Práctica Estadística

Fernández Hernández, Alberto. 54003003S

08/12/2020

Contents

Ejercicio 1

Apartado 1

Dado el siguiente conjunto de datos, obtener con R las diferentes medidas de centralización y dispersión estudiadas. Así mismo obtener el diagrama de caja y bigotes.

Inicialmente, partimos de los siguientes valores de estatura, recogidos en un DataFrame formado por las columnas **alumnos** y **estaturas**:

```
datos.estatura <- data.frame(alumnos = c("Alumno1", "Alumno2", "Alumno3", "Alumno4",
                             "Alumno5", "Alumno6", "Alumno7", "Alumno8", "Alumno9",
                             "Alumno10", "Alumno11", "Alumno12", "Alumno13", "Alumno14",
                             "Alumno15", "Alumno16", "Alumno17", "Alumno18", "Alumno19",
                             "Alumno20", "Alumno21", "Alumno22", "Alumno23", "Alumno24",
                             "Alumno25", "Alumno26", "Alumno27", "Alumno28", "Alumno29",
                             "Alumno30"),
                             estatura = c(1.25, 1.28, 1.27, 1.21, 1.22, 1.29, 1.30, 1.24,
                              1.27, 1.29, 1.23, 1.26, 1.30, 1.21, 1.28, 1.30, 1.22, 1.25,
                              1.20, 1.28, 1.21, 1.29, 1.26, 1.22, 1.28, 1.27, 1.26, 1.23,
                              1.22, 1.21))
```

Comencemos con las medidas de posicionamiento central:

1. **MEDIA ARITMÉTICA**, empleando la función *mean* definida en R:

```
# Media artimetica
media.aritmetica <- mean(datos.estatura[, "estatura"])
media.aritmetica
```

```
## [1] 1.253333
```

Con el cálculo de las medidas de dispersión comprobaremos si la media es representativa o no de la muestra. Por otro lado, aunque no corresponde con la muestra empleada, también podemos calcular la **media geométrica**, basado en el producto de cada valor, obteniendo finalmente su raíz n-ésima (siendo n el total de datos de la muestra). Dado que R no dispone de una función específica, mediante la función *Reduce* calculamos el productorio, elevando el resultado a $\frac{1}{n}$:

```
# Media geometrica
# Podemos ver que la Media geometrica es ligeramente inferior a la aritmetica
media.geometrica <- Reduce(prod, datos.estatura["estatura"], init = 1) **
                           (1/nrow(datos.estatura))
media.geometrica
```

```
## [1] 1.252927
```

2. **MEDIANA**, empleando la función *median* definida en R:

```
mediana <- median(datos.estatura[, "estatura"])
mediana
```

```
## [1] 1.26
```

Es decir, la mitad de los alumnos miden 1.26 o menos, mientras que el 50 % restante miden 1.26 o más.

3. **MODA**. Por desgracia, R no dispone de una función específica para el cálculo de la moda. Para ello, mediante la función *table* creamos una tabla con las frecuencias absolutas de cada estatura:

```
frecuencias.estaturas <- as.data.frame(table(Estatura = datos.estatura[, "estatura"]))
frecuencias.estaturas
```

```
##    Estatura Freq
```

```
## 1       1.2    1
## 2      1.21    4
## 3      1.22    4
## 4      1.23    2
## 5      1.24    1
## 6      1.25    2
## 7      1.26    3
## 8      1.27    3
## 9      1.28    4
## 10     1.29    3
## 11      1.3    3
```

A continuación, ordenamos las frecuencias:

```
# Lo pasamos a tipo de dato numeric (por defecto esta en tipo factor)
frecuencias.estaturas[, "Estatura"] <- as.numeric(levels(frecuencias.estaturas[, "Estatura"]))
frecuencias.estaturas <- frecuencias.estaturas[order(-frecuencias.estaturas[, "Freq"]),]
frecuencias.estaturas
```

```
##    Estatura Freq
## 2      1.21    4
## 3      1.22    4
## 9      1.28    4
## 7      1.26    3
## 8      1.27    3
## 10     1.29    3
## 11     1.30    3
## 4      1.23    2
## 6      1.25    2
## 1      1.20    1
## 5      1.24    1
```

Una vez ordenadas, mediante la función *which* recuperamos aquellas estaturas cuya frecuencia absoluta corresponda con la frecuencia máxima en el DataFrame. Dado que el máximo corresponde a varias estaturas, la moda resultante será más de un valor:

```
moda <- as.double(frecuencias.estaturas[which(frecuencias.estaturas[, "Freq"] ==
                                        max(frecuencias.estaturas[, "Freq"])), "Estatura"])
moda
```

```
## [1] 1.21 1.22 1.28
```

Por tanto, las estaturas más repetidas son 1.21, 1.22 y 1.28 metros.

A continuación, analizamos las medidas de dispersión con el objetivo de estudiar si los datos se encuentran más o menos concentrados o dispersos:

4. **RANGO**. Para ello, R dispone de una función denominada *range* que NO calcula el rango, sino que devuelve los valores máximo y mínimo de la muestra. Por tanto, una vez obtenidos ambos valores, se restan mediante la función *diff*:

```
# Range devuelve los valores maximo y minimo, NO el rango
range(datos.estatura[, "estatura"])
```

```
## [1] 1.2 1.3
```

```
rango <- diff(range(datos.estatura[, "estatura"]))

rango
```

```
## [1] 0.1
```

En este caso, la amplitud obtenida es de 10 centímetros entre la estatura máxima y mínima.

5. **VARIANZA**. Para el cálculo de la varianza, R dispone de la función *var* que permite obtener la **cuasi-varianza**, es decir, en lugar de obtener:

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

Calcula:

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

Por ello, si deseamos obtener la **varianza** debemos multiplicar el resultado obtenido en la función *var* por $\frac{(n-1)}{n}$:

```
# Cuasi-varianza
var(datos.estatura[, "estatura"])
```

```
## [1] 0.001050575
```

```
# Varianza
varianza <- var(datos.estatura[, "estatura"]) *
                    ((nrow(datos.estatura) - 1 )/ nrow(datos.estatura))
varianza
```

```
## [1] 0.001015556
```

Sin embargo, la varianza nos devuelve el resultado en las unidades de medida al **cuadrado**, por lo que hay que calcular su raíz cuadrada, es decir, su **desviación típica**, con el objetivo de obtener las mismas unidades que la media.

