

CASO PRÁCTICO: CURVAS ROC.

Para nuestra práctica vamos a utilizar el archivo aSHA, de un estudio publicado en *IntensiveCare Medicine* 36(1), 107–115. DOI:10.1007/s00134-009-1641-en 2010.

Nuestro archivo contiene siete variables y 113 muestras. De las siete variables nos vamos a centrar en la variable "outcome", en la "ndka" y en la "s100b". El estudio trata de ver cómo predecir las complicaciones en los pacientes con hemorragia subaracnoidea tras sufrir una rotura de un aneurisma.

La variable "outcome" es dicotómica con dos categorías: "Good" y "Poor". La variable "ndka" es la medida de nucleótido difostatasa A, y la variable "s100b" es un marcador de laboratorio bien conocido.

Vamos a comenzar el ejemplo con Rcmdr. Para ello, tenemos que instalar el paquete RcmdrPlugin.ROC, y cargarlo. Nos aparece una ventana igual que la de Rcmdr pero como podéis observar hay algunas opciones más en el menú.

Como siempre, comenzamos cargando el archivo como ya hemos explicado en temas anteriores. Una vez cargado, podemos comenzar haciendo un análisis descriptivo de los que queremos estudiar.

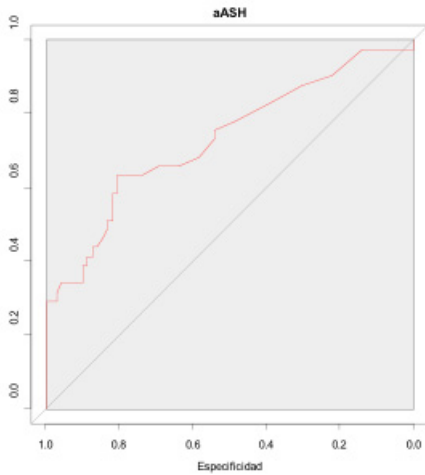
```
> summary(aSAH)
gos6 outcome gender age wfn s100b ndka
1:28 Good:72 Male :42 Min. :18.0 1:39 Min. :0.030 Min. : 3.01
2: 0 Poor:41 Female:71 1st Qu.:42.0 2:32 1st Qu.:0.090 1st Qu.: 9.01
3:13 Median :51.0 3: 4 Median :0.140 Median : 12.22
4: 6 Mean :51.1 4:16 Mean :0.247 Mean : 19.66
5:66 3rd Qu.:61.0 5:22 3rd Qu.:0.330 3rd Qu.: 17.30
Max. :81.0 Max. :2.070 Max. :419.19
```

Ahora vamos a construir nuestra curva ROC para ver cómo se comportan las variables que hemos dicho que íbamos a estudiar. Para ello, seleccionamos la opción "ROC", en la que aparecen dos desplegados: "pROC" y "ROCR". Abrimos la primera opción ("pROC"). Después seleccionamos "plot ROC curve for data...". Nos aparece la siguiente ventana:

The screenshot shows the RcmdrPlugin.ROC interface with the following settings:

- Data:**
 - Predictions variable (pick one): s100b
 - Outcome variable (pick one): outcome
- Options:**
 - Remove NAs
 - Show/input % instead of 0-1
 - Direction:**
 - auto
 - Control > cases
 - Control <= cases

