SAD_PEC_2

Marc Bañuls Tornero

13/11/2019

Contents

Ejercicio 1: Código de programación en R. Funciones

Lee el código fuente para cada una de las siguientes dos funciones genéricas, interpreta qué hacen y luego propón nombres mejores, más adecuados.

La primera función presentada, llamada a priori f1:

```
f1<- function(string, prefix) {
  substr(string,1,nchar(prefix)) == prefix
}
```

Esta función está hecha con la función de comprobar si una palabra tiene el mismo prefijo que el que se espera. Esto puede tener utilidad para buscar prefijos en un conjunto de palabras.

```
f1(string = c("interminable","interior","contraproducente","incapaz","inimaginable","desalentado"), pre
```

```
## [1]  TRUE  TRUE FALSE  TRUE  TRUE FALSE
```

A primera vista esto no parece útil en exceso, pero si queremos por ejemplo cuantas palabras tienen el prefijo en un texto extenso, se puede separar el texto por palabras y mediante la función for loop y if..else podemos contar fácilmente el número de palabras con este prefijo.
Podemos mejorar el nombre de la función y las variables ahora que sabemos su propósito:

```
detector_prefijo<- function(palabra, prefijo) {
  substr(palabra,1,nchar(prefijo)) == prefijo
}
```

También podríamos llamar a la función "buscador" o similares, ya que ésta también podría usarse para buscar palabras en un texto que empiecen por lo que hayamos definido.

La segunda función presentada, llamada a priori f2:

```r
f2<- function(x) {
  if (length(x)<=1) return(NULL)
  x[-length(x)]
}
```

Esta función tiene la utilidad de descartar el último valor de la variable. Si la variable tan solo tiene un valor, devuelve la variable con valor nulo (NULL). Por ejemplo, podemos descartar el último valor de la lista de la compra.

```r
comprar<-c("aguacates", "chocolate", "cereales", "plutonio")
f2(comprar)
```

```
## [1] "aguacates" "chocolate" "cereales"
```

Un buen nombre para la función podría ser:

```r
eliminar_ultimo<-function(x) {
  if (length(x)<=1) return(NULL)
  x[-length(x)]
}
```

Ejercicio 2: Generar funciones para estadística

Este ejercicio consiste en construir una función que calcule los estadísticos descriptivos básicos y un histograma de una variable continua. ¿La función sería diferente si la hacemos con una variable discreta? ¿Podríamos crear una función única para cualquier tipo de variable?

La función 'summary()' nos permite mostar un resumen de los parámetros estadísticos de un conjunto de datos, y la función 'hist()' realiza un histograma con el conjunto de datos que se de. Por lo tanto, juntando estas dos variables en una función podemos obtener lo que se pide en el enunciado. Para realizar la prueba de la función con una variable discreta y otra continua, las creamos previamente:

```r
discreta<-c(1,1,4,6,2,6)
continua<-c(1.1123,1.5534,6.3324,7.2213,9.3576,5.2231)
```

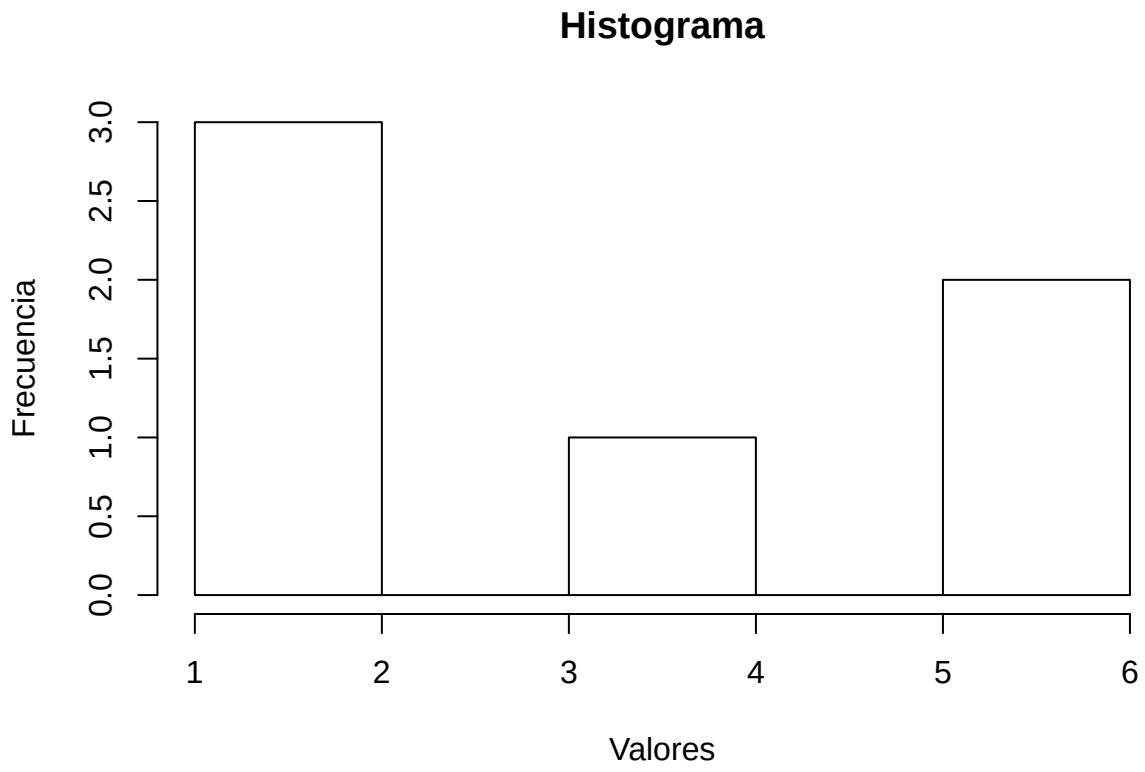
Realmente las variables continuas pueden tener valores tanto enteros como decimales. La función a crear posteriormente entonces no debería verse afectada si utilizamos una variable discreta en vez de una continua en la función, ya que los valores enteros también son usados en variables continuas.
Igualmente, para que la función trabaje siempre con variables continuas, podemos transformar siempre la variable a variable continua en la propia función.

