Práctica Estadística

Fernández Hernández, Alberto

Ejercicio 1.

a) 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* calcularemos 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, por debajo 1.26 metros se encuentra el 50 % de los alumnos de la muestra y por encima el 50 % restante.

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*:

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

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

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

rango
```

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

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 $\frac{(n-1)}{n}$:

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

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

```r
# 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 $\sqrt{\frac{(n-1)}{n}}$:

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

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

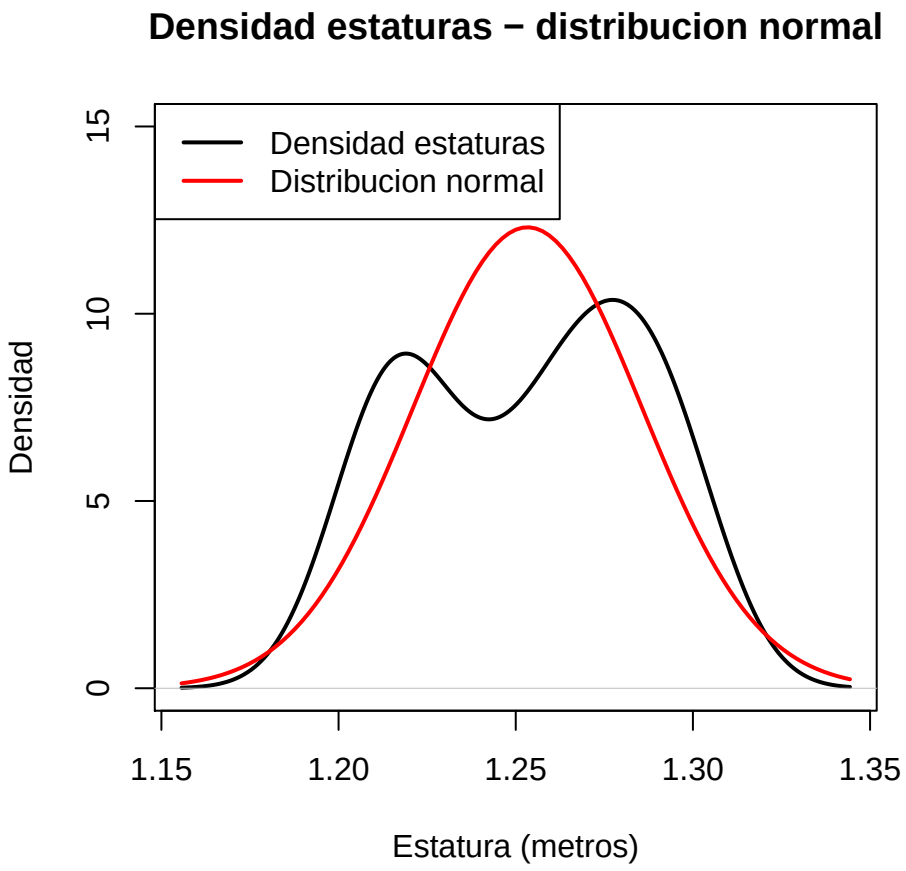
```r
# Desviacion tipica obtenida a partir de la varianza
desv.tipica <- sd(datos.estatura[, "estatura"]) *
                    sqrt((nrow(datos.estatura) - 1 )/ nrow(datos.estatura))
desv.tipica
```

```
## [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 obtenidos anteriormente.

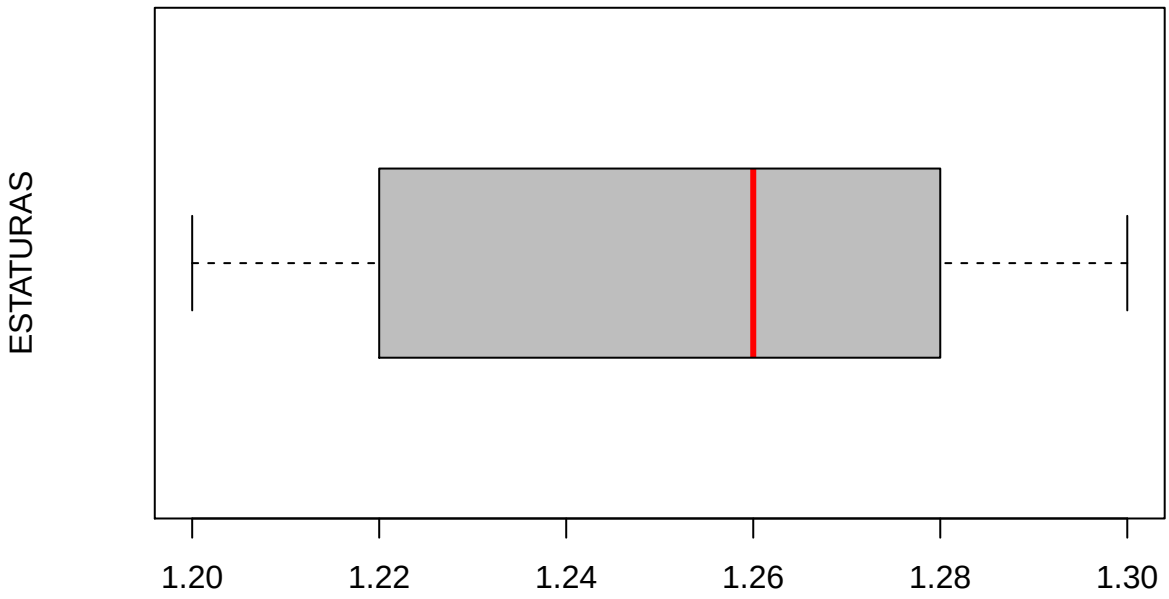
Por el contrario, la densidad al aproximarse a la media (1.25) es menor, por lo que no parece ser **un valor representativo de la muestra**:

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



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