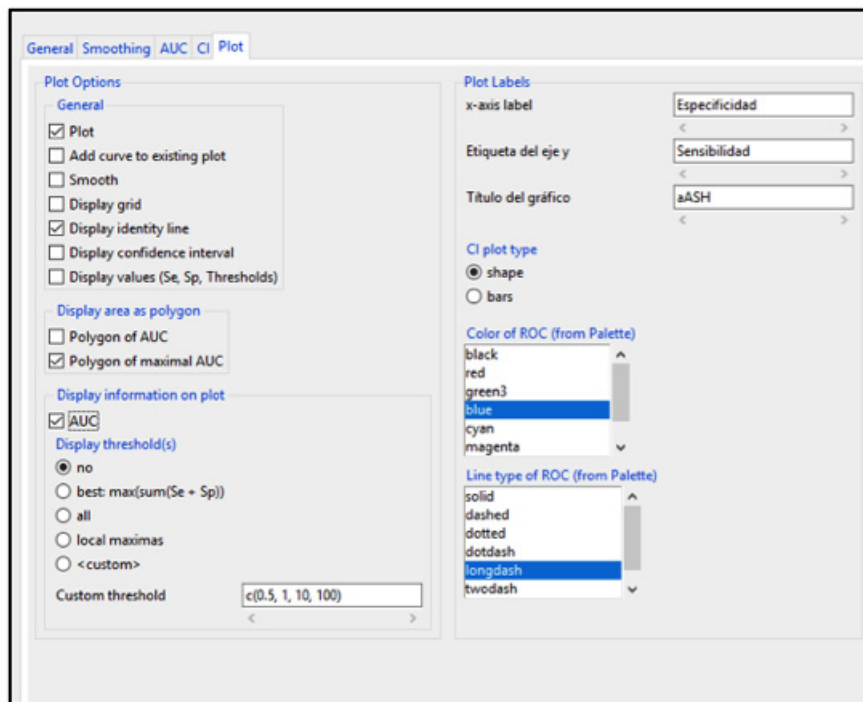
Seleccionamos nuestras variables de estudio, "s100b" y "outcome". En la ventana "AUC" marcamos que nos calcule el área bajo la curva; en la ventana CI le decimos que nos calcule el intervalo de confianza y, por último, en la ventana plot seleccionamos las opciones que queramos para construir el gráfico. El resultado es el siguiente:



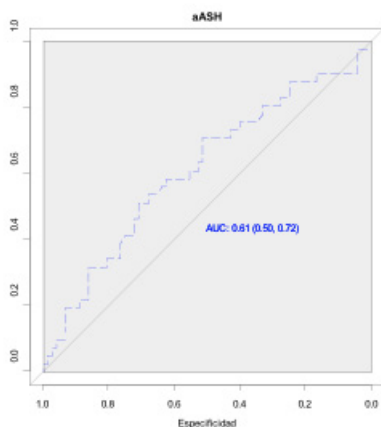
En Rcmdr, en la ventana de salida, tenemos los datos del área bajo la curva y su intervalo de confianza:

```
Data: s100b in 72 controls (outcome Good) < 41 cases (outcome Poor).
Area under the curve: 0.7314
95% CI: 0.6301-0.8326 (DeLong)
```

Como podéis ver el AUC es de 0,73 (IC95 % 0,63-0,83). Vamos a construir ahora la curva ROC con la variable "ndka". Para ello, seguimos el mismo proceso, solo que el lugar de marcar el s100b, ahora marcaremos la variable que queremos evaluar. Para diferenciarla dibujaremos la curva de color azul y con rayas discontinuas. Esto, como ya hemos dicho, está en la ventana de opciones. Os muestro la pantalla de las opciones seleccionadas para construir esta curva:

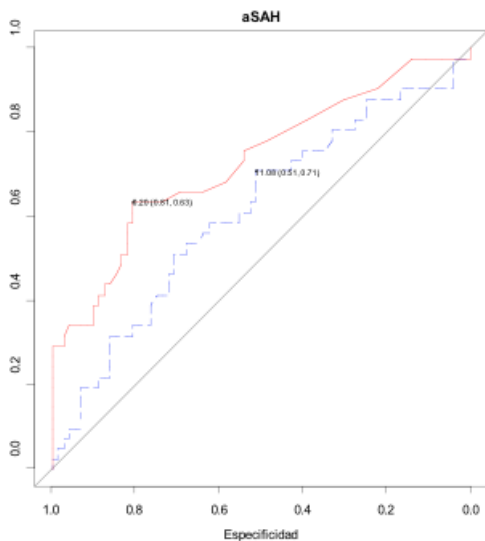


El resultado es el siguiente:



Observad que he marcado la opción de "Display information on plot" y he marcado la opción AUC. La gráfica resultante es la de la izquierda que contiene el AUC y su intervalo de confianza. Para ver el comportamiento de las dos variables estudiadas, nos interesaría verlo en la misma gráfica. Vamos, pues, a construirla. Primero construimos una curva ROC de igual manera que lo hemos construido con la variable "s100b", después, construimos la curva con

la variable "ndka", pero, en esta última, no se nos puede olvidar marcar la casilla "Add curve to existingplot". La gráfica es la siguiente:

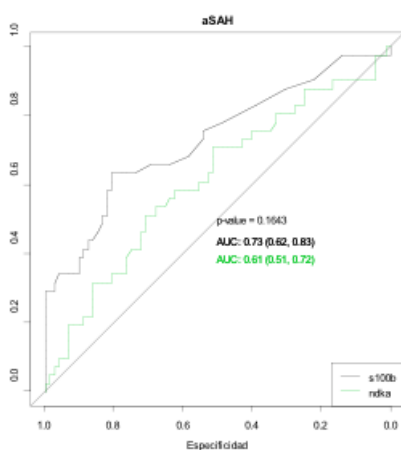


¿Veis las dos curvas? No le he dicho que me dibuje el dato del AUC porque me los superpone y, por tanto, no se ven bien en la gráfica. En el caso de la variable "ndka" su área bajo la curva es de 0,61 (IC 95 % 0,50 a 0,72). Como veis, el intervalo contiene el 0,50, lo que implica que no discrimina bien, por tanto, no sería útil para discriminar las complicaciones.

Podéis ver en la gráfica también el valor de las variables en el que la sensibilidad y la especificidad son mayores.

Por último, nos queda comparar las curvas para ver si son iguales o diferentes. Para ello, nos vamos a la pestaña ROC y seleccionamos "Paired ROC curves comparison...". Seleccionamos nuestras variables de estudio, y en la pestaña de plot le marcamos las opciones que deseamos que nos aparezcan. El resultado del test es el siguiente:

```
DeLong's test for two correlated ROC curves
data: roc.obj and roc.obj2
Z = 1.3908, p-value = 0.1643
alternative hypothesis: true difference in AUC is not equal to 0
sample estimates:
AUC of roc1 AUC of roc2
0.7313686 0.6119580
```

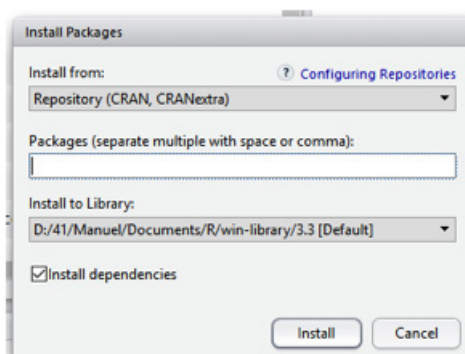


Vemos que emplea el test de DeLong. Nos calcula el estadístico "Z" y su "p" asociada, que este caso es muy grande (0,1643), por lo que no podemos rechazar la hipótesis nula y por tanto tendremos que afirmar que las dos curvas son iguales. Rcmdr nos construye la gráfica de comparación que es la siguiente:

Vamos a ver ahora cómo podemos trabajar con Rstudio. Lo primero sería cargar el paquete pROC. Para ello, en el cuadrado inferior derecho hay una pestaña denominada "Packages" y dentro de esta, hay dos pestañas "update" e "install", marcamos esta última y nos

aparece la ventana de la izquierda, en el cursor escribimos "pROC" y marcamos la casilla "installdependencies" y a continuación marcamos "install". El paquete se instalará junto con otros paquetes que necesita. Una vez instalado procedemos a cargar nuestro archivo,

en nuestro caso es una base de datos en formato Excel con 113 registros, es el mismo archivo que hemos usado con Rcmdr.



Como ya sabéis, comenzamos con las órdenes en Rstudio. La primera cargar el paquete y la segunda cargar el archivo:

```
library(pROC)
data("aSAH")
```

Empleamos la orden *data* porque el archivo viene incluido en el paquete pROC.

Una vez tenemos los elementos procedemos a construir la curva ROC. La estructura de esta orden es la siguiente:

Nombre objeto<-roc(respuesta, predictor). En nuestro caso vamos a crear dos objetos. El primero le llamamos roc1, en el que nuestra variable respuesta es "outcome", y la predictora "ndka", y, el segundo objeto, roc2, con la misma variable respuesta y usando como predictora la variable "s100b".

