

Apuntes LAB4

Inferencia

Objetivos

En este laboratorio vamos a aprender a hacer inferencias en R, así como a profundizar en algunos aspectos clave de la inferencia estadística.

El primer aspecto a tratar será la variabilidad en el muestreo. Utilizamos simulación para evaluar la diferencia entre un estimador puntual y un parámetro poblacional.

Construiremos intervalos de confianza y contrastes de hipótesis para una y dos muestras.

Utilizamos los datos `yrbss.dat` que contienen información sobre la encuesta anual del Sistema de Vigilancia del Comportamiento del Riesgo Juvenil (*Youth Risk Behavioral Surveillance System*, YRBSS), llevada a cabo por los centros de control de enfermedades americanos para medir la actividad relacionada con la salud en las escuelas de secundaria. Este conjunto de datos contiene respuestas de los 13572 participantes del 2013 para un subconjunto de variables que se detallan en la siguiente tabla.

Variable	Descripción
age	Edad en años.
gender	Sexo del individuo. Es una variable cualitativa con 2 niveles: <code>male</code> y <code>female</code> .
grade	Curso en la escuela secundaria: 9, 10, 11, 12 y <code>other</code> .
height	Altura en metros.
weight	Peso en Kg.
idealweight	Peso ideal en Kg.

Cargamos los datos en R utilizando el código,

```
workingDir <- "D:/Ingeniería Biomedica/Material/Laboratorios"
setwd(workingDir)
yrbss <- read.table(file.path(workingDir, "datos/yrbss.dat"),
header=TRUE, sep = "\t")
```

Consideramos que estos datos constituyen la población completa, en consecuencia, los parámetros poblacionales de, por ejemplo, la variable $X = \{\text{peso en kg}\}$ serán:

```
# recordad que el argumento na.rm=TRUE no considera los missing
# si el FALSE estas funciones devolverían NA
```

```
mu <- mean(yrbss$weight, na.rm=TRUE)
mu
```

```
[1] 67.9065
```

```
sigma <- sd(yrbss$weight, na.rm=TRUE)
```

```

si gma
[1] 16.89821

```

Variabilidad en el muestreo

Podemos obtener una muestra aleatoria de tamaño 10 para estimar los parámetros media y varianza poblacional. Para ello seleccionaremos aleatoriamente 10 filas del conjunto de datos yrbss y seleccionaremos los individuos correspondientes utilizando [].

```

# establecemos el tamaño de muestra
sample.size <- 10
# muestra aleatoria
set.seed(5011)
sample.rows <- sample(1:nrow(yrbss), sample.size)
yrbss.sample <- yrbss[sample.rows, ]
head(yrbss.sample)
##      age gender grade height weight
## 3495   14   male     9    1.68  49.90
## 2627   17   male    11    1.80  61.24
##  704   15 female    10    1.50  52.16
## 5648   17   male    11    1.83  84.37
## 8031   16   male    11    1.78  69.85
## 1070   17   male    11    1.68  63.50

```

Utilizamos la función nrow() para contar el número de individuos en la población: número de filas del conjunto de datos, y sample() para extraer una secuencia aleatoria sin reemplazamiento. La muestra está compuesta por 10 individuos, y las estimaciones de la media y la desviación típica poblacional son,

```

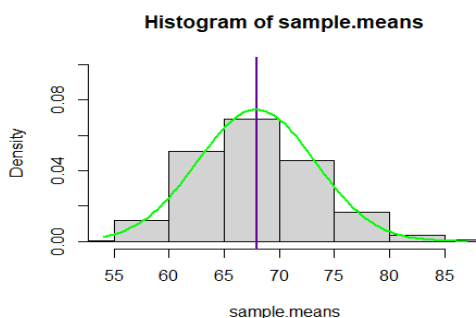
# individuos
sample.rows
## [1] 3495 2627  704 5648 8031 1070 8120 1302  425 12773
# distribución del sexo
table(yrbss.sample$gender)
##
# female  male
##      3      7
# estimación para la media poblacional
mean(yrbss.sample$weight)
## [1] 63.639
# estimación para la desviación típica poblacional
sd(yrbss.sample$weight)
## [1] 10.03137