6. **DESVIACIÓN TÍPICA**. Nuevamente, R dispone de la función *sd* que obtiene la desviación a partir de la cuasi-varianza, por lo que hay que multiplicar el resultado por $\sqrt{\frac{(n-1)}{n}}$:

```
# Desviacion tipica obtenida a partir de la cuasi-varianza
sd(datos.estatura[, "estatura"])
```

```
## [1] 0.03241257
```

```
# Desviacion tipica obtenida a partir de la varianza
desv.tipica <- sd(datos.estatura[, "estatura"]) *
                    sqrt((nrow(datos.estatura) - 1 )/ nrow(datos.estatura))
desv.tipica
```

```
## [1] 0.03186778
```

Analizando el resultado obtenido, podemos comprobar como la dispersión de las medidas con respecto a la media es de unos centímetros de diferencia. No obstante, si realizamos un gráfico de densidad y lo comparamos con el de una distribución normal con la media y desviación típica obtenidas, vemos que muchas de las estaturas no se concentran en torno a la media, sino que observamos una mayor "concentración" de valores en torno a estaturas más bajas y más altas, correspondientes con los valores de la moda (en torno a 1.21, 1.22 y 1.28 metros). Por el contrario, la densidad al aproximarse a la media (1.25) es menor, por lo que no parece ser **un valor muy representativo de la muestra**:

```
plot(density(datos.estatura[, "estatura"]), type = 'l', ylim = c(0,15),
     lwd = 2, xlab = "Estatura (metros)", ylab = "Densidad",
     main = "Densidad estaturas - distribucion normal")
curve(dnorm(x, mean = media.aritmetica, sd = desv.tipica),
```

```
      col = 'red', lwd = 2, type = 'l', add = TRUE)
legend("topleft", legend = c("Densidad estaturas", "Distribucion normal"),
       col = c("black", "red"), lty = 1, lwd = 2)
```

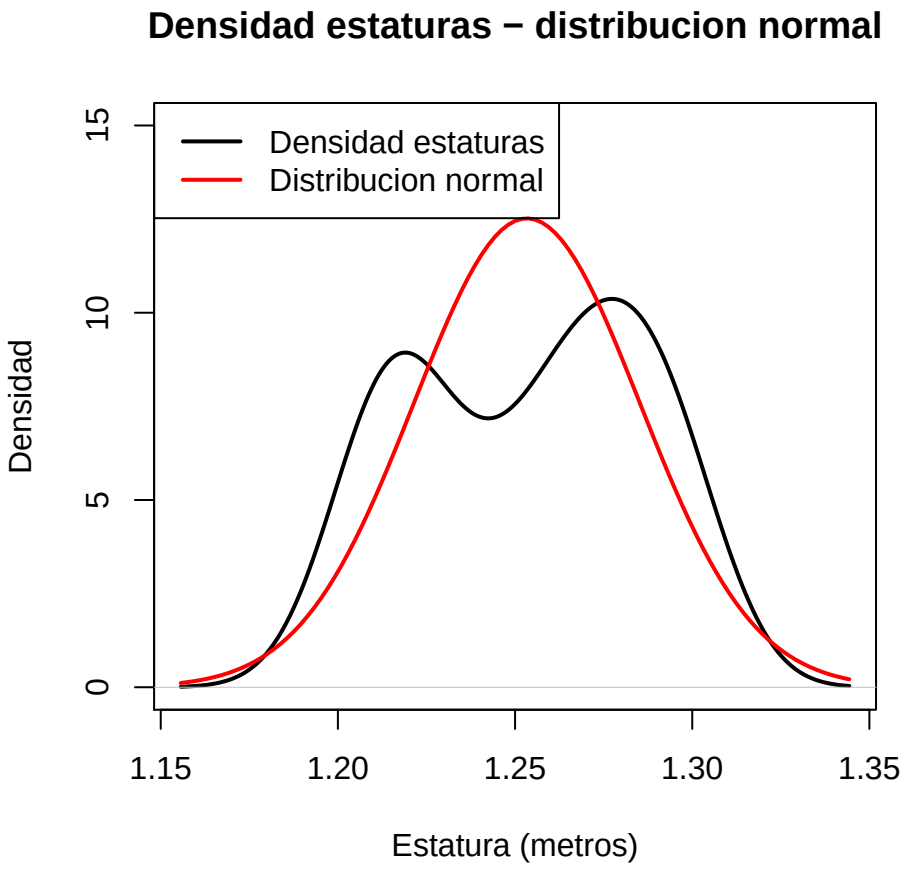


Figure 1: Comparativa entre la densidad estaturas y la distribución normal con la media y desviación típica obtenidas

Una mejor forma de observar dicha dispersión es mediante un **diagrama de caja y bigotes**, empleando la función *boxplot* de R:

```r
boxplot(datos.estatura[, "estatura"],
        las = 1, ylab = "ESTATURAS", horizontal = TRUE)
```