Como hemos ya hemos descrito las órdenes son:

```
roc1<- roc(aSAH$outcome, aSAH$ndka)
roc2<- roc(aSAH$outcome, aSAH$s100b)
```

Una vez ejecutadas estas órdenes podemos ir obteniendo los datos que nos interesan. Por ejemplo, queremos saber cuál es el área bajo la curva:

```
auc(roc1)
auc(roc2)
```

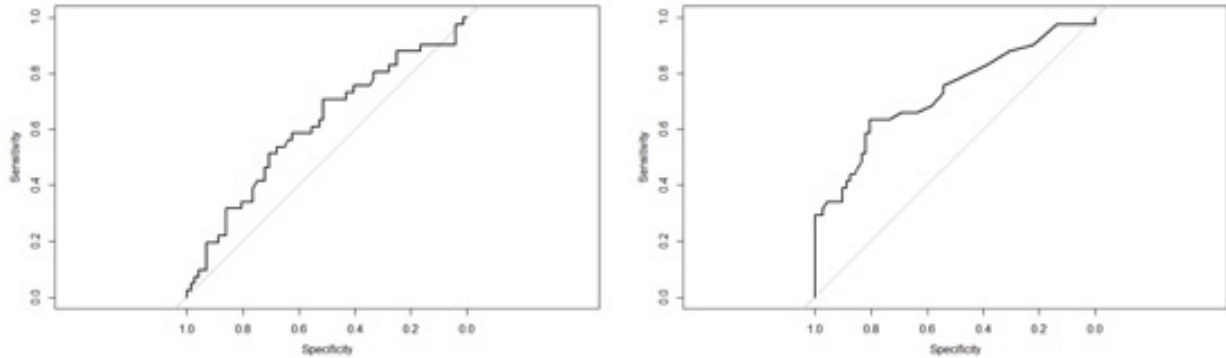
Los resultados obtenidos son:

```
> auc(roc1)
Area under the curve: 0.612
> auc(roc2)
Area under the curve: 0.7314
> |
```

En muchas ocasiones, en los trabajos sólo figura este dato. Sin embargo, ya hemos comentado que lo correcto es informarlo siempre con su intervalo de confianza, este se obtiene:

```
ci(roc1)
ci(roc2)
```

Una vez conocido estos datos nos interesa construir la gráfica correspondiente. Esto se hace con la orden "plot": plot(roc1) y plot(roc2). El resultado es el siguiente:



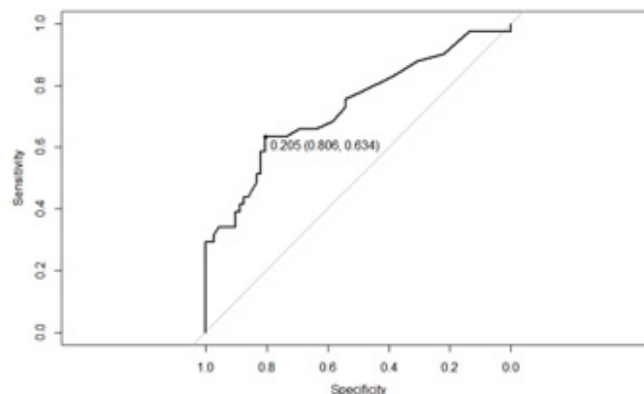
A la izquierda está la correspondiente a "roc1" y a la derecha la correspondiente a "roc2". Ahora nos interesa conocer qué punto es el de mayor sensibilidad y especificidad, ello lo conseguimos con la orden:

```
ci.coords(aSAH$outcome, aSAH$s100b, x="best", input =
"threshold"ret=c("specificity", "ppv", "tp"))
```

```
> ci.coords(aSAH$outcome, aSAH$s100b, x="best", input = "threshold",
+           ret=c("specificity", "ppv", "tp"))
95% CI (2000 stratified bootstrap replicates):
                2.5%    50% 97.5%
threshold best: specificity 0.6389 0.8194    1
threshold best: ppv        0.5246 0.6667    1
threshold best: tp         15.0000 26.0000   33
```

Como veis, nos muestra el intervalo de confianza y el valor de la especificidad, valor predictivo positivo (vpp) y verdaderos positivos (tp=true positive).