```
boxplot(datos.estatura[, "estatura"], medcol = "red",
        col = "grey", las = 1, ylab = "ESTATURAS", horizontal = TRUE)
```




```
# 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 %)**. 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.

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

```
coef.var.pearson <- desv.tipica / 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).

b) 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, para mayor facilidad, se agruparán en función de los cuartiles, además de los valores máximo y mínimo. 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:

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

Esto permite remarcar los dos últimos intervalos, donde se concentran el mayor número de muestras, esto es, entre 1.27 y 1.30. Para ello, la función *cut* de R nos permite dividir el rango de un vector en intervalos de longitud N pasado como parámetro. 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.

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
```

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 comprobar si los datos de ambas muestras están distribuidos normalmente, una primera aproximación es mediante un gráfico de cuantiles o **Gráfico Q-Q (*Quantile-Quantiles*)** , así como un gráfico de densidad, por medio de una función denominada **mostrar_graficos** :

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

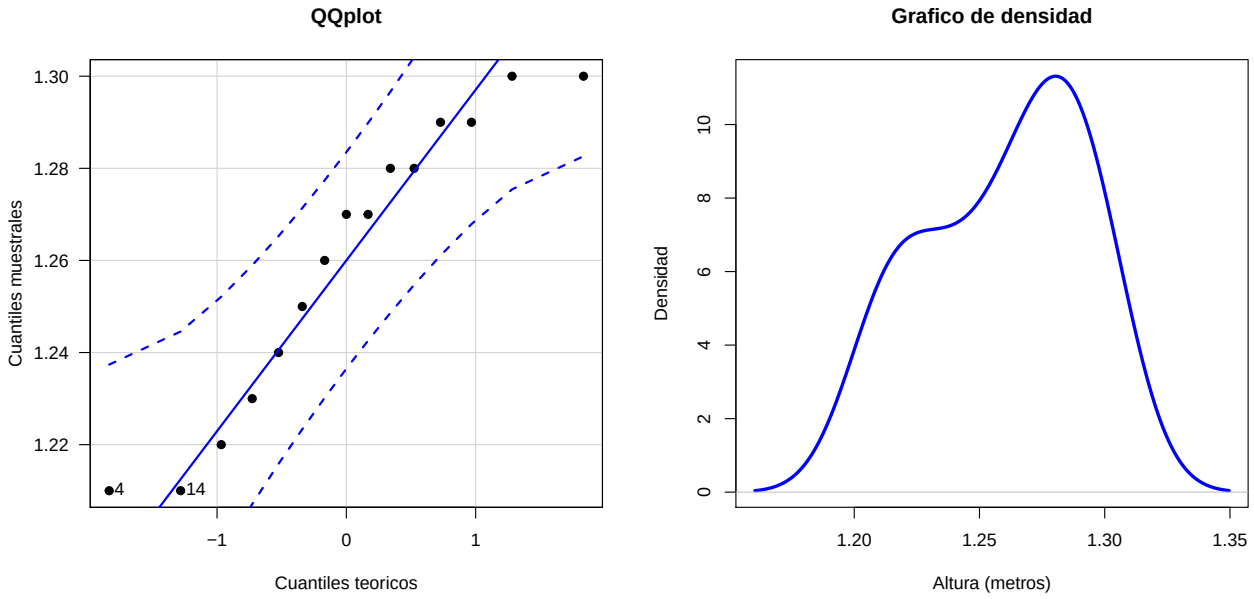
```
## Warning: package 'car' was built under R version 3.6.2
```

```
## Loading required package: carData
```

```r
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')
  plot(density(datos[, columna]), lwd = 3, col = 'blue',
       xlab = 'Altura (metros)', ylab = 'Densidad',
       main = 'Grafico de densidad')
}
```