Creamos la función:

```r
estadistico_hist<-function(x){
  hist(as.numeric(x), main = "Histograma", xlab = "Valores", ylab = "Frecuencia")
  summary(as.numeric(x))
}
```

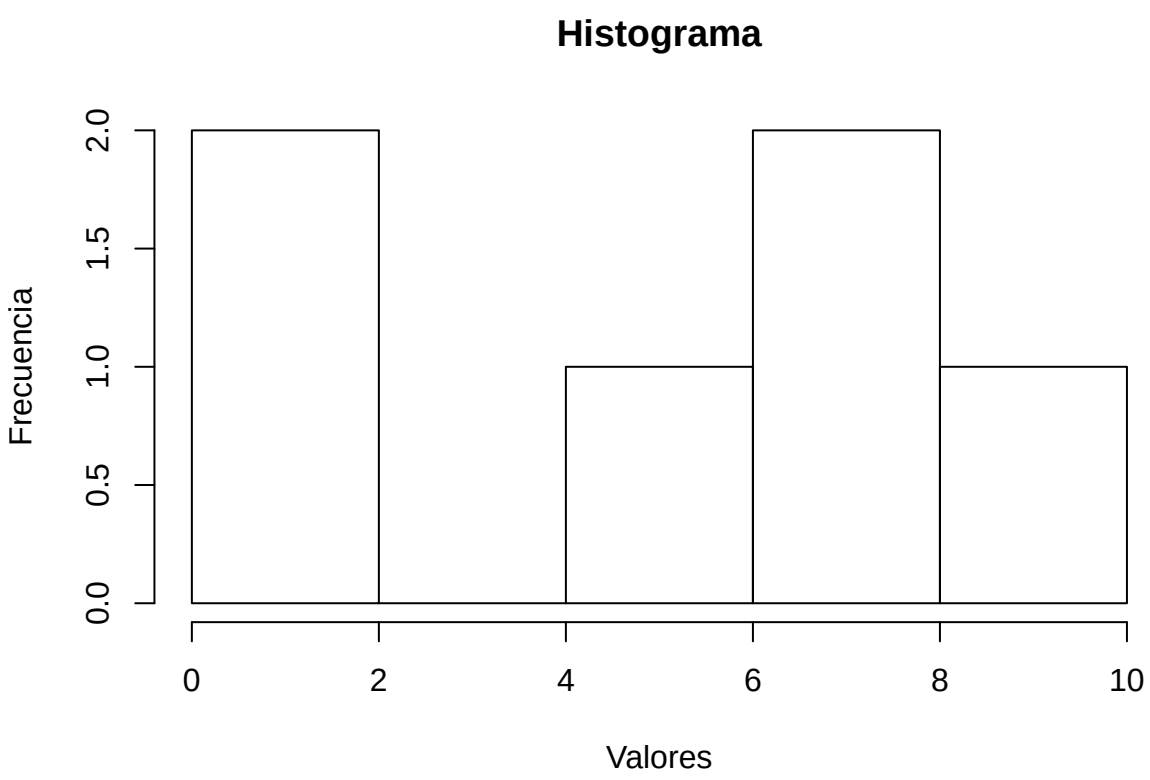
Comprobamos la función:

```
estadistico_hist(discreta)
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.250   3.000   3.333   5.500   6.000
```

```
estadistico_hist(continua)
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.112   2.471   5.778   5.133   6.999   9.358
```

En este ejercicio no habría una diferencia significativa entre tratar la variable como discreta o continua, ya que el histograma contaría los valores de igual manera como numéricos. En todo caso, donde mayor utilidad puede tener la función es en variables categóricas (interpretadas como factores en R) donde la función interpretaría esta variable como contínua para permitir su descripción estadística o realización del histograma de sus valores.

Ejercicio 3: Modelo de comportamiento de Ricker

La siguiente función nos muestra el comportamiento de una población bajo el modelo de Ricker:

$$N_{t+1} = N_t exp[r(1 - \frac{N_t}{K})]$$

Este modelo es usado ampliamente en ecología de poblaciones, particularmente en estudios demográficos de peces. El objetivo de este ejercicio es simular el modelo con respecto a la tasa de crecimiento r y el número inicial de individuos en la población N_0 (la capacidad de carga K es usualmente fijada en 1, y usaremos este como un valor por defecto). El tiempo (t) será igual a 100.

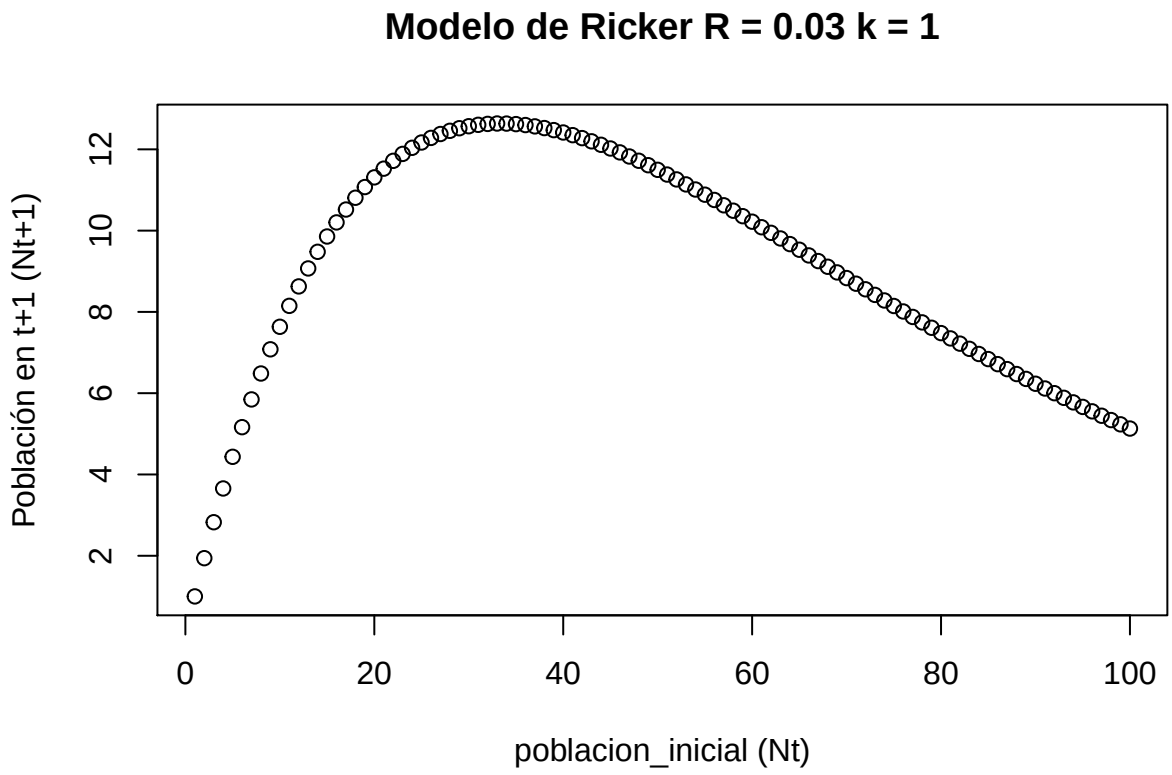
a)Debemos generar una función que nos permita mostrar los resultados de la fórmula del enunciado y una gráfica de individuos en función del tiempo.

Definimos la función en función de la población n, la tasa de crecimiento r, y la capacidad de carga k (que fijaremos en 1):

```r
ricker<- function(nt,r,k){
  n_t_mas_1<-nt * exp(r * (1-(nt/k)))
  return(n_t_mas_1)
}
```

Ahora podemos utilizar esta función para representar como cambia la población inicial en las siguientes generaciones teniendo en cuenta la tasa de crecimiento r.

```r
poblacion_inicial <- seq(1, 100)
modelo_ricker <- ricker(poblacion_inicial,r = 0.03, k = 1)
plot(poblacion_inicial,modelo_ricker, xlab = 'poblacion_inicial (Nt)', ylab = 'Población en t+1 (Nt+1)'
```



Para obtener una gráfica representativa de un crecimiento poblacional dependiente de la densidad, hemos ido cambiando la tasa de crecimiento junto el número de individuos. Observamos que representando N_{t+1} en función de N_t en las primeras generaciones donde la población es pequeña (entre 0 y 30 individuos) se observa que la población aumentará en la próxima generación. Cuando la población se llega a tener una alta densidad de población (superior a 30 individuos) se llega al punto de equilibrio o plateau. En niveles altos de población, los mecanismos dependientes de la densidad (competencia) reducen el tamaño de la población en la siguiente generación.

b) Existe una funión para implementar el modelo de Ricker ya implementada en R?