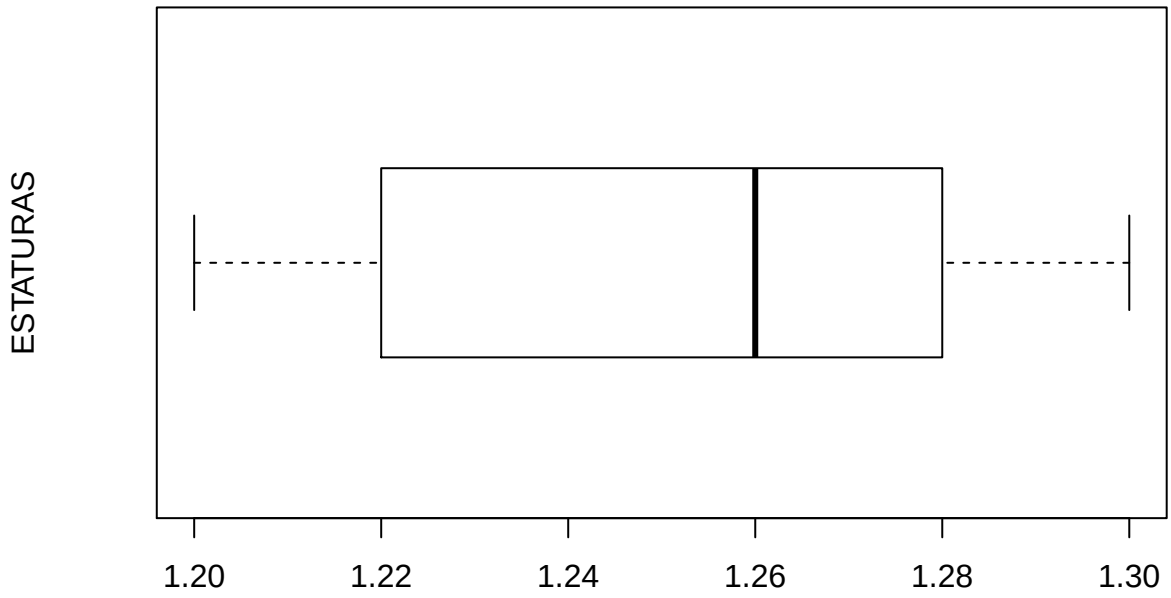


Figure 2: Diagrama de caja y bigotes de las medidas de la muestra

```r
# Mediante la funcion summary mostramos los valores de los cuartiles
summary(datos.estatura[, "estatura"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.200   1.220   1.260   1.253   1.280   1.300
```

Como podemos observar a partir del diagrama anterior, la parte izquierda del gráfico es significativamente mayor que la de la derecha, es decir, **las estaturas comprendidas entre el 25 y el 50 % de la muestra (1.22 y 1.26 metros) están mucho más dispersas con respecto a la mediana que las medidas situadas entre 1.26 y 1.28 metros (50 y 75 %)**. De hecho, el coeficiente de curtosis obtenido mediante la función *kurtosis* devuelve un valor menor que 0, lo que reafirma el bajo grado de concentración de los valores alrededor de la media (leptocúrtica):

```r
# Importamos el paquete EnvStats
library(EnvStats)
kurtosis(datos.estatura[, "estatura"])
```

```
## [1] -1.428355
```

Por otro lado, la amplitud de cada "bigote" es la misma (0.02), por lo que ambos extremos presentan la misma concentración:

$$\text{1er cuartil} - \text{Minimo} = 1.22 - 1.20 = 0.02$$

$$\text{Maximo} - \text{3er cuartil} = 1.30 - 1.28 = 0.02$$

Además, el **rango intercuartílico** es $Q_3 - Q_1 = 0.06$, es decir, el 50 % de las estaturas de la muestra están comprendidas entre 1.22 y 1.28 metros. Por otra parte, **no se han detectado valores atípicos fuera del rango**.

7. **COEFICIENTE DE VARIACIÓN DE PEARSON**. Se calcula como el cociente entre la desviación típica y la media en términos absolutos:

```
coef.var.pearson <- desv.tipica / abs(media.aritmetica)
coef.var.pearson
```

```
## [1] 0.02542642
```

Por lo general, dicho coeficiente se emplea para comparar el nivel de **dispersión** entre dos muestras, especialmente cuando vienen expresadas en distintas unidades (lo cual no ocurre con la desviación típica).

Apartado 2

Dado el siguiente conjunto de datos, obtener la tabla de correspondencias, con R, agrupando cada variable en cuatro clases o intervalos. Estos deberán ser elegidos por el alumno.

Como paso previo a la tabla de correspondencias, cargamos las medidas de estatura en un DataFrame con las columnas alumnos, estatura y peso:

```
datos.estatura.peso <- data.frame(alumnos = c("Alumno1", "Alumno2", "Alumno3", "Alumno4",
                        "Alumno5", "Alumno6", "Alumno7", "Alumno8", "Alumno9",
                        "Alumno10", "Alumno11", "Alumno12", "Alumno13", "Alumno14",
                        "Alumno15", "Alumno16", "Alumno17", "Alumno18", "Alumno19",
                        "Alumno20", "Alumno21", "Alumno22", "Alumno23", "Alumno24",
                        "Alumno25", "Alumno26", "Alumno27", "Alumno28", "Alumno29",
                        "Alumno30"),

                         estatura = c(1.25, 1.28, 1.27, 1.21, 1.22, 1.29, 1.30, 1.24, 1.27,
                         1.29, 1.25, 1.28, 1.27, 1.21, 1.22, 1.29, 1.30, 1.24,
                         1.27, 1.29, 1.25, 1.28, 1.27, 1.21, 1.22, 1.29, 1.30,
                         1.24, 1.27, 1.29),

                         peso = c(32, 33, 31, 34, 32, 31, 34, 32, 32, 35, 31, 35, 34,
                         33, 33, 31, 35, 32, 31, 33, 33, 32, 34, 34, 35, 31, 34,
                         33, 35, 34))
```

1. **ESTATURA**: En primer lugar, las columnas de estatura se agruparán en función de los cuartiles, además de los valores máximo y mínimo. De este modo, agrupando los cuartiles por intervalos es posible analizar con mayor rapidez la distribución o dispersión de los alumnos en base a la estatura. Para obtener dichos valores, empleamos la función *quantile* disponible en R:

```
# Estatura
breaks <- as.vector(quantile(datos.estatura.peso[, "estatura"]))
breaks
```

```
## [1] 1.21 1.24 1.27 1.29 1.30
```

Una vez recuperados, el objetivo es obtener los siguientes intervalos:

$$\text{Intervalo peso: } [1.21, 1.24); [1.24, 1.27); [1.27, 1.29); [1.29, 1.30]$$

Como podemos observar, la amplitud de los intervalos varía ligeramente. Esto permite remarcar los dos últimos donde se concentran el mayor número de muestras, esto es, entre 1.27 y 1.30, tal y como podemos comprobar en la siguiente tabla de frecuencias:

```
# Tabla con las frecuencias de estatura
table(datos.estatura.peso[, "estatura"])
```

```
##
## 1.21 1.22 1.24 1.25 1.27 1.28 1.29  1.3
##    3    3    3    3    6    3    6    3
```

Para ello, la función *cut* de R nos permite dividir el rango de un vector en intervalos de longitud N pasado como parámetro, además de agrupar el número de datos por cada intervalo. Por tanto, el conjunto de intervalos para la estatura queda de la siguiente forma:

```
intervalo.estatura <- cut(datos.estatura.peso[, "estatura"], breaks = breaks,
                          include.lowest = TRUE, right = FALSE)
levels(intervalo.estatura)
```

```
## [1] "[1.21,1.24)" "[1.24,1.27)" "[1.27,1.29)" "[1.29,1.3]"
```

Caben destacar los parámetros *include.lowest* y *right*, los cuales permiten configurar el intervalo para incluir el valor más bajo, además de establecer el intervalo abierto por la derecha (a excepción del último).