Con la orden `plot(roc2, print.thres="best", print.thres.best.method="youden")`, puedo dibujar este dato en la gráfica:



Vamos ahora a personalizar las gráficas, los comandos son los ya descritos en temas anteriores y otros propios de este paquete. Si queremos mostrar el intervalo de confianza en cada uno de los puntos, lo podemos hacer de la sensibilidad o de la especificidad o de ambos, además, lo podemos hacer mediante barras o mediante una sombra. Veamos el primer caso

```
> ci.thresholds(roc1)
95% CI (2000 stratified bootstrap replicates):
thresholds sp.low sp.median sp.high se.low se.median se.high
3.440 0.00000 0.01389 0.04167 1.00000 1.00000 1.00000
5.105 0.00000 0.04167 0.09722 0.92680 0.97560 1.00000
7.240 0.08333 0.16670 0.26390 0.80490 0.90240 0.97560
8.160 0.15280 0.25000 0.36110 0.75610 0.87800 0.97560
8.720 0.18060 0.27780 0.38890 0.70730 0.82930 0.92680
9.520 0.23610 0.33330 0.44440 0.65850 0.80490 0.92680
9.840 0.29170 0.40280 0.51390 0.60980 0.75610 0.87800
10.365 0.31940 0.43060 0.55560 0.58540 0.73170 0.85370
11.080 0.40280 0.51390 0.63890 0.56100 0.70730 0.82930
11.700 0.41670 0.52780 0.65280 0.48780 0.63410 0.78050
12.095 0.44440 0.55560 0.66670 0.46340 0.60980 0.75610
12.730 0.51390 0.62500 0.73610 0.43900 0.58540 0.73170
12.850 0.52780 0.63890 0.75000 0.41460 0.56100 0.70730
13.160 0.56940 0.68060 0.79170 0.39020 0.53660 0.68290
```

Lo primero es calcular estos intervalos de confianza de cada punto. Ello lo hacemos con la orden: *ci.thresholds(roc1)*. Obtenemos una tabla con los distintos puntos y su especificidad y sensibilidad correspondientes. Ya hemos comentado que a lo mejor no nos interesa el punto de mejor especificidad y sensibilidad,

ya que puede que nos interese el test para cribado o para confirmar, por ello nos da la opción de escoger el punto que más se adapta a nuestras necesidades.

Ahora vamos a escribir estos datos en la curva ROC. Hay dos formas, bien con líneas o bien con sombras.

Vamos a construir la curva ROC con los intervalos de confianza de la sensibilidad en forma de barras. Las órdenes correspondientes son:

1. Dibujar la curva ROC

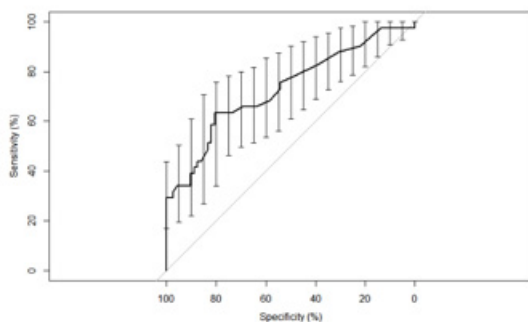
```
plot(roc1)
```

2. Calcular el intervalo de confianza de la sensibilidad

```
sens.ci <- ci.se(roc1, specificities=seq(0, 100, 5))
```

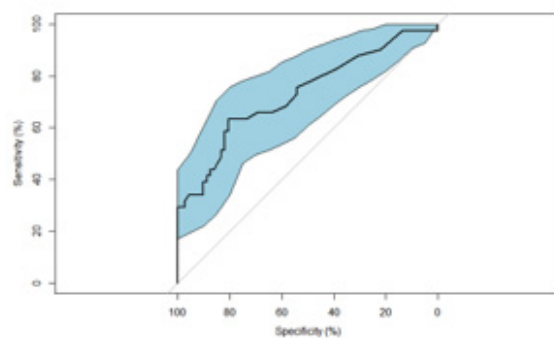
3.- Dibujar las barras del intervalo de confianza

```
plot(sens.ci, type="bars")
```



