < 2.2e-16
```

Figura 39: Salida del apartado d del ejercicio 2.

indices.muestra guarda una muestra de 150 pseudovalores de los 10.000 que componía la población total. Esos valores se buscan en las muestras independientes que componían la población total.

tabla.indep devuelve la tabla creada con las frecuencias absolutas de la muestra, y test.indep realiza el test de independencia (test de la chi cuadrado) de dicha tabla.

*Se obtiene un X^2 = 150, 6 grados de libertad (nºfilas-1 * nºcolumnas-1= 2*3=6), y se obtiene un p-valor ínfimo (<2.2e-16 < 0.001, por lo que las diferencias serían significativas a un nivel de confianza del 99.9%, es decir, los datos contradicen la hipótesis de la independencia).*

La $X^2_{calculada}$ (150) es muchísimo mayor que cualquier $X^2_{crítica}$, por lo que los parámetros son seguro no independientes (por eso el p-valor es tan bajo). Se rechaza la hipótesis nula, y se acepta la alternativa (los parámetros son dependientes).

Ahora, por curiosidad, gustaría saber la naturaleza de las desviaciones:

E <- test.indep$expected

O <- test.indep$observed

test.indep$expected

test.indep$observed

(O-E)^2/E

```
> E <- test.indep$expected
> O <- test.indep$observed
> test.indep$expected

           a    c        g         t
  A  9.246667 0.76 1.646667  7.346667
  B  7.786667 0.64 1.386667  6.186667
  C 55.966667 4.60 9.966667 44.466667
> test.indep$observed

    a  c  g  t
  A 0  6 13  0
  B 9  0  0  7
  C 64 0  0 51
> (O-E)^2/E

            a          c          g         t
  A 9.2466667 36.1284211 78.2782456 7.3466667
  B 0.1890639  0.6400000  1.3866667 0.1069253
  C 1.1530872  4.6000000  9.9666667 0.9599200
```

Figura 40: En E y O se guardan las cantidades esperadas y observadas, respectivamente.

La operación (O-E)^2/E sirve para calcular el estadístico de contraste (X^2). Es interesante observar las contribuciones de cada celda al chi-cuadrado total.

4.2.5. Apartado e)

Ahora vamos a contrastar si la distribución de probabilidades de la base de mayor frecuencia es la misma para cada tipo de cadena o no, lo que vamos a hacer es tomar una muestra aleatoria de 50 cadenas de cada tipo y juntarlas en una sola muestra estratificada (fijamos la semilla de aleatoriedad set.seed(42)).

Muestre el código. ¿Como enunciaría las hipótesis nula y alternativa en este tipo de muestreo? Realice el test correspondiente. Interprete los resultados.

#Apartado e

```
set.seed(42) # fijamos la semilla de aleatorización para que sea reproducible

n2=50

tipo.A=region[region$tipo=="A",][sample(1:nrow(region[region$tipo=="A",]), size=n2),]

tipo.B=region[region$tipo=="B",][sample(1:nrow(region[region$tipo=="B",]), size=n2),]

tipo.C=region[region$tipo=="C",][sample(1:nrow(region[region$tipo=="C",]), size=n2),]

muestra.test.indep2=rbind(tipo.A,tipo.B,tipo.C)

tabla.indep2=table(muestra.test.indep2$tipo, muestra.test.indep2$max.frec)

test.indep2=chisq.test(tabla.indep2)

test.indep2
```

```
> set.seed(42) # fijamos la semilla de aleatorización para que sea reproducible
> n2=50
> tipo.A=region[region$tipo=="A",][sample(1:nrow(region[region$tipo=="A",]), size=n2),]
> tipo.B=region[region$tipo=="B",][sample(1:nrow(region[region$tipo=="B",]), size=n2),]
> tipo.C=region[region$tipo=="C",][sample(1:nrow(region[region$tipo=="C",]), size=n2),]
> tipo.A
     tipo max.frec
3880    A        c
2191    A        g
7655    A        c
```

...

```
4365    A        g
1708    A        c
2610    A        g
> tipo.B
     tipo max.frec
877     B        t
9580    B        a
7052    B        a
2639    B        a
```

...