2. **PESO**: A continuación, analizamos las frecuencias de la columna peso:

```
# Estatura
table(datos.estatura.peso[, "peso"])
```

```
##
## 31 32 33 34 35
##  6  6  6  7  5
```

En este caso, y dado que solo disponemos de 5 valores de peso, salvo el último intervalo el resto estará formado por un único valor. Para ello, desde la función *cut* basta con pasar como parámetro el número de intervalos a formar (4):

```
intervalo.peso <- cut(datos.estatura.peso[, "peso"], breaks = 4,
                      include.lowest = TRUE, right = FALSE)
levels(intervalo.peso)
```

```
## [1] "[31,32)" "[32,33)" "[33,34)" "[34,35]"
```

Una vez creados los intervalos, mediante un DataFrame obtenemos la frecuencia de aparición de cada dato, creando una **tabla de correspondencias**:

```
df.intervalos <- data.frame(estatura = intervalo.estatura,
                            peso = intervalo.peso)
tabla.correspondencias <- table(df.intervalos[, "estatura"], df.intervalos[, "peso"])
tabla.correspondencias
```

```
##
##               [31,32) [32,33) [33,34) [34,35]
##   [1.21,1.24)       0       1       2       3
##   [1.24,1.27)       1       3       2       0
##   [1.27,1.29)       2       2       1       4
##   [1.29,1.3]        3       0       1       5
```

De forma adicional, podemos calcular las frecuencias marginales de cada fila y columna, mediante la función *apply*, aplicando a cada fila/columna la función suma (*sum*):

```
tabla.correspondencias <- rbind(tabla.correspondencias, apply(tabla.correspondencias, 2, sum))
tabla.correspondencias <- cbind(tabla.correspondencias, apply(tabla.correspondencias, 1, sum))
tabla.correspondencias
```

```
##             [31,32) [32,33) [33,34) [34,35]
## [1.21,1.24)       0       1       2       3  6
## [1.24,1.27)       1       3       2       0  6
## [1.27,1.29)       2       2       1       4  9
## [1.29,1.3]        3       0       1       5  9
##                   6       6       6      12 30
```

Analizando esta última tabla, podemos comprobar como el intervalo de peso con mayor número de alumnos se sitúa entre los 34 y 35 kg, mientras que los intervalos de altura se mueven en torno a 1.27 y 1.30 metros.

Ejercicio 2

Considerando, de nuevo, los datos de la primera pregunta del ejercicio anterior, se pide obtener un intervalo de confianza para la diferencia de medias teóricas entre las observaciones de los primeros 15 casos y de los segundos 15 casos.

Inicialmente, nos encontramos con dos submuestras de alturas de diferentes personas:

```
# Ejercicio 2
datos.estatura.primeros.15 <- datos.estatura[1:15,]
datos.estatura.ultimos.15 <- datos.estatura[16:30,]
```

De cara al cálculo del intervalo de confianza, debemos preguntarnos dos cuestiones fundamentales:

1. **¿Los datos están distribuidos normalmente?**
2. **¿La varianza de ambas poblaciones, aunque desconocidas para nosotros, son iguales?**

Para estudiar la normalidad de ambas muestras, una primera aproximación es mediante un gráfico de cuantiles o **gráfico Q-Q (*Quantile-Quantile*)**, así como un gráfico de densidad, por medio de una función denominada **mostrar_graficos**:

```
# Importamos el paquete car, el cual contiene la funcion qqPlot
library(car)

mostrar_graficos <- function(datos, columna) {
  par(mfrow = c(1,2))
  qqPlot(datos[, columna], pch=19, las=1, main='QQplot',
         xlab='Cuantiles teoricos', ylab='Cuantiles muestrales',
         envelope=0.95)
  plot(density(datos[, columna]), lwd = 3, col = 'blue',
       xlab = 'Altura (metros)', ylab = 'Densidad',
       main = 'Grafico de densidad')
  abline(v = mean(datos[, columna]), lwd = 2, lty = 2, col = "red")
}
```

Una vez definida la función, mostramos los gráficos correspondientes a ambas submuestras:

```r
# Primeros 15 datos
mostrar_graficos(datos.estatura.primeros.15, "estatura")
```