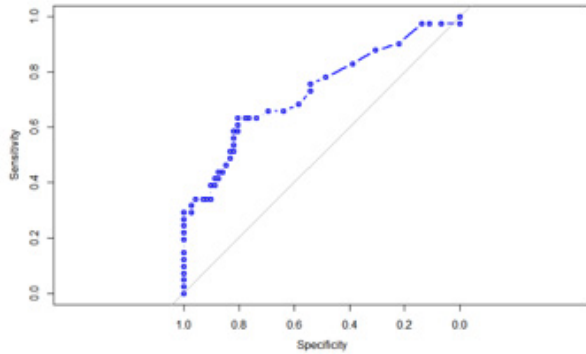
Si queremos dibujar una sombra la orden sería:

```
plot(sens.ci, type="shape", col="lightblue")
```



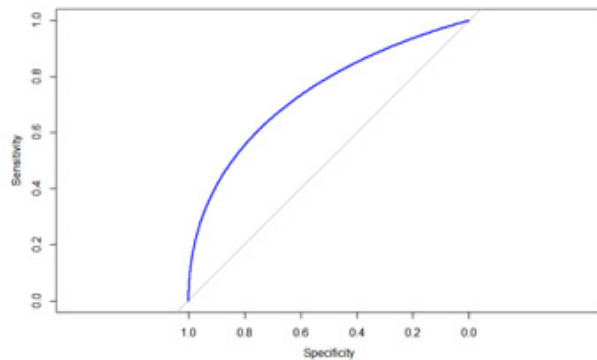
El resultado lo podéis ver en las gráficas. Otra posibilidad es construir la gráfica marcando los puntos para poder observar que distribución tienen. Para ello, las ordenes son las siguientes:

```
plot.roc(aSAH$outcome, aSAH$s100b, type="b", pch=21, col="blue", bg="grey")
```



Por ultimo, podemos construir la gráfica de forma suavizada. Para ello, utilizamos la siguiente orden:

```
plot.roc(smooth(roc1), add=FALSE, col="blue")
```



Con esto terminamos las distintas formas de construir las distintas modalidades de curvas. Vamos ahora a personalizarlas:

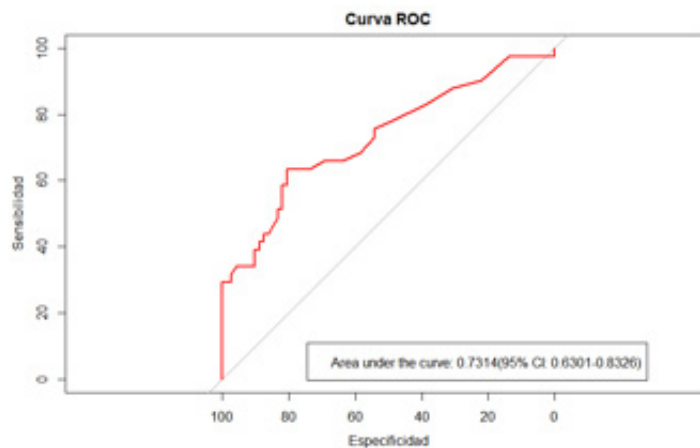
Vamos a escribir el título de los ejes en castellano, vamos a dibujar la curva en color rojo y vamos a poner en el gráfico el área bajo la curva y su intervalo de confianza:

```
plot(roc1, xlab="Especificidad", ylab="Sensibilidad", main="Curva ROC", col="red")
legend(locator(1), legend="Area under the curve: 0.7314(95% CI: 0.6301-0.8326)")
```

Con la orden "*plot*" dibujamos la curva y le escribimos las opciones de la misma forma que hemos hecho en temas anteriores. Como veis, los comandos son los mismos.

Con la orden "*legend*" escribimos la leyenda. Los datos del AUC y su IC los he copiado de la consola de R para dibujarlos en el gráfico. Con "*locator(1)*" le indicamos que marcaremos con el ratón donde quiero poner la leyenda.