Investigando entre los paquetes de R hemos el paquete 'datasets v3.6.1' que contiene la función 'ricker', la cual realiza la ecuación del modelo Ricker a partir de los valores de spawners= número de individuos (N), a = parámetro de productividad (r) y b = parámetro de densidad (K). También hemos encontrado el paquete 'pomp' (siglas de Partially Observed Markov Processes) que sirve para implementar varios modelos POMP, simularlos y tratar con sus datos. En este paquete existen varias funciones y, entre ellas, se encuentran

funciones para realizar simulaciones y gráficas del modelo de ricker o interpretar los datos del modelo.

Ejercicio 4: Aplicaciones de distribuciones de probabilidad en R

La probabilidad que, en un hospital, el próximo nacimiento sea niño es 0.52.

a) Calculemos la probabilidad de que haya más de 540 niños en los próximos 1000 nacimientos

Podemos definir la probabilidad de nacimiento de un niño como una distribución binomial con p = 0.52 (probabilidad de nacer niño) y q = 0.48 (probabilidad de nacer niña). Por lo tanto, podemos utilizar la función binomial de probabilidad acumulada para calcular la probabilidad (teniendo en cuenta que queremos la probabilidad de la cola derecha o 'upper tail'):

```
pbinom(540, 1000, 0.52, lower.tail = FALSE)
```

```
## [1] 0.09715473
```

El resultado es una probabilidad del 9.7%.

b) ¿Y si queremos pensar en 500 hospitales? Crea una pequeña función usando replicate() para buscar este valor.

En este caso podemos simular cuantos niños nacen en 500 hospitales distintos y calcular a partir de esto la probabilidad de que haya más de 540 niños de 1000 nacimientos. Para ello utilizaremos la función replicate(). Utilizaremos la función set.seed() para obtener unos valores constantes en la función.

```
set.seed(123)
nacimientos_hospitales <- replicate(500, {rbinom(1, 1000, 0.52)})
nacimientos_hospitales
```

```
##   [1] 532 530 488 524 509 559 507 544 510 536 522 512 514 511 514 534 536
##  [18] 513 527 513 517 524 535 532 525 501 539 492 511 527 549 505 522 507
##  [35] 530 526 529 487 502 535 525 511 530 515 503 528 526 506 489 548 546
##  [52] 525 506 500 513 535 516 536 520 513 517 481 519 525 514 511 532 513
##  [69] 510 527 500 506 508 502 521 541 509 497 509 520 515 550 508 526 518
##  [86] 484 489 522 532 540 509 530 512 518 506 532 541 517 538 530 516 505
## [103] 497 521 538 539 516 533 529 533 515 501 521 525 526 519 515 516 535
## [120] 506 522 482 538 536 509 530 524 539 509 545 541 529 506 513 515 517
## [137] 548 530 516 531 511 536 516 534 501 521 552 522 505 503 509 480 530
## [154] 545 501 523 502 532 520 500 529 512 517 511 508 526 532 537 517 528
## [171] 525 521 522 478 523 511 547 516 550 526 532 526 513 561 499 515 503
## [188] 518 541 557 537 529 531 521 526 543 508 536 525 507 511 538 517 481
## [205] 504 502 532 533 542 525 504 491 533 536 528 490 536 527 467 521 548
## [222] 510 514 515 509 509 511 561 501 509 542 524 544 518 507 518 498 526
## [239] 542 531 504 525 517 544 525 511 522 514 506 497 548 533 511 528 523
## [256] 524 506 555 501 510 511 506 529 483 497 545 507 500 531 520 538 540
## [273] 517 494 551 518 532 530 535 513 522 522 555 518 502 536 540 526 516
## [290] 496 508 538 511 489 495 536 525 519 523 513 557 501 506 539 499 526
## [307] 515 512 513 547 525 507 525 508 513 510 520 524 516 513 527 529 520
## [324] 522 505 537 522 526 518 515 515 510 526 525 504 509 498 527 509 515
## [341] 522 518 517 535 541 511 537 490 526 525 524 512 510 516 539 513 540
## [358] 532 532 510 511 498 524 540 501 528 545 522 508 510 495 503 499 513
## [375] 528 511 569 545 502 505 500 516 514 519 535 509 542 545 493 510 515
## [392] 518 507 531 544 511 505 522 508 537 510 512 523 531 552 522 539 502
## [409] 533 524 507 482 515 521 512 511 499 534 492 515 511 496 498 535 530
```

```
## [426] 528 517 522 519 525 523 506 521 503 549 527 540 513 532 531 510 528
## [443] 544 515 507 524 506 522 508 526 527 503 509 531 515 516 511 502 503
## [460] 497 530 544 518 540 499 483 511 508 512 507 519 536 516 523 524 519
## [477] 513 534 508 523 556 543 510 514 531 526 517 547 515 533 521 551 512
## [494] 504 517 548 523 511 509 507
```

Con estos valores podemos buscar los hospitales que tienen más de 540 niños de cada 1000 nacimientos.

```
mayor_540_nacimientos <- sum(540 < nacimientos_hospitales)
```

Ahora podemos encontrar la proporción de nacimientos entre el total de hospitales para observar la probabilidad obtenida:

```
probabilidad_nacimientos <- mayor_540_nacimientos / 500
probabilidad_nacimientos
```

```
## [1] 0.098
```

Observamos que este valor es similar al valor obtenido anteriormente en el apartado a), ya que la distribución utilizada en la generación de los 500 resultados es la misma en el apartado anterior. De esta manera, la repetición de los 500 valores ha dado lugar a una distribución en la que el 10% de hospitales han tenido más de 540 nacimientos de niños de un total de 1000 nacimientos en cada hospital.

Ejercicios 5 y 6: Distribuciones y cálculo de probabilidad con R o R-Commander

Ejercicio 5:

En una Universidad de California se estudian un determinado tipo de aves. Se comprueba que la longitud de las alas extendidas, X, es una variable aleatoria que se distribuye aproximadamente según una curva Normal, de media 110 cm. y desviación típica 4 cm. Elegida un ave al azar y suponiendo que las longitudes se distribuyen normalmente, calcular:

a) La probabilidad de que la longitud de las alas esté comprendida entre 110 y 115 cm.