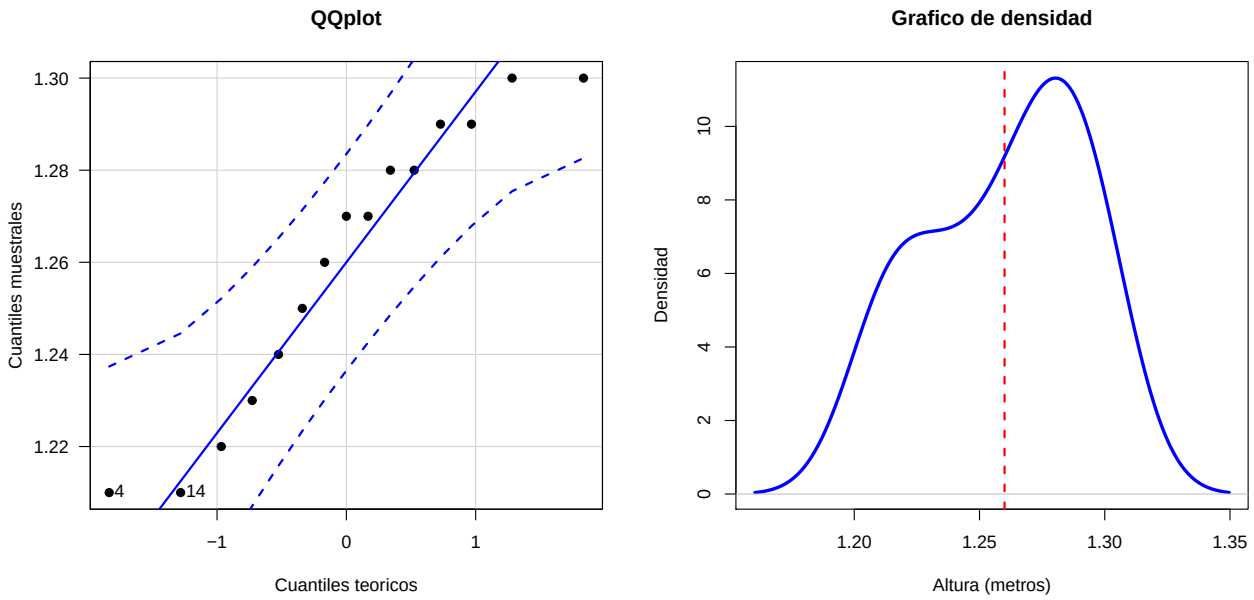
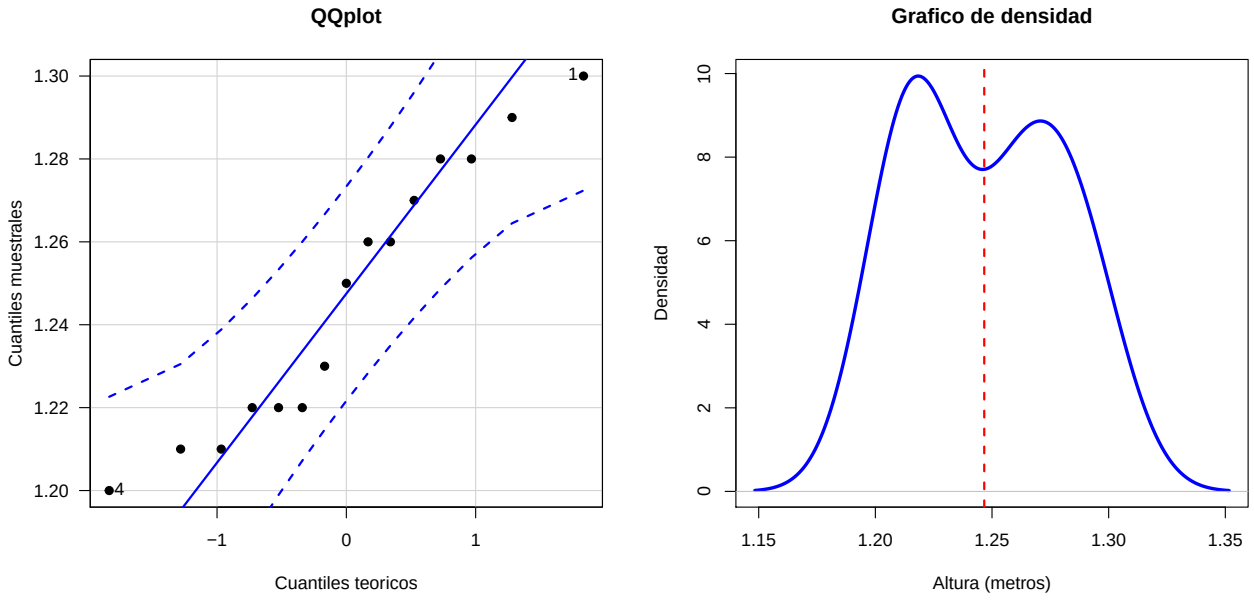


Figure 3: Grafico Q-Q y de densidad de los primeros 15 datos

```r
# Ultimos 15 datos
mostrar_graficos(datos.estatura.ultimos.15, "estatura")
```



Por un lado, el primer gráfico muestra la comparación entre la distribución de las estaturas de cada submuestra con los **cuantiles de una distribución teórica normal** (recta azul). En primera instancia, podemos comprobar que existen medidas en ambas submuestras que no se ajustan a la diagonal, es decir, **existen desviaciones con respecto a la recta**. No obstante, ¿Cómo de sustancial es dicho grado de desviación? Para ello, añadimos una **banda de confianza**: líneas discontínuas que representan los límites de confianza, tanto superior como inferior, para los puntos ajustados sobre la recta (por defecto del 95 %). Podemos comprobar como todos los puntos están contenidos dentro de dichos límites, por lo que no puede descartarse que ambas submuestras puedan provenir de una distribución normal.

Sin embargo, el gráfico de densidad tampoco nos aclara la distribución de los datos, aunque bien es cierto que la segunda submuestra presenta una mayor simetría con respecto a la primera. No obstante, como primera impresión no podemos descartar que los datos no provengan de una distribución normal.

Por ello, debemos plantear una alternativa "menos subjetiva" para determinar si los datos provienen de una distribución normal o no. Para ello, planteamos la siguiente hipótesis nula:

$$H_0 : \text{La muestra proviene de una distribución normal}$$
$$H_1 : \text{La muestra no proviene de una distribución normal}$$

Una posibilidad sería emplear el método de Kolmogorov-Smirnov visto en clase. Sin embargo, dicha técnica de inferencia asume que se conoce la media y desviación típica de la población, parámetros que por supuesto **desconocemos**. Por ello, R dispone de una función específica denominada *shapiro.test*, basada en la técnica Shapiro-Wilk que permite contrastar la normalidad de los datos sin necesidad de conocer los parámetros poblacionales, adecuada además cuando la muestra es pequeña. Dicha técnica se basa en el cálculo del estadístico W:

$$W = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Donde x_i es el i-ésimo valor de la muestra y a_i el coeficiente obtenido a través de la tabla de contraste de Shapiro-Wilks, planteando como hipótesis nula que los datos de una muestra **provienen de una población normalmente distribuida**:

```
# Prueba Shapiro-Wilk en R
# Primeros 15 datos
shapiro.test(datos.estatura.primeros.15[, "estatura"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos.estatura.primeros.15[, "estatura"]
## W = 0.91739, p-value = 0.1757
```

```
# Ultimos 15 datos
shapiro.test(datos.estatura.ultimos.15[, "estatura"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos.estatura.ultimos.15[, "estatura"]
## W = 0.92439, p-value = 0.2247
```

Como podemos observar en las salidas anteriores, los p-valores obtenidos en ambas submuestras son superiores al nivel de significación $\alpha = 0.05$, aunque la segunda submuestra con mayor p-valor que la primera. Dicha comparación implica que **estadísticamente no existe evidencia en contra de que ambas submuestras provengan de una distribución normal**. Por ello, con vistas a este apartado no podemos rechazar que los datos provienen de poblaciones normalmente distribuidas, por lo que lo asumiremos.