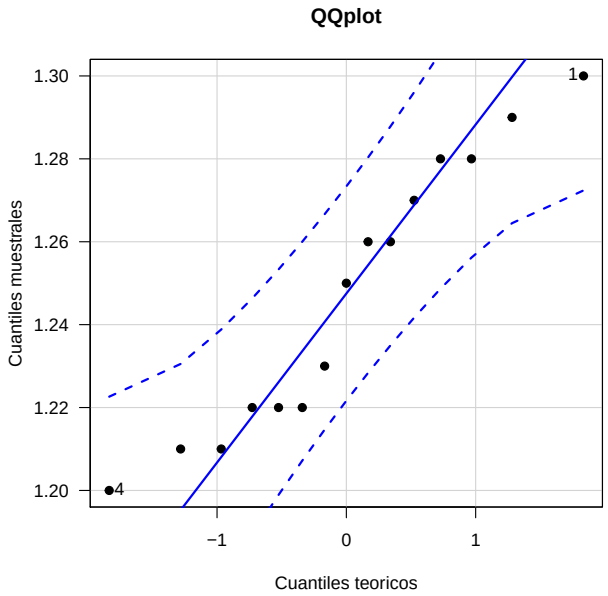
Una vez definida la función, mostramos los gráficos correspondientes a ambas submuestras, tanto con el primer subconjunto:

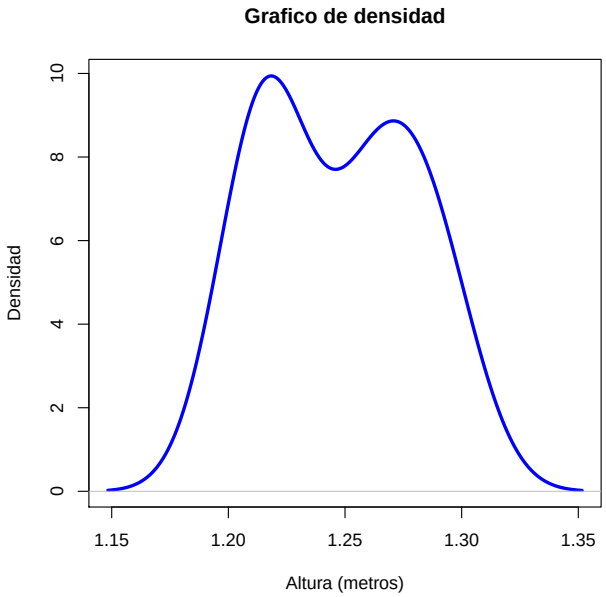
```r
# Primeros 15 datos
mostrar_graficos(datos.estatura.primeros.15, "estatura")
```



Como el segundo:

```r
# Primeros 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 cómo 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 que con respecto a la primera. No obstante, tampoco 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, plantearemos 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:

```
# 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 $\alpha = 0.05$, lo que significa que **estadísticamente no existe evidencia en contra de que ambas submuestras provengan de una distribución normal** . Por ello, con vistas a este apartado asumiremos que los datos se distribuyen normalmente.

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.