Teniendo en cuenta que se trata de una distribución normal, podemos buscar la probabilidad de que las alas sean menores a 100 y la probabilidad de que las alas sean menores a 115 cm. Entonces, la diferencia de estos dos valores resultará en la probabilidad de que la longitud de las alas se encuentre entre 100 y 115 cm. Utilizamos la función de probabilidad acumulada en una normal por la cola izquierda (lower.tail = TRUE):

```
normal_110<- pnorm(110, mean = 110, sd = 4, lower.tail = TRUE)
normal_115<- pnorm(115, mean = 110, sd = 4, lower.tail = TRUE)
p_110_115<- normal_115 - normal_110
p_110_115
```

```
## [1] 0.3943502
```

La probabilidad de que la longitud de las alas se encuentre entre 100 y 115 cm. es del 39.4%.

b) La probabilidad de que la longitud de las alas sea mayor que 105 cm.

Al estar buscando la probabilidad de que la longitud sea mayor y no menor a 105 cm. utilizaremos la función de la probabilidad para la cola derecha (lower.tail = FALSE).

```
normal_mayor_105<- pnorm(105, mean = 110, sd = 4, lower.tail = FALSE)
normal_mayor_105
```

```
## [1] 0.8943502
```

La probabilidad de que la longitud de las alas sea mayor a 105 cm. es del 89.4%.

d) La longitud mínima del 30% de las alas que más miden

El enunciado de este apartado se traduce a la búsqueda del percentil 30 por la cola derecha de la distribución normal, o lo que es lo mismo, el percentil 70 por la cola izquierda. Utilizamos entonces la función del cálculo de percentiles:

```
percentil_70<- qnorm(0.7, mean = 110, sd = 4)
percentil_70
```

```
## [1] 112.0976
```

La longitud mínima del 30% de las alas que más miden (o percentil 70) es de 112.09 cm.

e) Quince longitudes aleatorias que sigan dicha distribución

Con la función rnorm() generamos el número de longitudes deseado con la distribución deseada introduciendo la media y desviación típica de dicha distribución:

```
longitudes<- rnorm(15, mean = 110, sd = 4)
longitudes
```

```
##  [1] 103.9690 110.1044 108.7343 109.5906 105.2738 111.9946 105.8442
##  [8] 109.0951 111.5257 106.8659 112.3320 104.7340  98.7609 111.8599
## [15] 113.3622
```

Para comprobar que las longitudes generadas siguen la distribución podemos medir su media y desviación típica muestral.

```
mean(longitudes)
```

```
## [1] 108.2698
```

```
sd(longitudes)
```

```
## [1] 4.000479
```

Observamos que su media y desviación se acercan significativamente a la media y desviación típica de la distribución aunque no concuerdan exactamente. A mayor cantidad de muestras generadas, más se acercaran la media y desviación típica muestrales a la de la distribución.

Ejercicio 6:

Definid una variable aleatoria que cuente el número de supervivientes que ingresan en un Hospital con una enfermedad muy grave. Se ha determinado que produce una mortalidad del 75% en los bebés lactantes. Si en este Hospital ingresaron en un brote muy fuerte 20 lactantes con la enfermedad:

a) ¿Qué distribución sería la escogida para este caso? ¿Qué parámetros tendría?

En este caso se escogemos una distribución binomial con una probabilidad de que el bebé lactante sobreviva p = 0.25 y el tamaño de la muestra son los 20 lactantes ingresados en el hospital con la enfermedad.

b) Busca la probabilidad que sobrevivan todos los pacientes. ¿Qué nos dice esta probabilidad?

Utilizamos la función dbinom() para hallar la probabilidad de que 20 de los 20 niños sobrevivan.

```
p_todos_bebes<- dbinom(20, size = 20, prob = 0.25)
p_todos_bebes
```

```
## [1] 9.094947e-13
```

Con el resultado obtenido, la probabilidad de que sobrevivan todos los pacientes es similar al 0%, por lo que podemos concluir que es prácticamente imposible que sobrevivan todos los bebés.

c) Busca la probabilidad de que sobrevivan la mitad de los pacientes

En este caso utilizaremos la misma función que en el apartado b) pero ahora teniendo en cuenta que queremos saber la probabilidad de que 10 de los 20 pacientes sobrevivan:

```
p_mitad_bebes<- dbinom(10, size = 20, prob = 0.25)
p_mitad_bebes
```

```
## [1] 0.009922275
```

La probabilidad de que sobrevivan 10 de 20 pacientes es del 0.1%. Esto no indica que sea imposible que vayan a sobrevivir la mitad de los pacientes, ya que esta no es la probabilidad acumulada, si no la probabilidad de que exactamente 10 bebés sobrevivan. Si observamos la probabilidad de que sobrevivan más de la mitad de bebés enfermos:

```
p_mas_mitad_bebes<- pbinom(10, size = 20, prob = 0.25, lower.tail = FALSE)
p_mas_mitad_bebes
```

```
## [1] 0.003942142
```

De esta manera se puede concluir que también es improbable que sobrevivan más de la mitad de los pacientes.

Ejercicio 7: ANOVA con R y RCommander:

En un tratamiento contra el asma se seleccionaron 40 enfermos de características similares y ataques de asma muy frecuentes. A cada enfermo se le administró un tratamiento: P, A, B, AB, al azar, formando 4 grupos. El grupo P tomó placebo, el grupo A tomó un fármaco "A", el grupo B un fármaco "B" y el grupo AB una asociación entre "A" y "B". Para valorar la eficacia de los tratamientos, se registró el descenso de los ataques de asma desde el estado basal (inicio del tratamiento) hasta el estado al cabo de una semana de tratamiento. Los resultados, después de registrarse algunos abandonos, fueron los siguientes:
P: 10, 0, 15, -20, 0, 15, -5
A: 20, 25, 33, 25, 30, 18, 27, 0, 35, 20 B: 15, 10, 25, 30, 15, 35, 25, 22, 11, 25 AB: 10, 5, -5, 15, 20, 20, 0, 10

a) ¿Hay diferencias entre tratamientos? (Hipótesis global)

Antes que todo debemos tener en cuenta que estos datos son independientes entre ellos.

Para observar una posible diferencia entre tratamientos primero ubicaremos los valores obtenidos en variables (añadiendo valores NA para los que dejaron el tratamiento) para posteriormente definir el conjunto de todos los datos en un data frame:

```
P<- c(10, 0, 15, -20, 0, 15, -5, NA, NA, NA)
A<- c(20, 25, 33, 25, 30, 18, 27, 0, 35, 20)
B<- c(15, 10, 25, 30, 15, 35, 25, 22, 11, 25)
AB<- c(10, 5, -5, 15, 20, 20, 0, 10, NA, NA)
tratamientos<- data.frame(P, A, B, AB)
head(tratamientos)
```

```
##     P  A  B AB
## 1  10 20 15 10
## 2   0 25 10  5
## 3  15 33 25 -5
## 4 -20 25 30 15
```

```
## 5   0 30 15 20
## 6  15 18 35 20
```

Ahora que tenemos el dataframe, podemos apilar todos los valores con su nombre de grupo para poder interpretar los tipos de tratamiento como factores para facilitar los posteriores tratamientos. También definimos cada columna como una variable:

```
estudio<-stack(tratamientos)
estudio$ind<-factor(estudio$ind)
farmacos<-estudio$ind
valores<-estudio$values
```