```

Repitiendo el muestro muchas veces, obtenemos la distribución en el muestreo de la media muestral: $\bar{X}_{10} \sim N\left(\mu, \frac{\sigma}{\sqrt{10}}\right)$. A partir de 10000 extracciones,

```
# parámetros de la simulación
sample.size <- 10
replicates <- 10000
# establecemos la semilla del generador de n° aleatorios
set.seed(5011)
# vector vacío para almacenar los resultados
sample.means <- vector("numeric", replicates)
## en cada repetición:
# extraemos una muestra de tamaño 10
# estimamos la media y almacenamos
for(k in 1:replicates){
  sample.rows <- sample(1:nrow(yrbss), sample.size)
  sample.means[k] <- mean(yrbss$weight[sample.rows], na.rm = TRUE)
}
## Comprobamos como es la distribución de las medias estimadas
# histograma
hist(sample.means, xlim = c(54, 87), ylim=c(0, 0.1), prob=TRUE)
# xlim, ylim para establecer el rango en el eje X e Y respectivamente
# línea azul en la media de las medias
abline(v = mean(sample.means), col = "blue")
# línea roja en la media poblacional
abline(v = mean(yrbss$weight, na.rm = TRUE), col = "red")
# curva de la distribución teórica N(mu, sigma/raiz(n))
curve(dnorm(x, mean=mean(yrbss$weight, na.rm = TRUE),
                        sd=sd(yrbss$weight, na.rm = TRUE)/sqrt(sample.size)),
      lwd=2, col="green", add=TRUE)
```

La función `abline()` se usa para añadir líneas rectas a un gráfico. Su estructura es `abline(a, b, h, v)`, donde `a` y `b` son el intercept y la pendiente de una recta del tipo $a + bX$; `h` se refiere al valor de `y` de una línea horizontal; y `v` al valor de `x` de una vertical.



Intervalos de confianza

Podemos construir intervalos de confianza a partir de las expresiones que hemos visto en clase, utilizando las funciones del tipo `qnorm()`, `qt()`, `qchisq()`, etc para calcular los percentiles asociados a una distribución concreta.

Vamos a calcular, por ejemplo, el IC para el peso medio poblacional (μ) asumiendo que el peso es una variable normalmente distribuida con σ desconocida. En estas condiciones, la expresión del intervalo de confianza para μ es $\mu \in \left(\bar{x} \pm t_{n-1; 1-\alpha/2} \frac{s}{\sqrt{n}} \right)$. Podemos usar R como una calculadora para calcular este intervalo de confianza, utilizando la función `qt()` para calcular el percentil correspondiente.

El primer paso será obtener una muestra de la población. Extraemos una muestra de tamaño 30,

```
# eliminamos los missing en la variable weight
yrbss.compl ete <- yrbss[compl ete.cases(yrbss$wei ght), ]
# tamaño muestral
n. si ze <- 30
# establecemos una semilla para poder replicar los resultados
set. seed(2014)
# obtenemos una secuencia de números aleatorios entre 1 y
# el nº de individuos completos de tamaño n. si ze.
sampl e. rows <- sampl e(1:nrow(yrbss.compl ete), n. si ze)
# extraemos la muestra y la guardamos en yrbss.sampl e
yrbss.sampl e <- yrbss.compl ete[sampl e. rows, ]
```

Hemos creado un nuevo data.frame `yrbss.sampl e` para almacenarlos. Previamente, se crea un data.frame en el que se eliminan los missings en la variable `wei ght` utilizando la función `compl ete.cases()`. Esta función devuelve un vector de valores lógicos con TRUE en las filas que no tiene ninguna columna (variable) missing. Por ejemplo,

```
# creamos un vector con dos posiciones mi ssi ng
x <- c(50, 3, 1.2, NA, NA)
compl ete.cases(x)
## [1] TRUE TRUE TRUE FALSE FALSE
# valores no mi ssi ng de x
x[compl ete.cases(x)]
## [1] 50.0 3.0 1.2
```

A partir de esa muestra, para construir el intervalo de confianza, primero calculamos la media y la desviación típica en la realización muestral,

```
x. bar <- mean(yrbss.sampl e$wei ght, na.rm=TRUE)
x. bar
## [1] 69.61167
```

```
s <- sd(yrbss. sample$weight, na.rm=TRUE)
s
## [1] 15.92963
```

Notad que no coinciden exactamente con los valores de los parámetros poblacionales $\mu = 67.9065$ y $\sigma = 16.89821$ debido a la variabilidad del muestreo. Los valores obtenidos son estimadores puntuales de los parámetros poblacionales.