Una vez asumida la distribución normal de los datos, de cara al cálculo del intervalo de confianza debemos preguntarnos ¿Cómo son las varianzas de ambas poblaciones? ¿Cómo están relacionadas? Dado que en función de la respuesta a esta pregunta, el intervalo de confianza será diferente. El objetivo de la estadística inferencial es inducir, a partir de las propiedades de la muestra, el comportamiento de la población. No obstante, a menos que obtengamos toda la información (lo cual es poco probable) **NO podemos conocer a la población con exactitud** (μ, σ), pero si un intervalo entre cuyos valores se estima que se encuentra cada uno de estos parámetros.

Por ello, comenzando con la varianza proponemos la siguiente hipótesis nula:

$$H_0 : \sigma_1 = \sigma_2 \text{ , es decir, } \frac{\sigma_1}{\sigma_2} = 1$$
$$H_1 : \sigma_1 \neq \sigma_2 \text{ , es decir, } \frac{\sigma_1}{\sigma_2} \neq 1$$

Para ello, emplearemos la prueba F de Fisher con el objetivo de comparar ambas varianzas. Por ello, se sabe que bajo hipótesis nula:

$\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2}$ presenta una distribución en el muestreo $F_{n-1,m-1}$ donde n y m son los tamaños de las muestras

En base a dicho estadístico de contraste, el objetivo será encontrar dos valores F de Fisher tales que:

$$P(F_{n-1,m-1,1-\alpha/2} < \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} < F_{n-1,m-1,\alpha/2}) = 1 - \alpha$$

Es decir, un intervalo de confianza para el cociente de las varianzas poblacionales con un nivel de confianza 1 - α:

$$(\frac{S_1^2}{S_2^2}\frac{1}{F_{n-1,m-1,\alpha/2}}, \frac{S_1^2}{S_2^2}\frac{1}{F_{n-1,m-1,1-\alpha/2}})$$

Una primera aproximación sería realizar el cálculo del intervalo anterior de forma manual, creando una función específica denominada **contrastar_varianzas**. En este caso, para obtener el valor F de Fisher de la tabla, R dispone de la función *qf* , la cual devuelve el valor correspondiente en función de los grados de libertad del numerador y denominador (df1 y df2), así como del valor α:

```
contrastar_varianzas <- function(x, y, columna, confianza) {
  n.1 <- nrow(x)
  n.2 <- nrow(y)
  # R calcula la cuasi-varianza, por lo que debemos multiplicar
  # la varianza obtenida por (n - 1) / n para obtener la varianza
  var.1 <- var(x[, columna]) * ((n.1 - 1 )/ n.1)
  var.2 <- var(y[, columna]) * ((n.2 - 1 )/ n.2)
  estadistico.f <- min(var.1, var.2) / max(var.1, var.2)
  lim.inf <- estadistico.f * (1 / qf(1- (1 - confianza) / 2, df1 = n.1 - 1, df2 = n.2 - 1))
  lim.sup <- estadistico.f * (1 / qf((1 - confianza) / 2, df1 = n.1 - 1, df2 = n.2 - 1))
  cat("Estadistico F: ", estadistico.f, ". Intervalo: [",lim.inf, ",", lim.sup, "]")
}
```

Por tanto, para un valor de confianza del 95 %:

```
contrastar_varianzas(datos.estatura.primeros.15, datos.estatura.ultimos.15, "estatura", 0.95)
```

```
## Estadistico F:  0.9251101 . Intervalo: [ 0.3105869 , 2.755521 ]
```

Para comprobar que el resultado obtenido es correcto, R dispone de una función predefinida denominada *var.test* , empleada para comparar las varianzas de dos poblaciones:

```
var.test(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"])
```

```
##
##  F test to compare two variances
##
## data:  datos.estatura.primeros.15[, "estatura"] and datos.estatura.ultimos.15[, "estatura"]
## F = 0.92511, num df = 14, denom df = 14, p-value = 0.8863
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3105869 2.7555215
## sample estimates:
## ratio of variances
##          0.9251101
```

Analizando la salida anterior, podemos comprobar como el intervalo resultante es prácticamente idéntico al obtenido en la función **contrastar_varianzas**:

Intervalo de confianza al 95 %: [0.3105869, 2.7555215]

De dicha salida caben destacar dos aspectos:

1. En primer lugar, el intervalo obtenido comprende desde 0.31 hasta 2.76, aproximadamente, intervalo de confianza en el que está incluido el 1, por lo que no descartamos que ambas varianzas sean iguales ($\frac{\sigma_1}{\sigma_2} = 1$).
2. Por otro lado, el p-valor obtenido a partir del estadístico F (0.8863) es significativamente superior al valor $\alpha = 0.05$.

Incluso reduciendo el intervalo de confianza a un 80 %, el valor de uno sigue contenido en el intervalo:

```
var.test(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"],
         conf.level = 0.80)
```

```
##
##  F test to compare two variances
##
## data:  datos.estatura.primeros.15[, "estatura"] and datos.estatura.ultimos.15[, "estatura"]
## F = 0.92511, num df = 14, denom df = 14, p-value = 0.8863
## alternative hypothesis: true ratio of variances is not equal to 1
## 80 percent confidence interval:
##  0.4574242 1.8709741
## sample estimates:
## ratio of variances
##          0.9251101
```

Por ello, se concluye que los resultados obtenidos **no muestran evidencia en contra de la homogeneidad de las varianzas poblacionales (Homocedasticidad). Por tanto, la hipótesis nula H_0 se mantiene**. Como conclusión, podemos asumir que ambas varianzas son iguales.