Calculamos ahora la media y desviación típica para cada grupo:

```
aggregate(valores~farmacos, data = estudio, FUN = mean)
```

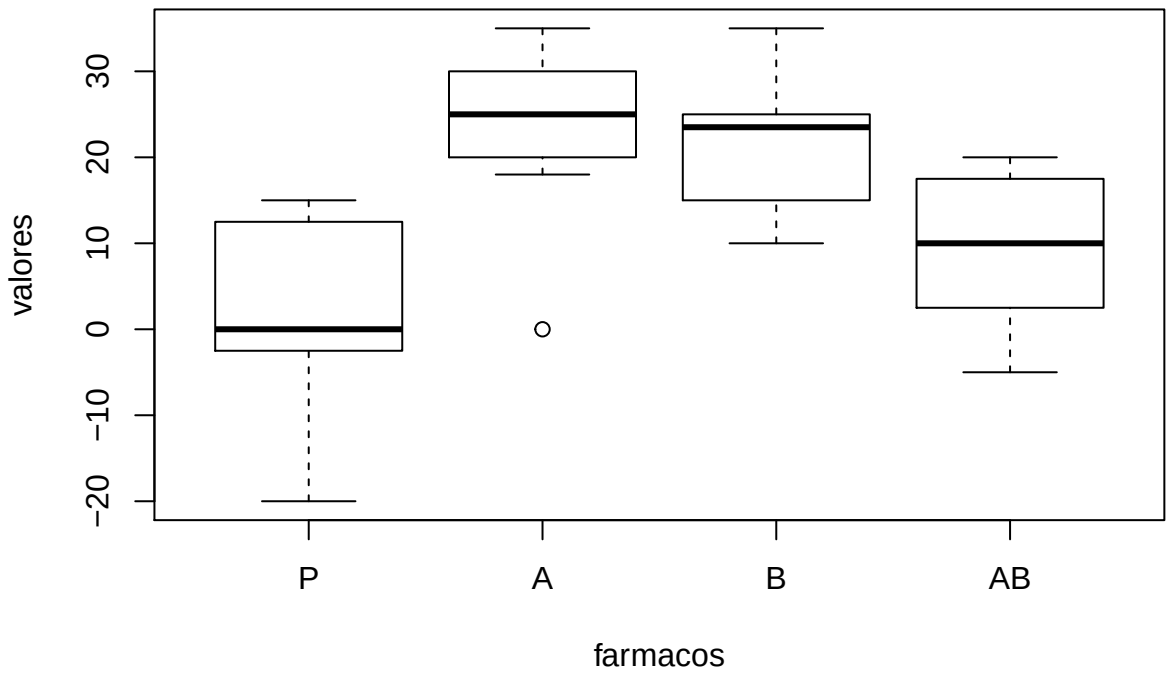
```
##   farmacos   valores
## 1        P  2.142857
## 2        A 23.300000
## 3        B 21.300000
## 4       AB  9.375000
```

```
aggregate(valores~farmacos, data = estudio, FUN = sd)
```

```
##   farmacos   valores
## 1        P 12.535663
## 2        A  9.933669
## 3        B  8.287206
## 4       AB  9.038608
```

Representamos gráficamente los valores de los distintos tipos de fármaco mediante un diagrama de cajas:

```
boxplot(valores~farmacos, data = estudio)
```

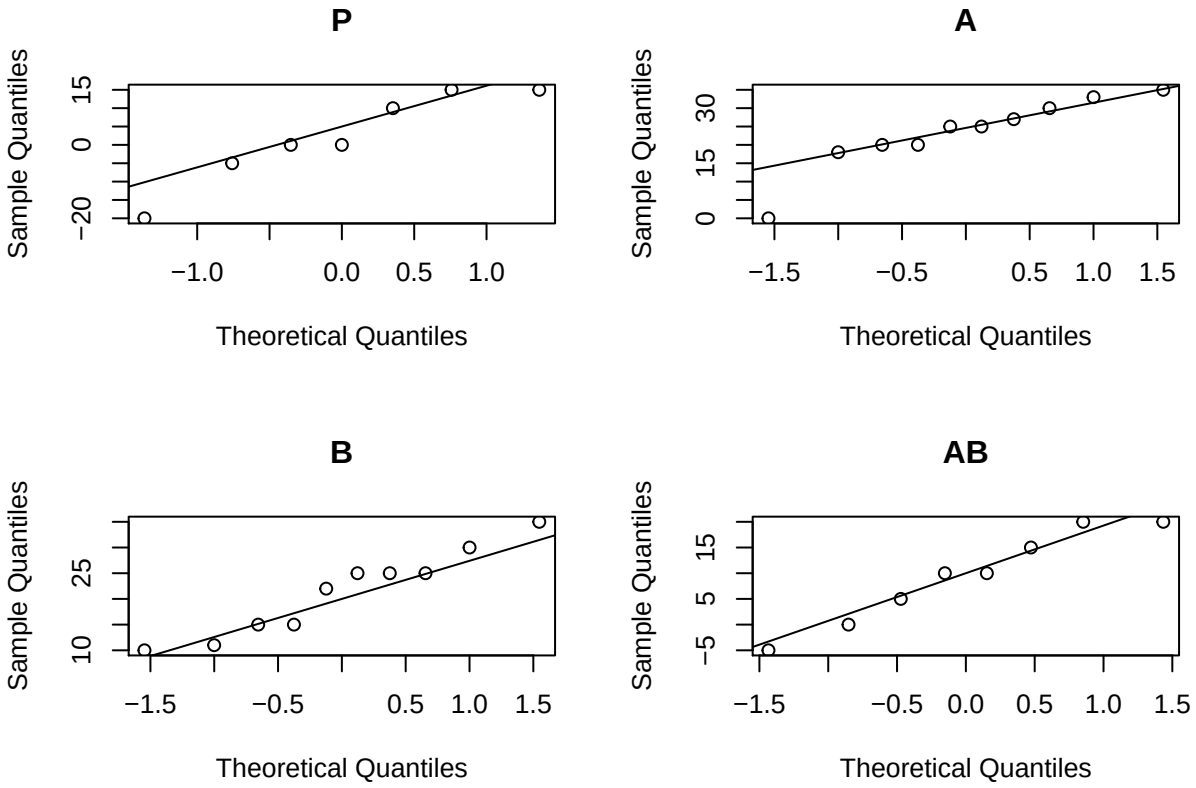


En el diagrama de cajas observamos visualmente una gran variación entre los valores de cada tipo de tratamiento, siendo los tratamientos A y B los que más se acercan entre ellos, y el tratamiento AB el que más se asemeja al tratamiento placebo P.

Debido a la baja cantidad de resultados las medias varían significativamente entre todos los tratamientos, aunque los tratamientos con A y con B tienen una media y desviación estándar similar entre ellos.