Comenzando con la varianza, proponemos la siguiente hipótesis nula:

$$H_0 : \sigma_1 = \sigma_2$$
$$H_1 : \sigma_1 \neq \sigma_2$$

O dicho de otro modo:

$$H_0 : \frac{\sigma_1}{\sigma_2} = 1$$
$$H_1 : \frac{\sigma_1}{\sigma_2} \neq 1$$

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)
  cociente.var <- min(var.1, var.2) / max(var.1, var.2)
  lim.inf <- cociente.var * (1 / qf(1- (1 - confianza) / 2, df1 = n.1 - 1, df2 = n.2 - 1))
  lim.sup <- cociente.var * (1 / qf((1 - confianza) / 2, df1 = n.1 - 1, df2 = n.2 - 1))
  c(lim.inf, lim.sup)
}
```

Por tanto, para un valor de confianza del 95 % tendremos el siguiente intervalo:

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

```
## [1] 0.3105869 2.7555215
```

Para comprobar que el resultado obtenido es correcto, R dispone de una función predefinada 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 cómo el intervalo resultante es idéntico al obtenido en la función **contrastar_varianzas** . 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, lo que supone que ambas varianzas pueden ser iguales ($\frac{\sigma_1}{\sigma_2} = 1$).
2. Por otro lado, el p-valor obtenido (0.8863) el cual es significativamente superior al valor $\alpha = 0.05$.

Por ello, se concluye que los resultados obtenidos **no muestran evidencia en contra de la homogeneidad de las varianzas (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:

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

Por ello, una primera aproximación sería, al igual que ocurría con las varianzas, implementar el intervalo de confianza a través de una función denominada **intervalo_confianza**, cuyos parámetros serán ambas submuestras (x,y), el nombre de la columna con las medidas así como el valor de confianza. Para obtener los valores correspondientes en la tabla t de Student, R dispone de una función específica denominada *qt* , cuyos parámetros son el nivel de significación α para cada cola, así como el número de grados de libertad (df):

```
intervalo_confianza <- 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])
  aux <- abs(qt(c((1 - confianza) / 2), df = n.1 + n.2 - 2)) *
          sqrt((n.1 * var.1 + n.2 * var.2) *
          (1/n.1 + 1/n.2))/sqrt(n.1 + n.2 - 2)
  c(media.1 - media.2 - aux, media.1 - media.2 + aux)
}
```

Una vez definida la función, realizamos la prueba correspondiente para ambas submuestras:

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

```
## [1] -0.01079358  0.03746024
```

Para comprobar el intervalo resultante, 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 (σ) sean iguales, marcamos el parámetro *var.equal* a TRUE, dado que por defecto R asume que las varianzas no son iguales (FALSE):

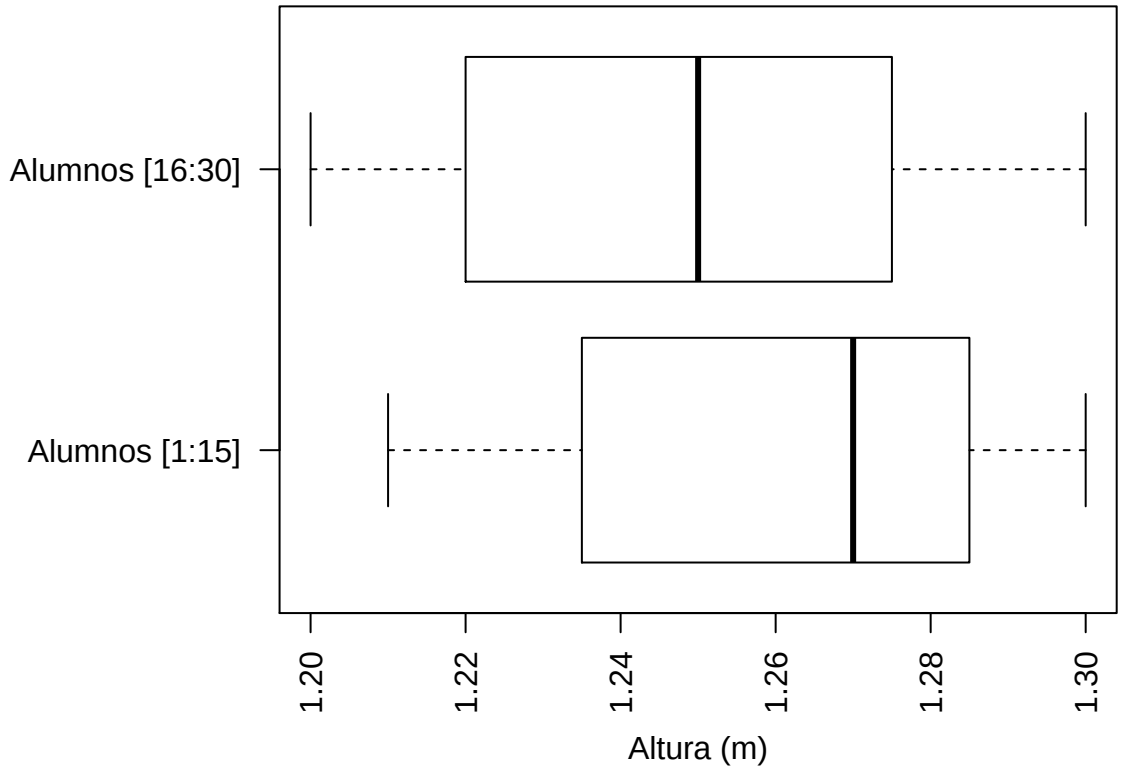
```
# Por defecto, conf.level esta a 0.95
t.test(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"],
       var.equal = TRUE, conf.level = 0.95)
```