En un intervalo de confianza del 95%, $t_{n-1;1-\alpha/2}$ es el percentil 0.975 de una t-Student con $n - 1 = 29$ grados de libertad. Calculamos este valor utilizando la función `qt()`.

```
tal pha <- qt(0.975, df = n.size-1, lower.tail=TRUE)
# equivalentemente qt(0.025, df = n.size-1, lower.tail=FALSE)
tal pha
## [1] 2.04523
```

Calculamos el margen de error $m = t_{n-1;1-\alpha/2} \frac{s}{\sqrt{n}}$.

```
m <- tal pha * s / sqrt(n.size)
m
## [1] 5.948223
```

Los límites del intervalo

```
ci.lower <- x.bar - m
ci.upper <- x.bar + m
c(ci.lower, ci.upper)
## [1] 63.66344 75.55989
```

Observad que el intervalo de confianza de nivel 95% para $\mu \in (63.66, 75.56)$. Esta realización muestral es una de las que forman parte del 95% que contienen al verdadero valor del parámetro.

Una forma alternativa de obtener este intervalo de confianza es utilizar la función `t.test()`. Por ejemplo, la siguiente sentencia devuelve el intervalo de confianza del 95% para el peso medio,

```
t.test(yrbss. sample$weight, conf.level = 0.95)$conf.int
[1] 63.66344 75.55989
attr(,"conf.level")
[1] 0.95
```

Con el argumento `conf.level` especificamos el nivel de confianza que nos interese.

Contrastes de hipótesis sobre la media poblacional

La función `t.test()` además permite hacer contrastes de hipótesis para la media poblacional de una variable normalmente distribuida con σ desconocida.

Esta función tiene la siguiente estructura:

```
t.test(x, alternative = "two.sided", mu = , conf.level = 0.95)
```

donde x es un vector numérico que contiene los valores muestrales, `alternative` especifica la forma de H_1 , μ es μ_0 y `conf.level` se refiere al nivel de confianza, es decir $1 - \alpha$. El argumento `alternative` puede ser "two.sided" cuando $H_1: \mu \neq \mu_0$, "less" si $H_1: \mu < \mu_0$ o "greater" si $H_1: \mu > \mu_0$. Por defecto el nivel de confianza será 95% y la alternativa de dos lados.

Por ejemplo, con la siguiente sentencia realizamos el contraste $H_0: \mu = 60$ vs $H_1: \mu > 60$,

```
t.test(yrbss$sampl e$wei ght, alternative = "greater", mu = 60, conf.level = 0.95)
```

```
One Sample t-test

data:  yrbss$sampl e$wei ght
t = 3.3049, df = 29, p-value = 0.001267
alternative hypothesis: true mean is greater than 60
95 percent confidence interval:
 64.67003      Inf
sample estimates:
mean of x
 69.61167
```

La salida de la función `t.test()` se organiza en forma de lista y se puede acceder a cada uno de los elementos utilizando el símbolo del dólar (\$) seguido del nombre del objeto. Estos objetos pueden ser: el valor del estadístico test observado (`statistic`), grados de libertad de la t-Student (`parameter`), p-valor (`p.value`), estimación de la media muestral (`estimate`), y el valor nulo (`null.value`). Por ejemplo, para acceder únicamente al p-valor del contraste anterior,

```
t.test(yrbss$sampl e$wei ght, alternative = "greater", mu = 60, conf.level = 0.95)$p.val
```

```
[1] 0.001267442
```

Notad que el intervalo de confianza calculado no tiene límite inferior, ya que en este caso la alternativa es `greater` se calcula una cota de confianza.

Contrastes de hipótesis para comparar medias de dos poblaciones

Es posible utilizar la función `t.test()` para llevar a cabo contrastes de hipótesis sobre medias de dos poblaciones. La estructura de esta función para ello es,

```
t.test(x, y, alternative = "two.sided", mu = , conf.level = 0.95)
```

donde ahora, además de los parámetros detallados anteriormente, tenemos x y y , los vectores numéricos que contienen los valores muestrales.