Para observar si las diferencias entre tratamientos son significativas, podemos realizar un test ANOVA. Para ello, debemos comprobar que se cumplen varias condiciones.
Una de las condiciones es que todos los grupos deben seguir una distribución normal, por lo que vamos a investigar esto en nuestros distintos fármacos:

```
par(mfrow= c(2,2))
qqnorm(subset(estudio$values,estudio$ind=="P"), main = "P")
qqline(subset(estudio$values,estudio$ind=="P"))

qqnorm(subset(estudio$values,estudio$ind=="A"), main = "A")
qqline(subset(estudio$values,estudio$ind=="A"))

qqnorm(subset(estudio$values,estudio$ind=="B"), main = "B")
qqline(subset(estudio$values,estudio$ind=="B"))

qqnorm(subset(estudio$values,estudio$ind=="AB"), main = "AB")
qqline(subset(estudio$values,estudio$ind=="AB"))
```



Con estos gráficos de los cuantiles observamos que todos los grupos parecen seguir una distribución normal. Para asegurarmos hacemos un test para determinar la normalidad de la distribución. Al tener menos de 50 observaciones aplicaremos el test *Shapiro-Wilk* ubicado en el paquete nortest:

```
require(nortest)
```

```
## Loading required package: nortest
```

```
by(data = estudio, INDICES = estudio$ind, FUN = function(x){shapiro.test(x$values)})
```

```
## estudio$ind: P
##
##  Shapiro-Wilk normality test
##
## data:  x$values
## W = 0.9093, p-value = 0.3911
##
## ------------------------------------------------------------
## estudio$ind: A
##
##  Shapiro-Wilk normality test
##
## data:  x$values
## W = 0.88389, p-value = 0.1446
##
## ------------------------------------------------------------
## estudio$ind: B
##
```

```
##  Shapiro-Wilk normality test
##
## data:  x$values
## W = 0.9385, p-value = 0.5364
##
## ------------------------------------------------------------
## estudio$ind: AB
##
##  Shapiro-Wilk normality test
##
## data:  x$values
## W = 0.94148, p-value = 0.6257
```

Observamos que cada grupo obtiene en el test un valor de p superior a 0.05, indicando que la hipótesis nula es correcta. Como la hipótesis nula de este test es que sí que hay normalidad en el grupo, podemos asegurar que en los cuatro grupos estudiados se sigue una distribución normal (siendo el grupo del fármaco A el que menos certeza hay que se distribuya normalmente debido a un p valor cercano a 0.05).
Otra suposición para poder realizar un test ANOVA es que los grupos deben tener una varianza común. Para ello, realizamos un test en estos grupos. Al saber que tratamos con grupos que siguen una distribución normal, podemos utilizar el *test de Bartlett*:

```
bartlett.test(estudio$values,estudio$ind)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  estudio$values and estudio$ind
## Bartlett's K-squared = 1.33, df = 3, p-value = 0.722
```

El *test de Bartlett* nos indica que la varianza es constante entre los grupos, ya que el valor de p es mayor a 0.05, aceptando la hipótesis nula (la hipótesis nula es que la varianza entre grupos es homogénea).

Al cumplirse las suposiciones necesarias, podemos realizar el test ANOVA entre el placebo y los distintos fármacos. Realizamos el test ANOVA:

```
anova_farmacos<-aov(estudio$values~estudio$ind)
summary(anova_farmacos)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## estudio$ind  3   2493   830.9   8.526 0.000282 ***
## Residuals   31   3021    97.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 5 observations deleted due to missingness
```

Como tenemos como hipótesis nula que no hay diferencias entre las medias de los grupos, y el valor de p es inferior a 0.05, (teniendo en cuenta que la hipótesis nula se confirma cuando p>0.05) el test nos confirma entonces que debemos descartar la hipótesis nula y seguir la alternativa, es decir, podemos asegurar entonces que sí hay diferencia entre los tratamientos.

b) Si la respuesta a la pregunta 1 es afirmativa realiza comparaciones de cada tratamiento con el placebo y determina si algún tratamiento difiere significativamente del placebo.

Como ya hemos confirmado la pregunta 1, ahora compararemos las diferencias entre el placebo con los otros tratamientos mediante esl *test de Tukey*:

```
TukeyHSD(anova_farmacos)
```

```
##   Tukey multiple comparisons of means
```

```
##     95% family-wise confidence level
##
## Fit: aov(formula = estudio$values ~ estudio$ind)
##
## $`estudio$ind`
##            diff        lwr        upr     p adj
## A-P   21.157143   7.953723 34.3605630 0.0007555
## B-P   19.157143   5.953723 32.3605630 0.0023316
## AB-P   7.232143  -6.634222 21.0985076 0.4994858
## B-A   -2.000000 -13.981909  9.9819085 0.9685408
## AB-A -13.925000 -26.633733 -1.2162668 0.0274523
## AB-B -11.925000 -24.633733  0.7837332 0.0721469
```

Observando los valores ajustados de p para cada comparación podemos determinar que fármacos difieren significativamente del placebo (P), es decir, que variables rechazan la hipótesis nula de que no hay diferencia entre los dos tratamientos.Los valores de p que tienen un valor menor a 0.05 (este valor se encuentra en "p adj") son la comparación entre P y el fármaco A y entre P y el fármaco B, indicando que existen diferencias significativas entre el placebo y estos fármacos (A y B). En cambio, el fármaco AB tiene un p valor superior a 0.05 y, por lo tanto, implica que se acepta la hipótesis nula y por lo tanto no hay diferencias significativas entre este fármaco y el placebo.

c) ¿Se cumplen suposiciones necesarias para poder realizar un test ANOVA?

Como ya hemos comentado en el apartado A, en estos datos se cumplen las suposiciones para poder realizar un test ANOVA (grupos normalmente distribuido y con varianzas constantes entre grupos).

d) ¿Podrías hacer este ejercicio con RCommander? En caso afirmativo, explica los pasos que tendrías que hacer para realizar el ejercicio con RCommander.

Para realizar el ejercicio en RCommander, primero debemos instalar y utilizar el paquete 'rcmdr'. A partir de la nueva interfaz, debemos importar la tabla de datos creada anteriormente llamada "estudio":

```
require(Rcmdr)
```