```
##
##  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
```

Como podemos observar en los intervalos resultantes (los cuales coinciden), abarcan 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 puede suponer que la media de ambas poblaciones puedan ser iguales. 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: la diferencia entre ambas medias es distinto de cero.

Dicho intervalo puede verse de forma gráfica, a través de un diagrama de caja y bigotes, donde la diferencia entre ambas medias es de apenas unos centímetros

```
par(xpd = TRUE, mar = par()$mar + c(0,7,0,0))
boxplot(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"],
        names = c("Alumnos [1:15]", "Alumnos [16:30]"),
        xlab = "Altura (m)",
        las = 2, horizontal = TRUE)
```



Incluso disminuyendo el valor de confianza de un 95 a un 70 % (por ejemplo), el intervalo de confianza es cada vez más cerrado:

```
t.test(datos.estatura.primeros.15[, "estatura"], datos.estatura.ultimos.15[, "estatura"],
       var.equal = TRUE, conf.level = 0.70)
```

```
##
##  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
## 70 percent confidence interval:
##  0.0008955006 0.0257711661
## sample estimates:
## mean of x mean of y
##  1.260000  1.246667
```

En este último intervalo, dado que los extremos son positivos, la media de población de la primera submuestra (los primeros 15 datos) es ligeramente mayor a la de la segunda submuestra. En cualquiera de los casos, el p-valor obtenido es superior a 0.05, por lo que no existiría evidencia estadística en contra de que ambas medias sean iguales.

De igual modo, podemos realizar el contraste de hipóstesis "a mano", planteando H_0 y H_1:

$$H_0 : \mu_1 = \mu_2$$
$$H_1 : \mu_1 \neq \mu_2$$

Para esta prueba, operaremos con un valor convencional α del 0.05 % (zona de rechazo de la hipótesis nula). Por ello, la zona central comprende el 95 % de los casos, es decir, la **zona de aceptación** de la hipótesis nula. Si la diferencia entre ambas medias se encuentra dentro de H_0 , aceptaremos la hipótesis nula. Por el contrario, si se encuentra dentro de la zona de rechazo, lo más conveniente es descartar H_0, ya que sería más viable asegurar que ambas muestras proceden de distintas poblaciones, con sólo un riesgo del 0.05. Por tanto, se trata de comprobar en qué zona se encuentra la diferencia de 0 entre ambas medias. Dado que en el intervalo de confianza hemos comprobado que el valor de 0 cae dentro del intervalo, lo confirmaremos con la siguiente prueba:

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

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**, la cual es una modificación de la función **intervalo_confianza** que devuelve tanto el valor del estadístico como el valor t-Student:

```
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)
  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))
  print(data.frame("T" = t, "t.Student" = t.student), row.names = FALSE)
}
```

Una vez definida dicha función, realizamos la prueba con las dos submuestras:

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

```
##         T t.Student
##  1.132018  2.048407
```

Analizando el resultado obtenido, $T < t_{m+n+2,\alpha/2}$, por lo que **no podemos rechazar H_0 para un nivel de significación del 5 %. Por tanto, asumimos que ambas muestras pertenecen a la misma población, es decir, presentan la misma media**. Del mismo modo que la función *t.test* anterior, el p-valor (0.2672) es superior al valor $\alpha = 0.05$, evidenciando de nuevo la hipótesis nula.