```
419     B         t
8979    B         t
5888    B         a
6673    B         t
> tipo.C
     tipo max.frec
9292    C         a
2650    C         a
9661    C         t
607     C         a
```

...

```
6188    C         a
856     C         a
> muestra.test.indep2=rbind(tipo.A,tipo.B,tipo.C)
> muestra.test.indep2
     tipo max.frec
3880    A         c
2191    A         g
7655    A         c
7291    A         c
```

...

```
1768    C         t
3967    C         a
6188    C         a
856     C         a
```

Figura 41: Salida del apartado e del ejercicio 2.

Sobre la población total, se buscan 50 submuestras de manera pseudoaleatoria. Primero se realiza por separado, buscando en las filas de los tipos 'A', 'B' y 'C', y después se unifican mediante rbind, obteniendo entonces una muestra de 150 submuestras.

Una vez llegados a este punto, se realiza el ejercicio como en el apartado d.

```
> tabla.indep2=table(muestra.test.indep2$tipo, muestra.test.indep2$max.frec)
> test.indep2=chisq.test(tabla.indep2)
> tabla.indep2

     a  c  g  t
  A  0 24 26  0
  B 24  0  0 26
  C 26  0  0 24
> test.indep2

        Pearson's Chi-squared test

data:  tabla.indep2
X-squared = 150.24, df = 6, p-value < 2.2e-16
```

Figura 42: Se crea una tabla de contingencia con las frecuencias absolutas de cada subtipo muestral. Véase que la tabla es diferente a la del apartado d, dado que las filas se eligen aleatoriamente.

Después, se aplica el test de la chi-cuadrado, obteniendo un X^2 = 150.24, 6 grados de libertad, y un p-valor ínfimo (<2.2e-16 < 0.001, por lo que las diferencias serían significativas a un nivel de confianza del 99.9%, es decir, los datos contradicen la hipótesis de la independencia).

La $X^2_{calculada}$ (150.24) es muchísimo mayor que cualquier $X^2_{crítica}$, por lo que los parámetros son seguro no independientes (por eso el p-valor es tan bajo). Se rechaza la hipótesis nula (los parámetros son independientes), y se acepta la alternativa (los parámetros son no independientes, es decir, dependientes).

Código completo Ejercicio 2:

```
#Ejercicio 2

#Apartado a

setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P-3")

region <- read.csv("MuestraTotalBases.csv", sep=",") #se abre el fichero csv.

#Apartado b

options(max.print=999999)

region

head(region)

#Apartado c

tabla=table(region)

addmargins(tabla)

prop.table(tabla)

addmargins(prop.table(tabla))

#Apartado d

set.seed(42) # fijamos la semilla de aleatorización para que sea reproducible

n=150

indices.muestra=sample(1:10000, size=n, replace=TRUE)

muestra.test.indep= region[indices.muestra, ]

tabla.indep=table(muestra.test.indep$tipo, muestra.test.indep$max.frec)

test.indep=chisq.test(tabla.indep)

test.indep

#Apartado e

set.seed(42) # fijamos la semilla de aleatorización para que sea reproducible

n2=50

tipo.A=region[region$tipo=="A",][sample(1:nrow(region[region$tipo=="A",]), size=n2),]

tipo.B=region[region$tipo=="B",][sample(1:nrow(region[region$tipo=="B",]), size=n2),]

tipo.C=region[region$tipo=="C",][sample(1:nrow(region[region$tipo=="C",]), size=n2),]

muestra.test.indep2=rbind(tipo.A,tipo.B,tipo.C)

tabla.indep2=table(muestra.test.indep2$tipo, muestra.test.indep2$max.frec)

test.indep2=chisq.test(tabla.indep2)

test.indep2
```

5. CONCLUSIONES

Tal y como era de esperar, RStudio agiliza los cálculos referidos a probabilidad, distribuciones e intervalos de confianza, que hechos a mano pueden resultar engorrosos y puede haber errores de cálculo por aproximaciones.

En esta práctica se han repasado conceptos de la práctica pasada, pero era más importante entender la teoría, que entender cómo funcionaba R (siguiendo los ejercicios guiados, los ejercicios propuestos eran bastante parecidos, y sencillos).