También es posible utilizarla de la forma,

```
t.test(x ~ y, ...)
```

donde y es una variable cualitativa con dos niveles, los grupos a comparar.

Por ejemplo, para tratar de contestar la pregunta: ¿Es mayor el peso en hombres que en mujeres? Tenemos dos variables de interés: $X_1 = \{peso\}$ e $Y = \{sexo\}$. La variable Y agrupa a X_1 en dos grupos de individuos independientes. Utilizando la misma muestra de la sección anterior, almacenada en el `data.frame` `yrbss.sample`, realizar un contraste de hipótesis sobre los parámetros pesos medios para cada una de esas variables. Para responder a la pregunta, tenemos que llevar a cabo el contraste $H_0: \mu_1 - \mu_2 = 0$ vs $H_1: \mu_1 - \mu_2 < 0$.

En el caso de **muestras independientes de poblaciones normales con varianzas desconocidas**, necesitamos saber si podemos asumir que estas varianzas son iguales o no. Para ello podemos hacer un contraste de igualdad de varianzas en el que $H_0: \frac{\sigma_1^2}{\sigma_2^2} = 1$ vs $H_0: \frac{\sigma_1^2}{\sigma_2^2} \neq 1$. Este contraste podemos llevarlo a cabo en R con la función `var.test()` cuya estructura general es,

```
var.test(x, y, ratio = 1, alternative = "two.sided", conf.level = 0.95)
```

donde x e y son los vectores numéricos que contienen los valores observados en nuestras variables, `ratio` es el valor nulo del parámetro $\frac{\sigma_1^2}{\sigma_2^2}$ por defecto 1, `alternative` especifica la dirección de la alternativa como indicamos en las secciones anteriores y `conf.level` se refiere al nivel de confianza. En nuestro contraste podríamos escribir,

```
var.test(yrbss.sample$weight[yrbss.sample$gender=="female"],
yrbss.sample$weight[yrbss.sample$gender=="male"])
```

F test to compare two variances

```
data:      yrbss.sample$weight[yrbss.sample$gender == "female"] and
yrbss.sample$weight[yrbss.sample$gender == "male"]
```

```
F = 0.91993, num df = 13, denom df = 15, p-value = 0.8887
```

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

```
95 percent confidence interval:
```

```
0.3145177 2.8082955
```

```
sample estimates:
```

```
ratio of variances
```

```
0.9199343
```

Notad que hemos dejado todos los argumentos de la función en sus valores por defecto.

Podríamos por tanto asumir que las dos varianzas son iguales. Para indicar esto en la función `t.test()` podemos utilizar el argumento `var.equal=TRUE`. Cuando este argumento es `FALSE` se realiza el contraste de Welch, que tiene en cuenta que no es posible asumir que las varianzas son iguales.

```
t.test(yrbss.sample$weight[yrbss.sample$gender=="female"],
yrbss.sample$weight[yrbss.sample$gender=="male"],
alternative = "less", mu = 0, conf.level = 0.95,
```

```
var.equal = TRUE)
```

```
Two Sample t-test
```

```
data:      yrbss.sample$weight[yrbss.sample$gender == "female"] and
yrbss.sample$weight[yrbss.sample$gender == "male"]
t = -1.2579, df = 28, p-value = 0.1094
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf 2.558181
sample estimates:
mean of x mean of y
 65.73929  73.00000
```

En el caso de **muestras dependientes**, podemos llevar a cabo un contraste de hipótesis para una media a partir de la nueva variable diferencia, o bien utilizar la función `t.test()` con el argumento `paired=TRUE`. La estructura general de esta función en este caso sería:

```
t.test(x, y, alternative = "two.sided", mu = , conf.level = 0.95,
paired = FALSE)
```

donde `x` y `y` son los vectores numéricos que contienen los valores observados en nuestras variables, `alternative` especifica la dirección de la alternativa como indicamos en la sección anterior, `mu` es el valor nulo del parámetro μ_D , `conf.level` se refiere al nivel de confianza y el argumento `paired` tendrá el valor `TRUE` cuando las dos muestras sean dependientes y `FALSE` (valor por defecto) en caso contrario.