Por tanto, una vez determinada la normalidad de los datos así como la igualdad de las varianzas poblacionales, ya podemos calcular el **intervalo de confianza para la diferencia de medias con varianzas desconocidas pero iguales**, visto en clase:

$$IC_\alpha(\mu_1-\mu_2)=\bar{X}-\bar{Y}\pm t_{n_1+n_2-2,\alpha/2}\frac{\sqrt{(n_1S_x^2+n_2S_y^2)[(1/n_1)+(1/n_2)]}}{\sqrt{n_1+n_2-2}}$$

Para ello, y en lugar de calcular a mano el intervalo, R dispone nuevamente de una función predefinida: *t.test*, obteniendo con ella el intervalo de confianza. Asumiendo que no existe evidencia en contra de que ambas varianzas poblcionales sean iguales (**aunque desconocidas**), marcamos el parámetro *var.equal* a TRUE, dado que por defecto R asume lo contrario (FALSE):

```
# Por defecto, conf.level esta a 0.95
x <- t.test(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"],
       var.equal = TRUE, conf.level = 0.95)
x
```

```
##
##  Two Sample t-test
##
## data:  datos.estatura.primeros.15[, "estatura"] and datos.estatura.ultimos.15[, "estatura"]
## t = 1.132, df = 28, p-value = 0.2672
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01079358  0.03746024
## sample estimates:
## mean of x mean of y
##  1.260000  1.246667
```

SOLUCIÓN. Como podemos observar, obtenemos el siguiente intervalo:

Intervalo de confianza (al 95 %) para la diferencia de medias teóricas: $[-0.011; 0.037]$

Dicho intervalo abarca tantos valores positivos como negativos, por lo que cabe la posibilidad de que la diferencia entre ambas medias sea positiva, negativa o incluso cero (pues está incluido en el intervalo), lo que supone que la media de ambas poblaciones pueden ser iguales. Por otra parte, entre la información del contraste de hipótesis se incluye el valor del estadístico T (1.132), el número de grados de libertad (28), así como el p-valor obtenido: 0.2672, el cual es superior al nivel de significación $\alpha = 0.05$. Esto último supone que **no podemos rechazar, con un 95 % de confianza, la hipótesis nula de que ambas medias sean idénticas**.

De hecho, *t.test* considera como hipótesis nula que **la media entre ambas submuestras son iguales**, ya que si nos fijamos en la salida anterior vemos que considera como hipótesis alternativa o H_1 justo lo contrario: *true difference in means is not equal to 0* (*two.sided*).

De forma gráfica, podemos observar que el valor del estadístico T "cae" dentro de la zona de aceptación de la hipótesis nula, por lo que a un 95 % de confianza se puede asumir que la diferencia entre los promedios de estatura $(\mu_1-\mu_2)$ es cero [1] [2]

```
# Cargamos para ello dos librerias adicionales
library(moonBook)
library(webr)

plot(x)
```

[1] https://cran.r-project.org/web/packages/moonBook/moonBook.pdf
[2] https://cran.r-project.org/web/packages/webr/webr.pdf

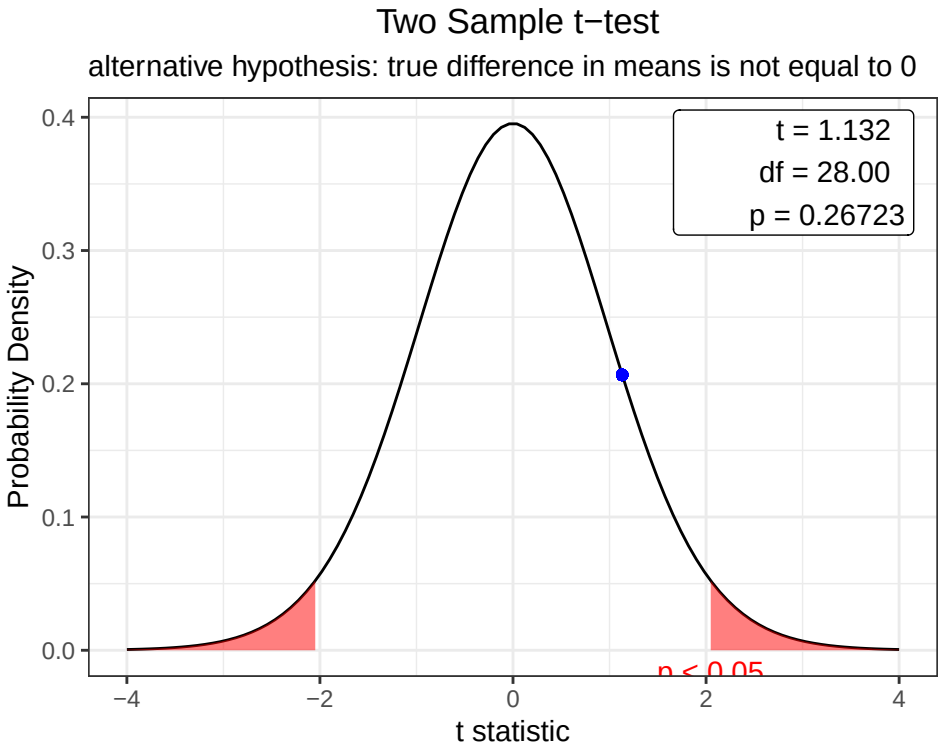


Figure 4: Valor del estadístico T sobre un intervalo de confianza del 95 %

Sin embargo, ¿Qué ocurriría si reducimos el intervalo de confianza de un 95 a un 80 %, por ejemplo?

```
x <- t.test(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"],
       var.equal = TRUE, conf.level = 0.80)
x
```

```
##
##  Two Sample t-test
##
## data:  datos.estatura.primeros.15[, "estatura"] and datos.estatura.ultimos.15[, "estatura"]
## t = 1.132, df = 28, p-value = 0.2672
## alternative hypothesis: true difference in means is not equal to 0
## 80 percent confidence interval:
##  -0.002126102  0.028792768
## sample estimates:
## mean of x mean of y
##  1.260000  1.246667
```

En este último intervalo, a un 80 % de confianza, el valor cero **sigue contenido dentro del intervalo** (más cerrado que en el caso anterior), por lo que podemos seguir asegurando que no existe evidencia en contra de la igualdad entre ambas medias, dado que el cero se encuentra dentro del intervalo, además de que el p-valor es superior al nivel de significancia $\alpha = 0.20$. Generalizando, **sólo se rechazaría H_0 para tamaños α mayor al p-valor = 0.2672 (por ejemplo, $\alpha = 0.3$)**.