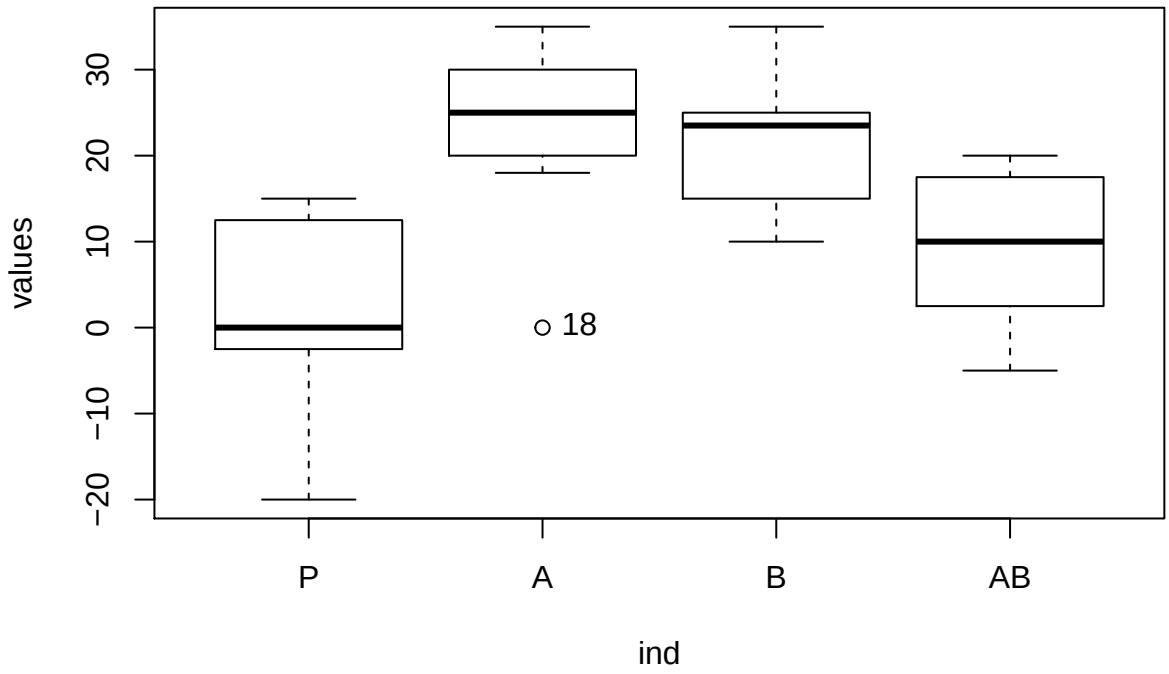
```
## Loading required package: Rcmdr

## Loading required package: splines

## Loading required package: RcmdrMisc

## Loading required package: car

## Loading required package: carData

## Loading required package: sandwich

## Loading required package: effects

## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.

## La interfaz R-Commander sólo funciona en sesiones interactivas
```

```
##
## Attaching package: 'Rcmdr'
```

```
## The following object is masked from 'package:base':
##
##     errorCondition
```

Cuando tenemos como conjunto de datos activo la variable estudio que contiene los 40 valores y los 4 grupos marcados como factores (variable categórica) podemos empezar a realizar el ejercicio.
Para observar el diagrama de cajas vamos a Gráficas<Diagrama de caja... y seleccionamos que queremos realizar el diagrama separado por grupos, obteniendo este código:

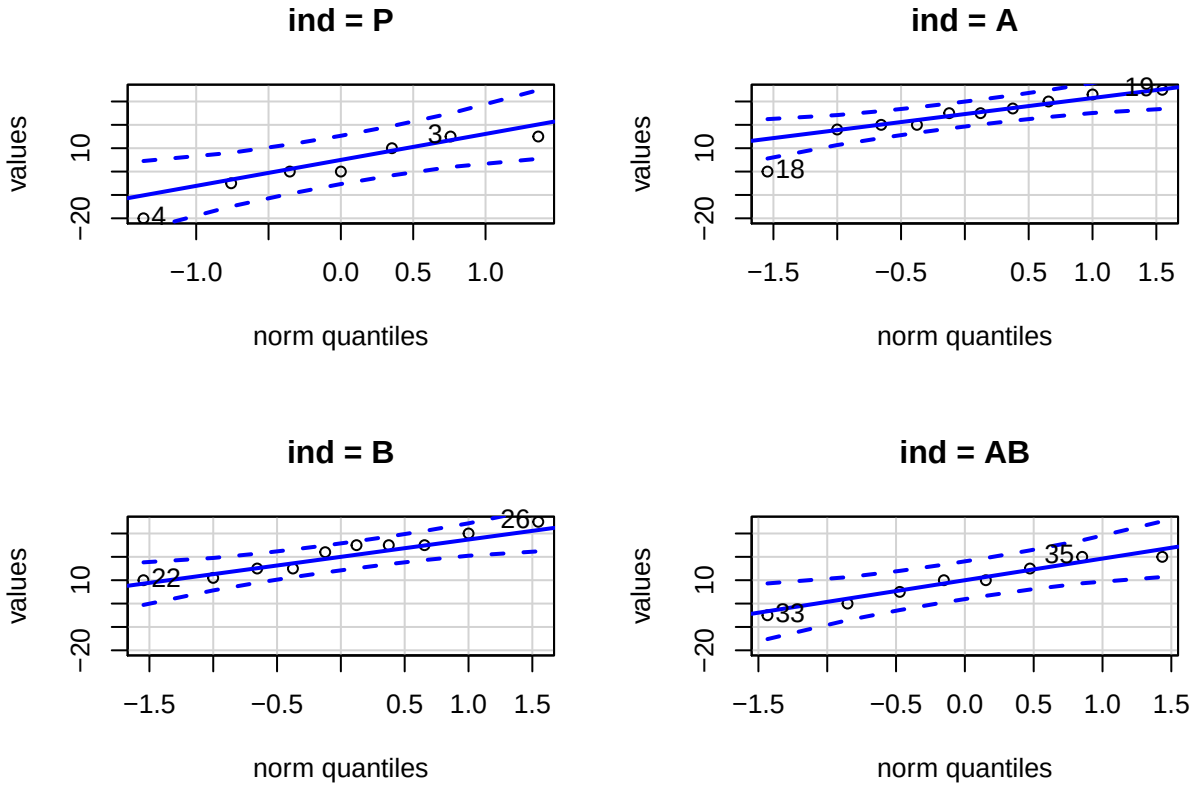
```
Boxplot(values ~ ind, data = estudio, id = list(method = "y"))
```




```
## [1] "18"
```

Para obtener las gráficas de los cuartiles de cada grupo para observar la normalidad de los valores, vamos a Gráficas<Gráfica de comparación de cuartiles... y seleccionamos que queremos realizar la gráfica por grupos del conjunto de datos activo. Esto resulta en la obtención de este código:

```
with(estudio, qqPlot(values, dist = "norm", id = list(method = "y", n = 2, labels = rownames(estudio)),
    groups = ind))
```



Para comprobar la normalidad de los grupos nos dirigimos a la pestaña Estadísticos<resumenes<Test de Normalidad. En la nueva pestaña elegimos "Test por grupos" y elegimos el test Shapiro-Francia. Esto nos da esta línea de código:

```
normalityTest(values ~ ind, test = "shapiro.test", data = estudio)
```

```
##
##  --------
##  ind = P
##
##  Shapiro-Wilk normality test
##
## data:  values
## W = 0.9093, p-value = 0.3911
##
##  --------
##  ind = A
##
##  Shapiro-Wilk normality test
##
## data:  values
## W = 0.88389, p-value = 0.1446
##
##  --------
##  ind = B
##
##  Shapiro-Wilk normality test
```

```
##
## data:  values
## W = 0.9385, p-value = 0.5364
##
##  --------
##  ind = AB
##
##  Shapiro-Wilk normality test
##
## data:  values
## W = 0.94148, p-value = 0.6257
##
##  --------
##
##  p-values adjusted by the Holm method:
##    unadjusted adjusted
## P  0.39107    1.00000
## A  0.14459    0.57837
## B  0.53642    1.00000
## AB 0.62573    1.00000
```

Observamos que los resultados obtenidos son los mismos a los realizados en el anterior ejercicio.