Por ejemplo para evaluar las diferencias entre el peso y el peso ideal,

```
t.test(yrbss.sample$weight, yrbss.sample$wtdesired, alternative =
"two.sided", mu = 0, conf.level = 0.95, paired = TRUE)
```

```
Paired t-test
```

```
data:      yrbss.sample$weight and yrbss.sample$ideal weight
t = 1.6949, df = 29, p-value = 0.1008
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.057166 11.286992
sample estimates:
mean of the differences
      5.114913
```

Contrastes de proporciones

Inferencias para una proporción

Supongamos que X es una v.a. binomial con parámetros n y p , donde p es la probabilidad de éxito. Las inferencias sobre p se basan en la proporción muestral \hat{p} . Podemos utilizar la distribución normal para aproximar este estadístico muestral siempre que las observaciones sean independientes y se cumpla que $np \geq 5$ y $n(1-p) \geq 5$. Bajo estas condiciones,

$$\hat{p} \rightsquigarrow N\left(p, \sqrt{\frac{p(1-p)}{n}}\right)$$

Los intervalos de confianza se construyen con la expresión,

$$p \in \left(\hat{p} \pm z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right)$$

Y los contrastes del tipo $H_0: p = p_0$ se basan en el estadístico test,

$$T = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \xrightarrow{H_0} N(0,1)$$

Podemos utilizar la función `prop.test()` para hacer inferencias sobre proporciones bajo la asunción de que la proporción muestral es aproximadamente normal. Su estructura general es, `prop.test(x, n, alternative = "two.sided", p = 0.5, conf.level = 0.95)` donde x es el número de éxitos, n el tamaño muestral, `alternative` especifica la forma de la alternativa, p es el valor nulo p_0 y `conf.level` el nivel de confianza. El argumento `alternative` puede ser "two.sided" cuando $H_1: p \neq p_0$, "less" si $H_1: p < p_0$ o "greater" si $H_1: p > p_0$. Por defecto el nivel de confianza será 95% y la alternativa de dos lados.

La salida de la función `prop.test()` se organiza en forma de lista y se puede acceder a cada uno de los elementos utilizando el símbolo del dólar (\$) seguido del nombre del objeto. Los más interesantes son: el p-valor (`p.val`) y el intervalo de confianza (`conf.int`).

Por ejemplo, si queremos estimar la proporción de mujeres en la población utilizando un intervalo de confianza del 95%, escribimos:

```
# contamos cuantas mujeres hay
n.female <- sum(yrbss.sample$gender == "female")
prop.test(x = n.female, n = n.size)$conf.int
[1] 0.2879653 0.6536010
attr(,"conf.level")
[1] 0.95
```

El IC del 95% para p es $p \in (0.29, 0.65)$, es decir que con una confianza del 95% el porcentaje de mujeres estará entre el 27% y el 55%.

Si queremos por ejemplo llevar a cabo el contraste $H_0: p = 0.5$ vs $H_1: p < 0.5$

```
prop.test(x = n.female, n = n.size, alternative = "greater", p = 0.5)
1-sample proportions test with continuity correction
```

```

data:  n. female out of n. size, null probability 0.5
X-squared = 0.033333, df = 1, p-value = 0.5724
alternative hypothesis: true p is greater than 0.5
95 percent confidence interval:
 0.3109881 1.0000000
sample estimates:
      p
0.4666667

```

Inferencias para la diferencia de dos proporciones

La distribución normal puede aplicarse a la diferencia de dos proporciones muestrales $\hat{p}_1 - \hat{p}_2$ si la distribución de cada una de ellas es aproximadamente normal, es decir, si las observaciones son independientes y se cumple que $n_1 p_1 \geq 5$, $n_1(1 - p_1) \geq 5$, $n_2 p_2 \geq 5$ y $n_2(1 - p_2) \geq 5$. Bajo estas condiciones,

$$\hat{p}_1 - \hat{p}_2 \rightsquigarrow N\left(p_1 - p_2, \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}\right)$$

Los intervalos de confianza se construyen con la expresión,

$$p_1 - p_2 \in \left(\hat{p}_1 - \hat{p}_2 \pm z_{1-\alpha/2} \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}} \right)$$

Y los contrastes del tipo $H_0: p_1 = p_2$ se basan en el estadístico test,

$$T = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}_p(1-\hat{p}_p)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \xrightarrow{H_0} N(0,1)$$

Donde $\hat{p}_p = \frac{n_1 \cdot \hat{p}_1 + n_2 \cdot \hat{p}_2}{n_1 + n_2}$. Para llevar a cabo estas inferencias se puede utilizar la función `prop.test()`, con los argumentos `x` y `n` vectores de longitud 2, `x`: el número de éxitos en cada uno de los dos grupos y `n`: el tamaño de cada uno de los grupos.