El resultado es el siguiente:



Vamos ahora a comparar dos curvas: en nuestro ejemplo la variable "ndka", que corresponde al objeto roc2 y la variable "s100b" que corresponde al objeto roc1. Como ya hemos visto, podemos calcular su AUC y su IC 95 %.

Los resultados son los siguientes roc1, variable s100b tiene un AUC de 0,7314 (95 % CI 0.6301-0.8326), y para roc2, variable ndka, los resultados son AUC 0.612 (95 % CI: 0,5012-0,7227).

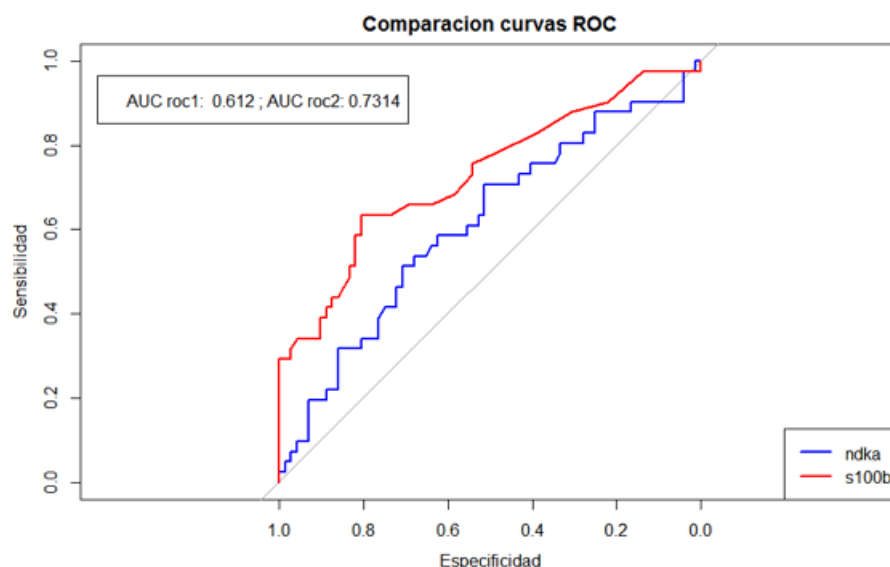
La pregunta que nos hacemos ahora es si estas curvas son iguales. Para ello nos planteamos nuestra hipótesis nula, que sería que las dos curvas son iguales, y la alternativa que son diferentes. Para ello utilizamos una orden muy sencilla "roc.test". La orden sería:

```
roc.test(roc1, roc2)
```

El resultado es:

```
data: roc1 and roc2
Z = -1.3908, p-value = 0.1643
alternative hypothesis: true difference in AUC is not equal to 0
sample estimates:
AUC of roc1 AUC of roc2
0.6119580   0.7313686
```

Como veis la p es mayor de 0,05, error muy grande, por lo que nos quedamos con la hipótesis nula o, lo que es lo mismo, podemos afirmar que las dos curvas son iguales. Vamos ahora a dibujar las curvas, primero dibujamos una, y después la otra, y a esta última le añadimos la orden `add=TRUE`.



Esta curva está construida con las siguientes órdenes:

```
plot(roc1, col="blue", xlab="Especificidad", ylab="Sensibilidad",
main="Comparacioncurvas ROC")
```



```
plot(roc2, col="red", add=TRUE)
legend("bottomright", legend=c("ndka", "s100b"), col=c("blue", "red"), lwd=2)
legend(locator(1), legend="AUC roc1: 0.612 ; AUC roc2: 0.7314")
```

Por último, solo nos resta conocer la potencia del test que hemos realizado, para ello vamos a utilizar la orden

```
power.roc.test(roc1, roc2, power=0.9)
```

El resultado es

```
Two ROC curves power calculation
      ncases = 41
      ncontrols = 72
      auc1 = 0.611958
      auc2 = 0.7313686
      sig.level = 0.05
      power = 0.2619268
      alternative = two.sided
```

La potencia o error beta es de 0,262, algo baja. Para aumentarla, habría que hacer el estudio con más individuos.

Como habéis observado con Rstudio tenemos más opciones y posibilidades, de todas formas Rcmdr también está bastante completo y como habéis visto es muy fácil de usar.