Ahora investigamos si la varianza entre grupos es constante, y para ello utilizamos el *test de Bartlett*. En RCommander debemos ir a Estadísticos<Varianzas<Test de Bartlett y aceptar cuando las variables estén donde corresponden. Esto nos genera este código:

```
with(estudio, tapply(values, ind, var, na.rm = TRUE))
```

```
##         P         A         B        AB
## 157.14286  98.67778  68.67778  81.69643
```

```
bartlett.test(values ~ ind, data = estudio)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  values by ind
## Bartlett's K-squared = 1.33, df = 3, p-value = 0.722
```

Obtenemos así el mismo resultado que en el ejercicio anterior (la varianza es constante entre grupos).

Ahora realizamos el test ANOVA para comprobar la diferencia de medias entre distintos grupos. Para ello vamos a Estadísticos<Medias<ANOVA de un factor. Introducimos el nombre del test que elijamos (en nuestro caso "anova_farmacos_rcmdr") y ejecutamos el código administrado:

```
anova_farmacos_rcmdr <- aov(values ~ ind, data = estudio)
summary(anova_farmacos_rcmdr)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## ind          3   2493   830.9   8.526 0.000282 ***
## Residuals   31   3021    97.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 5 observations deleted due to missingness
```

```
with(estudio, numSummary(values, groups = ind, statistics = c("mean", "sd")))
```

```
##         mean        sd data:n data:NA
```

```
## P   2.142857 12.535663      7       3
## A  23.300000  9.933669     10       0
## B  21.300000  8.287206     10       0
## AB  9.375000  9.038608      8       2
```

Obtenemos el mismo valor que en el ejercicio anterior, aunque además muestra la media y desviación estándar de cada grupo.

Si además queremos observar si hay diferencias entre medias en los distintos grupos, podemos marcar la casilla "Comparaciones dos a dos de las medias" en la misma ventana de creación del test ANOVA a la que hemos accedido anteriormente. De esta manera obtenemos este código este código requiere del paquete "multcomp" activado):

```
require(multcomp)
```

```
## Loading required package: multcomp
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
AnovaModel.6 <- aov(values ~ ind, data = estudio)
summary(AnovaModel.6)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## ind          3   2493   830.9   8.526 0.000282 ***
## Residuals   31   3021    97.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 5 observations deleted due to missingness
```

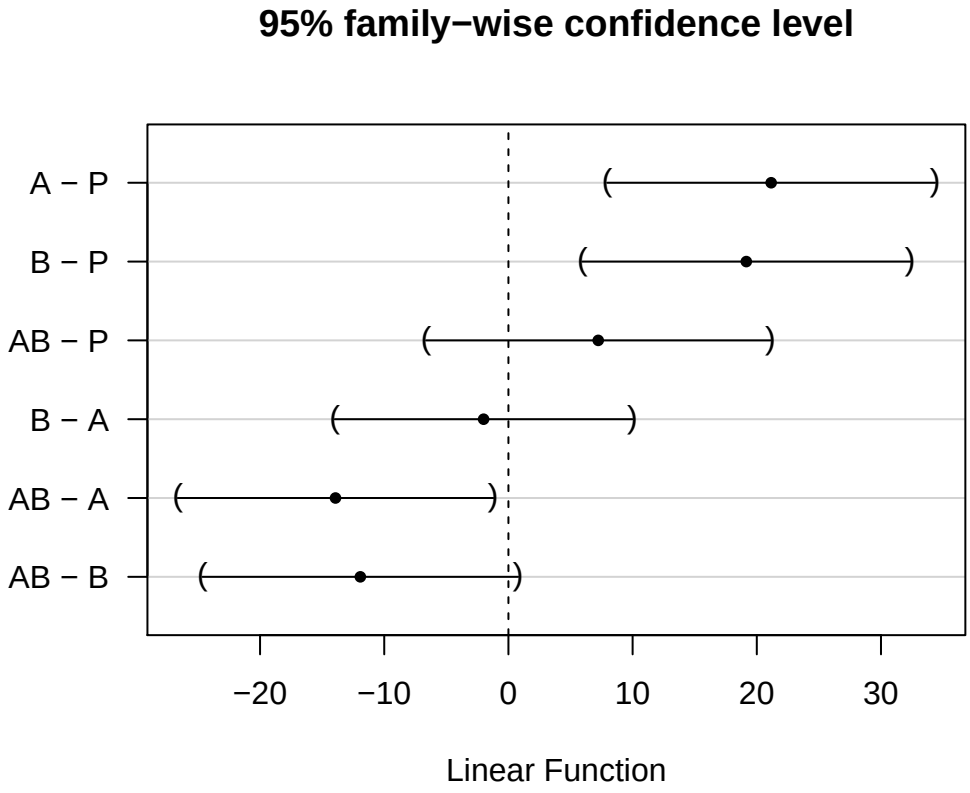
```
with(estudio, numSummary(values, groups = ind, statistics = c("mean", "sd")))
```

```
##         mean        sd data:n data:NA
## P   2.142857 12.535663      7       3
## A  23.300000  9.933669     10       0
## B  21.300000  8.287206     10       0
## AB  9.375000  9.038608      8       2
```

```
local({
    .Pairs <- glht(AnovaModel.6, linfct = mcp(ind = "Tukey"))
    print(summary(.Pairs))  # pairwise tests
    print(confint(.Pairs))  # confidence intervals
    print(cld(.Pairs))  # compact letter display
    old.oma <- par(oma = c(0, 5, 0, 0))
    plot(confint(.Pairs))
    par(old.oma)
})
```

```
##
```

```
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = values ~ ind, data = estudio)
##
## Linear Hypotheses:
##             Estimate Std. Error t value Pr(>|t|)
## A - P == 0    21.157      4.865   4.349  < 0.001 ***
## B - P == 0    19.157      4.865   3.938  0.00226 **
## AB - P == 0    7.232      5.109   1.416  0.49878
## B - A == 0    -2.000      4.415  -0.453  0.96844
## AB - A == 0  -13.925      4.683  -2.974  0.02735 *
## AB - B == 0  -11.925      4.683  -2.547  0.07207 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
##
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = values ~ ind, data = estudio)
##
## Quantile = 2.7144
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##             Estimate lwr      upr
## A - P == 0   21.1571   7.9522  34.3621
## B - P == 0   19.1571   5.9522  32.3621
## AB - P == 0   7.2321  -6.6358  21.1001
## B - A == 0   -2.0000 -13.9833   9.9833
## AB - A == 0 -13.9250 -26.6352  -1.2148
## AB - B == 0 -11.9250 -24.6352   0.7852
##
##    P    A    B   AB
##  "a"  "c" "bc" "ab"
```



Encontramos que seleccionando esa casilla, realiza automáticamente el *test Tukey* obteniendo los mismos resultados que en el ejercicio anterior. Además, también realiza automáticamente el nivel de confidencia "family-wise", el cual nos indica la diferencia de las medias de cada par de grupos. Aqui podemos observar visualmente los grupos con evidencia de que sus medias son diferentes entre ellas (la diferencia de sus valores no se encuentra dentro del intervalo de confianza en el valor 0, que indicaria que su no diferencia). De esta manera, confirmamos que los fármacos A y B difieren significativamente del placebo con un 95% de confidencia.