Contraste chi-cuadrado

En el contraste chi-cuadrado se comparan las frecuencias observadas con las frecuencias esperadas en tablas de contingencia. Estas frecuencias esperadas se calculan, suponiendo que H_0 : independencia es cierta, como $n_{ij}^* = \frac{n_{i+} n_{+j}}{n}$ con $i = 1, \dots, r$ y $j = 1, \dots, c$. El estadístico test en el que se basa este contraste es,

$$T = \sum_{i=1}^r \sum_{j=1}^c \frac{(n_{ij} - n_{ij}^*)^2}{n_{ij}^*} \xrightarrow{H_0} \chi_{(r-1)(c-1)}^2$$

Siempre que se verifique que todas las frecuencias esperadas son mayores que 1 y al menos el 80% mayores que 4. Cuando $r = c = 2$ esta condición es que todas las frecuencias esperadas sean mayores que 4.

Valores muy grandes de este estadístico test apoyan la hipótesis alternativa de asociación.

Para calcular el p-valor podemos utilizar la función `pchi sq()` cuya estructura es,

`pchi sq(q, df, lower.tail = TRUE)`

donde q es k en las expresiones $P(\chi_{df}^2 \leq k)$ o $P(\chi_{df}^2 > k)$, df son los grados de libertad, y `lower.tail` es `TRUE` para calcular $P(\chi_{df}^2 \leq k)$ (valor por defecto) o `FALSE` para calcular $P(\chi_{df}^2 > k)$.

El contraste chi-cuadrado puede resolverse utilizando la función `chi sq. test()`,

`chi sq. test(x)`

donde x es la tabla de contingencia. La salida de la función `chi sq. test()` se organiza en forma de lista y se puede acceder a cada uno de los elementos utilizando el símbolo del dólar (\$) seguido del nombre del objeto. Los más interesantes son: las frecuencias esperadas (`expected`) y el p-valor (`p.val`).

Por ejemplo, para evaluar la relación entre medir más de 1.70 y el sexo, lo primero será construir la tabla de contingencia,

```
tbl.obs <- table(yrbss.sample$height > 1.7, yrbss.sample$gender)
tbl.obs
```

	female	male
FALSE	13	4
TRUE	1	12

Y el contraste,

`chi sq. test(tbl.obs)`

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

```
data: table(yrbss.sample$height > 1.7, yrbss.sample$gender)
X-squared = 11.374, df = 1, p-value = 0.0007447
```

El valor obtenido puede ser un poco diferente al que obtendríamos manualmente, ya que por defecto se aplica una corrección de continuidad. Con el argumento `correct = FALSE`, no se aplica esa corrección,

`chi sq. test(tbl.obs, correct = FALSE)`

Pearson's Chi-squared test

```
data: tbl.obs
X-squared = 14.001, df = 1, p-value = 0.0001827
```

Cuando no se verifican las condiciones de aplicación del contraste chi-cuadrado, aparece un mensaje de aviso. Por ejemplo, si el lugar de utilizar un umbral para la altura de 1.7 utilizamos 1.8,

```
tbl.obs <- table(yrbss.sample$height>1.8, yrbss.sample$gender)
chi.sq.test(tbl.obs, correct = FALSE)
```

Pearson's Chi-squared test

data: tbl.obs

X-squared = 0.90517, df = 1, p-value = 0.3414

Warning message:

In chi.sq.test(tbl.obs, correct = FALSE) :

Chi-squared approximation may be incorrect

la tabla de frecuencias esperadas,

```
chi.sq.test(tbl.obs, correct = FALSE)$expected
```

	female	male
FALSE	13.5333333	15.4666667
TRUE	0.4666667	0.5333333

Observad que dos frecuencias esperadas son menores que 1.