Intervalo de confianza (al 80 %) para la diferencia de medias teóricas: $[-0.002; 0.029]$

```
# Mostramos graficamente el valor del estadistico T
plot(x)
```

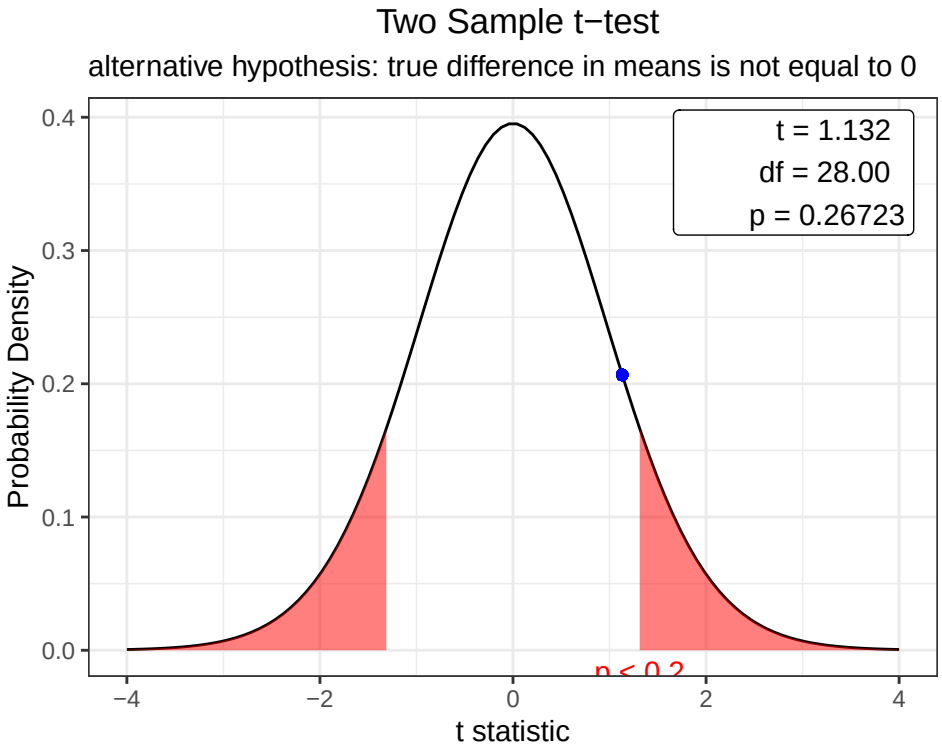


Figure 5: Valor del estadístico T sobre un intervalo de confianza del 80 %

Así mismo, se pide contrastar la hipótesis nula de que ambas sub-muestras tienen la misma media, es decir, proceden de la misma población. Detallar las hipótesis necesarias para hacer tal contraste, aunque no es preciso comprobarlas. El análisis debe realizarse con R.

Aunque hemos comprobado que el p-valor es superior al valor α en cada caso, de igual modo podemos realizar el contraste de hipótesis con ambos porcentajes "a mano", planteando H_0 y H_1 de forma bilateral, es decir, la hipótesis nula defiende la igualdad de ambas medias, mientras que la hipótesis alternativa defiende la desigualdad entre ambas (sea mayor o menor):

$$H_0 : \mu_1 = \mu_2 \text{ , es decir, } \mu_1 - \mu_2 = 0$$
$$H_1 : \mu_1 \neq \mu_2 \text{ , es decir, } \mu_1 - \mu_2 \neq 0$$

El objetivo es realizar el cálculo del estadístico T:

$$T = \frac{|\bar{X} - \bar{Y} - 0|}{\sqrt{\frac{(n_1 S_x^2 + n_2 S_y^2)}{n_1 + n_2 - 2}(\frac{1}{n_1} + \frac{1}{n_2})}}$$

Una vez calculado el valor de T, debemos probar que sea estrictamente menor al valor t-student $t_{m+n-2,\alpha/2}$. Para ello, creamos una función denominada **contrastar_hipotesis** que devuelve tanto el valor del estadístico como el valor t-Student además del p-valor mediante la función pt, el cual devuelve el valor correspondiente de la función de distribución:

```
contrastar_hipotesis <- function(x, y, columna, confianza) {
  n.1 <- nrow(x)
  n.2 <- nrow(y)
  var.1 <- var(x[, columna]) * ((n.1 - 1 )/ n.1)
  var.2 <- var(y[, columna]) * ((n.2 - 1 )/ n.2)
```

```
  media.1 <- mean(x[, columna])
  media.2 <- mean(y[, columna])
  s.c <- (n.1 * var.1 + n.2 * var.2) / (n.1 + n.2 - 2)
  estadistico.t <- (media.1 - media.2) / sqrt(s.c *  (1/n.1 + 1/n.2))
  t.student <-  abs(qt(c((1 - confianza) / 2), df = n.1 + n.2 - 2))
  p.valor <- 2 * pt(estadistico.t, n.1 + n.2 - 2, lower = FALSE)
  print(data.frame("T" = estadistico.t, "t.Student" = t.student,
        "p-valor" = p.valor, "alfa" = 1 - confianza),
        row.names = FALSE)
}
```

Una vez definida dicha función, realizamos la prueba con las dos submuestras. En función cada una, el nivel de significación será diferente ($\alpha = 0.05$ para un 95 %, $\alpha = 0.20$ para un 80 %):

```
# Con un 95 % de confianza
contrastar_hipotesis(datos.estatura.primeros.15, datos.estatura.ultimos.15, "estatura", 0.95)
```

```
##         T t.Student   p.valor alfa
##  1.132018  2.048407 0.2672288 0.05
```

```
# Con un 80 % de confianza
contrastar_hipotesis(datos.estatura.primeros.15, datos.estatura.ultimos.15, "estatura", 0.80)
```

```
##         T t.Student   p.valor alfa
##  1.132018  1.312527 0.2672288  0.2
```

Analizando los resultados obtenidos (coincidiendo tanto el estadístico T como el p-valor con lo obtenido en la función *t.test*), en ambos casos $T < t_{m+n+2,\alpha/2}$, además de que $p - valor > \alpha$ (dicho de otro modo, no cae dentro de región de rechazo de H_1) por lo que **no podemos rechazar H_0 tanto para un nivel de significación del 5 % como incluso del 20 %. Generalizando, los datos muestrales (para un $\alpha < 0.2672$, esto es, el p-valor) no contienen suficiente evidencia para rechazar la hipótesis nula de que la diferencia de ambas medias sean cero, es decir, iguales.**